

(最大8個)を記憶することにより、そのプロバイダ(ローカルコンプロバイダ)は、この領域定義ブロックのアロケーションテーブルで割り当てられるユーザブロックを、共通領域として使用することができる。

【0214】また、同一のユーザブロックを割り当てる2つの領域定義ブロックに、ローカルコンプロバイダを登録し、領域定義ブロック毎に、異なるアクセス権を設定することにより、そのユーザブロックに対するアクセス権を、ローカルコンプロバイダ毎に設定することができる。

【0215】このように、オーバラッププロバイダおよびローカルコンプロバイダを設定することにより、複数のプロバイダ(即ち、R/W)に対応して個別の処理を行うことができる。

【0216】なお、本発明は、電波(非接触)で信号を授受する場合の他、物理的に結合された状態(接触)で信号を授受する場合にも適用することができる。停電の場合、あるいは、電池で動作する装置において、電池が取り外されてしまったような場合に、データを確保しておくことができる。

【0217】また、上記各処理を行うプログラムは、磁気ディスク、CD-ROMなどの記録媒体よりなる伝送媒体に記録してユーザに提供したり、ネットワークなどの伝送媒体を介してユーザに伝送し、ハードディスク、固体メモリなどの記録媒体よりなる伝送媒体に記録し、利用させるようにすることができる。

【0218】

【発明の効果】以上のごとく、請求項1に記載の情報処理方法、請求項3に記載の伝送媒体、および請求項4に記載の情報処理装置によれば、1以上の利用者のデータを記憶する第1の領域と、第1の領域に記憶されている1以上の利用者により使用され、第1の領域として使用されていない空き領域に設定される第2の領域とを含む記憶部を利用してコマンドを処理するので、メモリの利用効率を高くすることができる。

【0219】請求項6に記載の情報処理方法、請求項10に記載の伝送媒体、および請求項11に記載の情報処理装置によれば、物理ブロックに記憶されるデータに、論理ブロック番号を割り当てるとともに、所定の論理ブロック番号を有する新たなデータを、その論理ブロック番号を有するデータが記憶されている物理ブロック以外の物理ブロックに記憶するようにしたので、メモリコラプションの発生を論理的に抑制することができる。

【0220】請求項15に記載の情報処理方法、請求項17に記載の伝送媒体、および請求項18に記載の情報処理装置によれば、第2の領域の所定のブロックのデータが、認識番号を有し、利用者により供給されたコマンドが有する認識番号と、データが有する認識番号を比較して、同一のコマンドを繰り返し処理しないようにするので、各コマンドを誤って複数回処理しないようにする

ことができる。

【0221】請求項20に記載の情報処理方法、請求項22に記載の伝送媒体、および請求項23に記載の情報処理装置によれば、ブロックに記憶されるデータに、記憶される順番に対応する番号を割り当てるとともに、最後の番号を有するブロックが、割り当てられた領域の最後のブロックである場合、新たなデータを、先頭のブロックに記憶し、最後の番号を有するブロックが、最後のブロックではない場合、新たなデータを、最後の番号を有するブロックの次のブロックに記憶するようにしたので、メモリコラプションの発生を論理的に抑制することができる。

【0222】請求項25に記載の情報処理方法、請求項26に記載の伝送媒体、および請求項27に記載の情報処理装置によれば、第2の領域における所定の領域、および、1利用者に対してそれぞれ異なるアクセス権を規定する複数のデータを、第1の領域に記憶する記憶部を利用して、コマンドを処理するようにしたので、所定の利用者に対して、所定の記憶領域における複数のアクセス権を与えることができる。

【0223】請求項28に記載の情報処理方法、請求項29に記載の伝送媒体、および請求項30に記載の情報処理装置によれば、第2の領域における所定の領域を複数の利用者が共同して使用するデータを、第1の領域に記憶する記憶部を利用してコマンドを処理するようにしたので、複数の利用者により、同一の記憶領域を割り当てることができる。

【0224】請求項31に記載の情報処理方法、請求項32に記載の伝送媒体、および請求項33に記載の情報処理装置によれば、第2の領域における所定の領域、および、複数の利用者のそれぞれ異なるアクセス権を規定する複数のデータを、第1の領域に記憶する記憶部を利用して、コマンドを処理するようにしたので、複数の利用者に対して、所定の記憶領域における異なるアクセス権を与えることができる。

【図面の簡単な説明】

【図1】本発明の情報処理装置の一実施の形態であるICカード2を利用した非接触カードシステムの一例を示すブロック図である。

【図2】図1のリーダー/ライター1の構成例を示すブロック図である。

【図3】本発明の情報処理装置の一実施の形態であるICカード2の構成を示すブロック図である。

【図4】図3のEEPROM66のメモリの割り当ての一例を示す図である。

【図5】図4のシステムIDブロックの各領域の割り当ての一例を示す図である。

【図6】図5のアトリビュート部の一例を示す図である。

【図7】図3の領域定義ブロックの各領域の割り当ての

一例を示す図である。

【図8】図3のユーザブロックの割り当ての一例を示す図である。

【図9】図7のパスブロックパーミッションの一例を示す図である。

【図10】図3のユーザブロックの各領域の割り当ての一例を示す図である。

【図11】図8のランダムアクセス領域のユーザブロックのアトリビュート部の一例を示す図である。

【図12】パスブロックの各領域の割り当ての一例を示す図である。

【図13】図8のシーケンシャルアクセス領域のユーザブロックのアトリビュート部の一例を示す図である。

【図14】図1の非接触カードシステムの動作の説明するフローチャートである。

【図15】図1の非接触カードシステムの動作の説明するタイミングチャートである。

【図16】BPSK変調の一例を示す図である。

【図17】図8のランダムアクセス領域のユーザブロックに対する書込時における、ICカード2の動作について説明するフローチャートである。

【図18】図8のランダムアクセス領域のユーザブロッ

*クに対する書込時における、ICカード2の動作について説明するフローチャートである。

【図19】図8のランダムアクセス領域のユーザブロックに対する書込時における、ICカード2の動作について説明するフローチャートである。

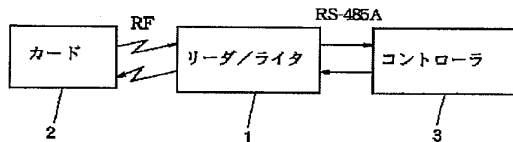
【図20】図8のシーケンシャルアクセス領域のユーザブロックに対する書込時における、ICカード2の動作について説明するフローチャートである。

【図21】図8のシーケンシャルアクセス領域のユーザブロックに対する書込時における、ICカード2の動作について説明するフローチャートである。

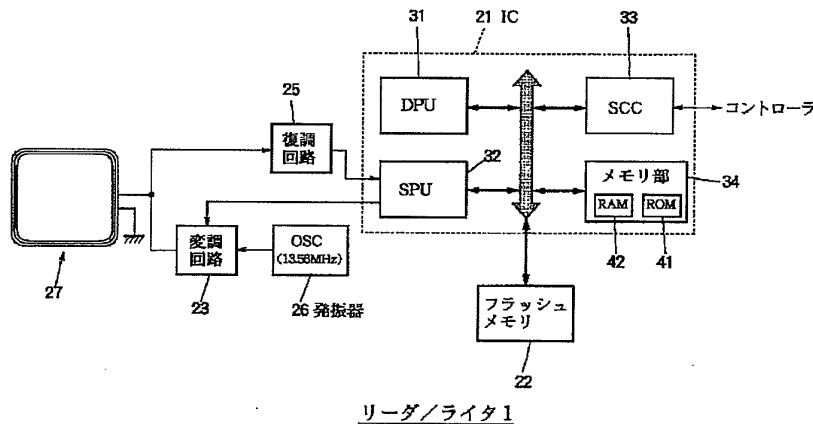
【符号の説明】

- 1 リーダ/ライタ, 2 ICカード, 3 コントローラ,
- 21 IC, 23 変調回路, 25 復調回路,
- 27 アンテナ, 51 IC, 52 コンデンサ,
- 53 アンテナ, 61 RFインタフェース部,
- 62 BPSK復調回路, 63 PLL部,
- 64 演算部, 65 ROM, 66 EEPROM,
- 67 RAM, 68 BPSK変調回路, 81 ASK復調部,
- 82 電圧レギュレータ, 83 発振回路,
- 84 ASK変調部, 91 シーケンサ,
- 92 暗号/復号部, 93 パリティ演算部

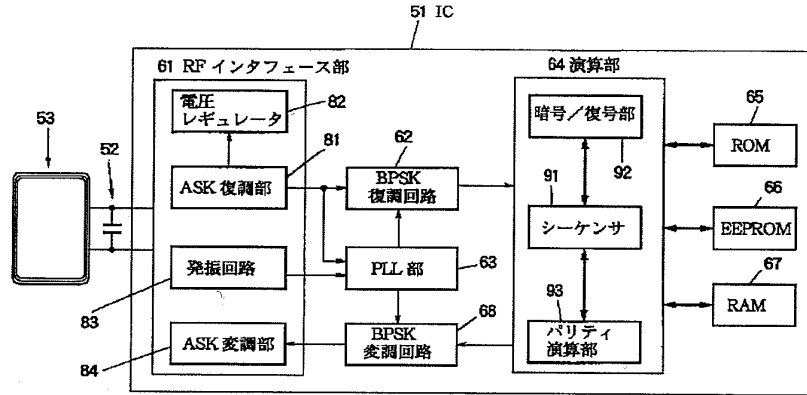
【図1】



【図2】

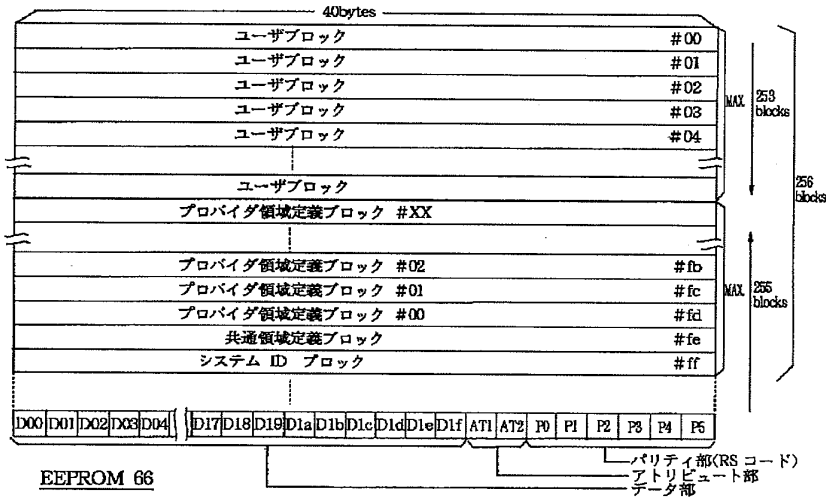


【図3】

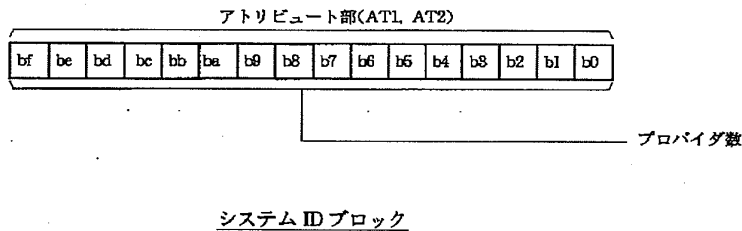


カード 2

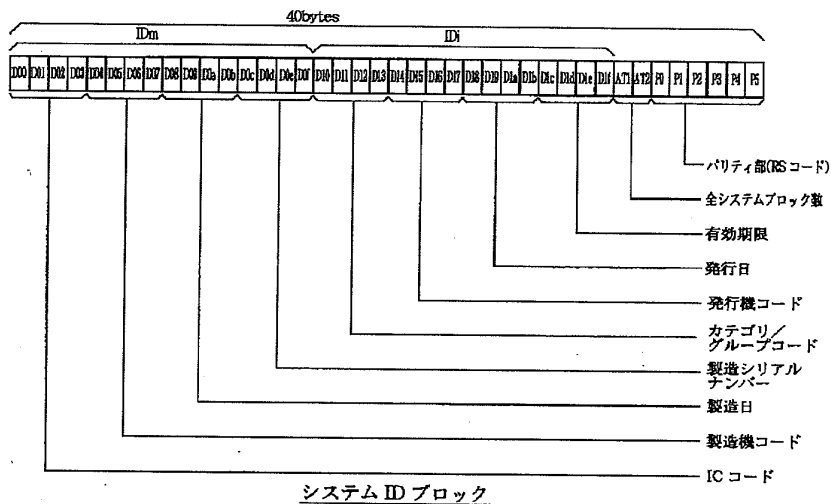
【図4】



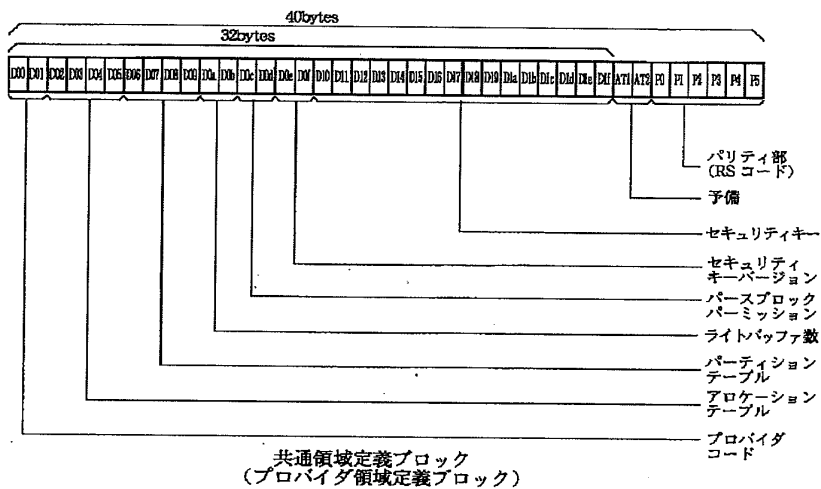
【図6】



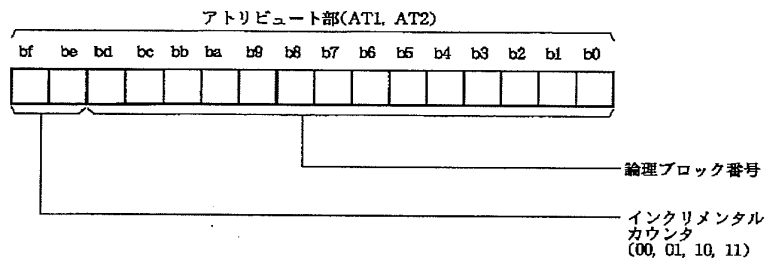
【図5】



【図7】

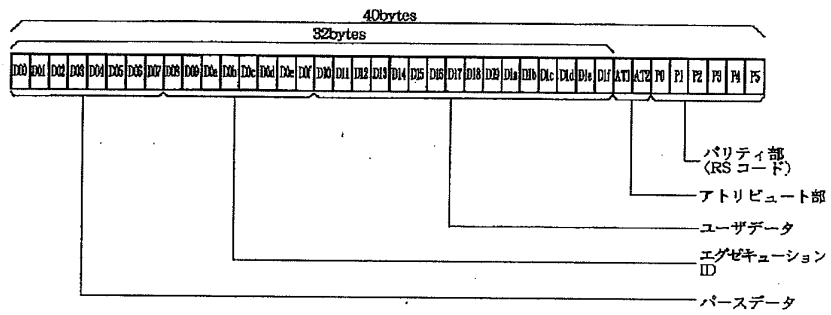


【図11】



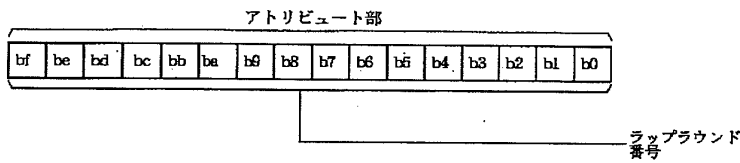
ランダムアクセス領域のユーザブロック

【図12】



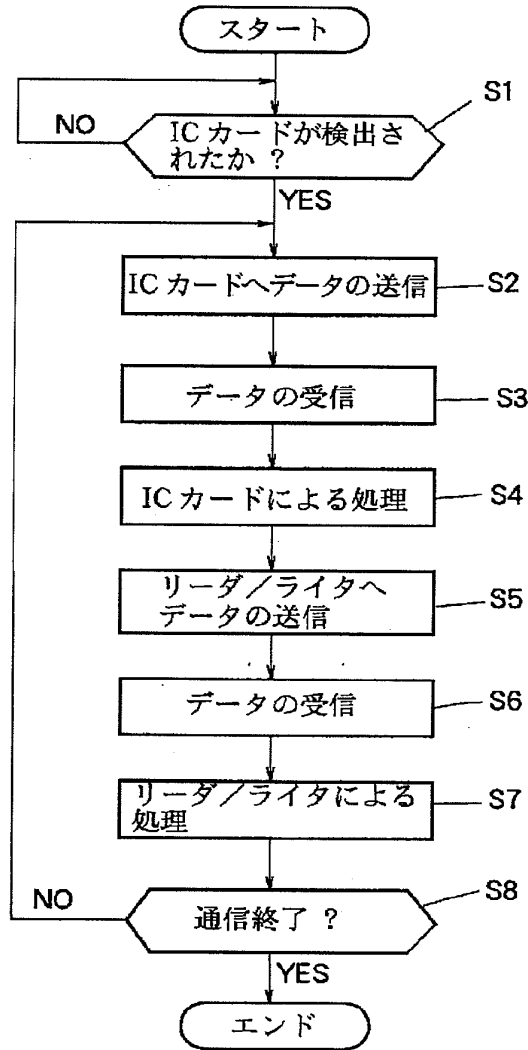
パースブロック

【図13】

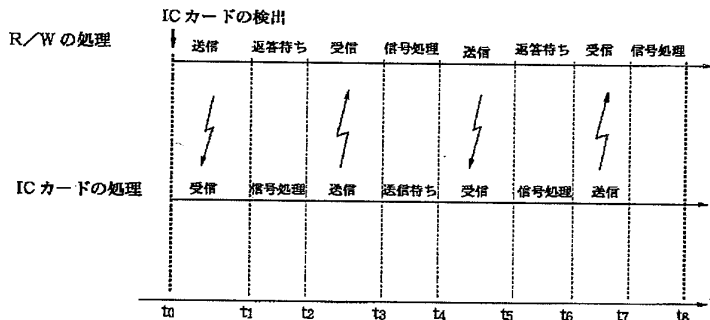


シーケンシャルアクセス領域のユーザブロック

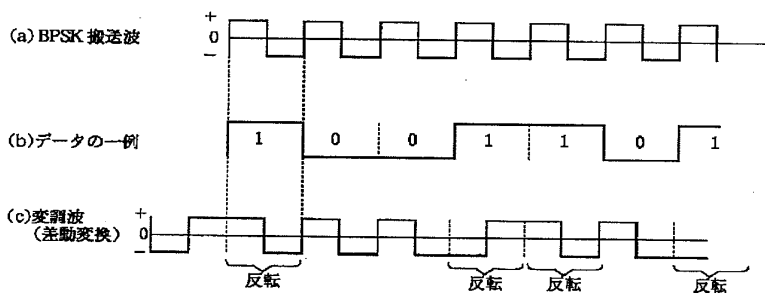
【図14】



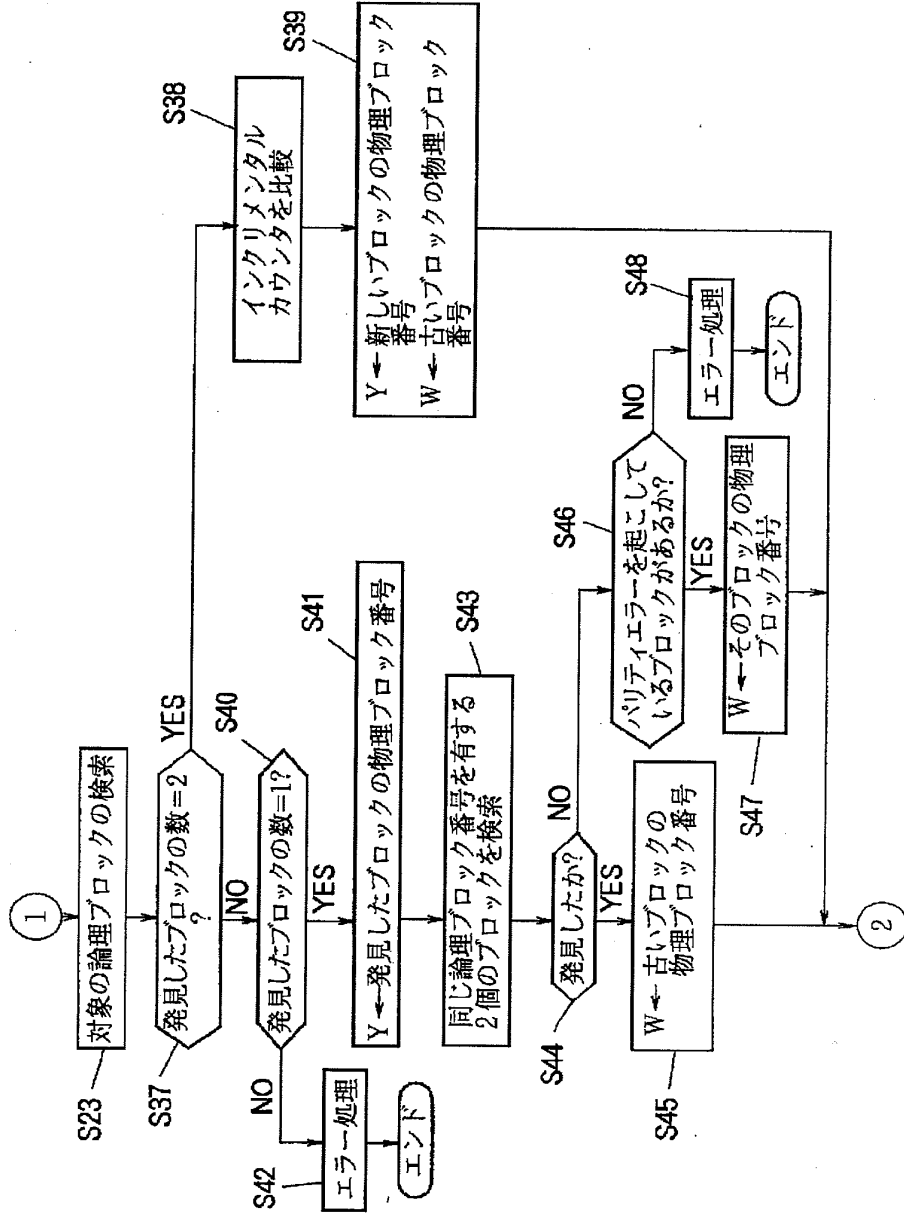
【図15】



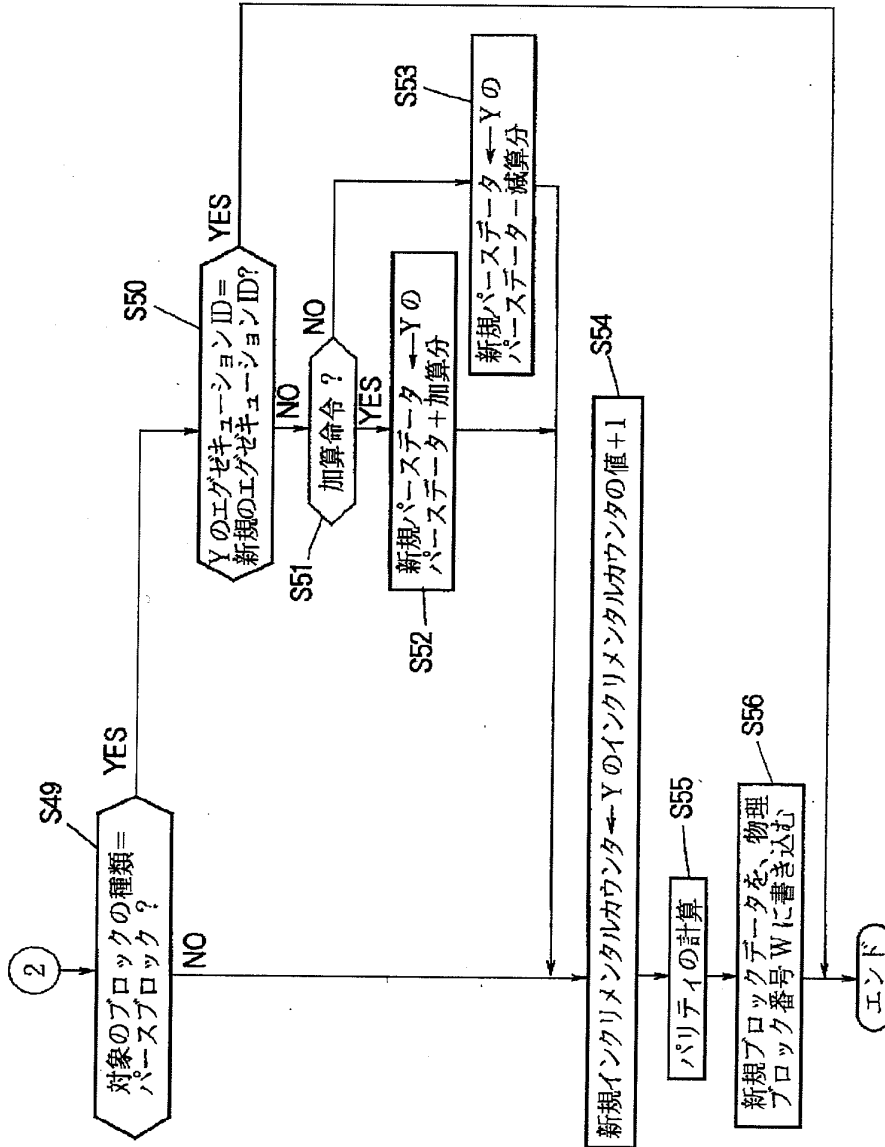
【図16】



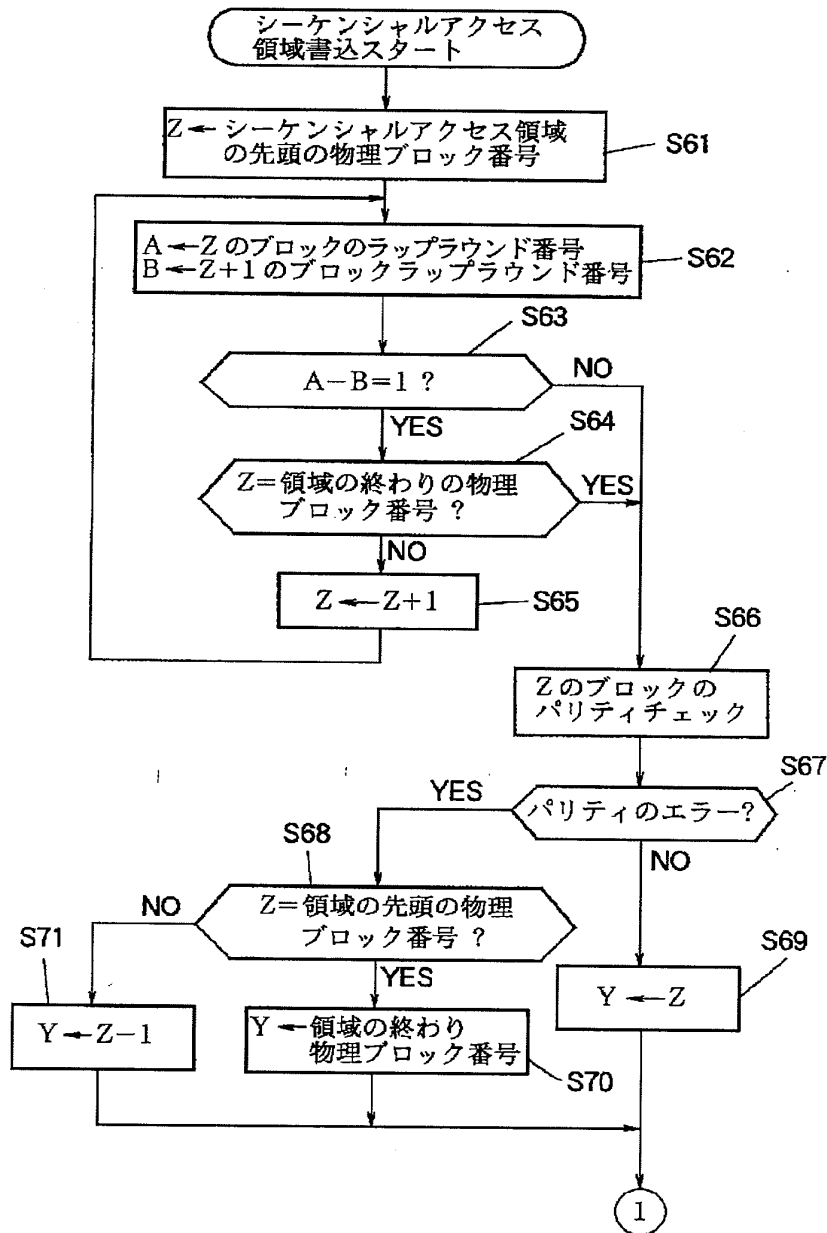
【図18】



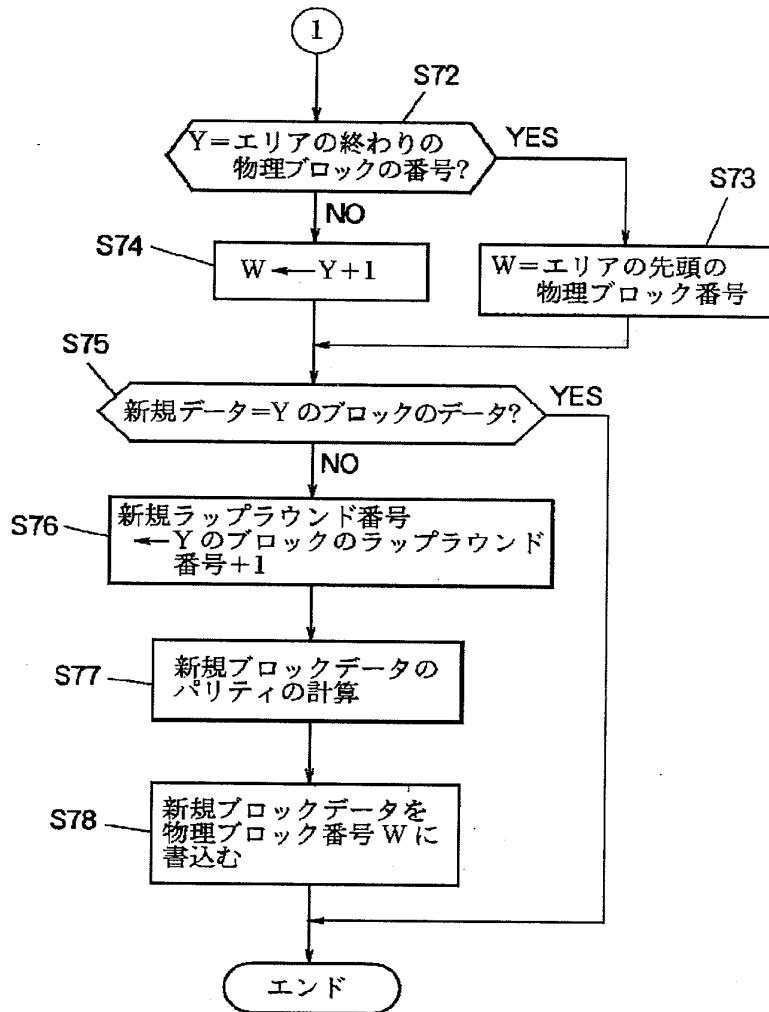
【図19】



【図20】



【図21】



フロントページの続き

(51) Int. Cl.⁶
G 0 6 K 19/07

識別記号

F I
G 0 6 K 19/00

N

【公報種別】 特許法第 17 条の 2 の規定による補正の掲載
【部門区分】 第 6 部門第 3 区分
【発行日】 平成 17 年 5 月 19 日 (2005.5.19)

【公開番号】 特開平 10-105661
【公開日】 平成 10 年 4 月 24 日 (1998.4.24)
【出願番号】 特願平 9-171430
【国際特許分類第 7 版】

G 0 6 K 17/00
G 0 6 F 3/06
G 0 6 F 3/08
G 0 6 F 12/02
G 0 6 K 19/07

【F I】

G 0 6 K	17/00		F
G 0 6 K	17/00		D
G 0 6 F	3/06	3 0 1 J	
G 0 6 F	3/08		C
G 0 6 F	12/02	5 1 0 A	
G 0 6 K	19/00		N

【手続補正書】

【提出日】 平成 16 年 6 月 25 日 (2004.6.25)

【手続補正 1】

【補正対象書類名】 明細書

【補正対象項目名】 発明の名称

【補正方法】 変更

【補正の内容】

【発明の名称】 情報処理装置および情報処理方法、リーダー/ライターおよびアクセス方法、並びに記録媒体

【手続補正 2】

【補正対象書類名】 明細書

【補正対象項目名】 特許請求の範囲

【補正方法】 変更

【補正の内容】

【特許請求の範囲】

【請求項 1】

所定のシステムを提供するプロバイダの装置からのコマンドを受信する受信手段と、前記コマンドを処理する処理手段と、前記処理の結果を送信する送信手段と、前記プロバイダの装置が利用するデータを記憶するとともに、1以上のユーザブロックが所定の大きさの単位で管理されるユーザブロック領域と、前記ユーザブロックの利用を定義する領域定義ブロック領域が形成される記憶手段とを備え、前記領域定義ブロック領域は、複数の領域定義ブロックからなり、各々の領域定義ブロックには、利用するユーザブロックを規定するデータおよび前記ユーザブロックへのアクセス権を規定するデータが記憶され、前記領域定義ブロック領域に記録されている前記データを利用して前記コマンドが処理される
ことを特徴とする情報処理装置。

【請求項 2】

前記ユーザブロック領域は、前記領域定義ブロック領域として使用されていない空き領域に割り当てられて形成される

ことを特徴とする請求項 1 に記載の情報処理装置。

【請求項 3】

複数の前記領域定義ブロックは、各々の前記プロバイダに対して、それぞれ異なるアクセス権を規定する

ことを特徴とする請求項 1 に記載の情報処理装置。

【請求項 4】

前記コマンドは、前記アクセス権を規定するデータが異なる 2 以上の領域定義ブロックを利用して処理される

ことを特徴とする請求項 3 に記載の情報処理装置。

【請求項 5】

前記アクセス権を規定するデータは、前記ユーザブロック領域のユーザブロックに対するリード/ライトまたはリードオンリのいずれかのアクセスを規定するデータである

ことを特徴とする請求項 4 に記載の情報処理装置。

【請求項 6】

所定のシステムを提供するプロバイダの装置が利用するデータを記憶するとともに、1 以上のユーザブロックが所定の大きさの単位で管理されるユーザブロック領域と、前記ユーザブロックの利用を定義する領域定義ブロック領域が形成される記憶手段であって、前記領域定義ブロック領域は、複数の領域定義ブロックからなり、各々の領域定義ブロックには、利用するユーザブロックを規定するデータおよび前記ユーザブロックへのアクセス権を規定するデータが記憶される記憶手段を備える情報処理装置の情報処理方法において、

前記プロバイダの装置からのコマンドを受信する受信ステップと、

前記領域定義ブロック領域を利用して、前記コマンドを処理する処理ステップと、

前記処理の結果を送信する送信ステップと

を含むことを特徴とする情報処理方法。

【請求項 7】

所定のシステムを提供するプロバイダの装置が利用するデータを記憶するとともに、1 以上のユーザブロックが所定の大きさの単位で管理されるユーザブロック領域と、前記ユーザブロックの利用を定義する領域定義ブロック領域が形成される記憶手段であって、前記領域定義ブロック領域は、複数の領域定義ブロックからなり、各々の領域定義ブロックには、利用するユーザブロックを規定するデータおよび前記ユーザブロックへのアクセス権を規定するデータが記憶される記憶手段を備える情報処理装置の情報処理用のプログラムにおいて、

前記プロバイダの装置からのコマンドの受信を制御する受信制御ステップと、

前記領域定義ブロック領域を利用して、前記コマンドを処理する処理ステップと、

前記処理の結果の送信を制御する送信制御ステップと

を含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項 8】

1 以上のユーザブロックが所定の大きさのブロック単位で管理され、所定のシステムを提供するプロバイダの装置が利用するデータが記憶されるユーザブロック領域が形成される記憶手段を備える情報処理装置の前記ユーザブロック領域にアクセスするためのリーダー/ライターであって、

前記情報処理装置に送信する送信データおよび前記情報処理装置から受信した受信データの処理を行うデータ処理手段と、

前記送信データを変調する変調手段と、

前記受信データを復調する復調手段と

を含み、

前記送信データは、前記記憶手段の領域定義ブロックに記憶されている、前記ユーザブロックを規定するデータおよび前記ユーザブロックへのアクセス権を規定するデータに基づいて制御される前記ユーザブロックにアクセスするためのコマンドである

ことを特徴とするリーダ/ライタ。

【請求項 9】

前記コマンドは、互いに異なる 2 以上の前記領域定義ブロックに記憶されている、前記アクセス権を規定するデータを基に前記ユーザブロック領域にアクセスするためのコマンドである

ことを特徴とする請求項 8 に記載の情報処理装置。

【請求項 10】

1 以上のユーザブロックが所定の大きさのブロック単位で管理され、所定のシステムを提供するプロバイダの装置が利用するデータが記憶されるユーザブロック領域が形成される記憶手段を備える情報処理装置の前記ユーザブロック領域にアクセスするアクセス方法であって、

前記情報処理装置に呼びかけを行う呼びかけステップと、

前記呼びかけに応答した前記情報処理装置に対し、コマンドを送信する送信ステップと、

前記情報処理装置により処理された前記コマンドに対する送信結果を受信する受信ステップと、

前記送信結果を処理する処理ステップと

を含み、

前記送信データは、前記記憶手段の領域定義ブロックに記憶されている、前記ユーザブロックを規定するデータおよび前記ユーザブロックへのアクセス権を規定するデータに基づいて制御される前記ユーザブロックにアクセスするためのコマンドである

ことを特徴とするアクセス方法。

【請求項 11】

前記コマンドは、互いに異なる 2 以上の前記領域定義ブロックに記憶されている、前記アクセス権を規定するデータを基に、前記ユーザブロック領域にアクセスするためのコマンドである

ことを特徴とする請求項 10 に記載のアクセス方法。

【請求項 12】

前記アクセス権を規定するデータは、前記ユーザブロック領域の前記ユーザブロックに対するリード/ライトまたはリードオンのいずれかのアクセスを規定するデータである

ことを特徴とする請求項 10 に記載のアクセス方法。

【請求項 13】

1 以上のユーザブロックが所定の大きさのブロック単位で管理され、所定のシステムを提供するプロバイダの装置が利用するデータが記憶されるユーザブロック領域が形成される記憶手段を備える情報処理装置の前記ユーザブロック領域にアクセスするアクセス処理用のプログラムであって、

前記情報処理装置に呼びかけを行う呼びかけステップと、

前記呼びかけに応答した前記情報処理装置に対し、コマンドの送信を制御する送信制御ステップと、

前記情報処理装置により処理された前記コマンドに対する送信結果の受信を制御する受信制御ステップと、

前記送信結果を処理する処理ステップと

を含み、

前記送信データは、前記記憶手段の領域定義ブロックに記憶されている、前記ユーザブロックを規定するデータおよび前記ユーザブロックへのアクセス権を規定するデータに基づいて制御される前記ユーザブロックにアクセスするためのコマンドである

ことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体

【手続補正 3】

【補正対象書類名】明細書

【補正対象項目名】0001

【補正方法】変更

【補正の内容】

【0001】

【発明の属する技術分野】

本発明は、情報処理装置および情報処理方法、リーダー/ライターおよびアクセス方法、並びに記録媒体に関し、特に、所定の利用者からのコマンドを受信し、そのコマンドを処理し、処理の結果を送信する情報処理装置および情報処理方法、リーダー/ライターおよびアクセス方法、並びに記録媒体に関する。

【手続補正 4】

【補正対象書類名】明細書

【補正対象項目名】0012

【補正方法】変更

【補正の内容】

【0012】

【課題を解決するための手段】

請求項 1 に記載の 情報処理装置は、所定のシステムを提供するプロバイダの装置からのコマンドを受信する受信手段と、コマンドを処理する処理手段と、処理の結果を送信する送信手段と、プロバイダの装置が利用するデータを記憶するとともに、1以上のユーザブロックが所定の大きさの単位で管理されるユーザブロック領域と、ユーザブロックの利用を定義する領域定義ブロック領域が形成される記憶手段とを含み、領域定義ブロック領域は、複数の領域定義ブロックからなり、各々の領域定義ブロックには、利用するユーザブロックを規定するデータおよびユーザブロックへのアクセス権を規定するデータが記憶され、領域定義ブロック領域に記録されているデータを利用してコマンドが処理されることを特徴とする。

【手続補正 5】

【補正対象書類名】明細書

【補正対象項目名】0013

【補正方法】変更

【補正の内容】

【0013】

請求項 6 に記載の 情報処理方法は、プロバイダの装置からのコマンドを受信する受信ステップと、領域定義ブロック領域を利用して、コマンドを処理する処理ステップと、処理の結果を送信する送信ステップとを含むことを特徴とする。

【手続補正 6】

【補正対象書類名】明細書

【補正対象項目名】0014

【補正方法】変更

【補正の内容】

【0014】

請求項 7 に記載の 記録媒体のプログラムは、プロバイダの装置からのコマンドの受信を制御する受信制御ステップと、領域定義ブロック領域を利用して、コマンドを処理する処理ステップと、処理の結果の送信を制御する送信制御ステップとを含むことを特徴とする。

【手続補正 7】

【補正対象書類名】明細書

【補正対象項目名】 0015

【補正方法】 変更

【補正の内容】

【0015】

請求項8に記載のリーダ/ライタは、情報処理装置に送信する送信データおよび情報処理装置から受信した受信データの処理を行うデータ処理手段と、送信データを変調する変調手段と、受信データを復調する復調手段とを含み、送信データは、記憶手段の領域定義ブロックに記憶されている、ユーザブロックを規定するデータおよびユーザブロックへのアクセス権を規定するデータに基づいて制御されるユーザブロックにアクセスするためのコマンドであることを特徴とする。

【手続補正8】

【補正対象書類名】 明細書

【補正対象項目名】 0016

【補正方法】 変更

【補正の内容】

【0016】

請求項10に記載のアクセス方法は、情報処理装置に呼びかけを行う呼びかけステップと、呼びかけに回答した情報処理装置に対し、コマンドを送信する送信ステップと、情報処理装置により処理されたコマンドに対する送信結果を受信する受信ステップと、送信結果を処理する処理ステップとを含み、送信データは、記憶手段の領域定義ブロックに記憶されている、ユーザブロックを規定するデータおよびユーザブロックへのアクセス権を規定するデータに基づいて制御されるユーザブロックにアクセスするためのコマンドであることを特徴とする。

【手続補正9】

【補正対象書類名】 明細書

【補正対象項目名】 0017

【補正方法】 変更

【補正の内容】

【0017】

請求項13に記載の記録媒体のプログラムは、情報処理装置に呼びかけを行う呼びかけステップと、呼びかけに回答した情報処理装置に対し、コマンドの送信を制御する送信制御ステップと、情報処理装置により処理されたコマンドに対する送信結果の受信を制御する受信制御ステップと、送信結果を処理する処理ステップとを含み、送信データは、記憶手段の領域定義ブロックに記憶されている、ユーザブロックを規定するデータおよびユーザブロックへのアクセス権を規定するデータに基づいて制御されるユーザブロックにアクセスするためのコマンドであることを特徴とする。

【手続補正10】

【補正対象書類名】 明細書

【補正対象項目名】 0018

【補正方法】 削除

【補正の内容】

【手続補正11】

【補正対象書類名】 明細書

【補正対象項目名】 0019

【補正方法】 削除

【補正の内容】

【手続補正12】

【補正対象書類名】 明細書

【補正対象項目名】 0 0 2 0
【補正方法】 削除
【補正の内容】

【手続補正 1 3】
【補正対象書類名】 明細書
【補正対象項目名】 0 0 2 1
【補正方法】 削除
【補正の内容】

【手続補正 1 4】
【補正対象書類名】 明細書
【補正対象項目名】 0 0 2 2
【補正方法】 削除
【補正の内容】

【手続補正 1 5】
【補正対象書類名】 明細書
【補正対象項目名】 0 0 2 3
【補正方法】 削除
【補正の内容】

【手続補正 1 6】
【補正対象書類名】 明細書
【補正対象項目名】 0 0 2 4
【補正方法】 削除
【補正の内容】

【手続補正 1 7】
【補正対象書類名】 明細書
【補正対象項目名】 0 0 2 5
【補正方法】 削除
【補正の内容】

【手続補正 1 8】
【補正対象書類名】 明細書
【補正対象項目名】 0 0 2 6
【補正方法】 削除
【補正の内容】

【手続補正 1 9】
【補正対象書類名】 明細書
【補正対象項目名】 0 0 2 7
【補正方法】 削除
【補正の内容】

【手続補正 2 0】
【補正対象書類名】 明細書
【補正対象項目名】 0 0 2 8
【補正方法】 削除

【補正の内容】

【手続補正 2 1】

【補正対象書類名】明細書
【補正対象項目名】0029
【補正方法】削除
【補正の内容】

【手続補正 2 2】

【補正対象書類名】明細書
【補正対象項目名】0030
【補正方法】削除
【補正の内容】

【手続補正 2 3】

【補正対象書類名】明細書
【補正対象項目名】0031
【補正方法】削除
【補正の内容】

【手続補正 2 4】

【補正対象書類名】明細書
【補正対象項目名】0032
【補正方法】削除
【補正の内容】

【手続補正 2 5】

【補正対象書類名】明細書
【補正対象項目名】0033
【補正方法】変更
【補正の内容】

【0033】

請求項 1 に記載の情報処理装置においては、所定のシステムを提供するプロバイダの装置からのコマンドが受信され、コマンドが処理され、処理の結果が送信され、プロバイダの装置が利用するデータを記憶するとともに、1以上のユーザブロックが所定の大きさの単位で管理されるユーザブロック領域と、ユーザブロックの利用を定義する領域定義ブロック領域が形成される。領域定義ブロック領域は、複数の領域定義ブロックからなり、各々の領域定義ブロックには、利用するユーザブロックを規定するデータおよびユーザブロックへのアクセス権を規定するデータが記憶され、領域定義ブロック領域に記録されているデータを利用してコマンドが処理される。

【手続補正 2 6】

【補正対象書類名】明細書
【補正対象項目名】0034
【補正方法】変更
【補正の内容】

【0034】

請求項 6 に記載の情報処理方法および請求項 7 に記載の記録媒体においては、プロバイダの装置からのコマンドが受信され、領域定義ブロック領域を利用して、コマンドが処理され、処理の結果が送信される。

【手続補正 2 7】

【補正対象書類名】明細書

【補正対象項目名】0035

【補正方法】変更

【補正の内容】

【0035】

請求項8に記載のリーダ/ライタにおいては、情報処理装置に送信する送信データおよび情報処理装置から受信した受信データの処理が行われ、送信データが変調され、受信データが復調される。また、送信データは、記憶手段の領域定義ブロックに記憶されている、ユーザブロックを規定するデータおよびユーザブロックへのアクセス権を規定するデータに基づいて制御されるユーザブロックにアクセスするためのコマンドとされる。

【手続補正28】

【補正対象書類名】明細書

【補正対象項目名】0036

【補正方法】変更

【補正の内容】

【0036】

請求項10に記載のアクセス方法および請求項13に記載の記録媒体においては、情報処理装置に呼びかけが行われ、呼びかけに応答した情報処理装置に対し、コマンドが送信され、情報処理装置により処理されたコマンドに対する送信結果が受信され、送信結果が処理される。送信データは、記憶手段の領域定義ブロックに記憶されている、ユーザブロックを規定するデータおよびユーザブロックへのアクセス権を規定するデータに基づいて制御されるユーザブロックにアクセスするためのコマンドとされる。

【手続補正29】

【補正対象書類名】明細書

【補正対象項目名】0037

【補正方法】削除

【補正の内容】

【手続補正30】

【補正対象書類名】明細書

【補正対象項目名】0038

【補正方法】削除

【補正の内容】

【手続補正31】

【補正対象書類名】明細書

【補正対象項目名】0039

【補正方法】削除

【補正の内容】

【手続補正32】

【補正対象書類名】明細書

【補正対象項目名】0041

【補正方法】変更

【補正の内容】

【0041】

請求項1に記載の情報処理装置は、所定のシステムを提供するプロバイダの装置からのコマンドを受信する受信手段（例えば、図3のアンテナ53，RFインターフェース部61、およびBPSK復調回路62）と、コマンドを処理する処理手段（例えば、図3のシーケンス91）と、処理の結果を送信する送信手段（例えば、図3のアンテナ53，RFインター

フェース部 6 1、およびBPSK変調回路 6 8) と、プロバイダの装置が利用するデータを記憶するとともに、1以上のユーザブロックが所定の大きさの単位で管理されるユーザブロック領域と、ユーザブロックの利用を定義する領域定義ブロック領域が形成される記憶手段(例えば、図 3 のEEPROM 6 6) とを含み、領域定義ブロック領域は、複数の領域定義ブロックからなり、各々の領域定義ブロックには、利用するユーザブロックを規定するデータおよびユーザブロックへのアクセス権を規定するデータが記憶され、領域定義ブロック領域に記録されているデータを利用してコマンドが処理されることを特徴とする。

【手続補正 3 3】

【補正対象書類名】明細書

【補正対象項目名】0 0 4 2

【補正方法】変更

【補正の内容】

【0 0 4 2】

請求項 8 に記載のリーダ/ライタは、1以上のユーザブロックが所定の大きさのブロック単位で管理され、所定のシステムを提供するプロバイダの装置が利用するデータが記憶されるユーザブロック領域が形成される記憶手段(例えば、図 3 のEEPROM 6 6) を備える情報処理装置のユーザブロック領域にアクセスするリーダ/ライタであって、情報処理装置に送信する送信データおよび情報処理装置から受信した受信データの処理を行うデータ処理手段(例えば、図 2 のSPU 3 2) と、送信データを変調する変調手段(例えば、図 2 の変調回路 2 3) と、受信データを復調する復調手段(例えば、図 2 の復調回路 2 5) とを含み、送信データは、記憶手段の領域定義ブロックに記憶されている、ユーザブロックを規定するデータおよびユーザブロックへのアクセス権を規定するデータに基づいて制御されるユーザブロックにアクセスするためのコマンドであることを特徴とする。

【手続補正 3 4】

【補正対象書類名】明細書

【補正対象項目名】0 0 4 3

【補正方法】削除

【補正の内容】

【手続補正 3 5】

【補正対象書類名】明細書

【補正対象項目名】0 0 4 4

【補正方法】削除

【補正の内容】

【手続補正 3 6】

【補正対象書類名】明細書

【補正対象項目名】0 0 4 5

【補正方法】削除

【補正の内容】

【手続補正 3 7】

【補正対象書類名】明細書

【補正対象項目名】0 0 4 6

【補正方法】削除

【補正の内容】

【手続補正 3 8】

【補正対象書類名】明細書

【補正対象項目名】0 0 4 7

【補正方法】削除

【補正の内容】

【手続補正 39】

【補正対象書類名】明細書

【補正対象項目名】0218

【補正方法】変更

【補正の内容】

【0218】

【発明の効果】

以上のごとく、請求項1に記載の情報処理装置によれば、所定のシステムを提供するプロバイダの装置からのコマンドが受信され、コマンドが処理され、処理の結果が送信され、プロバイダの装置が利用するデータを記憶するとともに、1以上のユーザブロックが所定の大きさの単位で管理されるユーザブロック領域と、ユーザブロックの利用を定義する領域定義ブロック領域が形成される。また、領域定義ブロック領域は、複数の領域定義ブロックからなり、各々の領域定義ブロックには、利用するユーザブロックを規定するデータおよびユーザブロックへのアクセス権を規定するデータが記憶され、領域定義ブロック領域に記録されているデータを利用してコマンドが処理されるようにしたので、所定の利用者に対して、所定の記憶領域における複数のアクセス権を与えることができる。複数の利用者に、同一の記憶領域を割り当てることができる。複数の利用者に対して、所定の記憶領域における異なるアクセス権を与えることができる。

【手続補正 40】

【補正対象書類名】明細書

【補正対象項目名】0219

【補正方法】変更

【補正の内容】

【0219】

請求項6に記載の情報処理方法および請求項7に記載の記録媒体によれば、プロバイダの装置からのコマンドが受信され、領域定義ブロック領域を利用して、コマンドが処理され、処理の結果が送信されるようにしたので、所定の利用者に対して、所定の記憶領域における複数のアクセス権を与えることができる。複数の利用者に、同一の記憶領域を割り当てることができる。複数の利用者に対して、所定の記憶領域における異なるアクセス権を与えることができる。

【手続補正 41】

【補正対象書類名】明細書

【補正対象項目名】0220

【補正方法】変更

【補正の内容】

【0220】

請求項8に記載のリーダー/ライターによれば、情報処理装置に送信する送信データおよび情報処理装置から受信した受信データの処理が行われ、送信データが変調され、受信データが復調される。送信データは、記憶手段の領域定義ブロックに記憶されている、ユーザブロックを規定するデータおよびユーザブロックへのアクセス権を規定するデータに基づいて制御されるユーザブロックにアクセスするためのコマンドであるようにしたので、所定の利用者に対して、所定の記憶領域における複数のアクセス権を与えることができる。複数の利用者に、同一の記憶領域を割り当てることができる。複数の利用者に対して、所定の記憶領域における異なるアクセス権を与えることができる。

【手続補正 42】

【補正対象書類名】明細書

【補正対象項目名】0221

【補正方法】変更

【補正の内容】

【0221】

請求項10に記載のアクセス方法および請求項13の記録媒体によれば、情報処理装置に呼びかけが行われ、呼びかけに回答した情報処理装置に対し、コマンドが送信され、情報処理装置により処理されたコマンドに対する送信結果が受信され、送信結果が処理される。また、送信データは、記憶手段の領域定義ブロックに記憶されている、ユーザブロックを規定するデータおよびユーザブロックへのアクセス権を規定するデータに基づいて制御されるユーザブロックにアクセスするためのコマンドであるようにしたので、所定の利用者に対して、所定の記憶領域における複数のアクセス権を与えることができる。複数の利用者に、同一の記憶領域を割り当てることことができる。複数の利用者に対して、所定の記憶領域における異なるアクセス権を与えることができる。

【手続補正43】

【補正対象書類名】明細書

【補正対象項目名】0222

【補正方法】削除

【補正の内容】

【手続補正44】

【補正対象書類名】明細書

【補正対象項目名】0223

【補正方法】削除

【補正の内容】

【手続補正45】

【補正対象書類名】明細書

【補正対象項目名】0224

【補正方法】削除

【補正の内容】

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The step which detects the command from a predetermined user, and the 1st field which memorizes one or more users' data, The step which processes said command using the storage section which is used by said one or more users memorized to said 1st field, and in which the 2nd field set as the free area which is not used as said 1st field is formed, The information processing approach characterized by having the step which outputs the result of said processing.

[Claim 2] Said 2nd field is the information processing approach according to claim 1 which has one or more blocks and is characterized by memorizing the number corresponding to the block of the head of the field used by the user, and the number corresponding to the last block to said 1st field in said 2nd field as some said user's data.

[Claim 3] The step which detects the command from a predetermined user, and the 1st field which memorizes one or more users' data, The step which processes said command using the storage section which is used by said one or more users memorized to said 1st field, and in which the 2nd field set as the free area which is not used as said 1st field is formed, The transmission medium characterized by transmitting a program equipped with the step which outputs the result of said processing.

[Claim 4] A detection means to detect the input signal from the outside, and the 1st field which memorizes one or more users' data, A storage means used by said one or more users memorized to said 1st field by which the 2nd field set as the free area which is not used as said 1st field is formed, The information processor characterized by having a processing means to perform processing corresponding to said input signal using said storage means, and an output means to output the result of processing of said processing means outside.

[Claim 5] Said 2nd field is an information processor according to claim 4 which has one or more blocks and is characterized by memorizing the number corresponding to the block of the head of the field used by the user, and the number corresponding to the last block to said 1st field in said 2nd field as some said user's data.

[Claim 6] A detection means to detect the command from a predetermined user, and a processing means to process said command, An output means to output the result of processing of said processing means, and the 1st field which memorizes one or more users' data, In the information processing approach in an information processor equipped with a storage means used by said one or more users memorized to said 1st field by which the 2nd field managed per physical block of predetermined magnitude is formed The step at which said processing means assigns a logical-block number to the data memorized by said physical block, The information processing approach characterized by equipping said storage means with the step which memorizes the new data which have a predetermined logical-block number to physical blocks other than the physical block the data which have the logical-block number are remembered to be.

[Claim 7] Said 2nd field is the information processing approach according to claim 6 which has one or more physical blocks and is characterized by memorizing the number corresponding to the block of the head of the field used by the user, and the number corresponding to the last block to said 1st field in said

2nd field as some said user's data.

[Claim 8] The data memorized by said physical block have further the identification information which identifies the newness of data which has said same logical-block number. Said storage means The information processing approach according to claim 6 characterized by memorizing the new data which have a predetermined logical-block number with reference to the value of said identification information to physical blocks other than the physical block the newest data of the data which have said same logical-block number are remembered to be.

[Claim 9] Said identification information is the information processing approach according to claim 8 characterized by being the value of the counter at the time of the data storage which has the time of day or said logical-block number at the time of the data storage which has the value of the counter showing the number of updating of the data which have said logical-block number, and said logical-block number.

[Claim 10] A detection means to detect the command from a predetermined user, and a processing means to process said command, An output means to output the result of processing of said processing means, and the 1st field which memorizes one or more users' data, Are used by said one or more users memorized to said 1st field. In the transmission medium which transmits the program used for an information processor equipped with a storage means by which the 2nd field managed per physical block of predetermined magnitude is formed The step at which said processing means assigns a logical-block number to the data memorized by said physical block, The transmission medium characterized by transmitting the program said whose storage means is equipped with the step which memorizes the new data which have a predetermined logical-block number to physical blocks other than the physical block the data which have the logical-block number are remembered to be.

[Claim 11] A detection means to detect the command from a predetermined user, and a processing means to process said command, An output means to output the result of processing of said processing means, and the 1st field which memorizes one or more users' data, In an information processor equipped with a storage means used by said one or more users memorized to said 1st field by which the 2nd field managed per physical block of predetermined magnitude is formed Said processing means assigns a logical-block number to the data memorized by said physical block. Said storage means The information processor characterized by memorizing the new data which have a predetermined logical-block number to physical blocks other than the physical block the data which have the logical-block number are remembered to be.

[Claim 12] Said 2nd field is an information processor according to claim 11 which has one or more physical blocks and is characterized by memorizing the number corresponding to the block of the head of the field used by the user, and the number corresponding to the last block to said 1st field in said 2nd field as some said user's data.

[Claim 13] It is the information processor according to claim 11 which the data memorized by said physical block have further the identification information which identifies the newness of data which has said same logical-block number, and is characterized by for said storage means to memorize the new data which have a predetermined logical-block number with reference to the value of said identification information to physical blocks other than the physical block the newest data of the data which have said same logical-block number are remembered to be.

[Claim 14] Said identification information is an information processor according to claim 13 characterized by being the value of the counter at the time of the data storage which has the time of day or said logical-block number at the time of the data storage which has the value of the counter showing the number of updating of the data which have said logical-block number, and said logical-block number.

[Claim 15] A detection means to detect the command from a predetermined user, and a processing means to process said command, An output means to output the result of processing of said processing means, and the 1st field which memorizes one or more users' data, In the information processing approach in an information processor equipped with a storage means used by said one or more users memorized to said 1st field by which the 2nd field managed per block of predetermined magnitude is

formed The data of a predetermined block of said 2nd field The information processing approach characterized by having a recognition number, comparing the recognition number which said command with which said processing means was supplied by said user has with the recognition number which said data have, and having the step which processes said command corresponding to the comparison result. [Claim 16] Said 2nd field is the information processing approach according to claim 15 which has one or more blocks and is characterized by memorizing the number corresponding to the block of the head of the field used by the user, and the number corresponding to the last block to said 1st field in said 2nd field as some said user's data.

[Claim 17] A detection means to detect the command from a predetermined user, and a processing means to process said command, An output means to output the result of processing of said processing means, and the 1st field which memorizes one or more users' data, Are used by said one or more users memorized to said 1st field. In the transmission medium which transmits the program used for an information processor equipped with a storage means by which the 2nd field managed per block of predetermined magnitude is formed The data of a predetermined block of said 2nd field It is the transmission medium characterized by having a recognition number, and for said processing means comparing the recognition number which said command supplied by said user has with the recognition number which said data have, and transmitting a program equipped with the step which processes said command corresponding to the comparison result.

[Claim 18] A detection means to detect the command from a predetermined user, and a processing means to process said command, An output means to output the result of processing of said processing means, and the 1st field which memorizes one or more users' data, In an information processor equipped with a storage means used by said one or more users memorized to said 1st field by which the 2nd field managed per block of predetermined magnitude is formed The data of a predetermined block of said 2nd field It is the information processor characterized by having a recognition number, and for said processing means comparing the recognition number which said command supplied by said user has with the recognition number which said data have, and processing said command corresponding to the comparison result.

[Claim 19] Said 2nd field is an information processor according to claim 18 which has one or more blocks and is characterized by memorizing the number corresponding to the block of the head of the field used by the user, and the number corresponding to the last block to said 1st field in said 2nd field as some said user's data.

[Claim 20] A detection means to detect the command from a predetermined user, and a processing means to process said command, An output means to output the result of processing of said processing means, and the 1st field which memorizes one or more users' data, In the information processing approach in an information processor equipped with a storage means used by said one or more users memorized to said 1st field by which the 2nd field managed per block of predetermined magnitude is formed The step at which said processing means assigns the number corresponding to the sequence memorized to the data memorized by said block, In said storage means which memorized the number corresponding to the block of the head of the field which said user uses for said 1st field, and the number corresponding to the last block When the block which has the number of said last is a block of said last, The information processing approach characterized by having the step which memorizes new data to the block of said head, and memorizes said new data to the block next to the block which has the number of said last when the block which has the number of said last is not a block of said last.

[Claim 21] It is the information processing approach according to claim 20 characterized by not memorizing said new data when there is a block which has the same data as said new data.

[Claim 22] A detection means to detect the command from a predetermined user, and a processing means to process said command, An output means to output the result of processing of said processing means, and the 1st field which memorizes one or more users' data, Are used by said one or more users memorized to said 1st field. In the transmission medium which transmits the program used for an information processor equipped with a storage means by which the 2nd field managed per block of predetermined magnitude is formed The step at which said processing means assigns the number

corresponding to the sequence memorized to the data memorized by said block, In said storage means which memorized the number corresponding to the block of the head of the field which said user uses for said 1st field, and the number corresponding to the last block When the block which has the number of said last is a block of said last, The block which memorizes new data to the block of said head, and has the number of said last The transmission medium characterized by transmitting a program equipped with the step which memorizes said new data to the block next to the block which has the number of said last when it is not the block of said last.

[Claim 23] A detection means to detect the command from a predetermined user, and a processing means to process said command, An output means to output the result of processing of said processing means, and the 1st field which memorizes one or more users' data, It has a storage means used by said one or more users memorized to said 1st field by which the 2nd field managed per block of predetermined magnitude is formed. Said processing means The number corresponding to the sequence memorized is assigned to the data memorized by said block. Said storage means The number corresponding to the block of the head of the field which said user uses for said 1st field, and the number corresponding to the last block are memorized. When the block which has the number of said last is a block of said last, The information processor characterized by memorizing new data to the block of said head, and memorizing said new data to the block next to the block which has the number of said last when the block which has the number of said last is not a block of said last.

[Claim 24] It is the information processor according to claim 23 characterized by not memorizing said new data when there is a block which has the same data as said new data.

[Claim 25] The step which detects the command from a predetermined user, and the 1st field which memorizes one or more users' data, As opposed to a predetermined field [in / the 2nd field managed per block of the predetermined magnitude used by said one or more users memorized to said 1st field is formed, and / said 2nd field] And the information processing approach characterized by having the step which processes said command, and the step which outputs the result of said processing to each user using the storage section which memorizes two or more data which specify an access privilege different, respectively to said 1st field.

[Claim 26] The step which detects the command from a predetermined user, and the 1st field which memorizes one or more users' data, As opposed to a predetermined field [in / the 2nd field managed per block of the predetermined magnitude used by said one or more users memorized to said 1st field is formed, and / said 2nd field] And the transmission medium characterized by transmitting a program equipped with the step which processes said command, and the step which outputs the result of said processing to each user using the storage section which memorizes two or more data which specify an access privilege different, respectively to said 1st field.

[Claim 27] A detection means to detect the command from a predetermined user, and a processing means to process said command, An output means to output the result of processing of said processing means, and the 1st field which memorizes one or more users' data, It has a storage means used by said one or more users memorized to said 1st field by which the 2nd field managed per block of predetermined magnitude is formed. Said storage means The information processor characterized by memorizing two or more data which specify an access privilege different, respectively to said 1st field to each user as opposed to the predetermined field in said 2nd field.

[Claim 28] The step which detects the command from a predetermined user, and the 1st field which memorizes two or more users' data, Are used by said two or more users memorized to said 1st field. The 2nd field managed per block of predetermined magnitude is formed, and the storage section which memorizes the data with which two or more users use the predetermined field in said 2nd field jointly to said 1st field is used. The information processing approach characterized by having the step which processes said command, and the step which outputs the result of said processing.

[Claim 29] The step which detects the command from a predetermined user, and the 1st field which memorizes two or more users' data, Are used by said two or more users memorized to said 1st field. The 2nd field managed per block of predetermined magnitude is formed, and the storage section which memorizes the data with which two or more users use the predetermined field in said 2nd field jointly to

said 1st field is used. The transmission medium characterized by transmitting a program equipped with the step which processes said command, and the step which outputs the result of said processing.

[Claim 30] A detection means to detect the command from a predetermined user, and a processing means to process said command, An output means to output the result of processing of said processing means, and the 1st field which memorizes two or more users' data, It has a storage means used by said two or more users memorized to said 1st field by which the 2nd field managed per block of predetermined magnitude is formed. Said storage means The information processor characterized by memorizing the data with which two or more users use the predetermined field in said 2nd field jointly to said 1st field.

[Claim 31] The step which detects the command from a predetermined user, and the 1st field which memorizes two or more users' data, Are used by said two or more users memorized to said 1st field. A predetermined field [in / the 2nd field managed per block of predetermined magnitude is formed and / said 2nd field], And the information processing approach characterized by having the step which processes said command, and the step which outputs the result of said processing using the storage section which memorizes two or more data which specify the access privilege from which two or more users differ, respectively to said 1st field.

[Claim 32] The step which detects the command from a predetermined user, and the 1st field which memorizes two or more users' data, Are used by said two or more users memorized to said 1st field. A predetermined field [in / the 2nd field managed per block of predetermined magnitude is formed and / said 2nd field], And the transmission medium characterized by transmitting a program equipped with the step which processes said command, and the step which outputs the result of said processing using the storage section which memorizes two or more data which specify the access privilege from which two or more users differ, respectively to said 1st field.

[Claim 33] A detection means to detect the command from a predetermined user, and a processing means to process said command, An output means to output the result of processing of said processing means, and the 1st field which memorizes two or more users' data, It has a storage means used by said two or more users memorized to said 1st field by which the 2nd field managed per block of predetermined magnitude is formed. Said storage means The information processor characterized by memorizing the predetermined field in said 2nd field, and two or more data which specify the access privilege from which two or more users differ, respectively to said 1st field.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to a transmission medium at the information processing approach of receiving the command from a predetermined user in the information processing approach and an information processor, and a list especially, processing the command in them, and transmitting the result of processing to them about a transmission medium and an information processor, and a list.

[0002]

[Description of the Prior Art] The IC card (smart card) used by the cybermoney system or the security system is developed.

[0003] Such an IC card contains the memory which memorizes CPU which performs various processings, data required for processing, etc., and is transmitting and receiving data in the condition of having made predetermined reader/writer (R/W) contacting.

[0004] Moreover, there is also an IC card of the dc-battery loess mold which does not have the dc-battery in an IC card itself. Power is supplied to the IC card of such a dc-battery loess mold from R/W.

[0005]

[Problem(s) to be Solved by the Invention] However, in such an IC card, since it is premised on using it in the condition of having made R/W contacting, when using it by non-contact, it has the problem that it is difficult to acquire power.

[0006] Moreover, although how to supply power required for an IC card by the electromagnetic wave is also considered while transmitting and receiving data between an IC card and R/W by non-contact using an electromagnetic wave While having accessed the memory which an IC card contains in such an approach When the receive state of an electromagnetic wave becomes a defect, it has the problem that there is possibility that sufficient power will no longer be obtained and a defect will arise for the adjustment of the data in memory (memory colla tempestade PUSHON (Memory Corruption) arises).

[0007] Furthermore, if information is held like FAT (FileAllocation Table) of MS-DOS (Microsoft-Disc Operating System) in every [data are remembered to be] unit (it is a sector when it is MS-DOS), the field proportional to the area size data are remembered to be is needed for data control, and it has the problem that the use effectiveness of memory falls. Moreover, if a storage region is managed in the predetermined unit data are remembered to be, when memorizing the data of the magnitude with which the unit is not filled, the storage region which is not used occurs and it has further the problem that the use effectiveness of memory falls.

[0008] Furthermore, in the above-mentioned IC card, since uniform processing is performed to R/W, it has the problem that it is difficult to perform processing according to individual corresponding to two or more R/W.

[0009] The 1st field which this invention was made in view of such a situation, and memorizes two or more users' data, While using the storage section including the 2nd field which is used by two or more users memorized to the 1st field, and is managed per physical block of predetermined magnitude A logical-block number is assigned to the data memorized by the physical block. Memorize the data to

physical blocks other than the physical block the data which have the logical-block number are remembered to be, or The number corresponding to the sequence memorized is assigned to the data memorized by the physical block. When the physical block which has the last number is the last physical block, By memorizing the data to a top physical block, and memorizing the data to the next physical block of the physical block which has a number at the tail end, when the physical block which has the last number is not the last physical block Generating of memory colla tempestade PUSHON in memory is controlled logically.

[0010] Moreover, this invention is not the area size used by the user, but is the information on the amount proportional to the number of users (the number corresponding to a top physical block, and number corresponding to the last physical block), and enables it to manage data by holding the number corresponding to the physical block of the head of the field used by each user, and the number corresponding to the last physical block.

[0011] Furthermore, a predetermined field [in / on the above-mentioned storage section and / in this invention / the 2nd field], By and the thing for which the data which specify a predetermined field [in / for two or more data which specify an access privilege different, respectively / in memorizing to the 1st field **** / the 2nd field] corresponding to one user are memorized to the 1st field corresponding to two or more users It enables it to perform processing according to individual corresponding to two or more users (R/W).

[0012]

[Means for Solving the Problem] The step to which the information processing approach according to claim 1 detects the command from a predetermined user, Are used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field. It is characterized by having the step which processes a command using the storage section in which the 2nd field set as the free area which is not used as the 1st field is formed, and the step which outputs the result of processing.

[0013] The step to which a transmission medium according to claim 3 detects the command from a predetermined user, Are used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field. It is characterized by transmitting a program equipped with the step which processes a command using the storage section in which the 2nd field set as the free area which is not used as the 1st field is formed, and the step which outputs the result of processing.

[0014] A detection means by which an information processor according to claim 4 detects the input signal from the outside, A storage means by which the 2nd field which is used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field and which is set as the free area which is not used as the 1st field is formed, It is characterized by having a processing means to perform processing corresponding to an input signal using a storage means, and an output means to output the result of processing of a processing means outside.

[0015] The information processing approach according to claim 6 is characterized by equipping the step at which a processing means assigns a logical-block number to the data memorized by the physical block, and a storage means with the step memorized to physical blocks other than the physical block the data which have are remembered [data / which have a predetermined logical-block number / new] to be in the logical-block number.

[0016] A transmission medium according to claim 10 is characterized by transmitting the program which the step at which a processing means assigns a logical-block number to the data memorized by the physical block, and a storage means equip with the step memorized to physical blocks other than the physical block the data which have are remembered [data / which have a predetermined logical-block number / new] to be in the logical-block number.

[0017] An information processor according to claim 11 assigns a logical-block number to the data with which a processing means is memorized by the physical block, and a storage means is characterized by memorizing the new data which have a predetermined logical-block number to physical blocks other than the physical block the data which have the logical-block number are remembered to be.

[0018] The information processing approach according to claim 15 is characterized by for the data of a predetermined block of the 2nd field having a recognition number, comparing the recognition number

which the command with which the processing means was supplied by the user has with the recognition number which data have, and equipping them with the step which processes a command corresponding to the comparison result.

[0019] It is characterized by for the data of a predetermined block of the 2nd field having a recognition number in a transmission medium according to claim 17, and for a processing means comparing the recognition number which the command supplied by the user has with the recognition number which data have, and transmitting a program equipped with the step which processes a command corresponding to the comparison result.

[0020] It is characterized by for the data of a predetermined block of the 2nd field having a recognition number in an information processor according to claim 18, and for a processing means comparing the recognition number which the command supplied by the user has with the recognition number which data have, and processing a command corresponding to the comparison result.

[0021] The step at which, as for the information processing approach according to claim 20, a processing means assigns the number corresponding to the sequence memorized to the data memorized by block, In the storage means which memorized the number corresponding to the block of the head of the field which a user uses for the 1st field, and the number corresponding to the last block When the block which has the last number is the last block, new data It is characterized by having the step which memorizes to a top block, and memorizes new data to the block next to the block which has the last number when the block which has the last number is not the last block.

[0022] The step at which a transmission medium according to claim 22 assigns the number corresponding to the sequence memorized to the data with which a processing means is memorized by block, In the storage means which memorized the number corresponding to the block of the head of the field which a user uses for the 1st field, and the number corresponding to the last block When the block which has the last number is the last block, new data It is characterized by transmitting a program equipped with the step which memorizes to a top block, and memorizes new data to the block next to the block which has the last number when the block which has the last number is not the last block.

[0023] A detection means by which an information processor according to claim 23 detects the command from a predetermined user, A processing means to process a command, and an output means to output the result of processing of a processing means, Are used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field. It has a storage means by which the 2nd field managed per block of predetermined magnitude is formed. A processing means The number corresponding to the sequence memorized is assigned to the data memorized by block. A storage means The number corresponding to the block of the head of the field which a user uses for the 1st field, and the number corresponding to the last block are memorized. It is characterized by memorizing new data to a top block, when the block which has the last number is the last block, and memorizing new data to the block next to the block which has the last number, when the block which has the last number is not the last block.

[0024] The step to which the information processing approach according to claim 25 detects the command from a predetermined user, As opposed to a predetermined field [in / the 2nd field managed per block of the predetermined magnitude used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field is formed, and / the 2nd field] And it is characterized by having the step which processes a command, and the step which outputs the result of processing to each user using the storage section which memorizes two or more data which specify an access privilege different, respectively to the 1st field.

[0025] The step to which a transmission medium according to claim 26 detects the command from a predetermined user, As opposed to a predetermined field [in / the 2nd field managed per block of the predetermined magnitude used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field is formed, and / the 2nd field] And it is characterized by transmitting a program equipped with the step which processes a command, and the step which outputs the result of processing to each user using the storage section which memorizes two or more data which specify an access privilege different, respectively to the 1st field.

[0026] A detection means by which an information processor according to claim 27 detects the command from a predetermined user, A processing means to process a command, and an output means to output the result of processing of a processing means, Are used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field. It has a storage means by which the 2nd field managed per block of predetermined magnitude is formed. A storage means It is characterized by memorizing two or more data which specify an access privilege different, respectively to the 1st field to each user as opposed to the predetermined field in the 2nd field.

[0027] The step to which the information processing approach according to claim 28 detects the command from a predetermined user, Are used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. The 2nd field managed per block of predetermined magnitude is formed, and the storage section which memorizes the data with which two or more users use the predetermined field in the 2nd field jointly to the 1st field is used. It is characterized by having the step which processes a command, and the step which outputs the result of processing.

[0028] The step to which a transmission medium according to claim 29 detects the command from a predetermined user, Are used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. The 2nd field managed per block of predetermined magnitude is formed, and the storage section which memorizes the data with which two or more users use the predetermined field in the 2nd field jointly to the 1st field is used. It is characterized by transmitting a program equipped with the step which processes a command, and the step which outputs the result of processing.

[0029] A detection means by which an information processor according to claim 30 detects the command from a predetermined user, A processing means to process a command, and an output means to output the result of processing of a processing means, Are used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. It has a storage means by which the 2nd field managed per block of predetermined magnitude is formed, and a storage means is characterized by memorizing the data with which two or more users use the predetermined field in the 2nd field jointly to the 1st field.

[0030] The step to which the information processing approach according to claim 31 detects the command from a predetermined user, Are used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. The 2nd field managed per block of predetermined magnitude is formed, and the storage section which memorizes the predetermined field in the 2nd field and two or more data which specify the access privilege from which two or more users differ, respectively to the 1st field is used. It is characterized by having the step which processes a command, and the step which outputs the result of processing.

[0031] The step to which a transmission medium according to claim 32 detects the command from a predetermined user, Are used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. The 2nd field managed per block of predetermined magnitude is formed, and the storage section which memorizes the predetermined field in the 2nd field and two or more data which specify the access privilege from which two or more users differ, respectively to the 1st field is used. It is characterized by transmitting a program equipped with the step which processes a command, and the step which outputs the result of processing.

[0032] A detection means by which an information processor according to claim 33 detects the command from a predetermined user, A processing means to process a command, and an output means to output the result of processing of a processing means, Are used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. It has a storage means by which the 2nd field managed per block of predetermined magnitude is formed, and a storage means is characterized by memorizing the predetermined field in the 2nd field, and two or more data which specify the access privilege from which two or more users differ, respectively to the 1st field.

[0033] In the information processing approach according to claim 1, a transmission medium according to claim 3, and an information processor according to claim 4 A command is processed using the storage section in which the 2nd field set as the 1st field which memorizes one or more users' data, and the free

area which is used by one or more users memorized to the 1st field, and is not used as the 1st field is formed.

[0034] In the information processing approach according to claim 6, a transmission medium according to claim 10, and an information processor according to claim 11, a processing means assigns a logical-block number to the data memorized by the physical block, and the new data with which a storage means has a predetermined logical-block number are memorized to physical blocks other than the physical block the data which have the logical-block number are remembered to be.

[0035] In the information processing approach according to claim 15, a transmission medium according to claim 17, and an information processor according to claim 18, it has a recognition number, a processing means compares the recognition number which the command supplied by the user has with the recognition number which data have, and the data of a predetermined block of the 2nd field process a command corresponding to the comparison result.

[0036] In the information processing approach according to claim 20, a transmission medium according to claim 22, and an information processor according to claim 23 The number corresponding to the sequence memorized can be assigned to the data memorized by block with a processing means. With a storage means When the block which has the last number is the last block, new data are memorized by top block, and new data are memorized by the block next to the block which has the last number when the block which has the last number is not the last block.

[0037] In the information processing approach according to claim 25, a transmission medium according to claim 26, and an information processor according to claim 27 As opposed to a predetermined field [in / the 2nd field managed per block of the predetermined magnitude used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field is formed, and / the 2nd field] And a command is processed using the storage section which memorizes two or more data which specify an access privilege which is different to each user, respectively to the 1st field.

[0038] In the information processing approach according to claim 28, a transmission medium according to claim 29, and an information processor according to claim 30 The 1st field which memorizes two or more users' data, and the 2nd field which is used by two or more users memorized to the 1st field, and is managed per block of predetermined magnitude are formed, and it sets to the 1st field. A command is processed using the storage section which memorizes the data with which two or more users use the predetermined field in the 2nd field jointly to the 1st field.

[0039] In the information processing approach according to claim 31, a transmission medium according to claim 32, and an information processor according to claim 33 It is used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. The 2nd field managed per block of predetermined magnitude is formed, and a command is processed using the storage section which memorizes the predetermined field in the 2nd field, and two or more data which specify the access privilege from which two or more users differ, respectively to the 1st field.

[0040]

[Embodiment of the Invention] Although the gestalt of operation of this invention is explained below, it is as follows, when the gestalt (however, an example) of operation [/ in the parenthesis after each means] is added and the description of this invention is described, in order to clarify correspondence relation between each means of invention given in a claim, and the gestalt of the following operations. However, of course, this publication does not mean limiting to what indicated each means.

[0041] A detection means by which an information processor according to claim 4 detects the input signal from the outside (for example, BPSK demodulator circuit 62 of drawing 3), Are used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field. A storage means by which the 2nd field set as the free area which is not used as the 1st field is formed (for example, EEPROM66 of drawing 3), It is characterized by having a processing means (for example, sequencer 91 of drawing 3) to perform processing corresponding to an input signal using a storage means, and an output means (for example, BPSK modulation circuit 68 of drawing 3) to output the result of processing of a processing means outside.

[0042] A processing means (for example, sequencer 91 of drawing 3) assigns a logical-block number to

the data memorized by the physical block, and an information processor according to claim 11 is characterized by memorizing the new data with which a storage means (for example, EEPROM66 of drawing 3) has a predetermined logical-block number to physical blocks other than the physical block the data which have the logical-block number are remembered to be.

[0043] An information processor according to claim 18 is characterized by for the data of a predetermined block of the 2nd field having a recognition number, comparing the recognition number which the command with which the processing means (for example, sequencer 91 of drawing 3) was supplied by the user has with the recognition number which data have, and processing a command corresponding to the comparison result.

[0044] A detection means by which an information processor according to claim 23 detects the command from a predetermined user, A processing means (for example, sequencer 91 of drawing 3) to process a command, and an output means to output the result of processing of a processing means, Are used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field. It has a storage means (for example, EEPROM66 of drawing 3) by which the 2nd field managed per block of predetermined magnitude is formed. A processing means The number corresponding to the sequence memorized is assigned to the data memorized by block. A storage means The number corresponding to the block of the head of the field which a user uses for the 1st field, and the number corresponding to the last block are memorized. It is characterized by memorizing new data to a top block, when the block which has the last number is the last block, and memorizing new data to the block next to the block which has the last number, when the block which has the last number is not the last block.

[0045] A detection means by which an information processor according to claim 27 detects the command from a predetermined user, A processing means (for example, sequencer 91 of drawing 3) to process a command, and an output means to output the result of processing of a processing means, Are used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field. It has a storage means (for example, EEPROM66 of drawing 3) by which the 2nd field managed per block of predetermined magnitude is formed. A storage means It is characterized by memorizing two or more data which specify an access privilege different, respectively to the 1st field to each user as opposed to the predetermined field in the 2nd field.

[0046] A detection means by which an information processor according to claim 30 detects the command from a predetermined user (for example, BPSK demodulator circuit 62 of drawing 3), A processing means (for example, sequencer 91 of drawing 3) to process a command, and an output means to output the result of processing of a processing means (for example, BPSK modulation circuit 68 of drawing 3), Are used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. It has a storage means (for example, EEPROM66 of drawing 3) by which the 2nd field managed per block of predetermined magnitude is formed, and a storage means is characterized by memorizing the data with which two or more users use the predetermined field in the 2nd field jointly to the 1st field.

[0047] A detection means by which an information processor according to claim 33 detects the command from a predetermined user (for example, BPSK demodulator circuit 62 of drawing 3), A processing means (for example, sequencer 91 of drawing 3) to process a command, and an output means to output the result of processing of a processing means (for example, BPSK modulation circuit 68 of drawing 3), Are used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. It has a storage means (for example, EEPROM66 of drawing 3) by which the 2nd field managed per block of predetermined magnitude is formed. A storage means It is characterized by memorizing the predetermined field in the 2nd field, and two or more data which specify the access privilege from which two or more users differ, respectively to the 1st field.

[0048] Drawing 1 shows an example using R/W1 and IC card 2 of a non-contact card system. Using an electromagnetic wave, R/W1 and IC card 2 are non-contact, and transmit and receive data.

[0049] If R/W1 transmits a predetermined command to IC card 2, IC card 2 receives the command and is made as [perform / processing corresponding to the command].

[0050] If R/W1 transmits data to IC card 2, the command is received, and IC card 2 which is the gestalt of 1 operation of the information processor of this invention processes the received command, and is made as [transmit / to R/W1 / the response data corresponding to the processing result].

[0051] Moreover, it connects with a controller 3 through a predetermined interface (for example, RS-485A), and a predetermined control signal is supplied to R/W1 from a controller 3, and it is made as [process] according to the control signal.

[0052] Drawing 2 shows the configuration of R/W1.

[0053] In IC21 Processing of data The communication link with SPU (Signal Processing Unit)32 and the controller 3 which process the data received from the data and IC card 2 which are transmitted to DPU (Data Processing Unit)31 to perform and IC card 2 It is ** to SCC (Serial Communication Controller)33 which carries out, and processing of data. The ROM section 41 which has memorized *** information beforehand, The memory section 34 which consists of RAM sections 42 which memorize the data in the middle of processing temporarily is connected through the bus.

[0054] Moreover, the flash memory 22 which memorizes predetermined data is also connected to this bus.

[0055] DPU31 is made as [receive / from SPU32 / the response data received from IC card 2] while outputting the command transmitted to IC card 2 to SPU32.

[0056] SPU32 is made in the response data transmitted with IC card 2 as [perform / to reception and its data / from a demodulator circuit 25 / predetermined processing] while outputting to a modulation circuit 23, after performing predetermined processing (for example, BPSK (BiPhase Shift Keying) modulation (after-mentioned)) to the command transmitted to IC card 2.

[0057] A modulation circuit 23 is data to which the subcarrier of the predetermined frequency (for example, 13.56MHz) supplied from the oscillator 26 was supplied from SPU32, and is made as [output / through an antenna 27 / the generated modulated wave / carry out an ASK (AmplitudeShift Keying) modulation and / to IC card 2 / as an electromagnetic wave]. At this time, a modulation circuit 23 makes a modulation factor less than one, and performs an ASK modulation. That is, when data are a low level, it is made for the maximum amplitude of a modulated wave not to become zero.

[0058] The demodulator circuit 24 is made as [output / restore to the modulated wave (ASK modulated wave) which received through the antenna 27, and / to SPU32 / the data to which it restored].

[0059] Drawing 3 shows the example of a configuration of IC card 2. In this IC card 2, IC51 is made as [receive / through an antenna 53 / the modulated wave transmitted by R/W1]. In addition, a capacitor 52 constitutes LC circuit with an antenna 53, and is made as [side / with the electromagnetic wave of a predetermined frequency (carrier frequency)].

[0060] In IC51 RF interface section 61 While detecting the modulated wave (ASK modulated wave) which received through the antenna 53, getting over in the ASK recovery section 81 and outputting the data after a recovery to the BPSK demodulator circuit 62 and the PLL (Phase Locked Loop) section 63 At a voltage regulator 82, it is ASK **. The signal which the tone section 81 detected is stabilized and it is made as [supply / each circuit / as direct current power].

[0061] Moreover, RF interface section 61 oscillates the signal of the same frequency as the clock frequency of data in an oscillator circuit 83, and is made as [output / to the PLL section 63 / the signal].

[0062] And the ASK modulation section 84 of RF interface section 61 Corresponding to the data supplied from operation part 64, fluctuate the load of the antenna 53 as a power source of IC card 2. (For example, correspond to data and a predetermined switching element is made to turn on / turn off, and only when a switching element is an ON state, a predetermined load is connected to an antenna 53 at juxtaposition) By things The modulated wave which has received through an antenna 53 (when transmitting data from IC card 2) maximum amplitude of a modulated wave is fixed -- **** -- an ASK modulation is carried out and the modulation component is transmitted to R/W1 through an antenna 53 - - it is made like (the terminal voltage of the antenna 27 of R/W1 is fluctuated).

[0063] From the data supplied from the ASK recovery section 81, the PLL section 63 generates the clock signal which synchronized with the data, and is made as [output / to the BPSK demodulator

circuit 62 and the BPSK modulation circuit 68 / the clock signal].

[0064] The BPSK demodulator circuit 62 is made as [output / according to the clock signal supplied from the PLL section 63, / restore to the data and / to operation part 64 / the data to which it restored], when the BPSK modulation of the data to which it restored in the ASK recovery section 81 is carried out.

[0065] When the data supplied from the BPSK demodulator circuit 62 are enciphered, operation part 64 is made as [process / the data / by the sequencer 91 / as a command], after decrypting the data in a code / decode section 92. In addition, when data are not enciphered, the data supplied from the BPSK demodulator circuit 62 are directly supplied to a sequencer 91 without a code / decode section 92.

[0066] The sequencer 91 is made as [perform / processing corresponding to the supplied command]. For example, a sequencer 91 processes the data memorized by EEPROM66 at this time.

[0067] The parity operation part 93 of operation part 64 is made as [compute / a Reed Solomon code] as parity from the data memorized by EEPROM66 and the data memorized by EEPROM66.

[0068] Furthermore, operation part 64 is made as [output / to the BPSK modulation circuit 68 / the response data (data transmitted to R/W1) corresponding to the processing], after performing processing predetermined by the sequencer 91.

[0069] The BPSK modulation circuit 68 carries out the BPSK modulation of the data supplied from operation part 64 (after-mentioned), and is made as [output / to the ASK modulation section 84 of RF interface section 61 / the data after a modulation].

[0070] RAM67 is made as [memorize / the data in the middle of processing etc. / temporarily], when a sequencer 91 processes.

[0071] EEPROM (Electrically Erasable and Programmable ROM)66 is the memory of a non-volatile, and even after IC card 2 ends the communication link with R/W1 and an electric power supply stops, it is made as [memorize / data / continue]. The fundamental program required for a sequencer 91 to process the command from R/W1 is memorized by ROM65.

[0072] Drawing 4 shows an example of assignment of the memory of EEPROM66.

[0073] EEPROM66 has 40 bytes of 256 physical blocks. Each physical block consists of a total of 40 bytes of 32 bytes of data division (D00 thru/or D1f), 2 bytes of attribute section (AT1, AT2), and 6 bytes of parity section (P0 thru/or P5).

[0074] The physical block number ffH (H expresses the hexadecimal) of EEPROM66 is assigned to the system ID block. The system ID block has memorized the information about the security of IC card 2.

[0075] Next, the physical block is assigned to the common area definition block (Common Area Definition Block) (the 1st field) or the provider domain-defined block (Provider Area Definition Block) (the 1st field) one by one toward 00H from the physical block number fdH.

[0076] When IC card 2 is published by EEPROM66, those (provider) who offer the system using this IC card 2 with predetermined equipment (issue machine) are registered into it. An issue machine is 1 physical block per one provider, carries out sequential use of the provider domain-defined block toward 00H from the physical block number fdH, and registers a provider.

[0077] The common area definition block and the provider domain-defined block have memorized the information on the location of the storage region which a provider uses etc.

[0078] And the physical block which is not used as a system ID block, a common area definition block, and a provider domain-defined block is assigned to the user block (User Block) used by the provider.

[0079] Drawing 5 shows an example of the assignment of each data to a system ID block.

[0080] As for D00 thru/or D0f of data division, the manufacture ID at the time of manufacture of EEPROM66 (Manufacture ID) (IDm) is memorized. A field D00 thru/or D03, a field D04 or D07, a field D08 or D0b and field D0c thru/or D0f have memorized the IC code of EEPROM66, the code (Manufacture Equipment Code) of the manufacture machine which created EEPROM66, the manufacture date (Manufacture Date) of EEPROM66, and the manufacture serial number (Manufacture Serial Number) of EEPROM66, respectively.

[0081] By using this information on IDm, all IC cards 2 (EEPROM66) are discriminable. In addition, a manufacture date sets January 1, 2000 to 0000H, and makes it the days from January 1, 2000. In

addition, when a manufacture date is the 1990 set, a manufacture date is expressed as negative days from January 1, 2000 using a two's complement.

[0082] The issue ID (Issue ID) (IDi) when D10 thru/or D1f of data division publishes this ID card 2 is memorized. A field D10 thru/or D13, a field D14 or D17, a field D18 or D1b and field D1c thru/or 1f of the codes of the issue machine which published the category / group number which shows the category and group to whom IC card 2 belongs, and this IC card 2, the dates which published IC card 2, and the expiration dates of IC card 2 are memorized, respectively.

[0083] Drawing 6 shows the attribute section of a system ID block. The attribute section has memorized the number of providers registered. In case an issue machine registers one provider, one physical block is used for it and it updates the value of this attribute section then.

[0084] The value of the attribute section is set as zero at the time of manufacture, and when an issue machine registers a provider into IC card 2 after that, it updates the value of the attribute section by the number of providers registered.

[0085] The parity section of a system ID block has memorized the Reed Solomon code (RS sign) calculated by the parity operation part 93 from the value of each bit of data division and the attribute section. Therefore, the value of the parity section is recalculated whenever data division or the attribute section is updated.

[0086] Drawing 7 shows an example of a common area definition block and a provider domain-defined block. In addition, these blocks are beforehand written in by the issue machine, when IC card 2 is published.

[0087] The common area definition block has been arranged at the physical block number feH of EEPROM66, and has memorized a setup of the storage region (common area (Common Area)) (the 2nd field) used by all providers.

[0088] From the physical block number fdH of EEPROM66, the provider domain-defined block has been arranged toward 00H, is 1 physical block per one provider, and has memorized a provider's information.

[0089] As shown in drawing 7, the data division D00 of a domain-defined block (a common area definition block and provider domain-defined block) thru/or the fields D00 and D01 of D1f have memorized the provider code (Provider Code) which shows provider's class. In the common area definition block, the value of fields D00 and D01 is set to 0000H, and, in the provider domain-defined block, let the value of fields D00 and D01 be the value of either 0001H thru/or FFFFH.

[0090] The field D02 of the data division of a domain-defined block thru/or D05 have memorized the allocation table (Allocation Table) which consists of numbers BN1 (fields D04 and D05) (BN1 > BN0) of the next physical block of the number BN0 (fields D02 and D03) of the physical block of the head of the storage region (provider field (Provider Area)) (the 2nd field) which this provider uses, and the number of the physical block of an end. a provider field is set as the position (the physical block number BN0 -- or (BN 1-1)) of EEPROM66 except a system block (a system ID block, domain-defined block), as shown in drawing 8.

[0091] Thus, since the provider field is specified by BN0 and BN1, it is not the area size used by the provider (user), and for the information on the amount proportional to the number of providers, data can be managed and use effectiveness of memory can be made high.

[0092] The field D06 of the data division of a domain-defined block thru/or D09 have remembered the partition table (Partition Table) which consists of the block counts BRW (fields D08 and D09) of the read/write block in a random access field to be block count BRA (fields D06 and D07) of a random access field (after-mentioned) among the storage regions which a provider uses. At this time, block count BRA of a random access field is set as formula $BRA=0$ or the value with which are satisfied of formula $2xn \leq BRA \leq BN1 - BN0$ (n is the number of light buffers (after-mentioned)), and the block count BRW of a read/write block is set as $BRW=0$, when it is $BRA=0$, and when it is $BRA \neq 0$, it is set as the value with which are satisfied of formula $n \leq BRW \leq BRA - n$.

[0093] Field D0a of the data division of a domain-defined block and D0b have memorized several n of the light buffer of a random access field. A n light buffer is used when making coincidence memorize n

data to logical-block number 00H thru/or (00+n (hexadecimal display)) H of a random access field. In addition, when memorizing data to the physical block which has other logical-block numbers among random access fields, only one light buffer is used.

[0094] as mentioned above, according to a domain-defined block, it is shown in drawing 8 -- as -- the physical block number BN0 -- a field (a provider field or common area) is assigned to the provider specified in provider code, further, the physical block of the BRA individual of the field (a provider field or common area) is assigned to a random access field, and the remaining physical blocks are assigned to the sequential access field (after-mentioned) for or (BN 1-1).

[0095] Furthermore, according to the domain-defined block, as shown in drawing 8, the random access field is logically assigned to the read/write block of a BRW individual, the read-only block, and the n light buffer. In addition, a read/write block and physical blocks other than a light buffer are assigned to a read-only block.

[0096] Field D0c of the data division of a domain-defined block and D0d have memorized the Perth block permission which has the information on the access privilege to the Perth block (Purse Block) (after-mentioned) in the storage region (random access field) which this provider uses.

[0097] Drawing 9 shows an example of the Perth block permission.

[0098] The Perth block permission (16 bits, b0, or bf) is read to the Perth block, and shows authorization or the disapproval of an add instruction and a subtraction instruction.

[0099] The Perth block permission of a common area definition block has memorized whether the Perth block is used in the storage region (common area) set up with a common area definition block to Field (bit) bb. That is, in the case of bb=0, the Perth block is not used. In the case of bb=1, the Perth block is used. And especially the field (bit) of others in the Perth block permission of a common area definition block is not used. In addition, in the case of bb=1, the read/write block whose logical-block number is 00H is used as a Perth block.

[0100] Next, in the Perth block permission of a provider domain-defined block, it has memorized whether the Perth block is used in the storage region set up with this provider domain-defined block to the field b3. That is, in the case of b3=0, the Perth block is not used. In the case of b3=1, the Perth block is used. In addition, in the case of b3=1, the read/write block whose logical-block number is 00H is used as a Perth block.

[0101] And the propriety of an add instruction to the Perth block was memorized to the field b2, the propriety of a subtraction instruction to the Perth block was memorized to the field b1, and the propriety of read-out to the Perth block is memorized to the field b0 (in the case of bi=1 (1 i= 0, 2), the instruction is permitted and, in the case of bi=0, the instruction is not permitted). Moreover, it has memorized to Field bb whether the Perth block is used in the storage region set up with a common area definition block. In addition, the same value as bb of the Perth block permission of a common area definition block is memorized by bb.

[0102] Furthermore, the propriety of an add instruction to the Perth block was memorized to Field ba, the propriety of a subtraction instruction to the Perth block was memorized to the field b9, and the propriety of read-out to the Perth block is memorized to the field b8 (in the case of bi=1 (8 i= 9a), the instruction is permitted and, in the case of bi=0, the instruction is not permitted).

[0103] Field D0e of the data division of a domain-defined block of drawing 7 and D0f memorized the version number of the security key (a common key and provider key) used for encryption and a decryption at a provider's (R/W1) authentication, and a list, and a field D10 thru/or 1f of the security key are memorized.

[0104] In addition, when R/W1 polls, IC card 2 returns the version number of these two keys (a common key and provider key). Therefore, in authentication between R/W1 and IC card 2, the security key of two or more versions can be used properly.

[0105] And the attribute sections AT1 and AT2 of a domain-defined block are formed as a reserve, and especially information is not memorized. The parity section of a domain-defined block has memorized the parity (RS sign) calculated from the value of all the bits of data division and the attribute section.

[0106] Thus, the domain-defined block set up by the issue machine has memorized a provider code, an

allocation table, a partition table, the Perth block permission, the security key version, and the security key.

[0107] Drawing 10 shows an example of a user block. As mentioned above with reference to drawing 4, physical blocks other than a system ID block, a common area definition block, and a provider domain-defined block are used by the provider as a user block among the rooms of EEPROM66.

[0108] For example, if eight providers are registered when room consists of 256 blocks as shown in drawing 4, 246 (= 256-10) blocks of those other than the system block of a total of ten (= 1+1+8) individuals of a system ID block, a common area definition block, and eight provider domain-defined blocks will be used as a user block. Moreover, if 40 providers are registered, a system block will become a total of 42 (= 1+1+40) individuals, and the user block of 214 (= 256-42) individuals will be secured.

[0109] A user block is assigned to each provider according to the allocation table (drawing 7) of a domain-defined block. In addition, since a provider uses the user block currently assigned beforehand with reference to an allocation table, he does not access other than the field (a provider field or common area) assigned on the allocation table.

[0110] The user block of the field (a provider field or common area) assigned on the allocation table is assigned to the random access field and the sequential access field according to the above-mentioned partition table (drawing 7).

[0111] Furthermore, the user block of a random access field is used as either a read/write block, a read-only block and a light buffer, and the number of these blocks is set up as mentioned above according to the number of a partition table and light buffers.

[0112] Thus, the data division D00 thru/or D1f of a user block currently assigned is used according to processing by the provider to whom the user block is assigned.

[0113] The attribute section of a user block of a random access field has memorized the incremental counter (Incremental Counter) (bits bf and be) and the logical-block number (Bit bd thru/or b0), as shown in drawing 11.

[0114] A logical-block number and an incremental counter are used when accessing the user block of a random access field.

[0115] When reading the data memorized to the random access field, it is a logical-block number, and the data (physical block) to read are searched and the newest data are read with reference to the incremental counter of the data which have the logical-block number.

[0116] On the other hand, when memorizing data to a random access field, after using as a light buffer the physical block (after-mentioned) which became unnecessary with reference to the logical-block number and incremental counter of data which have already been memorized to the random access field, data are written in the light buffer.

[0117] In addition, when the Perth block permission of an above-mentioned domain-defined block is set up so that the Perth block may be used, the read/write block whose logical-block number is 00H is used as a Perth block.

[0118] The Perth block is used to read the value already memorized when performing addition and subtraction of data frequently when setting up the access privilege to data finely (since possibility that information will be revealed increases).

[0119] Drawing 12 shows an example of the Perth block. The data division D00 of the Perth block the field D00 of D1f thru/or D07 are used as Perth data division. The data division D00 of the Perth block the field D08 of D1f thru/or D0f have memorized Execution ID (Execution ID). In addition, the field D10 thru/or D1f of data division of the Perth block is set up read-only, although used as user data division.

[0120] The Perth data division have memorized predetermined data. Execution ID is referred to when the add instruction or subtraction instruction to the Perth block is executed, and it is compared with the execution ID contained in the add instruction or a subtraction instruction.

[0121] On the other hand, the attribute section of a user block of a sequential access field has memorized the lap round number (Bit bf thru/or b0), as shown in drawing 13. In the sequential access field, if data (sequentially) are memorized in an order from the physical block of the head of a field and data are

memorized to the physical block of the last of a field, data are again memorized in an order from the physical block of the head of a field (overwritten). The lap round number has memorized the sequence. [0122] Therefore, when accessing the user block of a sequential access field, while being used, when memorizing data to a sequential access field, sequential reference of the lap round number is carried out. And data are memorized by the next physical block of the physical block which has a lap round number at the tail end till then. At this time, the lap round number of the physical block data were remembered to be is set as the number which added 1 to the lap round number at the tail end till then.

[0123] In addition, a failure occurs in the middle of a store at the time of the last store, and when the parity error (physical memory colla tempestade PUSHON) has arisen in the physical block which has a lap round number at the tail end, new data are memorized by the physical block, for example. Moreover, new data are memorized by the physical block of the head of a sequential access field when the physical block which has a lap round number at the tail end is a physical block of a sequential access end-of-region rate.

[0124] As mentioned above, EEPROM66 is suitably used for each provider.

[0125] Next, with reference to the flow chart of drawing 14 , and the timing chart of drawing 15 , actuation of IC card 2 and R/W1 is explained.

[0126] First, a predetermined electromagnetic wave is emitted from an antenna 27, the loaded condition of an antenna 27 is supervised, IC card 2 approaches, and R/W1 corresponding to the provider registered into IC card 2 in step S1 stands by until change of loaded condition is detected. In addition, R/W1 emits the electromagnetic wave which carried out the ASK modulation by the data of a short predetermined pattern, and you may make it repeat the appeal to IC card 2 in step S1 until the response from IC card 2 is obtained in fixed time amount.

[0127] When R/W1 detects approach of IC card 2 in step S1 (time of day t0 of drawing 15), it progresses to step S2. SPU32 of R/W1 The square wave of a predetermined frequency (a clock frequency twice the frequency [for example,] of data) as shown in drawing 16 (a) is made into a subcarrier. By the data (command corresponding to the processing which IC card 2 is made to perform) (for example, data shown in drawing 16 (b)) transmitted to IC card 2, a BPSK modulation is performed and the generated modulated wave (BPSK modulating signal) (drawing 16 (c)) is outputted to a modulation circuit 23.

[0128] In addition, as shown in drawing 16 (c) using differential conversion at the time of a BPSK modulation A value makes the same thing as the last BPSK modulating signal ("1", "0" or "0", "1") a BPSK modulating signal, when the data of 0 appear. The value makes what reversed the phase of the last BPSK modulating signal (thing which made "0" reverse "1" and made "1" reverse "0") the BPSK modulating signal, when the data of 1 appear.

[0129] Thus, since it gets over to the original data also when a BPSK modulating signal is reversed by holding data by change of the phase of a modulated wave using differential conversion, when getting over, the need of considering the polarity of a modulated wave is lost.

[0130] And a modulation circuit 23 is the BPSK modulating signal, the ASK modulation of the predetermined subcarrier is carried out with less than (for example, 0.1) one modulation factor (= maximum amplitude of the maximum amplitude/subcarrier of a data signal), and the generated modulated wave (ASK modulated wave) is transmitted to IC card 2 through an antenna 27 (during the time of day t0 of drawing 15 thru/or time of day t1).

[0131] In addition, when not transmitting, the modulation circuit 23 is made as [generate / with the high level of the two level (high level and low level) of a digital signal / a modulated wave].

[0132] Next, in step S3, IC cards 2 are an antenna 53 and a capacitor 52, transform into an electrical signal a part of electromagnetic wave which the antenna 27 of R/W1 emitted, and output the electrical signal (modulated wave) to RF interface section 61 of IC51. and the ASK recovery section 81 of RF interface section 61 controls the dc component of the generated signal, extracts a data signal, and outputs the data signal to the BPSK demodulator circuit 62 and the PLL section 63 while it supplies the signal generated in the modulated wave by rectifying and carrying out smooth (namely, envelope detection -- carrying out) to a voltage regulator 82.

[0133] A voltage regulator 82 stabilizes the signal supplied from the ASK recovery section 81, generates direct current power, and supplies it to each circuit.

[0134] In addition, the terminal voltage V_0 of an antenna 53 is as follows, for example at this time.

$$V_0 = V_{10}(1 + kxV_s(t)) \cos(\omega t)$$

[0135] It is here, and in V_{10} , k shows a modulation factor and $V_s(t)$ shows the signal component for the amplitude of a carrier component, respectively.

[0136] Moreover, the value VLR of a low level in the electrical potential difference V_1 after rectification by the ASK recovery section 81 is as follows, for example.

$$VLR = V_{10}(1 + kx(-1)) - V_f$$

[0137] Here, V_f shows the voltage drop in the diode D of a rectifier circuit. Usually, V_f is about 0.7 volts.

[0138] And a voltage regulator 82 stabilizes rectification and the signal by which smooth was carried out by the ASK recovery section 81, and supplies it to each circuits including operation part 64 as direct current power. In addition, since the modulation factor k of a modulated wave is less than one, its voltage variation after rectification (difference of high level and a low level) is small. Therefore, a voltage regulator 82 can generate direct current power easily.

[0139] When a modulation factor k receives 5% of modulated wave so that V_{10} may become 3 volts or more, for example, the low-level electrical potential difference VLR after rectification It becomes more than 2.15 ($= 3 \times (1 - 0.05) - 0.7$) volts. A voltage regulator 82 While being able to supply electrical potential difference sufficient as a power source to each circuit, amplitude $2kxV_{10}$ (Peak-to-Peak value) of the alternating current component (data component) of the electrical potential difference V_1 after rectification It becomes more than 0.3 ($= 2 \times 0.05 \times 3$) volts, and the ASK recovery section 81 can restore to data by the sufficiently high S/N ratio.

[0140] Thus, when a modulation factor k uses less than one ASK modulated wave, while performing the communication link with a low (in high condition of a S/N ratio) error rate, direct current voltage sufficient as a power source is supplied to IC card 2.

[0141] And according to the clock signal supplied from the PLL section 63, the BPSK demodulator circuit 62 restores to the data signal (BPSK modulating signal) from the ASK recovery section 81, and outputs the data to which it restored to operation part 64.

[0142] Next, in step S4, when the data supplied from the BPSK demodulator circuit 62 are deciphered, after decrypting operation part 64 in a code / decode section 92, it supplies the data (command) to a sequencer 91, and performs processing corresponding to the command (during the time of day t_1 of drawing 15 thru/or time of day t_2). In addition, while the value had transmitted the data of 1, R/W1 is standing by, until it receives the answerback from this period 2, i.e., an IC card. Therefore, in this period, IC card 2 has received the modulated wave with fixed maximum amplitude.

[0143] Next, in step S5, the sequencer 91 of operation part 64 outputs data (data transmitted to R/W1), such as a processing result, to the BPSK modulation circuit 68. Like SPU32 of R/W1, after the BPSK modulation circuit 68 carries out the BPSK modulation of the data, it is outputted to the ASK modulation section 84 of RF interface section 61.

[0144] The ASK modulation section 84 and by fluctuating the load connected to the both ends of an antenna 53 according to data using a switching element The modulated wave which has received (in the time of transmission of IC card 2) the maximum amplitude of a modulated wave becomes fixed -- **** - - an ASK modulation is carried out according to the data to transmit, the terminal voltage of the antenna 27 of R/W1 is fluctuated according to it, and the data is transmitted to R/W1 (during the time of day t_2 of drawing 15 thru/or time of day t_3).

[0145] In step S6, as for the modulation circuit 23 of R/W1, the value is continuing transmission of the data of 1 (high-level) at the time of reception of the data from IC card 2. And a demodulator circuit 25 detects the data transmitted with IC card 2 from minute fluctuation (for example, dozens of microvolts) of the terminal voltage of the antenna 27 of IC card 2, and the antenna 27 combined in electromagnetism.

[0146] And it gets over and a demodulator circuit 25 outputs the generated digital data to SPU32, after amplifying the detected signal (ASK modulated wave) with the amplifier of high interest profit.

[0147] And in step S7, after restoring to the data (BPSK modulating signal), it outputs to DPU31 and, as for SPU32 of R/W1, DPU31 processes the data (during the time of day t3 of drawing 15 thru/or time of day t4).

[0148] Furthermore, in step S8, when it judges whether a communication link is ended according to a processing result and it is judged again that it communicates, to step S2, DPU31 of R/W1 is return, step S2, or step S7, and communicates the following data (command) (time of day t4 thru/or time of day t8 of drawing 15). On the other hand, when it is judged that a communication link is ended, R/W1 ends the communication link with IC card 2.

[0149] As mentioned above, using the ASK modulation whose modulation factor k is less than one, R/W1 transmits a predetermined command to IC card 2, and IC card 2 performs reception and processing corresponding to the command for the command, and it returns the data corresponding to the result of the processing to R/W1.

[0150] Next, the actuation when writing in data to EEPROM66 as an example of processing by IC card 2 in above-mentioned step S4 is explained with reference to the flow chart of drawing 17 thru/or drawing 21 .

[0151] First, with reference to the flow chart of drawing 17 thru/or drawing 19 , the actuation when writing data in the random access field of EEPROM66 is explained.

[0152] In step S21, a sequencer 91 judges whether it is that the physical block which writes in data is a read/write block (the Perth block is not included) (the block to a BRW individual is considered as a read/write block from BN0 at sequence as shown in drawing 8), and when it is judged that it is a read/write block, it progresses to step S22.

[0153] A sequencer 91 progresses to step S23 (drawing 18), when it judges whether it is using the Perth block (b3=1) and the Perth block is not being used with reference to the Perth block permission (drawing 9) of a provider domain-defined block which has the provider code of R/W1 (in the case of b3=0).

[0154] On the other hand, when it is judged that the Perth block is used in step S22, a sequencer 91 judges whether the read/write block which writes in whether the logical-block number of the data (it writes in) to memorize is 00H and data in step S24 has lapped with the Perth block, and when it is judged that the read/write block which writes in data has not lapped with the Perth block, it progresses to step S23.

[0155] When it is judged that the read/write block which writes in data has lapped with the Perth block, a sequencer 91 ends processing in step S25, after performing error processing.

[0156] Moreover, when it is judged that the physical block which writes in data in step S21 is not a read/write block, it progresses to step S26, the physical block in which a sequencer 91 writes data judges whether it is the Perth block, and when it is judged that it is the Perth block, it progresses to step S27.

[0157] When it is judged that the physical block which writes in data is not the Perth block, a sequencer 91 ends processing in step S28, after performing error processing.

[0158] In step S27, in a random access field, a sequencer 91 progresses to step S29, when the Perth block (physical block whose logical-block number is 00H) is looked for and the Perth block is discovered.

[0159] Since the store to the Perth block cannot be performed when the Perth block is not discovered at step S27, a sequencer 91 ends processing in step S30, after performing error processing.

[0160] or [next, / that the add instruction is permitted with reference to the Perth block permission of a provider domain-defined block in step S29 by the instruction (command) to the Perth block judging whether it is an add instruction, and a sequencer 91 progressing to step S31 when it is judged that it is an add instruction] (b2=1) -- it judges whether it is no.

[0161] And when a sequencer 91 judges that the add instruction to the Perth block is permitted at step S31, it progresses to step S23.

[0162] On the other hand, when it is judged that the add instruction to the Perth block is not permitted at step S31, a sequencer 91 ends processing, after performing error processing in step S32, without executing an add instruction (when it is b2=0).

[0163] Moreover, in step S29, when the instruction to as opposed to [when it is judged that the instruction to the Perth block is not an add instruction / progress to step S33 and] the Perth block in a sequencer 91 judges that it judges whether it is a subtraction instruction and is a subtraction instruction, it progresses to step S34.

[0164] And in step S34, a sequencer 91 judges whether it is that the subtraction instruction is permitted ($b1=1$) with reference to the Perth block permission of a provider domain-defined block, and when it is judged that the subtraction instruction to the Perth block is permitted, it progresses to step S23.

[0165] On the other hand, when it is judged that the subtraction instruction to the Perth block is not permitted at step S34, a sequencer 91 ends processing, after performing error processing in step S35, without executing a subtraction instruction (when it is $b1=0$).

[0166] Moreover, in step S33, when it is judged that the instruction to the Perth block is not a subtraction instruction, a sequencer 91 ends processing in step S36, after performing error processing.

[0167] Next, in step S23 of drawing 18, a sequencer 91 searches the physical block of a random access field, and looks for the physical block which has the same logical-block number as the logical-block number of the data which write in.

[0168] And in step S37, a sequencer 91 judges whether the number of the physical blocks discovered at step S23 is two. That is, in this system, the last data and the data before last are memorized at least about each logical block. And when memorizing still newer data, new data are memorized on the data before last (it may memorize on the data of other logical-block numbers before last). When two physical blocks of the same logical-block number exist, it progresses to step S38, and the value (00, 01, 10, or 11) of the incremental counter in the two physical blocks is read and compared.

[0169] And make a physical block with the large value of an incremental counter into the physical block (new physical block) new data are remembered to be, and let a physical block with the small value of an incremental counter be the physical block (old physical block) old data are remembered to be.

[0170] However, when the values of two incremental counters are 00 and 11, make into a new physical block the physical block whose value of an incremental counter is 00, and let the physical block whose value of an incremental counter is 11 be an old physical block.

[0171] A sequencer 91 memorizes the number (physical block number) of a new physical block to RAM67 as a variable Y between two physical blocks, and makes RAM67 memorize the number of an old physical block in step S39 as a variable W (number of the physical block used as a light block).

[0172] Thus, a sequencer 91 progresses to step S49, after making Variable Y and Variable W memorize.

[0173] On the other hand, when it is judged that the number of the physical blocks discovered at step S23 is not two in step S37, it progresses to step S40 and a sequencer 91 judges whether the number of the physical blocks discovered at step S23 is one. And when it is judged that it is one piece, it progresses to step S41.

[0174] In step S40, when a sequencer 91 judges that the number of the physical blocks discovered at step S23 is not one, after performing error processing, processing is ended in step S42.

[0175] As for saying [that the same logical block exists only in one piece], the data before last will not exist for a certain reason. Then, in this case, it is the physical block of other logical-block numbers, and the physical block (namely, physical block whose number of the physical blocks which have the same logical-block number is two) which has data the last and before last is searched, and the physical block before last of them is used as a light block. For this reason, in step S41, a sequencer 91 progresses to step S43, after making RAM67 memorize the number of the discovered physical block (one piece) as a variable Y.

[0176] In step S43, a sequencer 91 searches the physical block of a random access field, and looks for two physical blocks which have the same predetermined logical-block number (arbitration) (logical-block number unrelated to the logical-block number now made into the write-in object).

[0177] In addition, since it retrieves sequentially from logical-block number 00H when searching a physical block, a smaller number, then retrieval time can be shortened for the logical-block number of the data which perform write-in processing frequently.

[0178] In step S44 and a sequencer 91 It judges whether the physical block whose logical-block number is the two same pieces was discovered at step S43. When it is judged that it was discovered, the incremental counter of two physical blocks discovered by progressing to step S45 is referred to. It progresses to step S49 (drawing 19), after making RAM67 memorize the number of the physical block of the older one as a variable W (number of a light block) between two physical blocks.

[0179] On the other hand, when it is judged in step S44 that two physical blocks were not discovered at step S43, it progresses to step S46, and a sequencer 91 carries out sequential count of the parity of each physical block of a random access field, and looks for the physical block which has started the parity error as compared with the value memorized by the parity section of each physical block.

[0180] And when it judges whether there is any physical block which has started the parity error and it is judged that there is a physical block which has started the parity error, it progresses to step S47, and a sequencer 91 progresses to step S49, after making RAM67 memorize the number of the physical block as a variable W (number of a light block).

[0181] In step S46, when it is judged that there is no physical block which has started the parity error, a sequencer 91 ends processing in step S48, after performing error processing.

[0182] In step S49 of drawing 19 next, a sequencer 91 It judges whether the physical block which writes in data is the Perth block (physical block whose logical-block number is 00H). When it is judged that it is the Perth block, the execution ID of the instruction which progresses to step S50 and is performed to the Perth block When it judges whether it is the same as that of the execution ID of the physical block of the number memorized as a variable Y at step S39 or step S41 (drawing 12) and it is judged that it is the same, it judges that this instruction is already processed and processing is ended.

[0183] Thus, since IC card 2 does not process the command when R/W1 carries out the retry of the same command by using Execution ID and the command is already processed, the same command is not processed twice.

[0184] In step S50, when the instruction with which a sequencer 91 is performed to the Perth block in step S51 when the execution ID of the instruction performed to the Perth block judges that it is not the same as that of the execution ID of the physical block of the number memorized as a variable Y judges whether it is an add instruction and it is an add instruction, it progresses to step S52.

[0185] In step S52, a sequencer 91 reads the Perth data of the physical block of the number of Variable Y, calculates the sum of the Perth data and the data contained in the instruction performed to the Perth block, and uses the sum as the Perth data (new Perth data) in new block data. Thus, it progresses to step S54, after processing. In addition, let execution ID of the physical block of the number of Variable Y be the execution ID of new block data at this time. This prevents processing of a duplex.

[0186] On the other hand, when it is judged that the instruction performed to the Perth block is not an add instruction in step S51 (that is, it is a subtraction instruction), it progresses to step S53, and a sequencer 91 reads the Perth data of the physical block of the number of Variable Y, calculates the difference of the Perth data and the data contained in the instruction performed to the Perth block, and uses the difference as the Perth data (new Perth data) in new block data. Thus, it progresses to step S54, after processing. In addition, let execution ID of the physical block of the number of Variable Y be the execution ID of new block data at this time. This prevents processing of a duplex.

[0187] Moreover, in step S49, a sequencer 91 progresses to step S54, when the physical block which writes in data judges that it is not the Perth block (that is, it is a read/write block).

[0188] And in step S54, a sequencer 91 makes the number which added 1 to the value of the incremental counter of the physical block of the number of Variable Y the value of the incremental counter of new block data. However, when the value of the incremental counter of the physical block of the number of Variable Y is 11, a sequencer 91 sets the value of the incremental counter of new block data to 00.

[0189] Next, in step S55, a sequencer 91 makes the parity of the data newly written in the parity operation part 93, an incremental counter, and a logical-block number calculate, and makes the value of the parity the value of the parity section of new block data.

[0190] And a sequencer 91 makes the physical block (light buffer) of the number of the variable W memorized at either step S39, step S45 or step S47 memorize new block data (the newly memorized

data (for them to be the Perth data and Execution ID in the Perth block), its logical-block number, incremental counters, and such parity) in step S56.

[0191] As mentioned above, since the data of the same logical-block number as the logical-block number of the data are left behind to memory by choosing the physical block (light buffer) which remembers data to be a logical-block number using an incremental counter when a failure occurs in the midst of the store of data, logically, memory colla tempestade PUSHON does not occur.

[0192] Although the incremental counter was used with the gestalt of the above-mentioned implementation in order to distinguish the block with which new data are recorded among the same logical blocks of a random access field, it is also possible to, distinguish the block with which new data are recorded by [at the time of record] securing 4 bytes of field in a random access field, and making time of day (the date, time of day, or value of a counter) then, record to it absolutely for example.

[0193] Next, with reference to the flow chart of drawing 20 and drawing 21 , the actuation when writing data in the sequential access field of EEPROM66 is explained.

[0194] A sequencer 91 makes RAM67 memorize the number of the physical block of the head of a sequential access field as a variable Z in step S61.

[0195] Next, a sequencer 91 reads the lap round number of the physical block whose physical block number is Z, while making RAM67 memorize, it reads the lap round number of the physical block whose physical block number is Z+1, and RAM67 is made to memorize it as a variable B as a variable A in step S62.

[0196] And in step S63, a sequencer 91 judges whether the difference (A-B) of the value of Variable A and the value of Variable B is 1, judges that it is the physical block the physical block of the physical block number Z remembers the data which have a lap round number at the tail end to be when it is not 1, and progresses to step S66.

[0197] When a sequencer 91 judges whether the physical block number Z is the same as the number of the physical block of a sequential access end-of-region rate in step S64 when it is judged that the difference (A-B) of the value of Variable A and the value of Variable B is 1, and it is judged that it is the same, the data with which the physical block of a sequential access end-of-region rate has a lap round number at the tail end are judged to be the physical block to memorize, and it progresses to step S66.

[0198] In step S64, when the physical block number Z judges that it is not the same as that of the number of the physical block of a sequential access end-of-region rate, a sequencer 91 returns to step S62 in step S65, after only 1 makes the value of the variable Z which RAM67 was made to memorize increase. And processing of step S62 thru/or step S65 is repeated successively, changing the value (value of the physical block number to search) of Variable Z.

[0199] Thus, the tail end of the lap round number of the data memorized sequentially is discovered. And in step S66, a sequencer 91 performs the parity check of a block of the number (number of the physical block at the tail end of = lap round number) of Variable Z.

[0200] And in step S67, it judges whether the parity error has produced the sequencer 91 in the physical block, and when it is judged that the parity error has arisen, it progresses to step S68.

[0201] In step S68 a sequencer 91 The value of Variable Z judges whether it is the same as that of the number of the physical block of the head of a sequential access field. When it is judged that it is the same, the tail end of data (what has started the parity error does not contain) judges that it is the physical block of a sequential access end-of-region rate, and sets to step S70. It progresses to step S72 (drawing 21), after making RAM67 memorize the number of the physical block of a sequential access end-of-region rate as a new variable Y.

[0202] When it is judged that the value of Variable Z is not the same as that of the number of the physical block of the head of a sequential access field, after a sequencer 91 makes RAM67 memorize the value (Z-1) which computed the number of the physical block at the tail end of data by having subtracted 1 from the value of Variable Z, and computed it as a variable Y in step S71, it progresses to step S72.

[0203] On the other hand, in step S69, when it is judged that the parity error has not arisen at step S67, a sequencer 91 progresses to step S72, after making RAM67 memorize the number (value of Variable Z in

this case) of the physical block at the tail end of data as a variable Y.

[0204] Next, in step S72, when it judges whether a sequencer 91 has the number (value of Variable Y) of the physical block at the tail end of data, and the same number of the physical block of a sequential access end-of-region rate and it is judged that it is the same, it progresses to step S73.

[0205] And in step S73, it progresses to step S75, after a sequencer's 91 making the number of the physical block of the head of a sequential access field the number of the physical block which writes in new data and making RAM67 memorize by making the number into Variable W.

[0206] It progresses to step S75, after a sequencer's 91 making the number which added 1 to the value of Variable Y the number of the physical block which writes in new data, making the number Variable W in step S74 and making RAM67 memorize, when it is judged in step S72 that the number (value of Variable Y) of the physical block at the tail end of data and the number of the physical block of a sequential access end-of-region rate are not the same.

[0207] Next, in step S75, since the newly memorized data and the data which judge whether the physical block (data at the tail end) of the number of Variable Y is the same, and are newly memorized when the same are already memorized, a sequencer 91 ends processing.

[0208] When it is judged that the physical block (data at the tail end) of the data newly memorized on the other hand and the number of Variable Y is not the same, in step S76, a sequencer 91 reads the lap round number of the physical block of the number of Variable Y, and makes the number which added 1 to the value the lap round number of the data (new block data) newly memorized.

[0209] Next, in step S77, a sequencer 91 makes the parity operation part 93 calculate the parity of the data to memorize and a lap round number (new block data), and writes new block data in it in step S78 at the physical block of a number W.

[0210] Thus, since the data of a lap round number smaller than the lap round number of the data which were being written in in the midst of the store of new data when a failure occurred since the lap round number in the data memorized sequentially is retrieved sequentially and new data are memorized to the next physical block (or physical block of the head of a sequential access field) of the data at the tail end remain, logically, memory colla tempestade PUSHON is not generated.

[0211] As mentioned above, EEPROM66 is made as [control / generating of memory colla tempestade PUSHON] using the information on the attribute section while being able to offer a storage region independently to two or more providers.

[0212] In addition, the same user block can also be assigned to two or more providers. In that case, the same user block is assigned on the allocation table of the provider domain-defined block with which those providers (overlap provider) are registered. at this time, a different access privilege (read/write -- or read-only) for every provider can be set up to the same user block by setting up the partition table of a provider domain-defined block for every provider. Furthermore, a predetermined provider can write in data to the user data division (read-only to other providers) of the Perth block which other providers use by setting up so that the Perth block may not be used to a predetermined provider, and setting up so that the Perth block may be used to other providers.

[0213] Moreover, the value of field D0e of a domain-defined block, and D0f (field where the version number of a security key is usually memorized) By setting it as a predetermined value (for example, FFFFH), and memorizing a predetermined provider's provider code (a maximum of eight pieces) to the field D10 thru/or D1f of a domain-defined block further That provider (local common provider) can use the user block assigned on the allocation table of this domain-defined block as a common area.

[0214] Moreover, the access privilege to the user block can be set up for every local common provider by registering a local common provider into two domain-defined blocks which assign the same user block, and setting up a different access privilege for every domain-defined block.

[0215] Thus, corresponding to two or more providers (namely, R/W), processing according to individual can be performed by setting up an overlap provider and a local common provider.

[0216] In addition, this invention can be applied also when delivering and receiving a signal in the condition (contact) of having been combined physically besides in the case of delivering and receiving a signal through radio (non-contact). In the case of interruption of service, or the equipment which

operates by the cell, when the cell has been removed, data can be secured.

[0217] Moreover, the program which performs each above-mentioned processing is recorded on the transmission medium which consists of record media, such as a magnetic disk and CD-ROM, it provides for a user, or is transmitted to a user through transmission media, such as a network, is recorded on the transmission medium which consists of record media, such as a hard disk and solid-state memory, and can be made to use.

[0218]

[Effect of the Invention] According to the information processing approach according to claim 1, a transmission medium according to claim 3, and the information processor according to claim 4, like the above It is used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field. Since a command is processed using the storage section including the 2nd field set as the free area which is not used as the 1st field, use effectiveness of memory can be made high.

[0219] Since the new data which have a predetermined logical-block number were memorized to physical blocks other than the physical block the data which have the logical-block number are remembered to be while assigning the logical-block number to the data memorized by the physical block according to the information processing approach according to claim 6, the transmission medium according to claim 10, and the information processor according to claim 11, generating of memory colla tempestade PUSHON can be controlled logically.

[0220] Since the data of a predetermined block of the 2nd field compare the recognition number which has a recognition number and the command supplied by the user has with the recognition number which data have and it is made not to repeat and process the same command according to the information processing approach according to claim 15, a transmission medium according to claim 17, and the information processor according to claim 18, it can avoid carrying out multiple-times processing accidentally [command / each].

[0221] While assigning the number corresponding to the sequence memorized to the data memorized by block according to the information processing approach according to claim 20, a transmission medium according to claim 22, and the information processor according to claim 23 When the block which has the last number is a block of the last of the assigned field, Since new data were memorized to the top block, and new data were memorized to the block next to the block which has the last number when the block which has the last number was not the last block, generating of memory colla tempestade PUSHON can be controlled logically.

[0222] According to the information processing approach according to claim 25, a transmission medium according to claim 26, and the information processor according to claim 27 Since the command was processed using the storage section which memorizes the predetermined field in the 2nd field, and two or more data which specify an access privilege which is different to one user, respectively to the 1st field Two or more access privileges which can be set to a predetermined storage region can be granted to a predetermined user.

[0223] Since the command was processed using the storage section which memorizes the data with which two or more users use the predetermined field in the 2nd field jointly to the 1st field according to the information processing approach according to claim 28, the transmission medium according to claim 29, and the information processor according to claim 30, the same storage region can be assigned to two or more users.

[0224] According to the information processing approach according to claim 31, a transmission medium according to claim 32, and the information processor according to claim 33 Since the command was processed using the storage section which memorizes the predetermined field in the 2nd field, and two or more data which specify the access privilege from which two or more users differ, respectively to the 1st field A different access privilege in a predetermined storage region can be granted to two or more users.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

TECHNICAL FIELD

[Field of the Invention] This invention relates to a transmission medium at the information processing approach of receiving the command from a predetermined user in the information processing approach and an information processor, and a list especially, processing the command in them, and transmitting the result of processing to them about a transmission medium and an information processor, and a list.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art] The IC card (smart card) used by the cybermoney system or the security system is developed.

[0003] Such an IC card contains the memory which memorizes CPU which performs various processings, data required for processing, etc., and is transmitting and receiving data in the condition of having made predetermined reader/writer (R/W) contacting.

[0004] Moreover, there is also an IC card of the dc-battery loess mold which does not have the dc-battery in an IC card itself. Power is supplied to the IC card of such a dc-battery loess mold from R/W.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

EFFECT OF THE INVENTION

[Effect of the Invention] According to the information processing approach according to claim 1, a transmission medium according to claim 3, and the information processor according to claim 4, like the above It is used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field. Since a command is processed using the storage section including the 2nd field set as the free area which is not used as the 1st field, use effectiveness of memory can be made high.

[0219] Since the new data which have a predetermined logical-block number were memorized to physical blocks other than the physical block the data which have the logical-block number are remembered to be while assigning the logical-block number to the data memorized by the physical block according to the information processing approach according to claim 6, the transmission medium according to claim 10, and the information processor according to claim 11, generating of memory colla tempestade PUSHON can be controlled logically.

[0220] Since the data of a predetermined block of the 2nd field compare the recognition number which has a recognition number and the command supplied by the user has with the recognition number which data have and it is made not to repeat and process the same command according to the information processing approach according to claim 15, a transmission medium according to claim 17, and the information processor according to claim 18, it can avoid carrying out multiple-times processing accidentally [command / each].

[0221] While assigning the number corresponding to the sequence memorized to the data memorized by block according to the information processing approach according to claim 20, a transmission medium according to claim 22, and the information processor according to claim 23 When the block which has the last number is a block of the last of the assigned field, Since new data were memorized to the top block, and new data were memorized to the block next to the block which has the last number when the block which has the last number was not the last block, generating of memory colla tempestade PUSHON can be controlled logically.

[0222] According to the information processing approach according to claim 25, a transmission medium according to claim 26, and the information processor according to claim 27 Since the command was processed using the storage section which memorizes the predetermined field in the 2nd field, and two or more data which specify an access privilege which is different to one user, respectively to the 1st field Two or more access privileges which can be set to a predetermined storage region can be granted to a predetermined user.

[0223] Since the command was processed using the storage section which memorizes the data with which two or more users use the predetermined field in the 2nd field jointly to the 1st field according to the information processing approach according to claim 28, the transmission medium according to claim 29, and the information processor according to claim 30, the same storage region can be assigned to two or more users.

[0224] According to the information processing approach according to claim 31, a transmission medium according to claim 32, and the information processor according to claim 33 Since the command was processed using the storage section which memorizes the predetermined field in the 2nd field, and two

or more data which specify the access privilege from which two or more users differ, respectively to the 1st field A different access privilege in a predetermined storage region can be granted to two or more users.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention] However, in such an IC card, since it is premised on using it in the condition of having made R/W contacting, when using it by non-contact, it has the problem that it is difficult to acquire power.

[0006] Moreover, although how to supply power required for an IC card by the electromagnetic wave is also considered while transmitting and receiving data between an IC card and R/W by non-contact using an electromagnetic wave While having accessed the memory which an IC card contains in such an approach When the receive state of an electromagnetic wave becomes a defect, it has the problem that there is possibility that sufficient power will no longer be obtained and a defect will arise for the adjustment of the data in memory (memory colla tempestade PUSHON (Memory Corruption) arises).

[0007] Furthermore, if information is held like FAT (FileAllocation Table) of MS-DOS (Microsoft-Disc Operating System) in every [data are remembered to be] unit (it is a sector when it is MS-DOS), the field proportional to the area size data are remembered to be is needed for data control, and it has the problem that the use effectiveness of memory falls. Moreover, if a storage region is managed in the predetermined unit data are remembered to be, when memorizing the data of the magnitude with which the unit is not filled, the storage region which is not used occurs and it has further the problem that the use effectiveness of memory falls.

[0008] Furthermore, in the above-mentioned IC card, since uniform processing is performed to R/W, it has the problem that it is difficult to perform processing according to individual corresponding to two or more R/W.

[0009] The 1st field which this invention was made in view of such a situation, and memorizes two or more users' data, While using the storage section including the 2nd field which is used by two or more users memorized to the 1st field, and is managed per physical block of predetermined magnitude A logical-block number is assigned to the data memorized by the physical block. Memorize the data to physical blocks other than the physical block the data which have the logical-block number are remembered to be, or The number corresponding to the sequence memorized is assigned to the data memorized by the physical block. When the physical block which has the last number is the last physical block, By memorizing the data to a top physical block, and memorizing the data to the next physical block of the physical block which has a number at the tail end, when the physical block which has the last number is not the last physical block Generating of memory colla tempestade PUSHON in memory is controlled logically.

[0010] Moreover, this invention is not the area size used by the user, but is the information on the amount proportional to the number of users (the number corresponding to a top physical block, and number corresponding to the last physical block), and enables it to manage data by holding the number corresponding to the physical block of the head of the field used by each user, and the number corresponding to the last physical block.

[0011] Furthermore, a predetermined field [in / on the above-mentioned storage section and / in this invention / the 2nd field], By and the thing for which the data which specify a predetermined field [in / for two or more data which specify an access privilege different, respectively / in memorizing to the 1st

field **** / the 2nd field] corresponding to one user are memorized to the 1st field corresponding to two or more users It enables it to perform processing according to individual corresponding to two or more users (R/W).

[Translation done.]

*** NOTICES ***

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

MEANS

[Means for Solving the Problem] The step to which the information processing approach according to claim 1 detects the command from a predetermined user, Are used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field. It is characterized by having the step which processes a command using the storage section in which the 2nd field set as the free area which is not used as the 1st field is formed, and the step which outputs the result of processing.

[0013] The step to which a transmission medium according to claim 3 detects the command from a predetermined user, Are used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field. It is characterized by transmitting a program equipped with the step which processes a command using the storage section in which the 2nd field set as the free area which is not used as the 1st field is formed, and the step which outputs the result of processing.

[0014] A detection means by which an information processor according to claim 4 detects the input signal from the outside, A storage means by which the 2nd field which is used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field and which is set as the free area which is not used as the 1st field is formed, It is characterized by having a processing means to perform processing corresponding to an input signal using a storage means, and an output means to output the result of processing of a processing means outside.

[0015] The information processing approach according to claim 6 is characterized by equipping the step at which a processing means assigns a logical-block number to the data memorized by the physical block, and a storage means with the step memorized to physical blocks other than the physical block the data which have are remembered [data / which have a predetermined logical-block number / new] to be in the logical-block number.

[0016] A transmission medium according to claim 10 is characterized by transmitting the program which the step at which a processing means assigns a logical-block number to the data memorized by the physical block, and a storage means equip with the step memorized to physical blocks other than the physical block the data which have are remembered [data / which have a predetermined logical-block number / new] to be in the logical-block number.

[0017] An information processor according to claim 11 assigns a logical-block number to the data with which a processing means is memorized by the physical block, and a storage means is characterized by memorizing the new data which have a predetermined logical-block number to physical blocks other than the physical block the data which have the logical-block number are remembered to be.

[0018] The information processing approach according to claim 15 is characterized by for the data of a predetermined block of the 2nd field having a recognition number, comparing the recognition number which the command with which the processing means was supplied by the user has with the recognition number which data have, and equipping them with the step which processes a command corresponding to the comparison result.

[0019] It is characterized by for the data of a predetermined block of the 2nd field having a recognition number in a transmission medium according to claim 17, and for a processing means comparing the recognition number which the command supplied by the user has with the recognition number which

data have, and transmitting a program equipped with the step which processes a command corresponding to the comparison result.

[0020] It is characterized by for the data of a predetermined block of the 2nd field having a recognition number in an information processor according to claim 18, and for a processing means comparing the recognition number which the command supplied by the user has with the recognition number which data have, and processing a command corresponding to the comparison result.

[0021] The step at which, as for the information processing approach according to claim 20, a processing means assigns the number corresponding to the sequence memorized to the data memorized by block, In the storage means which memorized the number corresponding to the block of the head of the field which a user uses for the 1st field, and the number corresponding to the last block When the block which has the last number is the last block, new data It is characterized by having the step which memorizes to a top block, and memorizes new data to the block next to the block which has the last number when the block which has the last number is not the last block.

[0022] The step at which a transmission medium according to claim 22 assigns the number corresponding to the sequence memorized to the data with which a processing means is memorized by block, In the storage means which memorized the number corresponding to the block of the head of the field which a user uses for the 1st field, and the number corresponding to the last block When the block which has the last number is the last block, new data It is characterized by transmitting a program equipped with the step which memorizes to a top block, and memorizes new data to the block next to the block which has the last number when the block which has the last number is not the last block.

[0023] A detection means by which an information processor according to claim 23 detects the command from a predetermined user, A processing means to process a command, and an output means to output the result of processing of a processing means, Are used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field. It has a storage means by which the 2nd field managed per block of predetermined magnitude is formed. A processing means The number corresponding to the sequence memorized is assigned to the data memorized by block. A storage means The number corresponding to the block of the head of the field which a user uses for the 1st field, and the number corresponding to the last block are memorized. It is characterized by memorizing new data to a top block, when the block which has the last number is the last block, and memorizing new data to the block next to the block which has the last number, when the block which has the last number is not the last block.

[0024] The step to which the information processing approach according to claim 25 detects the command from a predetermined user, As opposed to a predetermined field [in / the 2nd field managed per block of the predetermined magnitude used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field is formed, and / the 2nd field] And it is characterized by having the step which processes a command, and the step which outputs the result of processing to each user using the storage section which memorizes two or more data which specify an access privilege different, respectively to the 1st field.

[0025] The step to which a transmission medium according to claim 26 detects the command from a predetermined user, As opposed to a predetermined field [in / the 2nd field managed per block of the predetermined magnitude used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field is formed, and / the 2nd field] And it is characterized by transmitting a program equipped with the step which processes a command, and the step which outputs the result of processing to each user using the storage section which memorizes two or more data which specify an access privilege different, respectively to the 1st field.

[0026] A detection means by which an information processor according to claim 27 detects the command from a predetermined user, A processing means to process a command, and an output means to output the result of processing of a processing means, Are used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field. It has a storage means by which the 2nd field managed per block of predetermined magnitude is formed. A storage means It is characterized by memorizing two or more data which specify an access privilege different, respectively

to the 1st field to each user as opposed to the predetermined field in the 2nd field.

[0027] The step to which the information processing approach according to claim 28 detects the command from a predetermined user, Are used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. The 2nd field managed per block of predetermined magnitude is formed, and the storage section which memorizes the data with which two or more users use the predetermined field in the 2nd field jointly to the 1st field is used. It is characterized by having the step which processes a command, and the step which outputs the result of processing.

[0028] The step to which a transmission medium according to claim 29 detects the command from a predetermined user, Are used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. The 2nd field managed per block of predetermined magnitude is formed, and the storage section which memorizes the data with which two or more users use the predetermined field in the 2nd field jointly to the 1st field is used. It is characterized by transmitting a program equipped with the step which processes a command, and the step which outputs the result of processing.

[0029] A detection means by which an information processor according to claim 30 detects the command from a predetermined user, A processing means to process a command, and an output means to output the result of processing of a processing means, Are used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. It has a storage means by which the 2nd field managed per block of predetermined magnitude is formed, and a storage means is characterized by memorizing the data with which two or more users use the predetermined field in the 2nd field jointly to the 1st field.

[0030] The step to which the information processing approach according to claim 31 detects the command from a predetermined user, Are used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. The 2nd field managed per block of predetermined magnitude is formed, and the storage section which memorizes the predetermined field in the 2nd field and two or more data which specify the access privilege from which two or more users differ, respectively to the 1st field is used. It is characterized by having the step which processes a command, and the step which outputs the result of processing.

[0031] The step to which a transmission medium according to claim 32 detects the command from a predetermined user, Are used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. The 2nd field managed per block of predetermined magnitude is formed, and the storage section which memorizes the predetermined field in the 2nd field and two or more data which specify the access privilege from which two or more users differ, respectively to the 1st field is used. It is characterized by transmitting a program equipped with the step which processes a command, and the step which outputs the result of processing.

[0032] A detection means by which an information processor according to claim 33 detects the command from a predetermined user, A processing means to process a command, and an output means to output the result of processing of a processing means, Are used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. It has a storage means by which the 2nd field managed per block of predetermined magnitude is formed, and a storage means is characterized by memorizing the predetermined field in the 2nd field, and two or more data which specify the access privilege from which two or more users differ, respectively to the 1st field.

[0033] In the information processing approach according to claim 1, a transmission medium according to claim 3, and an information processor according to claim 4 A command is processed using the storage section in which the 2nd field set as the 1st field which memorizes one or more users' data, and the free area which is used by one or more users memorized to the 1st field, and is not used as the 1st field is formed.

[0034] In the information processing approach according to claim 6, a transmission medium according to claim 10, and an information processor according to claim 11, a processing means assigns a logical-block number to the data memorized by the physical block, and the new data with which a storage means has a predetermined logical-block number are memorized to physical blocks other than the

physical block the data which have the logical-block number are remembered to be.

[0035] In the information processing approach according to claim 15, a transmission medium according to claim 17, and an information processor according to claim 18, it has a recognition number, a processing means compares the recognition number which the command supplied by the user has with the recognition number which data have, and the data of a predetermined block of the 2nd field process a command corresponding to the comparison result.

[0036] In the information processing approach according to claim 20, a transmission medium according to claim 22, and an information processor according to claim 23 The number corresponding to the sequence memorized can be assigned to the data memorized by block with a processing means. With a storage means When the block which has the last number is the last block, new data are memorized by top block, and new data are memorized by the block next to the block which has the last number when the block which has the last number is not the last block.

[0037] In the information processing approach according to claim 25, a transmission medium according to claim 26, and an information processor according to claim 27 As opposed to a predetermined field [in / the 2nd field managed per block of the predetermined magnitude used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field is formed, and / the 2nd field] And a command is processed using the storage section which memorizes two or more data which specify an access privilege which is different to each user, respectively to the 1st field.

[0038] In the information processing approach according to claim 28, a transmission medium according to claim 29, and an information processor according to claim 30 The 1st field which memorizes two or more users' data, and the 2nd field which is used by two or more users memorized to the 1st field, and is managed per block of predetermined magnitude are formed, and it sets to the 1st field. A command is processed using the storage section which memorizes the data with which two or more users use the predetermined field in the 2nd field jointly to the 1st field.

[0039] In the information processing approach according to claim 31, a transmission medium according to claim 32, and an information processor according to claim 33 It is used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. The 2nd field managed per block of predetermined magnitude is formed, and a command is processed using the storage section which memorizes the predetermined field in the 2nd field, and two or more data which specify the access privilege from which two or more users differ, respectively to the 1st field.

[0040]

[Embodiment of the Invention] Although the gestalt of operation of this invention is explained below, it is as follows, when the gestalt (however, an example) of operation [/ in the parenthesis after each means] is added and the description of this invention is described, in order to clarify correspondence relation between each means of invention given in a claim, and the gestalt of the following operations. However, of course, this publication does not mean limiting to what indicated each means.

[0041] A detection means by which an information processor according to claim 4 detects the input signal from the outside (for example, BPSK demodulator circuit 62 of drawing 3), Are used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field. A storage means by which the 2nd field set as the free area which is not used as the 1st field is formed (for example, EEPROM66 of drawing 3), It is characterized by having a processing means (for example, sequencer 91 of drawing 3) to perform processing corresponding to an input signal using a storage means, and an output means (for example, BPSK modulation circuit 68 of drawing 3) to output the result of processing of a processing means outside.

[0042] A processing means (for example, sequencer 91 of drawing 3) assigns a logical-block number to the data memorized by the physical block, and an information processor according to claim 11 is characterized by memorizing the new data with which a storage means (for example, EEPROM66 of drawing 3) has a predetermined logical-block number to physical blocks other than the physical block the data which have the logical-block number are remembered to be.

[0043] An information processor according to claim 18 is characterized by for the data of a predetermined block of the 2nd field having a recognition number, comparing the recognition number

which the command with which the processing means (for example, sequencer 91 of drawing 3) was supplied by the user has with the recognition number which data have, and processing a command corresponding to the comparison result.

[0044] A detection means by which an information processor according to claim 23 detects the command from a predetermined user, A processing means (for example, sequencer 91 of drawing 3) to process a command, and an output means to output the result of processing of a processing means, Are used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field. It has a storage means (for example, EEPROM66 of drawing 3) by which the 2nd field managed per block of predetermined magnitude is formed. A processing means The number corresponding to the sequence memorized is assigned to the data memorized by block. A storage means The number corresponding to the block of the head of the field which a user uses for the 1st field, and the number corresponding to the last block are memorized. It is characterized by memorizing new data to a top block, when the block which has the last number is the last block, and memorizing new data to the block next to the block which has the last number, when the block which has the last number is not the last block.

[0045] A detection means by which an information processor according to claim 27 detects the command from a predetermined user, A processing means (for example, sequencer 91 of drawing 3) to process a command, and an output means to output the result of processing of a processing means, Are used by one or more users memorized to the 1st field which memorizes one or more users' data, and the 1st field. It has a storage means (for example, EEPROM66 of drawing 3) by which the 2nd field managed per block of predetermined magnitude is formed. A storage means It is characterized by memorizing two or more data which specify an access privilege different, respectively to the 1st field to each user as opposed to the predetermined field in the 2nd field.

[0046] A detection means by which an information processor according to claim 30 detects the command from a predetermined user (for example, BPSK demodulator circuit 62 of drawing 3), A processing means (for example, sequencer 91 of drawing 3) to process a command, and an output means to output the result of processing of a processing means (for example, BPSK modulation circuit 68 of drawing 3), Are used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. It has a storage means (for example, EEPROM66 of drawing 3) by which the 2nd field managed per block of predetermined magnitude is formed, and a storage means is characterized by memorizing the data with which two or more users use the predetermined field in the 2nd field jointly to the 1st field.

[0047] A detection means by which an information processor according to claim 33 detects the command from a predetermined user (for example, BPSK demodulator circuit 62 of drawing 3), A processing means (for example, sequencer 91 of drawing 3) to process a command, and an output means to output the result of processing of a processing means (for example, BPSK modulation circuit 68 of drawing 3), Are used by two or more users memorized to the 1st field which memorizes two or more users' data, and the 1st field. It has a storage means (for example, EEPROM66 of drawing 3) by which the 2nd field managed per block of predetermined magnitude is formed. A storage means It is characterized by memorizing the predetermined field in the 2nd field, and two or more data which specify the access privilege from which two or more users differ, respectively to the 1st field.

[0048] Drawing 1 shows an example using R/W1 and IC card 2 of a non-contact card system. Using an electromagnetic wave, R/W1 and IC card 2 are non-contact, and transmit and receive data.

[0049] If R/W1 transmits a predetermined command to IC card 2, IC card 2 receives the command and is made as [perform / processing corresponding to the command].

[0050] If R/W1 transmits data to IC card 2, the command is received, and IC card 2 which is the gestalt of 1 operation of the information processor of this invention processes the received command, and is made as [transmit / to R/W1 / the response data corresponding to the processing result].

[0051] Moreover, it connects with a controller 3 through a predetermined interface (for example, RS-485A), and a predetermined control signal is supplied to R/W1 from a controller 3, and it is made as [process] according to the control signal.

[0052] Drawing 2 shows the configuration of R/W1.

[0053] In IC21 Processing of data The communication link with SPU (Signal Processing Unit)32 and the controller 3 which process the data received from the data and IC card 2 which are transmitted to DPU (Data Processing Unit)31 to perform and IC card 2 It is ** to SCC (Serial Communication Controller)33 which carries out, and processing of data. The ROM section 41 which has memorized **** information beforehand, The memory section 34 which consists of RAM sections 42 which memorize the data in the middle of processing temporarily is connected through the bus.

[0054] Moreover, the flash memory 22 which memorizes predetermined data is also connected to this bus.

[0055] DPU31 is made as [receive / from SPU32 / the response data received from IC card 2] while outputting the command transmitted to IC card 2 to SPU32.

[0056] SPU32 is made in the response data transmitted with IC card 2 as [perform / to reception and its data / from a demodulator circuit 25 / predetermined processing] while outputting to a modulation circuit 23, after performing predetermined processing (for example, BPSK (BiPhase Shift Keying) modulation (after-mentioned)) to the command transmitted to IC card 2.

[0057] A modulation circuit 23 is data to which the subcarrier of the predetermined frequency (for example, 13.56MHz) supplied from the oscillator 26 was supplied from SPU32, and is made as [output / through an antenna 27 / the generated modulated wave / carry out an ASK (AmplitudeShift Keying) modulation and / to IC card 2 / as an electromagnetic wave]. At this time, a modulation circuit 23 makes a modulation factor less than one, and performs an ASK modulation. That is, when data are a low level, it is made for the maximum amplitude of a modulated wave not to become zero.

[0058] The demodulator circuit 24 is made as [output / restore to the modulated wave (ASK modulated wave) which received through the antenna 27, and / to SPU32 / the data to which it restored].

[0059] Drawing 3 shows the example of a configuration of IC card 2. In this IC card 2, IC51 is made as [receive / through an antenna 53 / the modulated wave transmitted by R/W1]. In addition, a capacitor 52 constitutes LC circuit with an antenna 53, and is made as [side / with the electromagnetic wave of a predetermined frequency (carrier frequency)].

[0060] In IC51 RF interface section 61 While detecting the modulated wave (ASK modulated wave) which received through the antenna 53, getting over in the ASK recovery section 81 and outputting the data after a recovery to the BPSK demodulator circuit 62 and the PLL (Phase Locked Loop) section 63 At a voltage regulator 82, it is ASK **. The signal which the tone section 81 detected is stabilized and it is made as [supply / each circuit / as direct current power].

[0061] Moreover, RF interface section 61 oscillates the signal of the same frequency as the clock frequency of data in an oscillator circuit 83, and is made as [output / to the PLL section 63 / the signal].

[0062] And the ASK modulation section 84 of RF interface section 61 Corresponding to the data supplied from operation part 64, fluctuate the load of the antenna 53 as a power source of IC card 2. (For example, correspond to data and a predetermined switching element is made to turn on / turn off, and only when a switching element is an ON state, a predetermined load is connected to an antenna 53 at juxtaposition) By things The modulated wave which has received through an antenna 53 (when transmitting data from IC card 2) maximum amplitude of a modulated wave is fixed -- **** -- an ASK modulation is carried out and the modulation component is transmitted to R/W1 through an antenna 53 - - it is made like (the terminal voltage of the antenna 27 of R/W1 is fluctuated).

[0063] From the data supplied from the ASK recovery section 81, the PLL section 63 generates the clock signal which synchronized with the data, and is made as [output / to the BPSK demodulator circuit 62 and the BPSK modulation circuit 68 / the clock signal].

[0064] The BPSK demodulator circuit 62 is made as [output / according to the clock signal supplied from the PLL section 63, / restore to the data and / to operation part 64 / the data to which it restored], when the BPSK modulation of the data to which it restored in the ASK recovery section 81 is carried out.

[0065] When the data supplied from the BPSK demodulator circuit 62 are enciphered, operation part 64

is made as [process / the data / by the sequencer 91 / as a command], after decrypting the data in a code / decode section 92. In addition, when data are not enciphered, the data supplied from the BPSK demodulator circuit 62 are directly supplied to a sequencer 91 without a code / decode section 92.

[0066] The sequencer 91 is made as [perform / processing corresponding to the supplied command]. For example, a sequencer 91 processes the data memorized by EEPROM66 at this time.

[0067] The parity operation part 93 of operation part 64 is made as [compute / a Reed Solomon code] as parity from the data memorized by EEPROM66 and the data memorized by EEPROM66.

[0068] Furthermore, operation part 64 is made as [output / to the BPSK modulation circuit 68 / the response data (data transmitted to R/W1) corresponding to the processing], after performing processing predetermined by the sequencer 91.

[0069] The BPSK modulation circuit 68 carries out the BPSK modulation of the data supplied from operation part 64 (after-mentioned), and is made as [output / to the ASK modulation section 84 of RF interface section 61 / the data after a modulation].

[0070] RAM67 is made as [memorize / the data in the middle of processing etc. / temporarily], when a sequencer 91 processes.

[0071] EEPROM (Electrically Erasable and Programmable ROM)66 is the memory of a non-volatile, and even after IC card 2 ends the communication link with R/W1 and an electric power supply stops, it is made as [memorize / data / continue]. The fundamental program required for a sequencer 91 to process the command from R/W1 is memorized by ROM65.

[0072] Drawing 4 shows an example of assignment of the memory of EEPROM66.

[0073] EEPROM66 has 40 bytes of 256 physical blocks. Each physical block consists of a total of 40 bytes of 32 bytes of data division (D00 thru/or D1f), 2 bytes of attribute section (AT1, AT2), and 6 bytes of parity section (P0 thru/or P5).

[0074] The physical block number ffH (H expresses the hexadecimal) of EEPROM66 is assigned to the system ID block. The system ID block has memorized the information about the security of IC card 2.

[0075] Next, the physical block is assigned to the common area definition block (Common Area Definition Block) (the 1st field) or the provider domain-defined block (Provider Area Definition Block) (the 1st field) one by one toward 00H from the physical block number fdH.

[0076] When IC card 2 is published by EEPROM66, those (provider) who offer the system using this IC card 2 with predetermined equipment (issue machine) are registered into it. An issue machine is 1 physical block per one provider, carries out sequential use of the provider domain-defined block toward 00H from the physical block number fdH, and registers a provider.

[0077] The common area definition block and the provider domain-defined block have memorized the information on the location of the storage region which a provider uses etc.

[0078] And the physical block which is not used as a system ID block, a common area definition block, and a provider domain-defined block is assigned to the user block (User Block) used by the provider.

[0079] Drawing 5 shows an example of the assignment of each data to a system ID block.

[0080] As for D00 thru/or D0f of data division, the manufacture ID at the time of manufacture of EEPROM66 (Manufacture ID) (IDm) is memorized. A field D00 thru/or D03, a field D04 or D07, a field D08 or D0b and field D0c thru/or D0f have memorized the IC code of EEPROM66, the code (Manufacture Equipment Code) of the manufacture machine which created EEPROM66, the manufacture date (Manufacture Date) of EEPROM66, and the manufacture serial number (Manufacture Serial Number) of EEPROM66, respectively.

[0081] By using this information on IDm, all IC cards 2 (EEPROM66) are discriminable. In addition, a manufacture date sets January 1, 2000 to 0000H, and makes it the days from January 1, 2000. In addition, when a manufacture date is the 1990 set, a manufacture date is expressed as negative days from January 1, 2000 using a two's complement.

[0082] The issue ID (Issue ID) (IDi) when D10 thru/or D1f of data division publishes this IC card 2 is memorized. A field D10 thru/or D13, a field D14 or D17, a field D18 or D1b and field D1c thru/or 1f of the codes of the issue machine which published the category / group number which shows the category and group to whom IC card 2 belongs, and this IC card 2, the dates which published IC card 2, and the

expiration dates of IC card 2 are memorized, respectively.

[0083] Drawing 6 shows the attribute section of a system ID block. The attribute section has memorized the number of providers registered. In case an issue machine registers one provider, one physical block is used for it and it updates the value of this attribute section then.

[0084] The value of the attribute section is set as zero at the time of manufacture, and when an issue machine registers a provider into IC card 2 after that, it updates the value of the attribute section by the number of providers registered.

[0085] The parity section of a system ID block has memorized the Reed Solomon code (RS sign) calculated by the parity operation part 93 from the value of each bit of data division and the attribute section. Therefore, the value of the parity section is recalculated whenever data division or the attribute section is updated.

[0086] Drawing 7 shows an example of a common area definition block and a provider domain-defined block. In addition, these blocks are beforehand written in by the issue machine, when IC card 2 is published.

[0087] The common area definition block has been arranged at the physical block number feH of EEPROM66, and has memorized a setup of the storage region (common area (Common Area)) (the 2nd field) used by all providers.

[0088] From the physical block number fdH of EEPROM66, the provider domain-defined block has been arranged toward 00H, is 1 physical block per one provider, and has memorized a provider's information.

[0089] As shown in drawing 7, the data division D00 of a domain-defined block (a common area definition block and provider domain-defined block) thru/or the fields D00 and D01 of D1f have memorized the provider code (Provider Code) which shows provider's class. In the common area definition block, the value of fields D00 and D01 is set to 0000H, and, in the provider domain-defined block, let the value of fields D00 and D01 be the value of either 0001H thru/or FFFFH.

[0090] The field D02 of the data division of a domain-defined block thru/or D05 have memorized the allocation table (Allocation Table) which consists of numbers BN1 (fields D04 and D05) (BN1 > BN0) of the next physical block of the number BN0 (fields D02 and D03) of the physical block of the head of the storage region (provider field (Provider Area)) (the 2nd field) which this provider uses, and the number of the physical block of an end. a provider field is set as the position (the physical block number BN0 -- or (BN 1-1)) of EEPROM66 except a system block (a system ID block, domain-defined block), as shown in drawing 8.

[0091] Thus, since the provider field is specified by BN0 and BN1, it is not the area size used by the provider (user), and for the information on the amount proportional to the number of providers, data can be managed and use effectiveness of memory can be made high.

[0092] The field D06 of the data division of a domain-defined block thru/or D09 have remembered the partition table (Partition Table) which consists of the block counts BRW (fields D08 and D09) of the read/write block in a random access field to be block count BRA (fields D06 and D07) of a random access field (after-mentioned) among the storage regions which a provider uses. At this time, block count BRA of a random access field is set as formula $BRA=0$ or the value with which are satisfied of formula $2xn \leq BRA \leq BN1 - BN0$ (n is the number of light buffers (after-mentioned)), and the block count BRW of a read/write block is set as $BRW=0$, when it is $BRA=0$, and when it is $BRA \neq 0$, it is set as the value with which are satisfied of formula $n \leq BRW \leq BRA - n$.

[0093] Field D0a of the data division of a domain-defined block and D0b have memorized several n of the light buffer of a random access field. A n light buffer is used when making coincidence memorize n data to logical-block number 00H thru/or (00+n (hexadecimal display)) H of a random access field. In addition, when memorizing data to the physical block which has other logical-block numbers among random access fields, only one light buffer is used.

[0094] as mentioned above, according to a domain-defined block, it is shown in drawing 8 -- as -- the physical block number BN0 -- a field (a provider field or common area) is assigned to the provider specified in provider code, further, the physical block of the BRA individual of the field (a provider field

or common area) is assigned to a random access field, and the remaining physical blocks are assigned to the sequential access field (after-mentioned) for or (BN 1-1).

[0095] Furthermore, according to the domain-defined block, as shown in drawing 8, the random access field is logically assigned to the read/write block of a BRW individual, the read-only block, and the n light buffer. In addition, a read/write block and physical blocks other than a light buffer are assigned to a read-only block.

[0096] Field D0c of the data division of a domain-defined block and D0d have memorized the Perth block permission which has the information on the access privilege to the Perth block (Purse Block) (after-mentioned) in the storage region (random access field) which this provider uses.

[0097] Drawing 9 shows an example of the Perth block permission.

[0098] The Perth block permission (16 bits, b0, or bf) is read to the Perth block, and shows authorization or the disapproval of an add instruction and a subtraction instruction.

[0099] The Perth block permission of a common area definition block has memorized whether the Perth block is used in the storage region (common area) set up with a common area definition block to Field (bit) bb. That is, in the case of bb=0, the Perth block is not used. In the case of bb=1, the Perth block is used. And especially the field (bit) of others in the Perth block permission of a common area definition block is not used. In addition, in the case of bb=1, the read/write block whose logical-block number is 00H is used as a Perth block.

[0100] Next, in the Perth block permission of a provider domain-defined block, it has memorized whether the Perth block is used in the storage region set up with this provider domain-defined block to the field b3. That is, in the case of b3=0, the Perth block is not used. In the case of b3=1, the Perth block is used. In addition, in the case of b3=1, the read/write block whose logical-block number is 00H is used as a Perth block.

[0101] And the propriety of an add instruction to the Perth block was memorized to the field b2, the propriety of a subtraction instruction to the Perth block was memorized to the field b1, and the propriety of read-out to the Perth block is memorized to the field b0 (in the case of bi=1 (1 i= 0, 2), the instruction is permitted and, in the case of bi=0, the instruction is not permitted). Moreover, it has memorized to Field bb whether the Perth block is used in the storage region set up with a common area definition block. In addition, the same value as bb of the Perth block permission of a common area definition block is memorized by bb.

[0102] Furthermore, the propriety of an add instruction to the Perth block was memorized to Field ba, the propriety of a subtraction instruction to the Perth block was memorized to the field b9, and the propriety of read-out to the Perth block is memorized to the field b8 (in the case of bi=1 (8 i= 9a), the instruction is permitted and, in the case of bi=0, the instruction is not permitted).

[0103] Field D0e of the data division of a domain-defined block of drawing 7 and D0f memorized the version number of the security key (a common key and provider key) used for encryption and a decryption at a provider's (R/W1) authentication, and a list, and a field D10 thru/or 1f of the security key are memorized.

[0104] In addition, when R/W1 polls, IC card 2 returns the version number of these two keys (a common key and provider key). Therefore, in authentication between R/W1 and IC card 2, the security key of two or more versions can be used properly.

[0105] And the attribute sections AT1 and AT2 of a domain-defined block are formed as a reserve, and especially information is not memorized. The parity section of a domain-defined block has memorized the parity (RS sign) calculated from the value of all the bits of data division and the attribute section.

[0106] Thus, the domain-defined block set up by the issue machine has memorized a provider code, an allocation table, a partition table, the Perth block permission, the security key version, and the security key.

[0107] Drawing 10 shows an example of a user block. As mentioned above with reference to drawing 4, physical blocks other than a system ID block, a common area definition block, and a provider domain-defined block are used by the provider as a user block among the rooms of EEPROM66.

[0108] For example, if eight providers are registered when room consists of 256 blocks as shown in

drawing 4 , 246 (= 256-10) blocks of those other than the system block of a total of ten (= 1+1+8) individuals of a system ID block, a common area definition block, and eight provider domain-defined blocks will be used as a user block. Moreover, if 40 providers are registered, a system block will become a total of 42 (= 1+1+40) individuals, and the user block of 214 (= 256-42) individuals will be secured.

[0109] A user block is assigned to each provider according to the allocation table (drawing 7) of a domain-defined block. In addition, since a provider uses the user block currently assigned beforehand with reference to an allocation table, he does not access other than the field (a provider field or common area) assigned on the allocation table.

[0110] The user block of the field (a provider field or common area) assigned on the allocation table is assigned to the random access field and the sequential access field according to the above-mentioned partition table (drawing 7).

[0111] Furthermore, the user block of a random access field is used as either a read/write block, a read-only block and a light buffer, and the number of these blocks is set up as mentioned above according to the number of a partition table and light buffers.

[0112] Thus, the data division D00 thru/or D1f of a user block currently assigned is used according to processing by the provider to whom the user block is assigned.

[0113] The attribute section of a user block of a random access field has memorized the incremental counter (Incremental Counter) (bits bf and be) and the logical-block number (Bit bd thru/or b0), as shown in drawing 11 .

[0114] A logical-block number and an incremental counter are used when accessing the user block of a random access field.

[0115] When reading the data memorized to the random access field, it is a logical-block number, and the data (physical block) to read are searched and the newest data are read with reference to the incremental counter of the data which have the logical-block number.

[0116] On the other hand, when memorizing data to a random access field, after using as a light buffer the physical block (after-mentioned) which became unnecessary with reference to the logical-block number and incremental counter of data which have already been memorized to the random access field, data are written in the light buffer.

[0117] In addition, when the Perth block permission of an above-mentioned domain-defined block is set up so that the Perth block may be used, the read/write block whose logical-block number is 00H is used as a Perth block.

[0118] The Perth block is used to read the value already memorized when performing addition and subtraction of data frequently when setting up the access privilege to data finely (since possibility that information will be revealed increases).

[0119] Drawing 12 shows an example of the Perth block. The data division D00 of the Perth block the field D00 of D1f thru/or D07 are used as Perth data division. The data division D00 of the Perth block the field D08 of D1f thru/or D0f have memorized Execution ID (Execution ID). In addition, the field D10 thru/or D1f of data division of the Perth block is set up read-only, although used as user data division.

[0120] The Perth data division have memorized predetermined data. Execution ID is referred to when the add instruction or subtraction instruction to the Perth block is executed, and it is compared with the execution ID contained in the add instruction or a subtraction instruction.

[0121] On the other hand, the attribute section of a user block of a sequential access field has memorized the lap round number (Bit bf thru/or b0), as shown in drawing 13 . In the sequential access field, if data (sequentially) are memorized in an order from the physical block of the head of a field and data are memorized to the physical block of the last of a field, data are again memorized in an order from the physical block of the head of a field (overwritten). The lap round number has memorized the sequence.

[0122] Therefore, when accessing the user block of a sequential access field, while being used, when memorizing data to a sequential access field, sequential reference of the lap round number is carried out. And data are memorized by the next physical block of the physical block which has a lap round number at the tail end till then. At this time, the lap round number of the physical block data were remembered

to be is set as the number which added 1 to the lap round number at the tail end till then.

[0123] In addition, a failure occurs in the middle of a store at the time of the last store, and when the parity error (physical memory colla tempestade PUSHON) has arisen in the physical block which has a lap round number at the tail end, new data are memorized by the physical block, for example. Moreover, new data are memorized by the physical block of the head of a sequential access field when the physical block which has a lap round number at the tail end is a physical block of a sequential access end-of-region rate.

[0124] As mentioned above, EEPROM66 is suitably used for each provider.

[0125] Next, with reference to the flow chart of drawing 14 , and the timing chart of drawing 15 , actuation of IC card 2 and R/W1 is explained.

[0126] First, a predetermined electromagnetic wave is emitted from an antenna 27, the loaded condition of an antenna 27 is supervised, IC card 2 approaches, and R/W1 corresponding to the provider registered into IC card 2 in step S1 stands by until change of loaded condition is detected. In addition, R/W1 emits the electromagnetic wave which carried out the ASK modulation by the data of a short predetermined pattern, and you may make it repeat the appeal to IC card 2 in step S1 until the response from IC card 2 is obtained in fixed time amount.

[0127] When R/W1 detects approach of IC card 2 in step S1 (time of day t0 of drawing 15), it progresses to step S2. SPU32 of R/W1 The square wave of a predetermined frequency (a clock frequency twice the frequency [for example,] of data) as shown in drawing 16 (a) is made into a subcarrier. By the data (command corresponding to the processing which IC card 2 is made to perform) (for example, data shown in drawing 16 (b)) transmitted to IC card 2, a BPSK modulation is performed and the generated modulated wave (BPSK modulating signal) (drawing 16 (c)) is outputted to a modulation circuit 23.

[0128] In addition, as shown in drawing 16 (c) using differential conversion at the time of a BPSK modulation A value makes the same thing as the last BPSK modulating signal ("1", "0" or "0", "1") a BPSK modulating signal, when the data of 0 appear. The value makes what reversed the phase of the last BPSK modulating signal (thing which made "0" reverse "1" and made "1" reverse "0") the BPSK modulating signal, when the data of 1 appear.

[0129] Thus, since it gets over to the original data also when a BPSK modulating signal is reversed by holding data by change of the phase of a modulated wave using differential conversion, when getting over, the need of considering the polarity of a modulated wave is lost.

[0130] And a modulation circuit 23 is the BPSK modulating signal, the ASK modulation of the predetermined subcarrier is carried out with less than (for example, 0.1) one modulation factor (= maximum amplitude of the maximum amplitude/subcarrier of a data signal), and the generated modulated wave (ASK modulated wave) is transmitted to IC card 2 through an antenna 27 (during the time of day t0 of drawing 15 thru/or time of day t1).

[0131] In addition, when not transmitting, the modulation circuit 23 is made as [generate / with the high level of the two level (high level and low level) of a digital signal / a modulated wave].

[0132] Next, in step S3, IC cards 2 are an antenna 53 and a capacitor 52, transform into an electrical signal a part of electromagnetic wave which the antenna 27 of R/W1 emitted, and output the electrical signal (modulated wave) to RF interface section 61 of IC51. and the ASK recovery section 81 of RF interface section 61 controls the dc component of the generated signal, extracts a data signal, and outputs the data signal to the BPSK demodulator circuit 62 and the PLL section 63 while it supplies the signal generated in the modulated wave by rectifying and carrying out smooth (namely, envelope detection -- carrying out) to a voltage regulator 82.

[0133] A voltage regulator 82 stabilizes the signal supplied from the ASK recovery section 81, generates direct current power, and supplies it to each circuit.

[0134] In addition, the terminal voltage V0 of an antenna 53 is as follows, for example at this time.

$$V_0 = V_{10}(1 + kxV_s(t)) \cos(\omega t)$$

[0135] It is here, and in V10, k shows a modulation factor and Vs (t) shows the signal component for the amplitude of a carrier component, respectively.

[0136] Moreover, the value VLR of a low level in the electrical potential difference V1 after rectification by the ASK recovery section 81 is as follows, for example.

$VLR = V10(1 + kx(-1)) - Vf$ [0137] Here, Vf shows the voltage drop in the diode D of a rectifier circuit. Usually, Vf is about 0.7 volts.

[0138] And a voltage regulator 82 stabilizes rectification and the signal by which smooth was carried out by the ASK recovery section 81, and supplies it to each circuits including operation part 64 as direct current power. In addition, since the modulation factor k of a modulated wave is less than one, its voltage variation after rectification (difference of high level and a low level) is small. Therefore, a voltage regulator 82 can generate direct current power easily.

[0139] When a modulation factor k receives 5% of modulated wave so that V10 may become 3 volts or more, for example, the low-level electrical potential difference VLR after rectification It becomes more than 2.15 ($= 3x(1 - 0.05) - 0.7$) volts. A voltage regulator 82 While being able to supply electrical potential difference sufficient as a power source to each circuit, amplitude $2kxV10$ (Peak-to-Peak value) of the alternating current component (data component) of the electrical potential difference V1 after rectification It becomes more than 0.3 ($= 2x0.05x3$) volts, and the ASK recovery section 81 can restore to data by the sufficiently high S/N ratio.

[0140] Thus, when a modulation factor k uses less than one ASK modulated wave, while performing the communication link with a low (in high condition of a S/N ratio) error rate, direct current voltage sufficient as a power source is supplied to IC card 2.

[0141] And according to the clock signal supplied from the PLL section 63, the BPSK demodulator circuit 62 restores to the data signal (BPSK modulating signal) from the ASK recovery section 81, and outputs the data to which it restored to operation part 64.

[0142] Next, in step S4, when the data supplied from the BPSK demodulator circuit 62 are deciphered, after decrypting operation part 64 in a code / decode section 92, it supplies the data (command) to a sequencer 91, and performs processing corresponding to the command (during the time of day t1 of drawing 15 thru/or time of day t2). In addition, while the value had transmitted the data of 1, R/W1 is standing by, until it receives the answerback from this period 2, i.e., an IC card. Therefore, in this period, IC card 2 has received the modulated wave with fixed maximum amplitude.

[0143] Next, in step S5, the sequencer 91 of operation part 64 outputs data (data transmitted to R/W1), such as a processing result, to the BPSK modulation circuit 68. Like SPU32 of R/W1, after the BPSK modulation circuit 68 carries out the BPSK modulation of the data, it is outputted to the ASK modulation section 84 of RF interface section 61.

[0144] The ASK modulation section 84 and by fluctuating the load connected to the both ends of an antenna 53 according to data using a switching element The modulated wave which has received (in the time of transmission of IC card 2) the maximum amplitude of a modulated wave becomes fixed -- **** - - an ASK modulation is carried out according to the data to transmit, the terminal voltage of the antenna 27 of R/W1 is fluctuated according to it, and the data is transmitted to R/W1 (during the time of day t2 of drawing 15 thru/or time of day t3).

[0145] In step S6, as for the modulation circuit 23 of R/W1, the value is continuing transmission of the data of 1 (high-level) at the time of reception of the data from IC card 2. And a demodulator circuit 25 detects the data transmitted with IC card 2 from minute fluctuation (for example, dozens of microvolts) of the terminal voltage of the antenna 27 of IC card 2, and the antenna 27 combined in electromagnetism.

[0146] And it gets over and a demodulator circuit 25 outputs the generated digital data to SPU32, after amplifying the detected signal (ASK modulated wave) with the amplifier of high interest profit.

[0147] And in step S7, after restoring to the data (BPSK modulating signal), it outputs to DPU31 and, as for SPU32 of R/W1, DPU31 processes the data (during the time of day t3 of drawing 15 thru/or time of day t4).

[0148] Furthermore, in step S8, when it judges whether a communication link is ended according to a processing result and it is judged again that it communicates, to step S2, DPU31 of R/W1 is return, step S2, or step S7, and communicates the following data (command) (time of day t4 thru/or time of day t8 of

drawing 15). On the other hand, when it is judged that a communication link is ended, R/W1 ends the communication link with IC card 2.

[0149] As mentioned above, using the ASK modulation whose modulation factor k is less than one, R/W1 transmits a predetermined command to IC card 2, and IC card 2 performs reception and processing corresponding to the command for the command, and it returns the data corresponding to the result of the processing to R/W1.

[0150] Next, the actuation when writing in data to EEPROM66 as an example of processing by IC card 2 in above-mentioned step S4 is explained with reference to the flow chart of drawing 17 thru/or drawing 21 .

[0151] First, with reference to the flow chart of drawing 17 thru/or drawing 19 , the actuation when writing data in the random access field of EEPROM66 is explained.

[0152] In step S21, a sequencer 91 judges whether it is that the physical block which writes in data is a read/write block (the Perth block is not included) (the block to a BRW individual is considered as a read/write block from BNO at sequence as shown in drawing 8), and when it is judged that it is a read/write block, it progresses to step S22.

[0153] A sequencer 91 progresses to step S23 (drawing 18), when it judges whether it is using the Perth block ($b3=1$) and the Perth block is not being used with reference to the Perth block permission (drawing 9) of a provider domain-defined block which has the provider code of R/W1 (in the case of $b3=0$).

[0154] On the other hand, when it is judged that the Perth block is used in step S22, a sequencer 91 judges whether the read/write block which writes in whether the logical-block number of the data (it writes in) to memorize is 00H and data in step S24 has lapped with the Perth block, and when it is judged that the read/write block which writes in data has not lapped with the Perth block, it progresses to step S23. [0155] When it is judged that the read/write block which writes in data has lapped with the Perth block, a sequencer 91 ends processing in step S25, after performing error processing.

[0156] Moreover, when it is judged that the physical block which writes in data in step S21 is not a read/write block, it progresses to step S26, the physical block in which a sequencer 91 writes data judges whether it is the Perth block, and when it is judged that it is the Perth block, it progresses to step S27.

[0157] When it is judged that the physical block which writes in data is not the Perth block, a sequencer 91 ends processing in step S28, after performing error processing.

[0158] In step S27, in a random access field, a sequencer 91 progresses to step S29, when the Perth block (physical block whose logical-block number is 00H) is looked for and the Perth block is discovered.

[0159] Since the store to the Perth block cannot be performed when the Perth block is not discovered at step S27, a sequencer 91 ends processing in step S30, after performing error processing.

[0160] or [next, / that the add instruction is permitted with reference to the Perth block permission of a provider domain-defined block in step S29 by the instruction (command) to the Perth block judging whether it is an add instruction, and a sequencer 91 progressing to step S31 when it is judged that it is an add instruction] ($b2=1$) -- it judges whether it is no.

[0161] And when a sequencer 91 judges that the add instruction to the Perth block is permitted at step S31, it progresses to step S23.

[0162] On the other hand, when it is judged that the add instruction to the Perth block is not permitted at step S31, a sequencer 91 ends processing, after performing error processing in step S32, without executing an add instruction (when it is $b2=0$).

[0163] Moreover, in step S29, when the instruction to as opposed to [when it is judged that the instruction to the Perth block is not an add instruction / progress to step S33 and] the Perth block in a sequencer 91 judges that it judges whether it is a subtraction instruction and is a subtraction instruction, it progresses to step S34.

[0164] And in step S34, a sequencer 91 judges whether it is that the subtraction instruction is permitted ($b1=1$) with reference to the Perth block permission of a provider domain-defined block, and when it is judged that the subtraction instruction to the Perth block is permitted, it progresses to step S23.

[0165] On the other hand, when it is judged that the subtraction instruction to the Perth block is not permitted at step S34, a sequencer 91 ends processing, after performing error processing in step S35, without executing a subtraction instruction (when it is $b1=0$).

[0166] Moreover, in step S33, when it is judged that the instruction to the Perth block is not a subtraction instruction, a sequencer 91 ends processing in step S36, after performing error processing.

[0167] Next, in step S23 of drawing 18, a sequencer 91 searches the physical block of a random access field, and looks for the physical block which has the same logical-block number as the logical-block number of the data which write in.

[0168] And in step S37, a sequencer 91 judges whether the number of the physical blocks discovered at step S23 is two. That is, in this system, the last data and the data before last are memorized at least about each logical block. And when memorizing still newer data, new data are memorized on the data before last (it may memorize on the data of other logical-block numbers before last). When two physical blocks of the same logical-block number exist, it progresses to step S38, and the value (00, 01, 10, or 11) of the incremental counter in the two physical blocks is read and compared.

[0169] And make a physical block with the large value of an incremental counter into the physical block (new physical block) new data are remembered to be, and let a physical block with the small value of an incremental counter be the physical block (old physical block) old data are remembered to be.

[0170] However, when the values of two incremental counters are 00 and 11, make into a new physical block the physical block whose value of an incremental counter is 00, and let the physical block whose value of an incremental counter is 11 be an old physical block.

[0171] A sequencer 91 memorizes the number (physical block number) of a new physical block to RAM67 as a variable Y between two physical blocks, and makes RAM67 memorize the number of an old physical block in step S39 as a variable W (number of the physical block used as a light block).

[0172] Thus, a sequencer 91 progresses to step S49, after making Variable Y and Variable W memorize.

[0173] On the other hand, when it is judged that the number of the physical blocks discovered at step S23 is not two in step S37, it progresses to step S40 and a sequencer 91 judges whether the number of the physical blocks discovered at step S23 is one. And when it is judged that it is one piece, it progresses to step S41.

[0174] In step S40, when a sequencer 91 judges that the number of the physical blocks discovered at step S23 is not one, after performing error processing, processing is ended in step S42.

[0175] As for saying [that the same logical block exists only in one piece], the data before last will not exist for a certain reason. Then, in this case, it is the physical block of other logical-block numbers, and the physical block (namely, physical block whose number of the physical blocks which have the same logical-block number is two) which has data the last and before last is searched, and the physical block before last of them is used as a light block. For this reason, in step S41, a sequencer 91 progresses to step S43, after making RAM67 memorize the number of the discovered physical block (one piece) as a variable Y.

[0176] In step S43, a sequencer 91 searches the physical block of a random access field, and looks for two physical blocks which have the same predetermined logical-block number (arbitration) (logical-block number unrelated to the logical-block number now made into the write-in object).

[0177] In addition, since it retrieves sequentially from logical-block number 00H when searching a physical block, a smaller number, then retrieval time can be shortened for the logical-block number of the data which perform write-in processing frequently.

[0178] In step S44 and a sequencer 91 It judges whether the physical block whose logical-block number is the two same pieces was discovered at step S43. When it is judged that it was discovered, the incremental counter of two physical blocks discovered by progressing to step S45 is referred to. It progresses to step S49 (drawing 19), after making RAM67 memorize the number of the physical block of the older one as a variable W (number of a light block) between two physical blocks.

[0179] On the other hand, when it is judged in step S44 that two physical blocks were not discovered at step S43, it progresses to step S46, and a sequencer 91 carries out sequential count of the parity of each

physical block of a random access field, and looks for the physical block which has started the parity error as compared with the value memorized by the parity section of each physical block.

[0180] And when it judges whether there is any physical block which has started the parity error and it is judged that there is a physical block which has started the parity error, it progresses to step S47, and a sequencer 91 progresses to step S49, after making RAM67 memorize the number of the physical block as a variable W (number of a light block).

[0181] In step S46, when it is judged that there is no physical block which has started the parity error, a sequencer 91 ends processing in step S48, after performing error processing.

[0182] In step S49 of drawing 19 next, a sequencer 91 It judges whether the physical block which writes in data is the Perth block (physical block whose logical-block number is 00H). When it is judged that it is the Perth block, the execution ID of the instruction which progresses to step S50 and is performed to the Perth block When it judges whether it is the same as that of the execution ID of the physical block of the number memorized as a variable Y at step S39 or step S41 (drawing 12) and it is judged that it is the same, it judges that this instruction is already processed and processing is ended.

[0183] Thus, since IC card 2 does not process the command when R/W1 carries out the retry of the same command by using Execution ID and the command is already processed, the same command is not processed twice.

[0184] In step S50, when the instruction with which a sequencer 91 is performed to the Perth block in step S51 when the execution ID of the instruction performed to the Perth block judges that it is not the same as that of the execution ID of the physical block of the number memorized as a variable Y judges whether it is an add instruction and it is an add instruction, it progresses to step S52.

[0185] In step S52, a sequencer 91 reads the Perth data of the physical block of the number of Variable Y, calculates the sum of the Perth data and the data contained in the instruction performed to the Perth block, and uses the sum as the Perth data (new Perth data) in new block data. Thus, it progresses to step S54, after processing. In addition, let execution ID of the physical block of the number of Variable Y be the execution ID of new block data at this time. This prevents processing of a duplex.

[0186] On the other hand, when it is judged that the instruction performed to the Perth block is not an add instruction in step S51 (that is, it is a subtraction instruction), it progresses to step S53, and a sequencer 91 reads the Perth data of the physical block of the number of Variable Y, calculates the difference of the Perth data and the data contained in the instruction performed to the Perth block, and uses the difference as the Perth data (new Perth data) in new block data. Thus, it progresses to step S54, after processing. In addition, let execution ID of the physical block of the number of Variable Y be the execution ID of new block data at this time. This prevents processing of a duplex.

[0187] Moreover, in step S49, a sequencer 91 progresses to step S54, when the physical block which writes in data judges that it is not the Perth block (that is, it is a read/write block).

[0188] And in step S54, a sequencer 91 makes the number which added 1 to the value of the incremental counter of the physical block of the number of Variable Y the value of the incremental counter of new block data. However, when the value of the incremental counter of the physical block of the number of Variable Y is 11, a sequencer 91 sets the value of the incremental counter of new block data to 00.

[0189] Next, in step S55, a sequencer 91 makes the parity of the data newly written in the parity operation part 93, an incremental counter, and a logical-block number calculate, and makes the value of the parity the value of the parity section of new block data.

[0190] And a sequencer 91 makes the physical block (light buffer) of the number of the variable W memorized at either step S39, step S45 or step S47 memorize new block data (the newly memorized data (for them to be the Perth data and Execution ID in the Perth block), its logical-block number, incremental counters, and such parity) in step S56.

[0191] As mentioned above, since the data of the same logical-block number as the logical-block number of the data are left behind to memory by choosing the physical block (light buffer) which remembers data to be a logical-block number using an incremental counter when a failure occurs in the midst of the store of data, logically, memory colla tempestade PUSHON does not occur.

[0192] Although the incremental counter was used with the gestalt of the above-mentioned

implementation in order to distinguish the block with which new data are recorded among the same logical blocks of a random access field, it is also possible to, distinguish the block with which new data are recorded by [at the time of record] securing 4 bytes of field in a random access field, and making time of day (the date, time of day, or value of a counter) then, record to it absolutely for example.

[0193] Next, with reference to the flow chart of drawing 20 and drawing 21 , the actuation when writing data in the sequential access field of EEPROM66 is explained.

[0194] A sequencer 91 makes RAM67 memorize the number of the physical block of the head of a sequential access field as a variable Z in step S61.

[0195] Next, a sequencer 91 reads the lap round number of the physical block whose physical block number is Z, while making RAM67 memorize, it reads the lap round number of the physical block whose physical block number is Z+1, and RAM67 is made to memorize it as a variable B as a variable A in step S62.

[0196] And in step S63, a sequencer 91 judges whether the difference (A-B) of the value of Variable A and the value of Variable B is 1, judges that it is the physical block the physical block of the physical block number Z remembers the data which have a lap round number at the tail end to be when it is not 1, and progresses to step S66.

[0197] When a sequencer 91 judges whether the physical block number Z is the same as the number of the physical block of a sequential access end-of-region rate in step S64 when it is judged that the difference (A-B) of the value of Variable A and the value of Variable B is 1, and it is judged that it is the same, the data with which the physical block of a sequential access end-of-region rate has a lap round number at the tail end are judged to be the physical block to memorize, and it progresses to step S66.

[0198] In step S64, when the physical block number Z judges that it is not the same as that of the number of the physical block of a sequential access end-of-region rate, a sequencer 91 returns to step S62 in step S65, after only 1 makes the value of the variable Z which RAM67 was made to memorize increase. And processing of step S62 thru/or step S65 is repeated successively, changing the value (value of the physical block number to search) of Variable Z.

[0199] Thus, the tail end of the lap round number of the data memorized sequentially is discovered. And in step S66, a sequencer 91 performs the parity check of a block of the number (number of the physical block at the tail end of = lap round number) of Variable Z.

[0200] And in step S67, it judges whether the parity error has produced the sequencer 91 in the physical block, and when it is judged that the parity error has arisen, it progresses to step S68.

[0201] In step S68 a sequencer 91 The value of Variable Z judges whether it is the same as that of the number of the physical block of the head of a sequential access field. When it is judged that it is the same, the tail end of data (what has started the parity error does not contain) judges that it is the physical block of a sequential access end-of-region rate, and sets to step S70. It progresses to step S72 (drawing 21), after making RAM67 memorize the number of the physical block of a sequential access end-of-region rate as a new variable Y.

[0202] When it is judged that the value of Variable Z is not the same as that of the number of the physical block of the head of a sequential access field, after a sequencer 91 makes RAM67 memorize the value (Z-1) which computed the number of the physical block at the tail end of data by having subtracted 1 from the value of Variable Z, and computed it as a variable Y in step S71, it progresses to step S72.

[0203] On the other hand, in step S69, when it is judged that the parity error has not arisen at step S67, a sequencer 91 progresses to step S72, after making RAM67 memorize the number (value of Variable Z in this case) of the physical block at the tail end of data as a variable Y.

[0204] Next, in step S72, when it judges whether a sequencer 91 has the number (value of Variable Y) of the physical block at the tail end of data, and the same number of the physical block of a sequential access end-of-region rate and it is judged that it is the same, it progresses to step S73.

[0205] And in step S73, it progresses to step S75, after a sequencer's 91 making the number of the physical block of the head of a sequential access field the number of the physical block which writes in new data and making RAM67 memorize by making the number into Variable W.

[0206] It progresses to step S75, after a sequencer's 91 making the number which added 1 to the value of Variable Y the number of the physical block which writes in new data, making the number Variable W in step S74 and making RAM67 memorize, when it is judged in step S72 that the number (value of Variable Y) of the physical block at the tail end of data and the number of the physical block of a sequential access end-of-region rate are not the same.

[0207] Next, in step S75, since the newly memorized data and the data which judge whether the physical block (data at the tail end) of the number of Variable Y is the same, and are newly memorized when the same are already memorized, a sequencer 91 ends processing.

[0208] When it is judged that the physical block (data at the tail end) of the data newly memorized on the other hand and the number of Variable Y is not the same, in step S76, a sequencer 91 reads the lap round number of the physical block of the number of Variable Y, and makes the number which added 1 to the value the lap round number of the data (new block data) newly memorized.

[0209] Next, in step S77, a sequencer 91 makes the parity operation part 93 calculate the parity of the data to memorize and a lap round number (new block data), and writes new block data in it in step S78 at the physical block of a number W.

[0210] Thus, since the data of a lap round number smaller than the lap round number of the data which were being written in in the midst of the store of new data when a failure occurred since the lap round number in the data memorized sequentially is retrieved sequentially and new data are memorized to the next physical block (or physical block of the head of a sequential access field) of the data at the tail end remain, logically, memory colla tempestade PUSHON is not generated.

[0211] As mentioned above, EEPROM66 is made as [control / generating of memory colla tempestade PUSHON] using the information on the attribute section while being able to offer a storage region independently to two or more providers.

[0212] In addition, the same user block can also be assigned to two or more providers. In that case, the same user block is assigned on the allocation table of the provider domain-defined block with which those providers (overlap provider) are registered. at this time, a different access privilege (read/write -- or read-only) for every provider can be set up to the same user block by setting up the partition table of a provider domain-defined block for every provider. Furthermore, a predetermined provider can write in data to the user data division (read-only to other providers) of the Perth block which other providers use by setting up so that the Perth block may not be used to a predetermined provider, and setting up so that the Perth block may be used to other providers.

[0213] Moreover, the value of field D0e of a domain-defined block, and D0f (field where the version number of a security key is usually memorized) By setting it as a predetermined value (for example, FFFFH), and memorizing a predetermined provider's provider code (a maximum of eight pieces) to the field D10 thru/or D1f of a domain-defined block further That provider (local common provider) can use the user block assigned on the allocation table of this domain-defined block as a common area.

[0214] Moreover, the access privilege to the user block can be set up for every local common provider by registering a local common provider into two domain-defined blocks which assign the same user block, and setting up a different access privilege for every domain-defined block.

[0215] Thus, corresponding to two or more providers (namely, R/W), processing according to individual can be performed by setting up an overlap provider and a local common provider.

[0216] In addition, this invention can be applied also when delivering and receiving a signal in the condition (contact) of having been combined physically besides in the case of delivering and receiving a signal through radio (non-contact). In the case of interruption of service, or the equipment which operates by the cell, when the cell has been removed, data can be secured.

[0217] Moreover, the program which performs each above-mentioned processing is recorded on the transmission medium which consists of record media, such as a magnetic disk and CD-ROM, it provides for a user, or is transmitted to a user through transmission media, such as a network, is recorded on the transmission medium which consists of record media, such as a hard disk and solid-state memory, and can be made to use.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the block diagram showing an example using IC card 2 which is the gestalt of 1 operation of the information processor of this invention of a non-contact card system.

[Drawing 2] It is the block diagram showing the example of a configuration of the reader/writer 1 of drawing 1 .

[Drawing 3] It is the block diagram showing the configuration of IC card 2 which is the gestalt of 1 operation of the information processor of this invention.

[Drawing 4] It is drawing showing an example of assignment of the memory of EEPROM66 of drawing 3 .

[Drawing 5] It is drawing showing an example of assignment of each field of a system ID block of drawing 4 .

[Drawing 6] It is drawing showing an example of the attribute section of drawing 5 .

[Drawing 7] It is drawing showing an example of assignment of each field of a domain-defined block of drawing 3 .

[Drawing 8] It is drawing showing an example of assignment of a user block of drawing 3 .

[Drawing 9] It is drawing showing an example of the Perth block permission of drawing 7 .

[Drawing 10] It is drawing showing an example of assignment of each field of a user block of drawing 3 .

[Drawing 11] It is drawing showing an example of the attribute section of a user block of the random access field of drawing 8 .

[Drawing 12] It is drawing showing an example of assignment of each field of the Perth block.

[Drawing 13] It is drawing showing an example of the attribute section of a user block of the sequential access field of drawing 8 .

[Drawing 14] It is the flow chart which actuation of the non-contact card system of drawing 1 explains.

[Drawing 15] It is the timing chart which actuation of the non-contact card system of drawing 1 explains.

[Drawing 16] It is drawing showing an example of a BPSK modulation.

[Drawing 17] It is a flow chart explaining actuation of IC card 2 at the time of the store to the user block of the random access field of drawing 8 .

[Drawing 18] It is a flow chart explaining actuation of IC card 2 at the time of the store to the user block of the random access field of drawing 8 .

[Drawing 19] It is a flow chart explaining actuation of IC card 2 at the time of the store to the user block of the random access field of drawing 8 .

[Drawing 20] It is a flow chart explaining actuation of IC card 2 at the time of the store to the user block of the sequential access field of drawing 8 .

[Drawing 21] It is a flow chart explaining actuation of IC card 2 at the time of the store to the user block of the sequential access field of drawing 8 .

[Description of Notations]

1 Reader/writer 2 IC Card, 3 Controller 21 IC, 23 Modulation circuit 25 A demodulator circuit, 27
Antenna 51 IC and 52 capacitor 53 Antenna 61 RF interface section 62BPSK demodulator circuit, 63
The PLL section 64 Operation part, 65 ROM, 66 EEPROM, 67 RAM 68 BPSK modulation circuit 81
ASK recovery section 82 Voltage regulator 83 Oscillator circuit 84 ASK modulation section 91
Sequencer 92 A code / decode section 93 Parity operation part

[Translation done.]

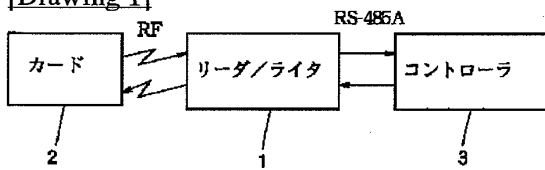
* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

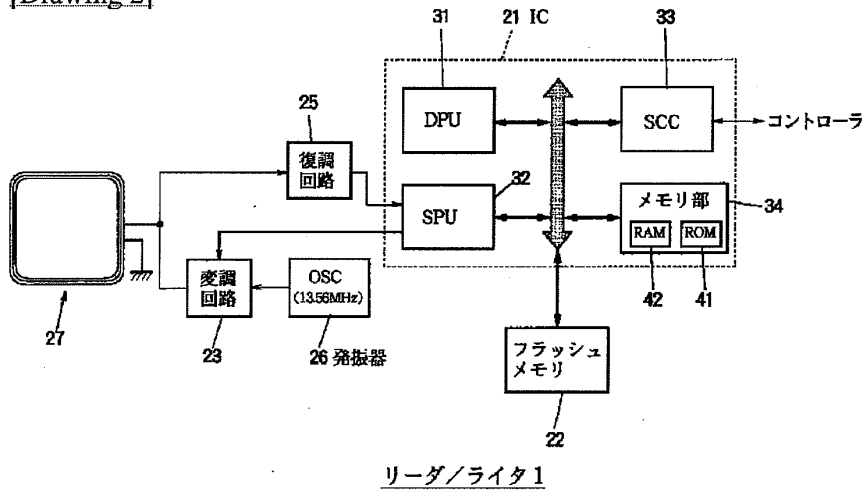
- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DRAWINGS

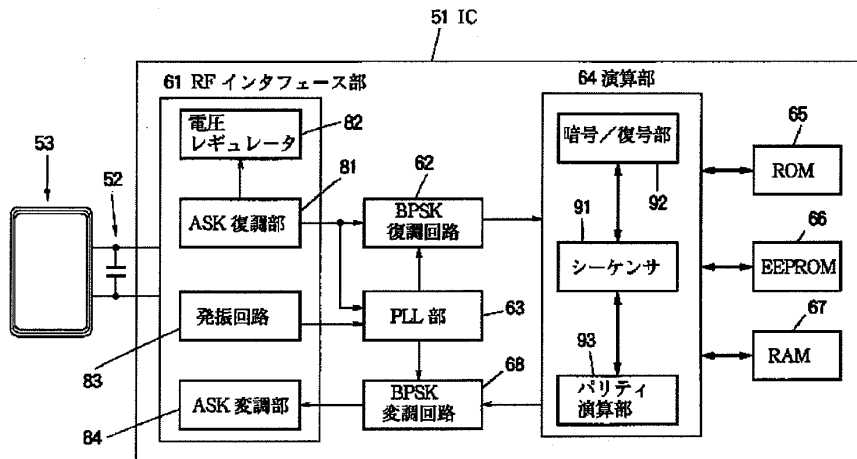
[Drawing 1]



[Drawing 2]

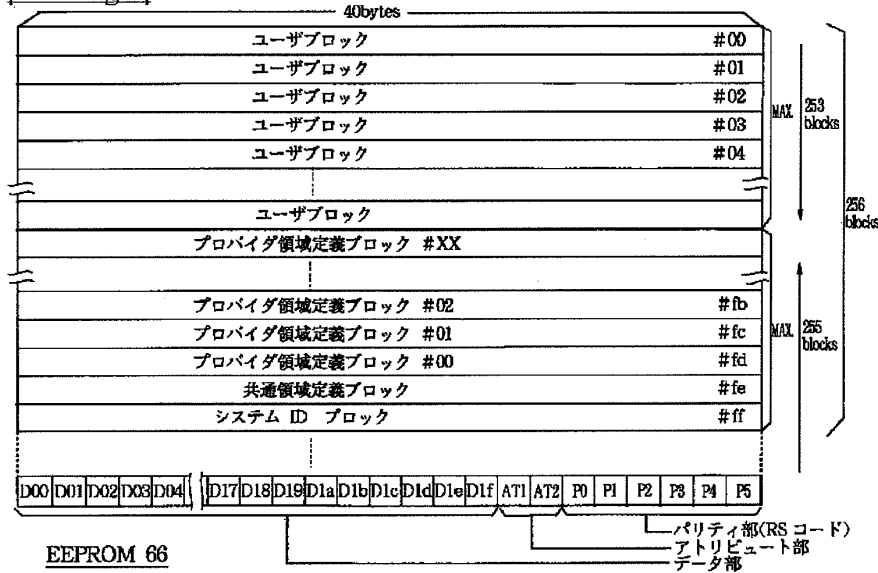


[Drawing 3]

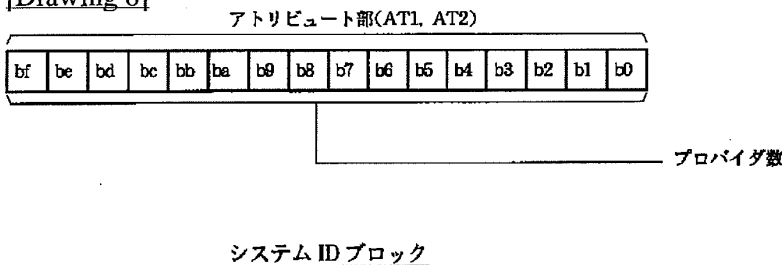


カード 2

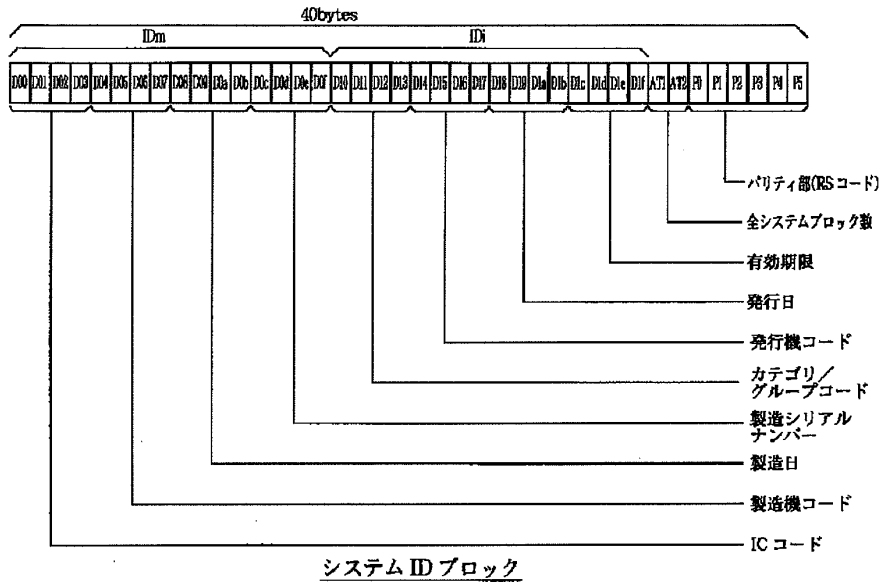
[Drawing 4]



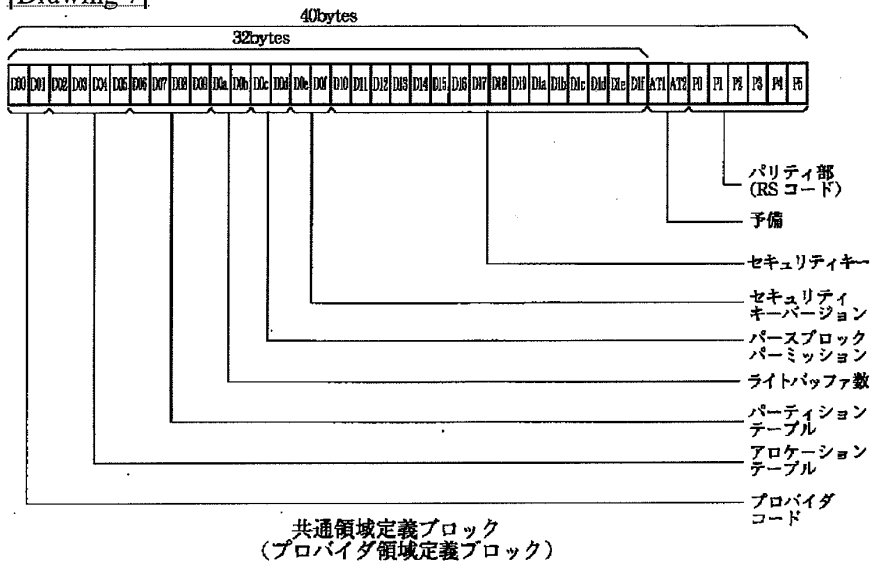
[Drawing 6]



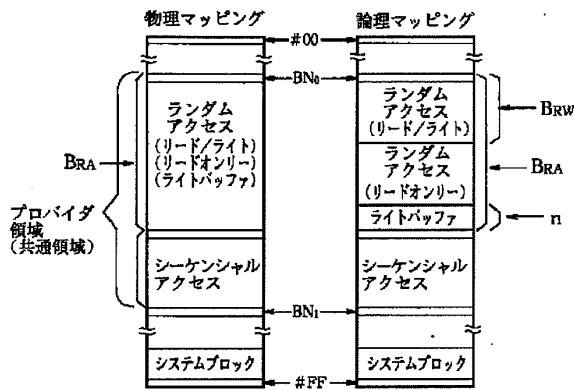
[Drawing 5]



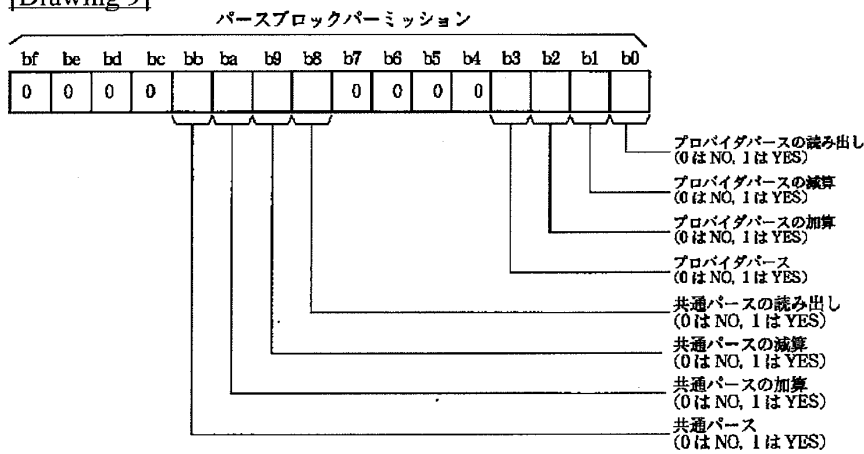
[Drawing 7]



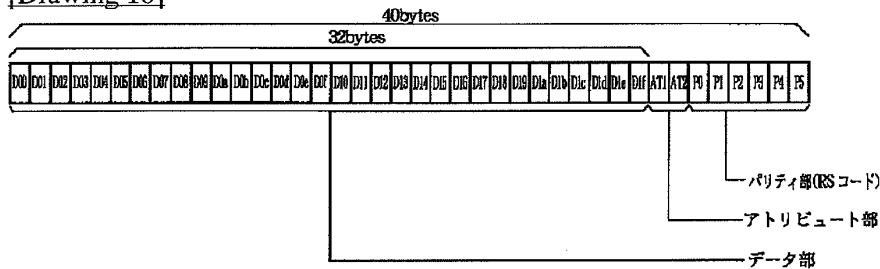
[Drawing 8]



[Drawing 9]

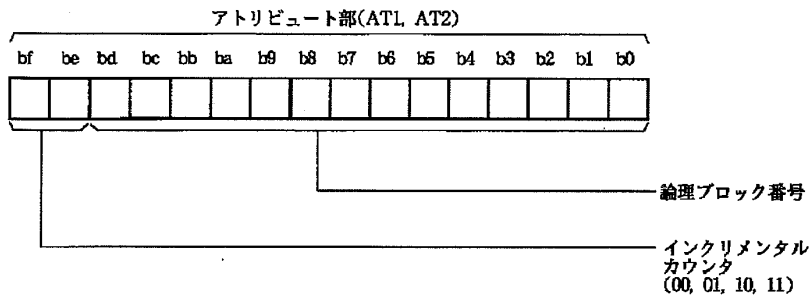


[Drawing 10]

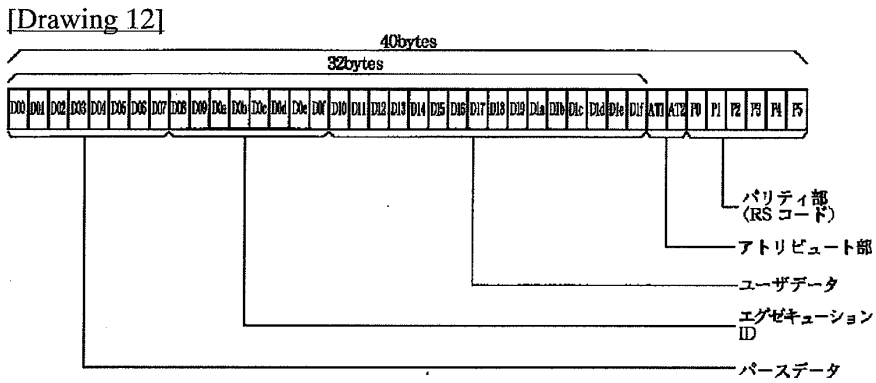


ユーザブロック

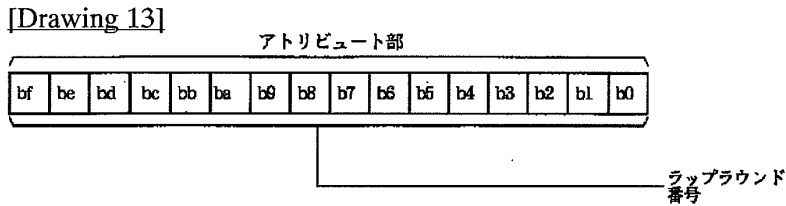
[Drawing 11]



ランダムアクセス領域のユーザブロック

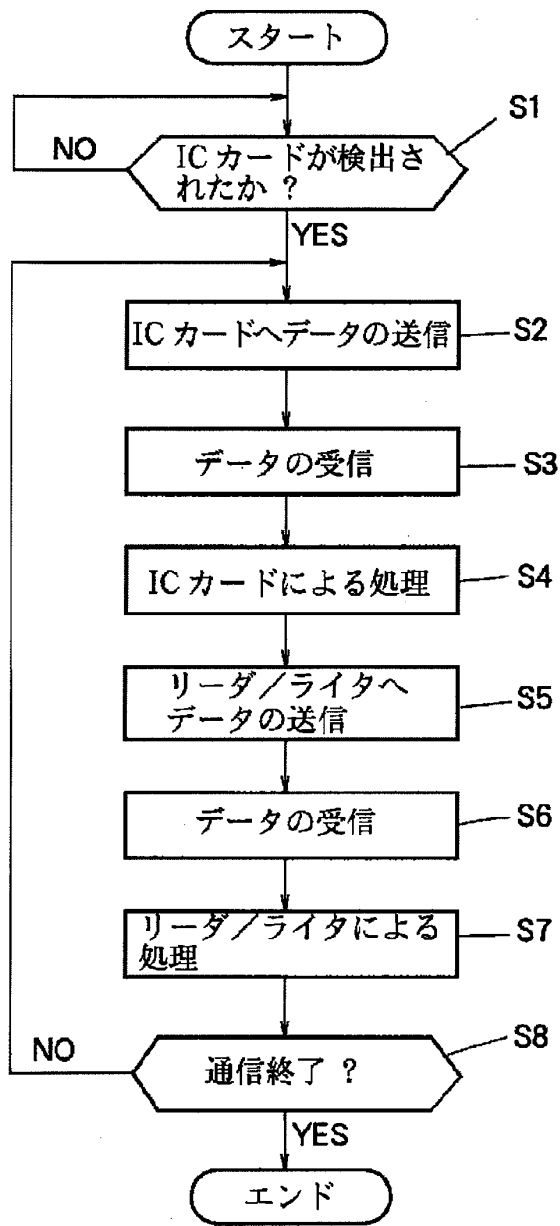


パースブロック

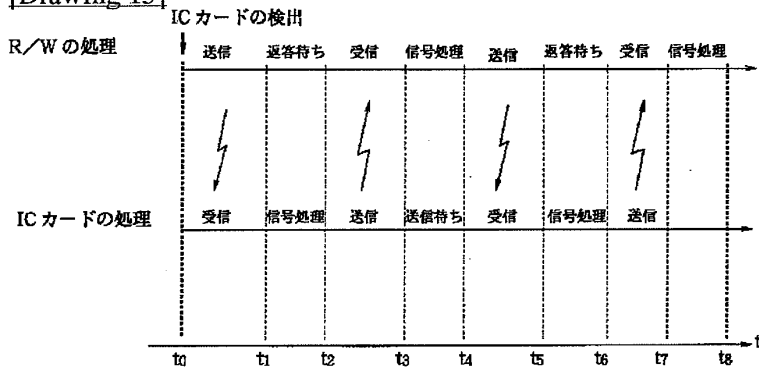


シーケンシャルアクセス領域のユーザブロック

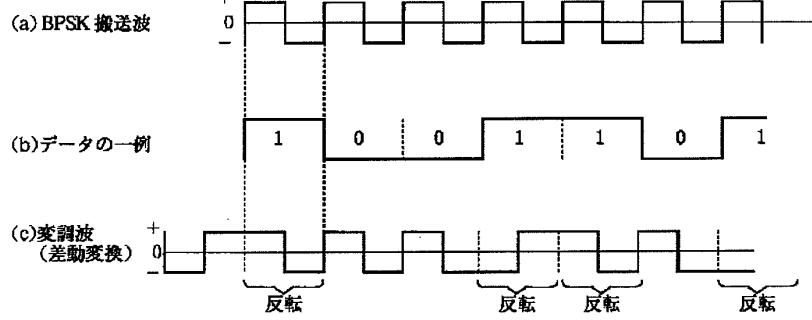
[Drawing 14]



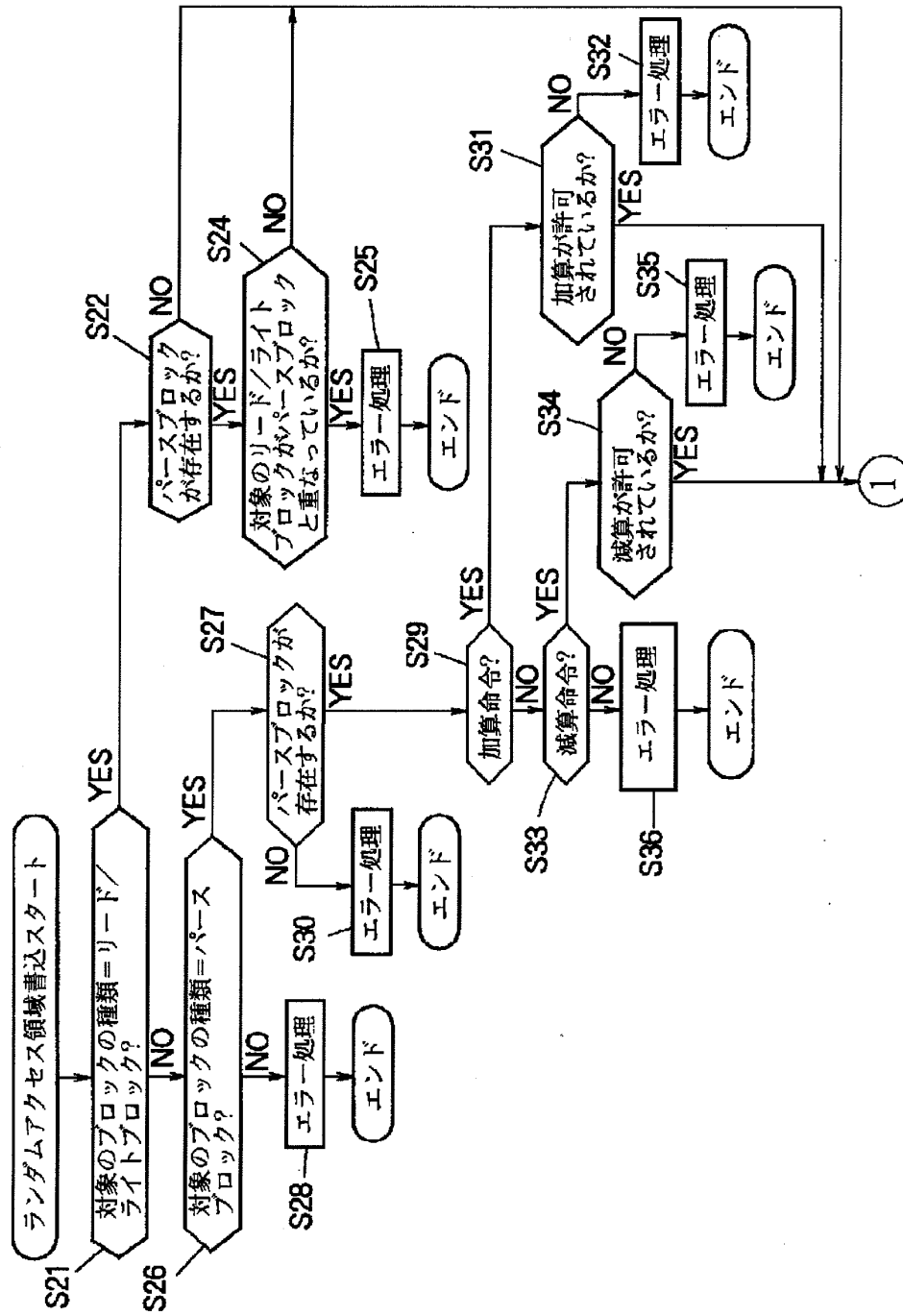
[Drawing 15]



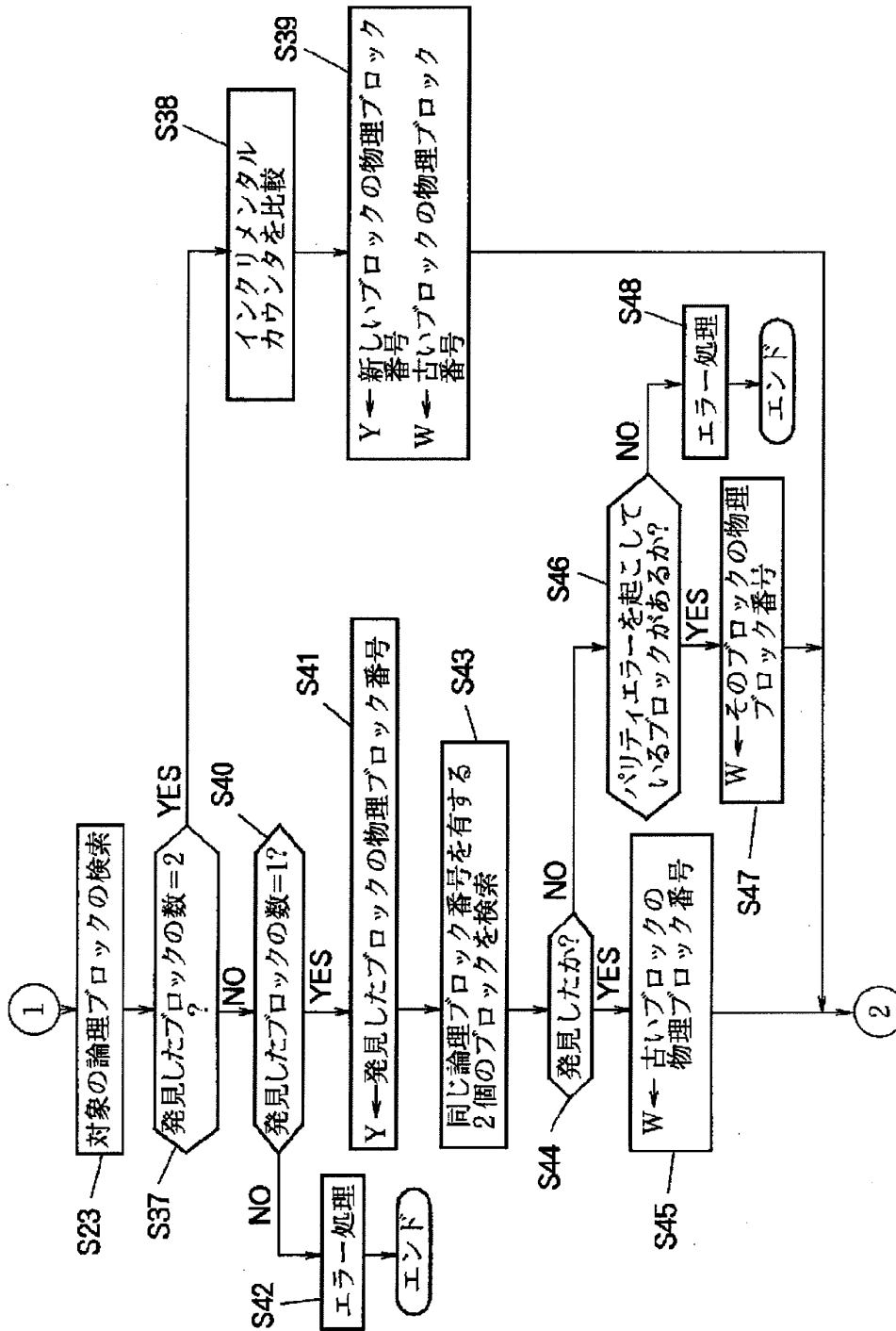
[Drawing 16]



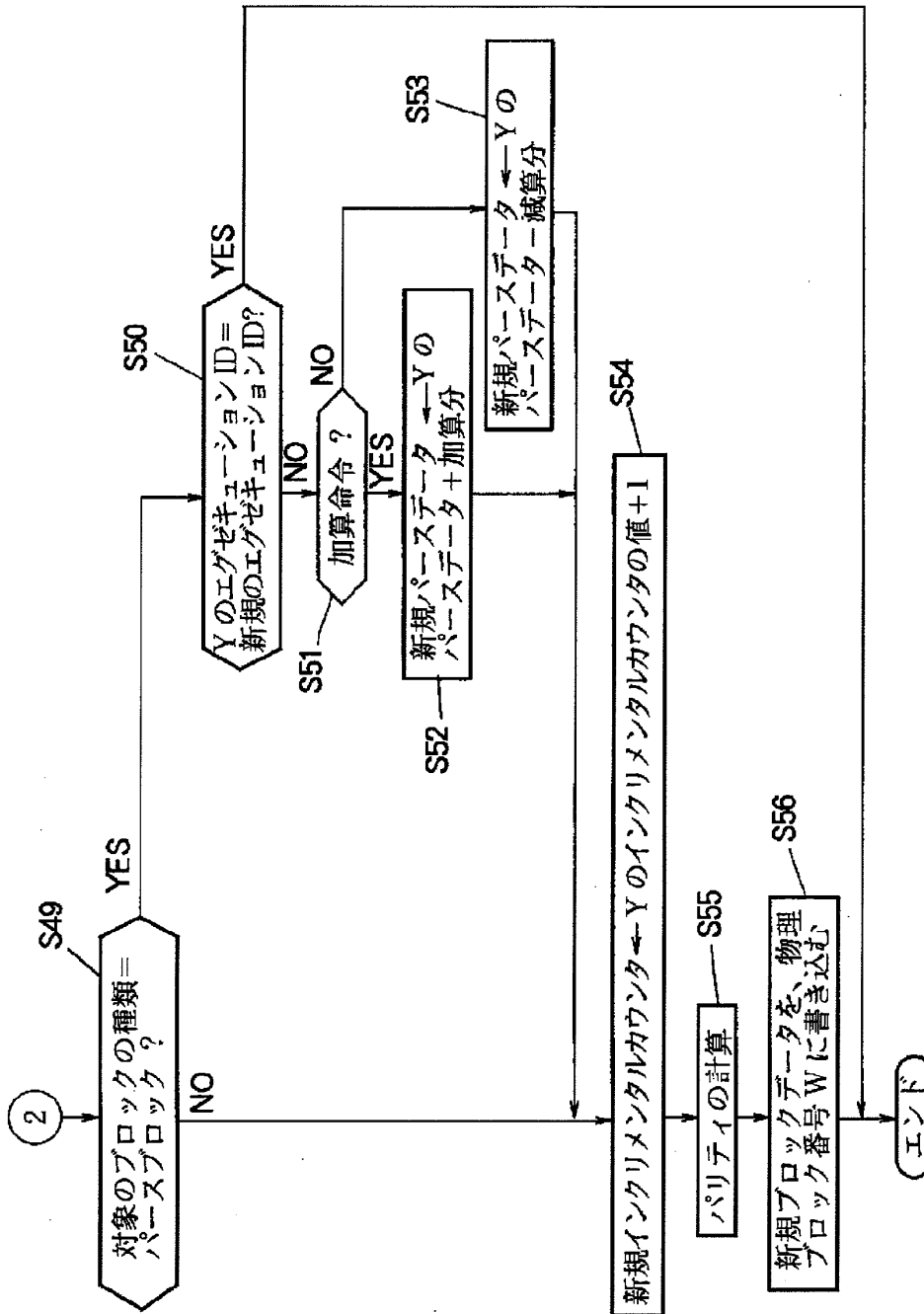
[Drawing 17]



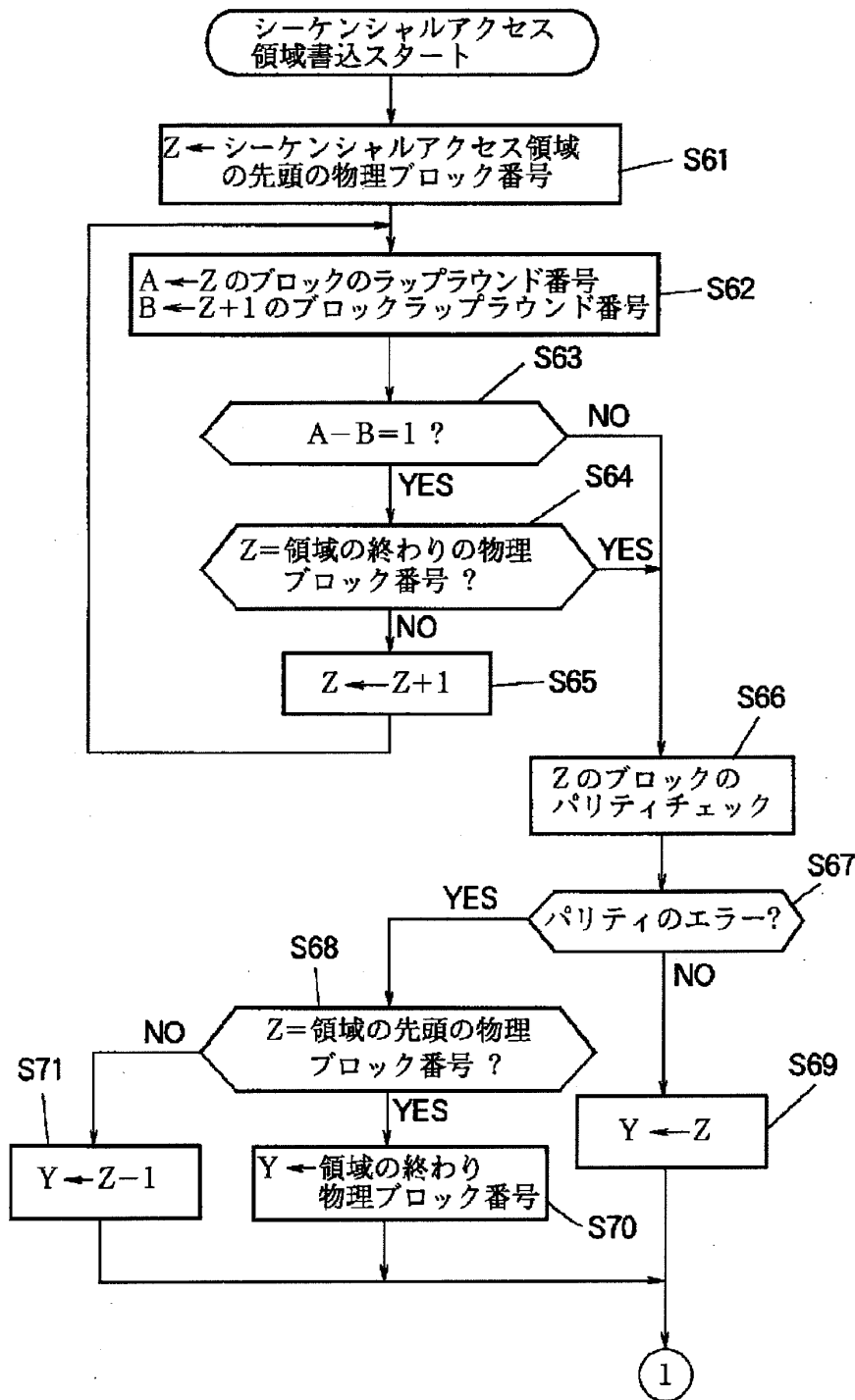
[Drawing 18]



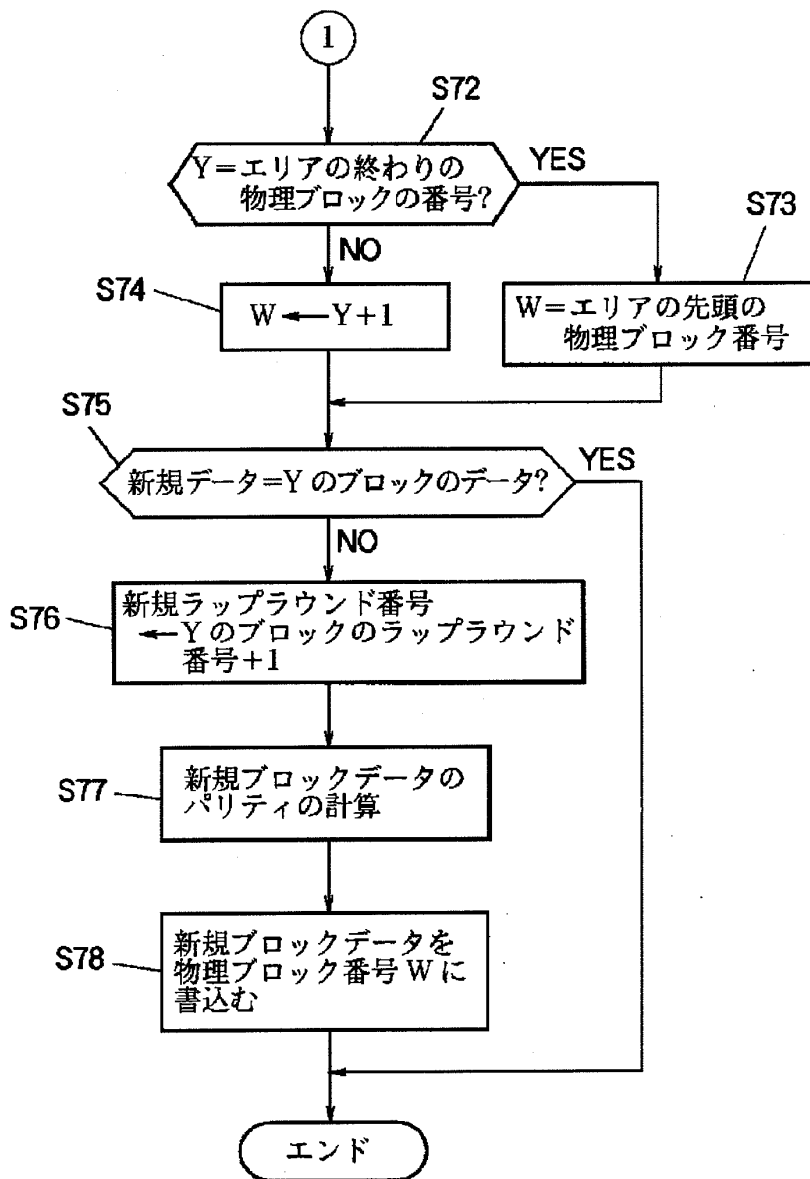
[Drawing 19]



[Drawing 20]



[Drawing 21]



[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

CORRECTION OR AMENDMENT

[Kind of official gazette] Printing of amendment by the convention of 2 of Article 17 of Patent Law
[Section partition] The 3rd partition of the 6th section
[Publication date] May 19, Heisei 17 (2005. 5.19)

[Publication No.] JP,10-105661,A
[Date of Publication] April 24, Heisei 10 (1998. 4.24)
[Application number] Japanese Patent Application No. 9-171430
[The 7th edition of International Patent Classification]

G06K 17/00
G06F 3/06
G06F 3/08
G06F 12/02
G06K 19/07

[FI]

G06K 17/00	F
G06K 17/00	D
G06F 3/06	301 J
G06F 3/08	C
G06F 12/02	510 A
G06K 19/00	N

[Procedure revision]
[Filing Date] June 25, Heisei 16 (2004. 6.25)
[Procedure amendment 1]
[Document to be Amended] Specification
[Item(s) to be Amended] The name of invention
[Method of Amendment] Modification
[The contents of amendment]
[Title of the Invention] It is a record medium to an information processor and the information processing approach, reader/writer and the access approach, and a list.
[Procedure amendment 2]
[Document to be Amended] Specification
[Item(s) to be Amended] Claim
[Method of Amendment] Modification
[The contents of amendment]
[Claim(s)]
[Claim 1]

A receiving means to receive the command from the equipment of the provider who offers a predetermined system,

A processing means to process said command,

A transmitting means to transmit the result of said processing,

A storage means by which the user block field where one or more user blocks are managed in the unit of predetermined magnitude, and the domain-defined block field which defines use of said user block are formed while memorizing the data which said provider's equipment uses

Preparation,

Said domain-defined block field consists of two or more domain-defined blocks,

The data which specify the access privilege to the data which specify the user block to be used in each domain-defined block, and said user block are memorized,

Said command is processed using said data currently recorded on said domain-defined block field.

The information processor characterized by things.

[Claim 2]

Said user block field is assigned and formed in the free area which is not used as said domain-defined block field.

The information processor according to claim 1 characterized by things.

[Claim 3]

Said two or more domain-defined blocks specify an each of said provider access privilege which is different, respectively.

The information processor according to claim 1 characterized by things.

[Claim 4]

Said command is processed using two or more domain-defined blocks with which the data which specify said access privilege differ.

The information processor according to claim 3 characterized by things.

[Claim 5]

The data which specify said access privilege are data which specify access of either [to the user block of said user block field] read/write or a read-only either.

The information processor according to claim 4 characterized by things.

[Claim 6]

While memorizing the data which the equipment of the provider who offers a predetermined system uses The user block field where one or more user blocks are managed in the unit of predetermined magnitude, It is a storage means by which the domain-defined block field which defines use of said user block is formed. Said domain-defined block field In the information processing approach of the information processor which consists of two or more domain-defined blocks, and equips each domain-defined block with a storage means by which the data which specify the access privilege to the data which specify the user block to be used, and said user block are memorized,

The receiving step which receives the command from said provider's equipment,

The processing step which processes said command using said domain-defined block field,

The transmitting step which transmits the result of said processing

***** -- the information processing approach characterized by things.

[Claim 7]

While memorizing the data which the equipment of the provider who offers a predetermined system uses The user block field where one or more user blocks are managed in the unit of predetermined magnitude, It is a storage means by which the domain-defined block field which defines use of said user block is formed. Said domain-defined block field In the program for information processing of the information processor which consists of two or more domain-defined blocks, and equips each domain-defined block with a storage means by which the data which specify the access privilege to the data which specify the user block to be used, and said user block are memorized,

The reception-control step which controls reception of the command from said provider's equipment,

The processing step which processes said command using said domain-defined block field,

The transmission-control step which controls the transmission as a result of said processing
***** -- the record medium with which the program which the computer characterized by things can read is recorded.

[Claim 8]

It is the reader/writer for accessing said user block field of an information processor equipped with a storage means by which the user block field where the data which the equipment of the provider who one or more user blocks are managed per block of predetermined magnitude, and offers a predetermined system uses are memorized is formed,

A data-processing means to process the received data received from the transmit data transmitted to said information processor, and said information processor,

A modulation means to modulate said transmit data,

A recovery means to restore to said received data

Implication,

Said transmit data is a command for accessing said user block controlled based on the data which specify the access privilege to the data which specify said user block memorized by the domain-defined block of said storage means, and said user block.

Reader/writer characterized by things.

[Claim 9]

Said command is a command for accessing said user block field based on the data which specify said access privilege memorized by said two or more [mutually different] domain-defined blocks.

The information processor according to claim 8 characterized by things.

[Claim 10]

It is the access approach which accesses said user block field of an information processor equipped with a storage means by which the user block field where the data which the equipment of the provider who one or more user blocks are managed per block of predetermined magnitude, and offers a predetermined system uses are memorized is formed,

The appeal step which performs appeal to said information processor,

The transmitting step which transmits a command to said information processor which answered said appeal,

The receiving step which receives the transmitting result of said command processed by said information processor,

The processing step which processes said transmitting result

Implication,

Said transmit data is a command for accessing said user block controlled based on the data which specify the access privilege to the data which specify said user block memorized by the domain-defined block of said storage means, and said user block.

The access approach characterized by things.

[Claim 11]

Said command is a command for accessing said user block field based on the data which specify said access privilege memorized by said two or more [mutually different] domain-defined blocks.

The access approach according to claim 10 characterized by things.

[Claim 12]

The data which specify said access privilege are data which specify access of either [to said user block of said user block field] read/write or a read-only either.

The access approach according to claim 10 characterized by things.

[Claim 13]

It is the program for access processing which accesses said user block field of an information processor equipped with a storage means by which the user block field where the data which the equipment of the provider who one or more user blocks are managed per block of predetermined magnitude, and offers a predetermined system uses are memorized is formed,

The appeal step which performs appeal to said information processor,

The transmission-control step which controls transmission of a command to said information processor which answered said appeal,

The reception-control step which controls the reception of a transmitting result to said command processed by said information processor,

The processing step which processes said transmitting result

Implication,

Said transmit data is a command for accessing said user block controlled based on the data which specify the access privilege to the data which specify said user block memorized by the domain-defined block of said storage means, and said user block.

The record medium with which the program which the computer characterized by things can read is recorded.

[Procedure amendment 3]

[Document to be Amended] Specification

[Item(s) to be Amended] 0001

[Method of Amendment] Modification

[The contents of amendment]

[0001]

[Field of the Invention]

This invention relates to a record medium at the information processor and the information processing approach of receiving the command from a predetermined user in an information processor and the information processing approach, reader/writer and the access approach, and a list especially, processing the command in them, and transmitting the result of processing to them about a record medium, reader/writer and the access approach, and a list.

[Procedure amendment 4]

[Document to be Amended] Specification

[Item(s) to be Amended] 0012

[Method of Amendment] Modification

[The contents of amendment]

[0012]

[Means for Solving the Problem]

A receiving means to receive the command from the equipment of the provider whom an information processor according to claim 1 provides with a predetermined system, While remembering the data which a provider's equipment uses to be a processing means to process a command, and a transmitting means to transmit the result of processing The user block field where one or more user blocks are managed in the unit of predetermined magnitude, A storage means by which the domain-defined block field which defines use of a user block is formed is included. A domain-defined block field It consists of two or more domain-defined blocks. For each domain-defined block The data which specify the access privilege to the data and the user block which specify the user block to be used are memorized, and it is characterized by processing a command using the data currently recorded on the domain-defined block field.

[Procedure amendment 5]

[Document to be Amended] Specification

[Item(s) to be Amended] 0013

[Method of Amendment] Modification

[The contents of amendment]

[0013]

The information processing approach according to claim 6 is characterized by including the receiving step which receives the command from a provider's equipment, the processing step which processes a command using a domain-defined block field, and the transmitting step which transmits the result of processing.

[Procedure amendment 6]

[Document to be Amended] Specification

[Item(s) to be Amended] 0014

[Method of Amendment] Modification

[The contents of amendment]

[0014]

The program of a record medium according to claim 7 is characterized by including the reception-control step which controls reception of the command from a provider's equipment, the processing step which processes a command using a domain-defined block field, and the transmission-control step which controls the transmission as a result of processing.

[Procedure amendment 7]

[Document to be Amended] Specification

[Item(s) to be Amended] 0015

[Method of Amendment] Modification

[The contents of amendment]

[0015]

Transmit data is characterized by to be a command for accessing the user block controlled based on the data which specify the access privilege to the data and the user block which are memorized by the domain-defined block of a storage means, and which specify a user block including a data-processing means process the received data which received from the transmit data and the information processor which reader/writer according to claim 8 transmits to an information processor, a modulation means modulate transmit data, and a recovery means restore to received data.

[Procedure amendment 8]

[Document to be Amended] Specification

[Item(s) to be Amended] 0016

[Method of Amendment] Modification

[The contents of amendment]

[0016]

The appeal step to which the access approach according to claim 10 performs appeal to an information processor, The transmitting step which transmits a command to the information processor which answered appeal, The receiving step which receives the transmitting result of the command processed by the information processor, and the processing step which processes a transmitting result are included. Transmit data It is characterized by being a command for accessing the user block controlled based on the data which specify the access privilege to the data and the user block which are memorized by the domain-defined block of a storage means, and which specify a user block.

[Procedure amendment 9]

[Document to be Amended] Specification

[Item(s) to be Amended] 0017

[Method of Amendment] Modification

[The contents of amendment]

[0017]

The program of a record medium according to claim 13 The appeal step which performs appeal to an information processor, and the transmission-control step which controls transmission of a command to the information processor which answered appeal, The reception-control step which controls the reception of a transmitting result to the command processed by the information processor, and the processing step which processes a transmitting result are included. Transmit data It is characterized by being a command for accessing the user block controlled based on the data which specify the access privilege to the data and the user block which are memorized by the domain-defined block of a storage means, and which specify a user block.

[Procedure amendment 10]

[Document to be Amended] Specification

[Item(s) to be Amended] 0018

[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 11]
[Document to be Amended] Specification
[Item(s) to be Amended] 0019
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 12]
[Document to be Amended] Specification
[Item(s) to be Amended] 0020
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 13]
[Document to be Amended] Specification
[Item(s) to be Amended] 0021
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 14]
[Document to be Amended] Specification
[Item(s) to be Amended] 0022
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 15]
[Document to be Amended] Specification
[Item(s) to be Amended] 0023
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 16]
[Document to be Amended] Specification
[Item(s) to be Amended] 0024
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 17]
[Document to be Amended] Specification
[Item(s) to be Amended] 0025
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 18]
[Document to be Amended] Specification
[Item(s) to be Amended] 0026
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 19]
[Document to be Amended] Specification
[Item(s) to be Amended] 0027
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 20]
[Document to be Amended] Specification
[Item(s) to be Amended] 0028
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 21]
[Document to be Amended] Specification
[Item(s) to be Amended] 0029
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 22]
[Document to be Amended] Specification
[Item(s) to be Amended] 0030
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 23]
[Document to be Amended] Specification
[Item(s) to be Amended] 0031
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 24]
[Document to be Amended] Specification
[Item(s) to be Amended] 0032
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 25]
[Document to be Amended] Specification
[Item(s) to be Amended] 0033
[Method of Amendment] Modification
[The contents of amendment]
[0033]

In an information processor according to claim 1, the command from the equipment of the provider who offers a predetermined system is received, a command is processed, the result of processing is transmitted, and while memorizing the data which a provider's equipment uses, the user block field where one or more user blocks are managed in the unit of predetermined magnitude, and the domain-defined block field which defines use of a user block are formed. A domain-defined block field consists of two or more domain-defined blocks, the data which specify the access privilege to the data and the user block which specify the user block to be used for each domain-defined block are memorized, and a command is processed using the data currently recorded on the domain-defined block field.

[Procedure amendment 26]

[Document to be Amended] Specification
[Item(s) to be Amended] 0034
[Method of Amendment] Modification
[The contents of amendment]
[0034]

In the information processing approach and a record medium according to claim 7 according to claim 6, the command from a provider's equipment is received, a command is processed using a domain-defined block field, and the result of processing is transmitted.

[Procedure amendment 27]
[Document to be Amended] Specification
[Item(s) to be Amended] 0035
[Method of Amendment] Modification
[The contents of amendment]
[0035]

In reader/writer according to claim 8, processing of the received data received from the transmit data and the information processor which are transmitted to an information processor is performed, transmit data is modulated, and received data get over. Moreover, let transmit data be a command for accessing the user block controlled based on the data which specify the access privilege to the data and the user block which are memorized by the domain-defined block of a storage means, and which specify a user block.

[Procedure amendment 28]
[Document to be Amended] Specification
[Item(s) to be Amended] 0036
[Method of Amendment] Modification
[The contents of amendment]
[0036]

In the access approach and a record medium according to claim 13 according to claim 10, appeal is performed to an information processor, a command is transmitted to the information processor which answered appeal, the transmitting result of the command processed by the information processor is received, and a transmitting result is processed. Let transmit data be a command for accessing the user block controlled based on the data which specify the access privilege to the data and the user block which are memorized by the domain-defined block of a storage means, and which specify a user block.

[Procedure amendment 29]
[Document to be Amended] Specification
[Item(s) to be Amended] 0037
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 30]
[Document to be Amended] Specification
[Item(s) to be Amended] 0038
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 31]
[Document to be Amended] Specification
[Item(s) to be Amended] 0039
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 32]

[Document to be Amended] Specification
 [Item(s) to be Amended] 0041
 [Method of Amendment] Modification
 [The contents of amendment]
 [0041]

A receiving means to receive the command from the equipment of the provider whom an information processor according to claim 1 provides with a predetermined system (for example, the antenna 53, RF interface section 61, and the BPSK demodulator circuit 62 of drawing 3), A processing means to process a command (for example, sequence 91 of drawing 3), While remembering the data which a provider's equipment uses to be a transmitting means (for example, the antenna 53, RF interface section 61, and the BPSK modulation circuit 68 of drawing 3) to transmit the result of processing The user block field where one or more user blocks are managed in the unit of predetermined magnitude, A storage means (for example, EEPROM66 of drawing 3) by which the domain-defined block field which defines use of a user block is formed is included. A domain-defined block field It consists of two or more domain-defined blocks. For each domain-defined block The data which specify the access privilege to the data and the user block which specify the user block to be used are memorized, and it is characterized by processing a command using the data currently recorded on the domain-defined block field.

[Procedure amendment 33]
 [Document to be Amended] Specification
 [Item(s) to be Amended] 0042
 [Method of Amendment] Modification
 [The contents of amendment]
 [0042]

Reader/writer according to claim 8 is managed per block of magnitude predetermined in one or more user blocks. A storage means by which the user block field where the data which the equipment of the provider who offers a predetermined system uses are memorized is formed It is the reader/writer which accesses the user block field of an information processor equipped with (for example, EEPROM66 of drawing 3). A data-processing means to process the received data received from the transmit data and the information processor which are transmitted to an information processor (for example, SPU32 of drawing 2), A modulation means (for example, modulation circuit 23 of drawing 2) to modulate transmit data, and a recovery means (for example, demodulator circuit 25 of drawing 2) to restore to received data are included. Transmit data It is characterized by being a command for accessing the user block controlled based on the data which specify the access privilege to the data and the user block which are memorized by the domain-defined block of a storage means, and which specify a user block.

[Procedure amendment 34]
 [Document to be Amended] Specification
 [Item(s) to be Amended] 0043
 [Method of Amendment] Deletion
 [The contents of amendment]

[Procedure amendment 35]
 [Document to be Amended] Specification
 [Item(s) to be Amended] 0044
 [Method of Amendment] Deletion
 [The contents of amendment]

[Procedure amendment 36]
 [Document to be Amended] Specification
 [Item(s) to be Amended] 0045
 [Method of Amendment] Deletion
 [The contents of amendment]

[Procedure amendment 37]
[Document to be Amended] Specification
[Item(s) to be Amended] 0046
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 38]
[Document to be Amended] Specification
[Item(s) to be Amended] 0047
[Method of Amendment] Deletion
[The contents of amendment]

[Procedure amendment 39]
[Document to be Amended] Specification
[Item(s) to be Amended] 0218
[Method of Amendment] Modification
[The contents of amendment]

[0218]

[Effect of the Invention]

Like the above, the command from the equipment of the provider who offers a predetermined system according to the information processor according to claim 1 is received, a command is processed, the result of processing is transmitted, and while memorizing the data which a provider's equipment uses, the user block field where one or more user blocks are managed in the unit of predetermined magnitude, and the domain-defined block field which defines use of a user block are formed. Moreover, a domain-defined block field consists of two or more domain-defined blocks, the data which specify the access privilege to the data and the user block which specify the user block to be used for each domain-defined block are memorized, and since the command was processed using the data currently recorded on the domain-defined block field, two or more access privileges which can be set to a predetermined storage region can be granted to a predetermined user. The same storage region can be assigned to two or more users. A different access privilege in a predetermined storage region can be granted to two or more users.

[Procedure amendment 40]
[Document to be Amended] Specification
[Item(s) to be Amended] 0219
[Method of Amendment] Modification
[The contents of amendment]

[0219]

Since according to the information processing approach and a record medium according to claim 7 according to claim 6 the command from a provider's equipment is received, a command is processed using a domain-defined block field and the result of processing was transmitted, two or more access privileges which can be set to a predetermined storage region can be granted to a predetermined user. The same storage region can be assigned to two or more users. A different access privilege in a predetermined storage region can be granted to two or more users.

[Procedure amendment 41]
[Document to be Amended] Specification
[Item(s) to be Amended] 0220
[Method of Amendment] Modification
[The contents of amendment]

[0220]

According to reader/writer according to claim 8, processing of the received data received from the

transmit data and the information processor which are transmitted to an information processor is performed, transmit data is modulated, and received data get over. Since it was made for transmit data to be a command for accessing the user block controlled based on the data which specify the access privilege to the data and the user block which are memorized by the domain-defined block of a storage means, and which specify a user block, it can grant two or more access privileges which can be set to a predetermined storage region to a predetermined user. The same storage region can be assigned to two or more users. A different access privilege in a predetermined storage region can be granted to two or more users.

[Procedure amendment 42]
 [Document to be Amended] Specification
 [Item(s) to be Amended] 0221
 [Method of Amendment] Modification
 [The contents of amendment]
 [0221]

According to the record medium of the access approach according to claim 10 and claim 13, appeal is performed to an information processor, a command is transmitted to the information processor which answered appeal, the transmitting result of the command processed by the information processor is received, and a transmitting result is processed. Moreover, since it was made for transmit data to be a command for accessing the user block controlled based on the data which specify the access privilege to the data and the user block which are memorized by the domain-defined block of a storage means, and which specify a user block, it can grant two or more access privileges which can be set to a predetermined storage region to a predetermined user. The same storage region can be assigned to two or more users. A different access privilege in a predetermined storage region can be granted to two or more users.

[Procedure amendment 43]
 [Document to be Amended] Specification
 [Item(s) to be Amended] 0222
 [Method of Amendment] Deletion
 [The contents of amendment]

[Procedure amendment 44]
 [Document to be Amended] Specification
 [Item(s) to be Amended] 0223
 [Method of Amendment] Deletion
 [The contents of amendment]

[Procedure amendment 45]
 [Document to be Amended] Specification
 [Item(s) to be Amended] 0224
 [Method of Amendment] Deletion
 [The contents of amendment]

[Translation done.]

Electronic Patent Application Fee Transmittal

Application Number:	11250238			
Filing Date:	13-Oct-2005			
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY			
First Named Inventor/Applicant Name:	Kevin M. Conley			
Filer:	Gerald Paul Parsons/Eileen Bowen (GPP)			
Attorney Docket Number:	SNDK.156US2			
Filed as Large Entity				
Utility Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Miscellaneous:				
Request for continued examination	1801	1	790	790
Total in USD (\$)				790

Electronic Acknowledgement Receipt

EFS ID:	1839564
Application Number:	11250238
International Application Number:	
Confirmation Number:	7727
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY
First Named Inventor/Applicant Name:	Kevin M. Conley
Customer Number:	66785
Filer:	Gerald Paul Parsons/Eileen Bowen (GPP)
Filer Authorized By:	Gerald Paul Parsons
Attorney Docket Number:	SNDK.156US2
Receipt Date:	05-JUN-2007
Filing Date:	13-OCT-2005
Time Stamp:	13:35:59
Application Type:	Utility

Payment information:

Submitted with Payment	yes
Payment was successfully received in RAM	\$ 790
RAM confirmation Number	6607
Deposit Account	040258

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)	Multi Part /.zip	Pages (if appl.)
-----------------	----------------------	-----------	------------------	------------------	------------------

1		SNDK156US2_Transmittal_RCE_IDS_6-5-07.pdf	199001	yes	6
Multipart Description/PDF files in .zip description					
	Document Description		Start	End	
	Miscellaneous Incoming Letter		1	1	
	Request for Continued Examination (RCE)		2	2	
	Information Disclosure Statement (IDS) Filed		3	6	
Warnings:					
Information:					
2	Foreign Reference	SNDK156US2_Reference12_WO9844420.pdf	4019372	no	59
Warnings:					
Information:					
3	Foreign Reference	SNDK156US2_Reference13_250876.pdf	695651	no	10
Warnings:					
Information:					
4	Foreign Reference	SNDK156US2_Reference14_WO0249039.pdf	1510193	no	27
Warnings:					
Information:					
5	Foreign Reference	SNDK156US2_Reference15_2742893.pdf	669917	no	15
Warnings:					
Information:					
6	Foreign Reference	SNDK156US2_Reference16_WO9907000.pdf	7572750	no	121
Warnings:					
Information:					
7	Foreign Reference	SNDK156US2_Reference17_0977121.pdf	2095210	no	33
Warnings:					
Information:					

8	Foreign Reference	SNDK156US2_Reference18_0896280.pdf	5517612	no	73
Warnings:					
Information:					
9	Foreign Reference	SNDK156US2_Reference19_10-091490.pdf	465033	no	8
Warnings:					
Information:					
10	Foreign Reference	SNDK156US2_Reference20_10-105661.pdf	6184559	no	73
Warnings:					
Information:					
11	Foreign Reference	SNDK156US2_Reference21_08-221223.pdf	986149	no	18
Warnings:					
Information:					
12	Foreign Reference	SNDK156US2_Reference22_08-221223.pdf	2315078	no	34
Warnings:					
Information:					
13	Foreign Reference	SNDK156US2_Reference23_WO9921093.pdf	3838346	no	84
Warnings:					
Information:					
14	Foreign Reference	SNDK156US2_Reference24_06-250798.pdf	1078548	no	18
Warnings:					
Information:					
15	Foreign Reference	SNDK156US2_Reference25_06-250798.pdf	2438394	no	33
Warnings:					
Information:					
16	Foreign Reference	SNDK156US2_Reference26_10-105661.pdf	2964975	no	45
Warnings:					
Information:					

17	Foreign Reference	SNDK156US2_Reference27_10-105661.pdf	6184559	no	73
Warnings:					
Information:					
18	NPL Documents	SNDK156US2_Reference28_NoticeOpposition.pdf	590238	no	20
Warnings:					
Information:					
19	NPL Documents	SNDK156US2_Reference_Japanese_OA.pdf	549168	no	10
Warnings:					
Information:					
20	Fee Worksheet (PTO-06)	fee-info.pdf	8226	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			49882979		
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					

Davis Wright Tremaine LLP



ANCHORAGE BELLEVUE LOS ANGELES NEW YORK PORTLAND SAN FRANCISCO SEATTLE SHANGHAI WASHINGTON, D.C.

SUITE 800
505 MONTGOMERY STREET
SAN FRANCISCO, CA 94111-6533

TEL (415) 276-6500
FAX (415) 276-6599
www.dwt.com

June 5, 2007

Mail Stop
Commissioner For Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Customer No. 66785

Re: Applicant(s): Conley
Title: Block Data Programming and Reading Operations in a Non-Volatile Memory
Application No.: 11/250,238 Filing Date: October 13, 2005
Examiner: Ngoc V. Dinh Group Art Unit: 2189
Docket No.: SNDK.156US2 Conf. No.: 7727

Dear Sir:

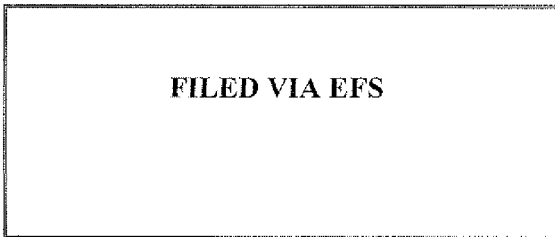
Transmitted herewith are the following documents in the above-identified application:

- (1) This Transmittal Letter (1 page);
- (2) Request for Continued Examination (1 page);
- (3) Information Disclosure Statement (2 pages);
- (4) PTO Form 1449 (2 pages); and
- (5) 18 references enclosed.

- No additional fee is required.
- The fee has been calculated as shown below:

<input checked="" type="checkbox"/>	Fee for Request for Continued Examination	\$ 790.00
Total additional fee:		\$ 790.00

- Conditional Petition for Extension of Time: If an extension of time is required for timely filing of the enclosed document(s) after all papers filed with this transmittal have been considered, an extension of time is hereby requested.
- The fee of \$790.00 has been authorized via EFS to Deposit Account 502664. The Commissioner is hereby authorized to charge any additional fees, which may be required, or credit any overpayment to Deposit Account 502664.



Respectfully submitted,

Gerald P. Parsons
Reg. No. 24,486

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1	"5924092".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/06/11 16:56
L2	1	"5388083".pn..pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/06/11 16:56
L3	1	"5479638".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/06/11 16:56
L4	1	"5568439".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/06/11 16:57
L5	1	"5835935".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/06/11 16:57
L6	1	"5838614".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/06/11 16:57
L7	1	"5860124".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/06/11 16:57
L8	1	"5924113".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/06/11 16:57
L9	1	"5937425".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/06/11 16:58
L10	1	"5598370".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/06/11 16:58
L11	1	"6023423".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/06/11 16:58

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Kevin M. Conley
Title: Partial Block Data Programming and Reading Operations in a Non-Volatile Memory
Application No.: 11/250,238 Filing Date: October 13, 2005
Examiner: Dinh, Ngoc V. Group Art Unit: 2189
Docket No.: SNDK.156US2 Conf. No.: 7727

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

VOLUNTARY AMENDMENT

Sir:

This Voluntary Amendment is being filed in the above-referenced application, in which a Request for Continued Examination was filed June 5, 2007, in order to amend the claims.

Claim Amendments are reflected in the listing of claims, which begins on page 2 of this paper.

Remarks begin on page 7 of this paper.

Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

CLAIM AMENDMENTS

Please amend claim 4, cancel dependent claims 5 and 6 and add new claims 19-25, all without prejudice, as indicated on the following listing of all the claims in the present application after this Amendment:

1. - 3. (Cancelled)

4. (Currently Amended) A method of operating a memory system having an array of reprogrammable non-volatile charge storage elements organized in blocks of storage elements that are erasable together and in pages of storage elements within the blocks that are individually programmable as a unit, comprising:

as part of writing data into pages, recording an indication of a time from a clock source that data are written into individual pages,

when updating data previously written into one or more initial pages, writing the updated data into one or more update pages and identify the initial and update data pages by common logical addresses, and

when reading data of two or more pages having the same logical addresses, read the indications of the times that data have been stored in the two or more pages and use the data in the two or more pages having more recent time indications without using data in the two or more pages having older time indications.

5. (Cancelled)

6. (Cancelled)

7. (Previously Presented) The method of claim 4, wherein recording the indication of the time that data are written into individual pages includes recording the indication within the pages wherein the data are written.

Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

8. (Previously Presented) The method of claim 4, wherein updating data previously written into one or more initial pages is limited to updating data in a number of pages less than all of the pages of a block while not updating data in a remaining one or more pages of the same block.

9. (Previously Presented) The method of claim 8, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

10. (Previously Presented) The method of claim 4, wherein as part of writing data into pages, logical addresses of the individual pages in which the data are written are also written in the individual pages.

11. (Previously Presented) The method of claim 4, wherein as part of writing data into pages, data are written in individual storage elements of the pages with more than two storage states, thereby storing more than one bit of data in the individual storage elements.

12. (Previously Presented) The method of claim 4, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

13. (Previously Presented) A method of operating a memory system having an array of reprogrammable non-volatile charge storage elements organized in blocks of storage elements that are erasable together and in pages of storage elements within the blocks that are individually programmable as a unit, comprising:

as part of writing data into pages, data are written into the pages of the blocks in sequence,

updating data previously written into one or more initial pages by writing the updated data into one or more update pages and identify the initial and update data pages by common logical addresses, and

reading data from the pages of the blocks in an order that is a reverse of the sequence in which they were written and ignore data in any page having the same logical address as a page from which data have already been read.

14. (Previously Presented) The method of claim 13, wherein updating data previously written into one or more initial pages is limited to updating data in a number of pages less than all of the pages of a block while not updating data in a remaining one or more pages of the same block.

15. (Previously Presented) The method of claim 14, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

16. (Previously Presented) The method of claim 13, wherein as part of writing data into pages, logical addresses of the individual pages in which the data are written are also written in the individual pages.

17. (Previously Presented) The method of claim 13, wherein as part of writing data into pages, data are written in individual storage elements of the pages with more than two storage states, thereby storing more than one bit of data in the individual storage elements.

18. (Previously Presented) The method of claim 13, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

19. (New) A method of operating a memory system having an array of reprogrammable non-volatile charge storage elements organized in blocks of storage elements that are erasable together and in pages of storage elements within the blocks that are individually programmable as a unit, comprising:

as part of writing data into pages, (1) recording an indication of a time that data are written into individual pages and (2) writing the data in individual storage elements of the pages with more than two storage states, thereby storing more than one bit of data in the individual storage elements,

when updating data previously written into one or more initial pages, writing the updated data into one or more update pages and identify the initial and update data pages by common logical addresses, and

when reading data of two or more pages having the same logical addresses, read the indications of the times that data have been stored in the two or more pages and use the data in the two or more pages having more recent time indications without using data in the two or more pages having older time indications,

wherein the memory system being operated includes a controller connected with the array of charge storage elements and is positioned within an enclosed card having externally accessible electrical contacts connected with the controller.

20. (New) The method of claim 19, wherein recording an indication of a time that data are written into individual pages includes recording a value of a clock within the memory system.

21. (New) The method of claim 19, wherein recording an indication of a time that data are written into individual pages includes recording a different value of a sequence of numbers.

22. (New) The method of claim 19, wherein recording the indication of the time that data are written into individual pages includes recording the indication within the pages wherein the data are written.

23. (New) The method of claim 19, wherein updating data previously written into one or more initial pages is limited to updating data in a number of pages less than all of the pages of a block while not updating data in a remaining one or more pages of the same block.

24. (New) The method of claim 23, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

25. (New) The method of claim 19, wherein as part of writing data into pages, logical addresses of the individual pages in which the data are written are also written in the individual pages.

REMARKS

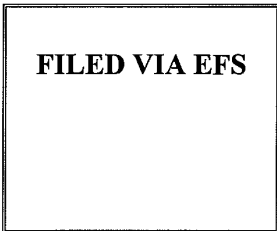
The previously allowed group of claims 4-12 are being amended by adding a limitation to independent claim 4 and cancelling dependent claims 5 and 6. New independent claim 19 combines allowed claims 4 and 11 and introduces other limitations in its last paragraph. Dependent claims 20-25 are taken from allowed dependent claims 5-12.

Information Disclosure Statement

A Supplemental Information Disclosure Statement is being filed herewith to cite a full English translation of Japanese patent publication no. H10-91490 for which only an English abstract was previously cited.

Conclusion

It is believed that this application is now in condition for allowance and an early indication of its allowance is solicited. However, if the Examiner has any further matters that need to be resolved, a telephone call to the undersigned at 415-276-6534 would be appreciated.



Respectfully submitted,


Gerald P. Parsons Date 7/20/07
Reg. No. 24,486

PARSONS HSUE & DE RUNTZ LLP
595 Market Street, Suite 1900
San Francisco, CA 94105
(415) 318-1160 (main)
(415) 318-1163 (direct)
(415) 693-0194 (fax)

Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Conley		
Title:	Block Data Programming and Reading Operations in a Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	October 13, 2005
Examiner:	Ngoc V. Dinh	Group Art Unit:	2189
Docket No.:	SNDK.156US2	Conf. No.:	7727

Mail Stop RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

INFORMATION DISCLOSURE STATEMENT

Dear Sir:

Pursuant to 37 C.F.R. §§ 1.56, 1.97 and 1.98, Applicant(s) call(s) the documents listed on the enclosed Form PTO-1449 to the Examiner's attention in this patent application.

A copy of the listed foreign patent document is enclosed.

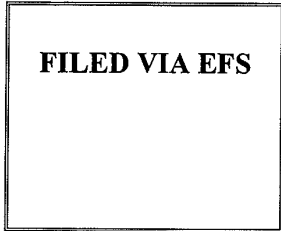
Citation of this document shall not be construed as (1) an admission that the document is prior art with respect to the invention or inventions claimed in this application, (2) a representation that a search has been made (other than as indicated by any cited document), or (3) an admission that the cited information is, or is considered to be, material to patentability as defined in § 1.56(b).

This information disclosure statement is submitted under 37 C.F.R. § 1.97(b) and consequently no fee should be required. The Commissioner is authorized, however, to charge

Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

any fee that may be required, or to credit any overpayment, against Deposit Account No. 04-0258.



Respectfully submitted,

Gerald P. Parsons 7/20/07
Gerald P. Parsons Date
Reg. No. 24,486

DAVIS WRIGHT TREMAINE LLP
505 Montgomery Street, Suite 800
San Francisco, California 94111-6533
Telephone: (415) 276-6500
Facsimile: (415) 276-6599
Email: geraldparsons@dwt.com

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.		Application No.				
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		SNDK.156US2		11/250,238				
		Applicants		Conf. No.				
(Use several sheets if necessary)		Conley		7727				
(Form PTO-1449)		Filing Date		Art Group				
		October 13, 2005		2189				
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
						Translation		
		Document	Date	Country	Class	Subclass	Yes	No
		H10-91490	4/10/98	Japan			X	
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
Examiner			Date Considered					
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

FILED VIA EFS
Sheet 1 of 1

(19) Japan Patent Office (JP) (12) Japanese Unexamined Patent Application Publication (A) (11) Japanese Unexamined Patent Application Publication Number H10-91490

(43) Publication date April 10, 1998

(51) Int. Cl. ⁶	Identification	FI
	symbols	
G06F 12/00	510	G06F 12/00 510A
G11C 16/06		G11C 17/00 510A 530C

Status of examination: Not yet requested Number of claims: 6 OL (Total of 6 pages)

(21) Application number	H8-242124	(71) Applicant	000001889 SANYO Electric Co., Ltd. 5-5, Keihan-hondori 2-chome, Moriguchi-shi, Osaka, Japan
(22) Date of application	September 12, 1996	(72) Applicant	000214892 Tottori SANYO Electric Co., Ltd. 201, 3-chome, Minamiyoshikata, Tottori-shi, Tottori, Japan
		(72) Inventor	Haruaki KANBARA C/o Tottori SANYO Electric Co., Ltd. 201, 3-chome, Minamiyoshikata, Tottori-shi, Tottori, Japan
		(72) Inventor	Shigemi KAHARA C/o Tottori SANYO Electric Co., Ltd. 201, 3-chome, Minamiyoshikata, Tottori-shi, Tottori, Japan
		(74) Representative	Patent Attorney, Koji YASUTOMI (and one other)

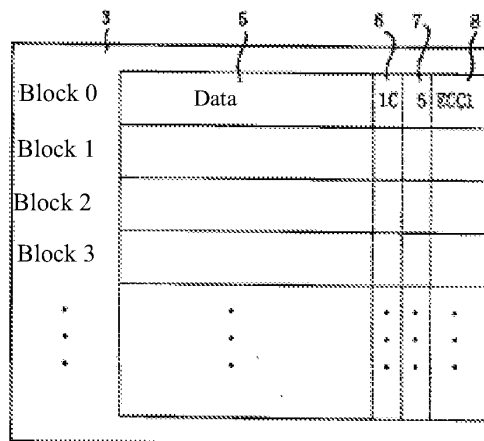
Continued on the last page.

(54) [Title of Invention]
STORAGE DEVICE USING FLASH MEMORY

(57) [Abstract]

PROBLEM TO BE SOLVED: To make it possible to perform data updates in a short time, even for NAND-type flash memories.

SOLUTION: Within a flash memory 3, form a data area 5 for storing data, an allocation area 6 that corresponds to each data area and stores the individual logical addresses for said data area, and a history information area 7 that stores the storage history information for each logical address stored in said area 6.



(2)

[Claims]

1. A storage device using flash memory, wherein a plurality of data areas for storing data, a plurality of allocation areas, which respectively corresponds to each of said data areas, for storing each logical address for each said data area; and a plurality of history information areas, which corresponds to each said data area, for storing storage history information for logical addresses stored in said allocation areas corresponding to each data area that corresponds to each said data area are formed in said flash memory.

2. A storage device using flash memory, comprising: the flash memory having a plurality of data areas for storing data, a plurality of allocation areas, which corresponds to each of said data areas, for storing each logical address for each said data area, and a plurality of history information areas, which corresponds to each said data area, for storing storage history information for logical addresses stored in said allocation areas corresponding to each data area; a free area detection means to detect free data areas in said memory; an area search means to search said allocation areas in which the same logical address data as entered address data is stored; a means to generate new history information according to history information stored in the history information area corresponding to the allocation area found by said search means; and a means to write said address data and new history information generated by said generating means to the allocation area and history information area corresponding to the detected data area detected by said free data area detection means.

3. A storage device using flash memory, wherein said writing means according to Claim 2 writes new data entered along with said address data into the data area detected by said free data area search means.

4. A storage device using flash memory, wherein said flash memory according to any one of Claims 1-3 is of the NAND type.

5. A storage device using flash memory according to Claim 4, wherein said data areas, allocation areas and history information areas that are corresponding to each other are located in the same block, which becomes a writing unit.

6. A storage device using flash memory according to any one of Claims 1-5, wherein said history information indicates how many times the same logical address is stored.

[Detailed Description of the Invention]

[0001]

[Technical Field of the Invention] The present

invention relates to a storage device using flash memory.

[0002]

[Prior Art] Since flash memory can keep its memory contents semi-permanently without battery backup and can access electrically same as semiconductor memory, there are expectations for it to be used as a storage device to replace disk type of memory devices such as FDs (Floppy Disks) or HDs (Hard Disks).

[0003] However, data in flash memory must be deleted as a unit of at least a few Kbytes. To delete or modify one byte data, after moving a few Kbytes data containing the one byte to be modified to another memory such as semiconductor RAM, the few Kbytes area must be deleted and then the moved few Kbytes data must be written onto the flash memory. Therefore, this was a problem since it took too much time.

[0004] Then, in the specification of U.S. patent No. 5,404,485, a method is disclosed to improve data update speed by skipping the process of re-writing the data area, which is accomplished by creating allocation areas corresponding to each data area to store the data in flash memory; by making it possible to store the logical address of the data area corresponding to this allocation area or data indicating that it has been already updated; by writing new data to free data areas, by not re-writing to the area where data has been already stored, when re-writing the already stored data in the data area, and writing a logical address of the already stored data above into the allocation area corresponding to this data area as well as re-writing data in the allocation area corresponding to the already stored data to indicate that the data is already updated.

[0005]

[Problem to be Solved by the Invention] However, the method disclosed in this U.S. patent is for the flash memory generally called the NOR type in which data is written per byte, and cannot be applied to the NAND type of flash memory wherein the writing unit is a few hundreds bytes, while the deleting unit is a few Kbytes.

[0006] In other word, flash memory is characterized in that for both NOR type and NAND type, it is easy to change its state from the non-charge state logic "1" to the charge state logic "0" compared to the so called "data deleting" of changing its state from charge to non-charge. Therefore, if the above data indicating it is already updated is defined as all "0"s data, for an example, it is easy to change the logical address data in the allocation area to all "0"s of already updated data for the NOR type.

(3)

However, since data cannot be written per one byte unit, the invention described in the U.S. patent above cannot be employed for the NAND type.

[0007]

[Means for Solving the Problem] The present invention is born by considering the problem above and basically characterized in that a plurality of data areas for storing data, a plurality of allocation areas, which corresponds to each of said data areas, for storing each logical address for each of said data areas, and a plurality of history information areas, which corresponds to said each of data areas, for storing storage history information for logical addresses stored in said allocation areas corresponding to each data area are formed in flash memory.

[0008]

[Embodiment of the Invention] Fig. 1 is a block diagram indicating the main circuit configuration of a system that the present invention is applied to. The main control unit 1 comprising micro computer, for an example, accesses to the semiconductor random access memory RAM2 based on built-in control programs or controls the flash memory control unit 4 which controls the flash memory 3.

[0009] The flash memory 3 according to this embodiment is the NAND type which only allows to write by a block unit comprising (512+16) bytes as shown in Fig. 2 and to delete by a unit comprising 8 blocks (4.125 Kbytes). In addition, each block described above is divided into the data area 5 where 512-byte data is stored, the allocation area 6 where 2-byte logical address is stored, and the history information area 7 where information is stored indicating how many times the same two-byte logical address is stored as shown in Fig. 3; the rest of the 12 bytes are used as storage area 8, such as this ECC data.

[0010] The flash memory control unit 4 reads or writes data into the flash memory 3 described above based on address data or new data supplied under control of the main control unit 1 and comprises dedicated gate arrays and one-chip microcomputers.

[0011] Next the operation of this embodiment is described. Fig. 4 is a flow chart showing the control operation when writing data into the flash memory 3 in the flash memory control unit 4.

[0012] The flash memory control unit 4 performs initialization for writing new data (including updating) in S1 step once an instruction is received from the main control unit 1, which tells to write new data DN into the logical address a.

[0013] Specifically, "0", the head block number,

is set to the variable i to specify each block in the flash memory 3; "-1" is set to the variable cnt to store data in the history information area 8; and "-1" is set to the variable blk to store the block number. However, the variables i, cnt and blk described above are kept in the RAM (not shown) in the flash memory control unit 4.

[0014] In the following S2 step, whether the logical address stored in the allocation area (hereinafter called alloc [i]) in the block i in the flash memory 3 is matched to the logical address a or not is decided.

[0015] If they are matched, in the S3 step, the data in the history information area (hereinafter called count [i]) in the block i in the flash memory 3 is compared to the variable cnt. If "count [i] > cnt", in the S4 step, the value of the variable blk is replaced with the value of the variable i as well as the value of the variable cnt is replaced with the value of the count [i], and then the process proceeds to the S5 step.

[0016] On the other hand, if it is decided that they are not matched in the S2 step, and if it is not "count [i] > cnt" in the S3 step, then the process proceeds to the S5 step immediately.

[0017] In the S5 step, the variable i is incremented, and in the following S6 step, whether the value of the variable i is more than the number of blocks (max blk) in the flash memory 3 is tested. If it is decided to be less, the process goes back to the S2 step.

[0018] On the other hand, if it is decided to be more, since the comparison between the logical addresses stored in the allocation area 6 for all the blocks in the flash memory 3 and the logical address a is finished, the process proceeds to the S7 step to finish the last process.

[0019] In the S7 step, whether the value of the variable blk is "-1" or not is decided. If the value of the variable blk is "-1", "0" is set to the variable cnt in the S8 step. On the other hand, if the value of the variable blk is not "-1", after the value of the variable cnt is incremented in the S9 step, the process proceeds to the S10 step.

[0020] In the S10 step, free blocks are detected in the flash memory 3, and new data DN, the logical address a and the value of the variable cnt are written into the data area 5, allocation area 6 and the history information area 8 respectively in the block to finish the process.

[0021] The free block detection described above is performed by detecting the non charge state logic "1" since the data in the block in the flash memory 3 described above is the non charge state logic "1" in a deleted state in all areas. In addition, since all areas for writing are located on the same

(4)

block, it is easily written even for the NAND type.

[0022] Next, to understand the control operation shown in the Fig. 4 flow chart more easily, a specific example is used to describe.

[0023] Now as shown in Fig. 5, there are 8 blocks, from block 0 to 7, in the flash memory 3, and all areas in all blocks are deleted and are all "1"s.

[0024] In this state, if an instruction is received to write new data d1 into the logical address a1 from the main control unit 1, the flash memory control unit 4 executes the flow shown in Fig. 4. However, at this timing, since all the blocks are free, the S2, S5 and S6 steps are executed and repeated 8 times, then, the process proceeds to the S7 step. In addition, since the variable blk is still "-1" at this timing, the S8 and S10 steps are processed sequentially and the data d1, the logical address a1 and the value "0" of the variable cnt are stored in the data area 5, the allocation area 6 and the history information area 7 in block 0, which is a free block as shown in Fig. 6 respectively.

[0025] Then if an instruction is received to write new data d2 and d3 (may be $d1 = d2 = d3$) into different logical addresses a2 and a3, since no same address is existed in the allocation area 6 in the flash memory 3, only S1, S2, S5, S6, S7, S8 and S10 steps are processed in the same way as the case described above, and the data d2 and d3 and the address a2 and a3 are stored in the block 1 and 2 which were free areas, as shown in Fig. 7.

[0026] Next, if an instruction is received to write new data d4 into the logical address a1 from the main control unit 1, since only the logical address (alloc [0]) in the allocation area in the block 0 is "a1", when this is tested in the S2 step, a value "0" is set in the variable blk via the process in S3 and S4 steps, which indicates the block number in the block 0 while "0" is set to the variable cnt, which is the value in the history information area in the block 0. Therefore, after testing for all blocks in the S2-S6 steps, since the S7, S9 and S10 steps are processed sequentially, the data d4, the logical address a1 and the value "1" are written into the data area 5, the allocation area 6 and the history information area 7 respectively in the block 3 as shown in Fig. 8.

[0027] Moreover, if an instruction is received again and again to write the new data d5 into the logical address a1 from the main control unit 1, when the variable i becomes "0" and "3", it is decided that it is a match in the S2 step, when the variable i is "0", the value "0" is set to the variables blk and cnt, and when the variable i is "0", the values "3" and "1" are set to the variables blk and cnt respectively. Therefore, as a

result of processing the S7 and following steps, as shown in Fig. 9, in the data area 5, the allocation area 6 and the history information area 7 in the block 4, the data d5, the logical address a1 and the value "2" are written respectively.

[0028] As described above, in the device according to this embodiment, new and update data are written sequentially into free blocks together with logical addresses, and whether the update data is new or old can be decided by the value in the history information area 7 in each block.

[0029] Fig. 10 is a flow chart showing data read control operation for the flash memory 3 in the flash memory control unit 4.

[0030] The flash memory control unit 4 processes the S20 and following steps shown in Fig. 10 once an instruction is received to read data corresponding to the logical address a from the main control unit 1.

[0031] In the S20 to S25 steps, all blocks are searched sequentially in the flash memory 3 same as the S1-S6 steps described above, a block is detected in which same address is stored as the logical address a above in the allocation area 6, and the block number and data (count [i]) in the history information area 7 are set to the variables blk and cnt respectively. If multiple blocks are detected, one with maximum history information value, i.e. the block where the latest update data is stored is set.

[0032] In the S26 step, the value of variable blk is tested same as the S7 step. At this timing, if the variable blk is still "-1" initialized in the S20 step, which means the block in which logical address a is stored was not found, this is processed as an error.

[0033] On the other hand, if the value of the variable blk is other than "-1", which means the block in which logical address a is stored was found, and changed to the block number in the S23 step described above, so data in the data area 5 in the block of the same block number as the variable blk is read and output to the main control unit 1 to finish the process in the following S27 step.

[0034]

[Effect of the Invention] According to the present invention, since it is possible to modify data without requiring deleting in the NAND type of flash memory, data update speed can be improved.

[Brief Description of Drawings]

[Fig. 1] A block circuit diagram showing the main section of a device that the present invention is applied to.

[Fig. 2] A view showing the configuration of

NAND type of flash memory intended for the present invention is intended to.

[Fig. 3] A view showing the flash memory in one embodiment of the present invention.

[Fig. 4] A flow chart for describing the operation in one embodiment of the present invention.

[Fig. 5] A view showing the flash memory in one embodiment of the present invention.

[Fig. 6] A view showing the flash memory in one embodiment of the present invention.

[Fig. 7] A view showing the flash memory in one embodiment of the present invention.

[Fig. 8] A view showing the flash memory in one embodiment of the present invention.

[Fig. 9] A view showing the flash memory in one embodiment of the present invention.

[Fig. 10] A flow chart for describing the operation in one embodiment of the present invention.

[Description of Symbols]

3 Flash memory

4 Flash memory control unit

5 Data area

6 Allocation area

*7 History information area

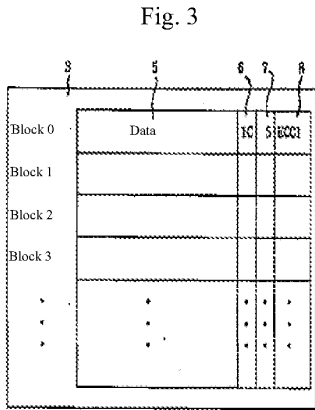
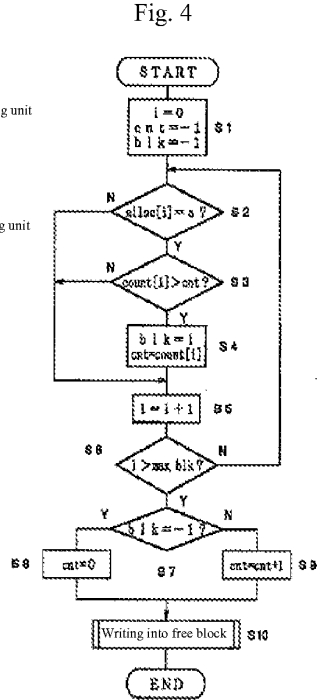
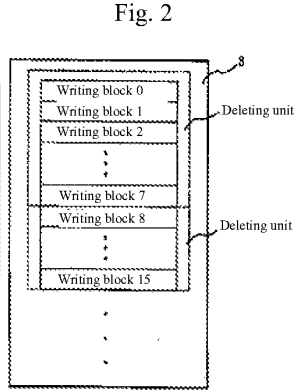
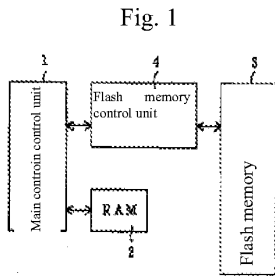


Fig. 6

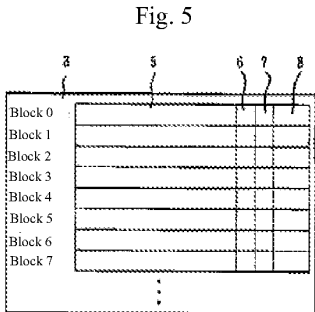
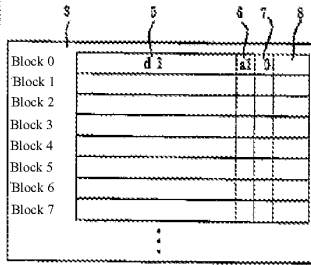


Fig. 8

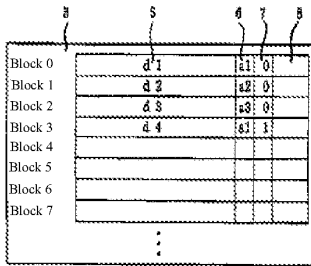


Fig. 7

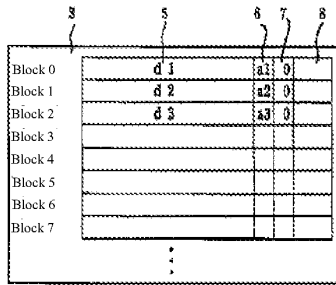


Fig. 9

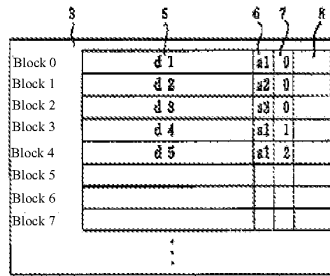
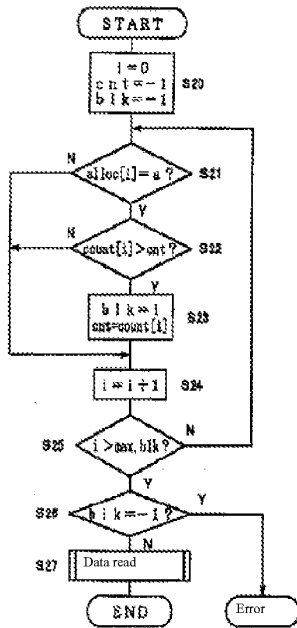


Fig. 10



Continued from the front page.

(72) Inventor Takahiro YAMASHITA
C/o Tottori SANYO Electric Co., Ltd.
201, 3-chome, Minamiyoshikata,
Tottori-shi, Tottori, Japan

(72) Inventor Masaya AMISAKI
C/o Tottori SANYO Electric Co., Ltd.
201, 3-chome, Minamiyoshikata,
Tottori-shi, Tottori, Japan

(72) Inventor Isao UMEZAWA
C/o Tottori SANYO Electric Co.,
Ltd.
201, 3-chome, Minamiyoshikata,
Tottori-shi, Tottori, Japan

(72) Inventor Fumitaka URAKAWA
C/o Tottori SANYO Electric Co.,
Ltd.
201, 3-chome, Minamiyoshikata,
Tottori-shi, Tottori, Japan

(72) Inventor Koji MAEDA
C/o Tottori SANYO Electric Co.,
Ltd.
201, 3-chome, Minamiyoshikata,
Tottori-shi, Tottori, Japan

Electronic Acknowledgement Receipt

EFS ID:	1995185
Application Number:	11250238
International Application Number:	
Confirmation Number:	7727
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY
First Named Inventor/Applicant Name:	Kevin M. Conley
Customer Number:	66785
Filer:	Gerald Paul Parsons/Eileen Bowen (GPP)
Filer Authorized By:	Gerald Paul Parsons
Attorney Docket Number:	SNDK.156US2
Receipt Date:	20-JUL-2007
Filing Date:	13-OCT-2005
Time Stamp:	15:50:10
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no
------------------------	----

File Listing:

Document Number	Document Description	File Name	File Size(Bytes) /Message Digest	Multi Part /.zip	Pages (if appl.)
1		SND156US2_Transmittal_Voluntary_Amendment_IDS_7-20-07.pdf	403748 <small>18d1d9eb3ba0acf5e5d0b6f131b9cda1039627be</small>	yes	11

Multipart Description/PDF files in .zip description					
Document Description			Start	End	
Miscellaneous Incoming Letter			1	1	
Preliminary Amendment			2	2	
Claims			3	7	
Applicant Arguments/Remarks Made in an Amendment			8	8	
Information Disclosure Statement (IDS) Filed			9	11	
Warnings:					
Information:					
2	Foreign Reference	SNDK156US2_Reference_J P10-091490_English_Transl ation1.pdf	128045	no	7
			bdbcc681b1c5e0e83a123671c6fd3e23 b9ee4c60		
Warnings:					
Information:					
Total Files Size (in bytes):			531793		
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					

Davis Wright Tremaine LLP



ANCHORAGE BELLEVUE LOS ANGELES NEW YORK PORTLAND SAN FRANCISCO SEATTLE SHANGHAI WASHINGTON, D.C.

SUITE 800
505 MONTGOMERY STREET
SAN FRANCISCO, CA 94111-6533

TEL (415) 276-6500
FAX (415) 276-6599
www.dwt.com

July 20, 2007

Mail Stop Amendment
Commissioner For Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Customer No. 66785

Re: Applicant(s): Conley
Title: Partial Block Data Programming and Reading Operations in a Non-Volatile Memory
Application No.: 11/250,238 Filing Date: October 13, 2005
Examiner: Dinh, Ngoc V. Group Art Unit: 2189
Docket No.: SNDK.156US2 Conf. No.: 7727

Dear Sir:

Transmitted herewith are the following documents in the above-identified application:

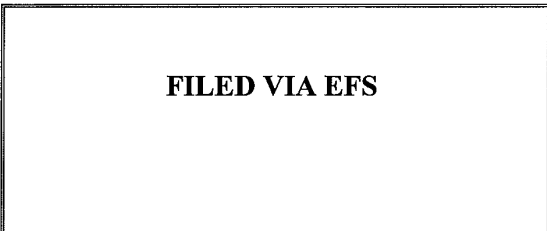
- (1) This Transmittal Letter (1 page); and
(2) Voluntary Amendment (7 pages);
(3) Information Disclosure Statement (2 pages);
(4) PTO Form 1449 (1 page); and
(5) 1 Reference enclosed.

- No additional fee is required.
The fee has been calculated as shown below:

CLAIMS AS AMENDED

Table with columns: Claims Remaining After Amendment, Highest No. Previously Paid For, Present Extra, Rate, Additional Fee. Rows include Total Claims, Independent Claims, and Total additional fee for this Amendment.

The fee of \$... has been authorized via EFS to Deposit Account 040258. The Commissioner is hereby authorized to charge any additional fees, which may be required, or credit any overpayment to Deposit Account 040258.



Respectfully submitted,

[Signature]

Gerald P. Parsons
Reg. No. 24,486

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD
 Substitutely for Form PTO-875 Effective December 1, 2004
 Application or Doctel Number: **11250237**

APPLICATION AS FILED - PART I
 (Column 1) (Column 2)

FOR	NUMBER FILED	NUMBER EXTRA	RATE (\$) SMALL ENTITY	FEE (\$) SMALL ENTITY	RATE (\$) OTHER THAN SMALL ENTITY	FEE (\$) OTHER THAN SMALL ENTITY
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	150.00	N/A	300.00
SEARCH FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	\$250	N/A	\$500
EXAMINATION FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	\$100	N/A	\$200
TOTAL CLAIMS (37 CFR 1.16(d))	3	minus 20 =	X\$ 25		X\$50	
INDEPENDENT CLAIMS (37 CFR 1.16(d))		minus 3 =	X100		X200	
APPLICATION SIZE FEE (37 CFR 1.16(e))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$250 (\$125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(f).					
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(i))			+180=		+360=	
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL		TOTAL	1000

APPLICATION AS AMENDED - PART II
 (Column 1) (Column 2) (Column 3)

AMENDMENT A	CLAMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$) SMALL ENTITY	ADDITIONAL FEE (\$) SMALL ENTITY	RATE (\$) OTHER THAN SMALL ENTITY	ADDITIONAL FEE (\$) OTHER THAN SMALL ENTITY
Total (37 CFR 1.16(d))	15	Minus	20	X\$ 25		X\$50	
Independent (37 CFR 1.16(d))	2	Minus	3	X100		X200	
Application Size Fee (37 CFR 1.16(f))							
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(i))				+180=		+360=	
			TOTAL ADD'L FEE		TOTAL ADD'L FEE		

AMENDMENT B
 (Column 1) (Column 2) (Column 3)

AMENDMENT B	CLAMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$) SMALL ENTITY	ADDITIONAL FEE (\$) SMALL ENTITY	RATE (\$) OTHER THAN SMALL ENTITY	ADDITIONAL FEE (\$) OTHER THAN SMALL ENTITY
Total (37 CFR 1.16(d))	20	Minus	20	X\$ 25		X\$50	
Independent (37 CFR 1.16(d))	3	Minus	3	X100		X200	
Application Size Fee (37 CFR 1.16(f))							
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(i))				+180=		+360=	
			TOTAL ADD'L FEE		TOTAL ADD'L FEE		

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
 * If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
 * If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
 The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.
 This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1460, Alexandria, VA 22313-1460.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L3	1	"6968421".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 13:48
L4	1	"6763424".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 13:48
L5	1463	("same" identical common) near5 (logical adj2 address)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 14:58
L6	378641	prom eeprom flash	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 14:58
L7	16	(out\$of\$order (reverse near3 order)) same 5	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:06
L8	3227	711/103	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 14:59
L9	7240	((original old) adj2 data) with ((updated new replacement) adj2 data)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:02
L11	4	7.clm.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:00
L12	1463	("same" identical common) near5 (logical adj2 address)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:01
L13	146	6 same I2	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:00
L14	792	(("same" identical common) with (logical adj2 address)) same (read write program programming)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:01
L15	126	6 same I4	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:01
L16	19	15.clm.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:01
L17	64	9 same I2	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:02

EAST Search History

<i>Interference Search</i>	L18	6	17.clm.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:03
	L19	785	12 same (stor\$3 updat\$3 identifying identify)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:04
	L20	1450	6 same "169"	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:04
	L21	112	6 same 19	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:04
	L22	23	21.clm.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:04
	L23	4	(out\$of\$order (reverse near3 order)) same 14	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:05
	L24	8305	(out\$of\$order (reverse near3 order)) same (stor\$3 updat\$3 programm\$3)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:06
	L25	78	24 same 6	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:06
	L26	2	25.clm.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/07/23 15:06

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1	"5479638".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 07:41
L2	1	"5924092".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 06:48
L3	29547	(old/new old new updated superseded) same logical	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 06:50
L4	1421	(non\$volatile eeprom flash cam) same 3	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 07:31
L5	11844	(old/new old new updated superseded) with logical	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 06:53
L6	977	(old/new old new updated superseded) with (logical adj address)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 06:53
L7	316	4 and 6	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 06:52
L8	314	(read read\$3 search search\$3) and 7	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 06:51
L9	244	4 same 6	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 06:52
L10	91124	(old/new old new) same (updated superseded)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 07:28
L11	339	10 same (logical adj address)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 06:53
L12	115	(non\$volatile eeprom flash cam) same 11	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 07:42
L13	6	12 same ((more most) near4 recent)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 07:34
L14	242580	(old/new old new) and (updated superseded)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 07:33

EAST Search History

L15	751	("same" common) near2 (logical adj address)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 07:29
L16	404	(non\$volatile eeprom flash cam) and 15	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 07:33
L17	2972579	(old/new old new) (updated superseded)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 07:33
L18	205	15 same 17	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 07:33
L19	30	18 same ((more most) near4 recent)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 07:34
L20	2385	(read\$3 search\$3) with (data file) with ((older young more most) near2 time)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 07:42
L21	84	(non\$volatile eeprom flash cam) same 20	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2007/08/06 07:42



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/250,238	10/13/2005	Kevin M. Conley	SNDK.156US2	7727
66785 7590 09/05/2007 DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION 505 MONTGOMERY STREET SUITE 800 SAN FRANCISCO, CA 94111			EXAMINER	
			DINH, NGOC V	
			ART UNIT	PAPER NUMBER
			2189	
			MAIL DATE	DELIVERY MODE
			09/05/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 11/250,238	Applicant(s) CONLEY, KEVIN M.	
	Examiner NGOC V. DINH	Art Unit 2189	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 03 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 20 July 2007.
- 2a) This action is **FINAL**.
- 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 4 and 7-25 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 4 and 7-25 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on 13 October 2005 is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All b) Some * c) None of:
- 1. Certified copies of the priority documents have been received.
- 2. Certified copies of the priority documents have been received in Application No. _____.
- 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) Notice of References Cited (PTO-892)
- 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date 02/20/07.
- 4) Interview Summary (PTO-413)
Paper No(s)/Mail Date _____.
- 5) Notice of Informal Patent Application
- 6) Other: _____.

DETAILED ACTION

1. This Office Action is responsive to RCE filed 06/05/07 in which claims 4, 7-25 are presented for examination.

INFORMATION DISCLOSURE STATEMENT

2. The Applicant's submission of the IDS filed 07/20/2007 has been considered. As required by M.P.E.P. 609 C(2), a copy of the PTOL-1449 is attached to the instant office action.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 4, 7, 10-13, 16-22, 25 are rejected under 35 U.S.C 103(a) as being unpatentable over Assar PN. 5,479,638.

Claims 4, 10, 25, Assar teaches a method of operating a memory system having an array of reprogrammable non-volatile charge storage elements [CAM, col. 4/12-25] organized in blocks of storage elements that are erasable together and in pages of storage elements within the blocks that are individually programmable as a unit [fig. 1-2], comprising:
as part of writing data into pages, recording an indication of a time from a clock source [old/new Flag, fig. 1] that data are written into individual pages,
when updating data [updated information files, col. 2/62-63] previously written into one or more initial pages, writing the updated data into one or more update pages and identify the initial and update data pages [old/new Flag, fig. 1-2; col. 9/20-23] by common logical addresses [correlating a logical address assigned to superseded data to a physical address of updated data, col. 9/28-33].

Art Unit: 2189

Assar does not teach when reading data of two or more pages having the same logical addresses, read the indications of the times that data have been stored in the two or more pages and use the data in the two or more pages having more recent time indications without using data in the two or more pages having older time indications.

However, it would have been obvious to one having ordinary skill in the art at the time the invention was made because pages having more recent time indications include the most up-to-date version data comparing to pages having older time indications. Therefore reading data from the pages having more recent time indications means reading the most up-to-date data. The time the file was most recently read and written is the time the data was most recently updated by the system.

Claims 7, 22, recording the indication of the time that data are written into individual pages includes recording the indication within the pages wherein the data are written [the old/new flag from the source block is then set, col. 8/8-9].

Claims 11, 17, as part of writing data into pages, data are written in individual storage elements of the pages with more than two storage states, thereby storing more than one bit of data in the individual storage elements [CAM, fig. 4; col. 5/28-40; fig. 6].

Claim 12, the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage element [CAM, col. 3/58-59, fig. 4].

Claims 13, 16 Assar teaches the claimed limitations as mentioned above.

Assar does not teach reading data from the pages of the blocks in an order that is reversed of the sequence in which they were written and ignore data in any page having the same logical address as a page from which data have already been read.

However, it would have been obvious to one having ordinary skill in the art at the time the invention was made because reading the block in reverse order (from younger time indication to older time indication) in which the pages have been written which means reading (as mentioned

in claim 4 above) the most up-to-date data, and ignoring a page from which data have already been read in order to speed up read operation of a non-volatile memory.

Claim 18, the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements [CAM, fig. 4].

Claims 19-20, Assar teaches the claimed limitations as mentioned above and further teaches: as part of writing data into pages, (1) recording an indication of a time that data are written into individual pages and (2) writing the data in individual storage elements of the pages with more than two storage states, thereby storing more than one bit of data in the individual storage elements [CAM, fig. 4; col. 5/28-40; fig. 6];

wherein the memory system being operated includes a controller connected with the array of charge storage elements and is positioned within an enclosed card having externally accessible electrical contacts connected with the controller [col. 10/claim 18].

Claim 21, recording an indication of a time that data are written into individual pages includes recording a different value of a sequence of numbers [a counter, col. 3/2-5].

4. Claims 8-9, 14-15, 23-24 are rejected under 35 U.S.C 103(a) as being unpatentable over Assar and in view of Estakhri et al PN. 6,223,308.

Claims 8, 14, 23, Assar teaches the claimed limitations as mentioned above.

Assar does not teach wherein updating data previously written into one or more initial pages is limited to updating data in a number of pages less than all of the pages of a block while not updating data in a remaining one or more pages of the same block.

Estakhri teaches updating data previously written into one or more initial pages is limited to updating data in a number of pages less than all of the pages of a block [fig. 23; the space manager block 544 determines that block 1000 or a portion thereof is free to be written into. The

LBA associated with block 1000 is then programmed into two locations 1030 and 1020 (within sectors 1006 and 1010, respectively) of block 1000, col. 19/65 to col. 20/5] while not updating data in a remaining one or more pages of the same block [to avoid having to perform an erase-before-write each time a data file is changed, col. 4/18-25].

It would have been obvious to one having ordinary skill in the art at the time the invention was made to combine Estakhri to Assar in order to avoid delay in performance due to the erase operation [Estakhri, col. 2/38-40].

Claims 9, 15, 24, Assar teaches the method is carried out utilizes electrically conductive floating gates as the charge storage elements [CAM, fig. 4].

CONCLUSION

5. Any response to this action should be mailed to:

Under Secretary of Commerce for intellectual Property and Director of the
United States Patent and Trademark Office
PO Box 1450

Alexandria, VA 22313-1450

or faxed to:

(571) 273-8300, (for Official communications intended for entry)

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PMR) system. Status information for published Applications may be obtained from either Private PMR or Public PMR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pak-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ngoc Dinh whose telephone number is (571) 272-4191. The examiner can normally be reached on Monday-Friday 8:30 AM-5:00 PM.

Application/Control Number: 11/250,238

Page 6

Art Unit: 2189

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor,
Reginald Bragdon, can be reached on (571) 272-4204.

ND

NGOC DINH

August 21, 2007

Reginald N. Bragdon
REGINALD BRAGDON
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

U.S. Department of Commerce, Patent and Trademark	Atty. Docket No.	Application No.
INFORMATION DISCLOSURE STATEMENT BY APPLICANT	SNDK.156US2	11/250,238
	Applicants	Conf. No.
(Use several sheets if necessary)	Conley	7727
(Form PTO-1449)	Filing Date	Art Group
	October 13, 2005	2189

U.S. Patent Documents

*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate
MD	1	5,924,092	7/13/99	Johnson			
	2	5,388,083	2/7/95	Assar et al.			
	3	5,479,638	12/26/95	Assart et al.			
	4	5,568,439	10/22/96	Harari			
	5	5,835,935	11/10/98	Estakhri et al.			
	6	5,838,614	11/17/98	Estakhri et al.			
	7	5,860,124	1/12/99	Mathews et al.			
	8	5,924,113	7/13/99	Estakhri et al.			
	9	5,937,425	8/10/99	Ban			
	10	5,598,370	1/28/97	Nijjima et al.			
MD	11	6,023,423	2/8/00	Aritome			

U.S. Published Patent Application Documents

*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate

Foreign Patent Documents

							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
MD	12	WO 98/44420	10/8/98	PCT				
	13	0 250 876	5/27/87	EPO				
	14	WO 02/49039	6/20/02	PCT				
	15	FR 2 742 893	12/20/95	France			Abstract	
	16	WO 99/07000	2/11/99	PCT				
	17	EP 0 977 121	7/26/99	EPO				
	18	EP 0 896 280	02/10/99	EPO				
	19	H10-091490	4/10/98	Japan			Abstract	
	20	H10-091490 ?	4/10/98	Japan			X	
	21	H08-221223	8/30/96	Japan			Abstract	
MD	22	H08-221223	8/30/96	Japan			X	

FILED VIA EFS
Sheet 1 of 2

U.S. Department of Commerce, Patent and Trademark				Atty. Docket No.			Application No.	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT				SNDK.156US2			11/250,238	
				Applicants			Conf. No.	
(Use several sheets if necessary)				Conley			7727	
(Form PTO-1449)				Filing Date			Art Group	
				October 13, 2005			2189	
<i>MP</i>	23	WO 99/21093	4/29/99	PCT				
	24	H06-250798	9/19/94	Japan			Abstract	
	25	H06-250798	9/19/94	Japan			X	
↓	26	H10-105661	4/24/98	Japan			Abstract	
<i>MP</i>	27	H10-105661	4/24/98	Japan			X	
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
<i>MP</i>	28	Notice of Opposition to a European Patent for SanDisk Corporation, Patent No. 1,352,394, Application No. 02703078.2, dated March 1, 2007, 17 pages.						
<i>MP</i>	29	Notification of Reasons for Refusal for SanDisk Corporation, Japanese Patent Application No. 2002-558275, mailed February 23, 2007, 10 pages.						
Examiner		<i>[Signature]</i>		Date Considered		<i>06/15/07</i>		
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

FILED VIA EFS
Sheet 2 of 2

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.		Application No.				
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		SNDK.156US2		11/250,238				
		Applicants		Conf. No.				
(Use several sheets if necessary)		Conley		7727				
(Form PTO-1449)		Filing Date		Art Group				
		October 13, 2005		2189				
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
						Translation		
		Document	Date	Country	Class	Subclass	Yes	No
<i>ND</i>		H10-91490	4/10/98	Japan			X	
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
Examiner <i>[Signature]</i>		Date Considered <i>07/23/07</i>						
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

FILED VIA EFS
Sheet 1 of 1

Notice of References Cited	Application/Control No. 11/250,238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.	
	Examiner NGOC V. DINH	Art Unit 2189	Page 1 of 1

U.S. PATENT DOCUMENTS

*	Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A US-6,223,308 B1	04-2001	Estakhri et al.	714/42
B	US-			
C	US-			
D	US-			
E	US-			
F	US-			
G	US-			
H	US-			
I	US-			
J	US-			
K	US-			
L	US-			
M	US-			

FOREIGN PATENT DOCUMENTS

*	Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
N					
O					
P					
Q					
R					
S					
T					

NON-PATENT DOCUMENTS

*	Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
U	
V	
W	
X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
 United States Patent and Trademark Office
 Address: COMMISSIONER FOR PATENTS
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 www.uspto.gov



Bib Data Sheet

CONFIRMATION NO. 7727

SERIAL NUMBER 11/250,238	FILING OR 371(c) DATE 10/13/2005 RULE	CLASS 711	GROUP ART UNIT 2189	ATTORNEY DOCKET NO. SNDK.156US2
------------------------------------	---	---------------------	-------------------------------	---

APPLICANTS
 Kevin M. Conley, San Jose, CA;
**** CONTINUING DATA ******* *Yes NY*
 This application is a CON of 10/841,388 05/07/2004 PAT 6,968,421 which is a CON of 09/766,436 01/19/2001 PAT 6,763,424
**** FOREIGN APPLICATIONS ******* *NA NY*
IF REQUIRED, FOREIGN FILING LICENSE GRANTED
**** 11/03/2005**

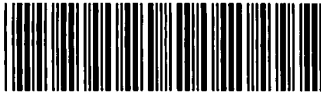
Foreign Priority claimed <input type="checkbox"/> yes <input checked="" type="checkbox"/> no	STATE OR COUNTRY CA	SHEETS DRAWING 9	TOTAL CLAIMS 3	INDEPENDENT CLAIMS 2
35 USC 119 (a-d) conditions met <input type="checkbox"/> yes <input checked="" type="checkbox"/> no <input type="checkbox"/> Met after Allowance				
Verified and Acknowledged	Examiner's Signature <i>[Signature]</i>	Initials <i>NY</i>		

ADDRESS
66785

TITLE
PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY

FILING FEE RECEIVED 1000	FEES: Authority has been given in Paper No. _____ to charge/credit DEPOSIT ACCOUNT No. _____ for following:	<input type="checkbox"/> All Fees
		<input type="checkbox"/> 1.16 Fees (Filing)
		<input type="checkbox"/> 1.17 Fees (Processing Ext. of time)
		<input type="checkbox"/> 1.18 Fees (Issue)
		<input type="checkbox"/> Other _____
		<input type="checkbox"/> Credit

Index of Claims



Application/Control No.

11/250,238

Examiner

NGOC V. DINH

Applicant(s)/Patent under Reexamination

CONLEY, KEVIN M.

Art Unit

2189

✓	Rejected
=	Allowed

-	(Through numeral) Cancelled
+	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

Claim		Date	
Final	Original		
	1		
	2		
	3		
	4	✓	
	5		
	6		
	7	✓	
	8		
	9		
	10		
	11		
	12		
	13		
	14		
	15		
	16		
	17		
	18		
	19		
	20		
	21		
	22		
	23		
	24	✓	
	25		
	26		
	27		
	28		
	29		
	30		
	31		
	32		
	33		
	34		
	35		
	36		
	37		
	38		
	39		
	40		
	41		
	42		
	43		
	44		
	45		
	46		
	47		
	48		
	49		
	50		

Claim		Date	
Final	Original		
	51		
	52		
	53		
	54		
	55		
	56		
	57		
	58		
	59		
	60		
	61		
	62		
	63		
	64		
	65		
	66		
	67		
	68		
	69		
	70		
	71		
	72		
	73		
	74		
	75		
	76		
	77		
	78		
	79		
	80		
	81		
	82		
	83		
	84		
	85		
	86		
	87		
	88		
	89		
	90		
	91		
	92		
	93		
	94		
	95		
	96		
	97		
	98		
	99		
	100		

Claim		Date	
Final	Original		
	101		
	102		
	103		
	104		
	105		
	106		
	107		
	108		
	109		
	110		
	111		
	112		
	113		
	114		
	115		
	116		
	117		
	118		
	119		
	120		
	121		
	122		
	123		
	124		
	125		
	126		
	127		
	128		
	129		
	130		
	131		
	132		
	133		
	134		
	135		
	136		
	137		
	138		
	139		
	140		
	141		
	142		
	143		
	144		
	145		
	146		
	147		
	148		
	149		
	150		

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Kevin M. Conley
Title: Partial Block Data Programming And Reading Operations In A Non-Volatile Memory
Application No.: 11/250,238 Filing Date: October 13, 2005
Examiner: Dinh, Ngoc V. Group Art Unit: 2189
Docket No.: SNDK.156US2 Conf. No.: 7727

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE TO OFFICE ACTION

Sir:

This is in response to the non-final Office Action dated September 5, 2007.

A Claims Listing begins on page 2 of this paper.

Remarks begin on page 7 of this paper.

Reconsideration is kindly requested in light of the following Remarks.

Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

CLAIMS LISTING

No amendments are being made to the present application claims, which are listed here for the convenience of the Examiner:

1 – 3. (Cancelled)

4. (Previously Presented) A method of operating a memory system having an array of reprogrammable non-volatile charge storage elements organized in blocks of storage elements that are erasable together and in pages of storage elements within the blocks that are individually programmable as a unit, comprising:

as part of writing data into pages, recording an indication of a time from a clock source that data are written into individual pages,

when updating data previously written into one or more initial pages, writing the updated data into one or more update pages and identify the initial and update data pages by common logical addresses, and

when reading data of two or more pages having the same logical addresses, read the indications of the times that data have been stored in the two or more pages and use the data in the two or more pages having more recent time indications without using data in the two or more pages having older time indications.

5 – 6. (Cancelled)

7. (Previously Presented) The method of claim 4, wherein recording the indication of the time that data are written into individual pages includes recording the indication within the pages wherein the data are written.

8. (Previously Presented) The method of claim 4, wherein updating data previously written into one or more initial pages is limited to updating data in a number of pages less than

all of the pages of a block while not updating data in a remaining one or more pages of the same block.

9. (Previously Presented) The method of claim 8, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

10. (Previously Presented) The method of claim 4, wherein as part of writing data into pages, logical addresses of the individual pages in which the data are written are also written in the individual pages.

11. (Previously Presented) The method of claim 4, wherein as part of writing data into pages, data are written in individual storage elements of the pages with more than two storage states, thereby storing more than one bit of data in the individual storage elements.

12. (Previously Presented) The method of claim 4, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

13. (Previously Presented) A method of operating a memory system having an array of reprogrammable non-volatile charge storage elements organized in blocks of storage elements that are erasable together and in pages of storage elements within the blocks that are individually programmable as a unit, comprising:

as part of writing data into pages, data are written into the pages of the blocks in sequence,

updating data previously written into one or more initial pages by writing the updated data into one or more update pages and identify the initial and update data pages by common logical addresses, and

reading data from the pages of the blocks in an order that is a reverse of the sequence in which they were written and ignore data in any page having the same logical address as a page from which data have already been read.

14. (Previously Presented) The method of claim 13, wherein updating data previously written into one or more initial pages is limited to updating data in a number of pages less than all of the pages of a block while not updating data in a remaining one or more pages of the same block.

15. (Previously Presented) The method of claim 14, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

16. (Previously Presented) The method of claim 13, wherein as part of writing data into pages, logical addresses of the individual pages in which the data are written are also written in the individual pages.

17. (Previously Presented) The method of claim 13, wherein as part of writing data into pages, data are written in individual storage elements of the pages with more than two storage states, thereby storing more than one bit of data in the individual storage elements.

18. (Previously Presented) The method of claim 13, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

19. (Previously Presented) A method of operating a memory system having an array of reprogrammable non-volatile charge storage elements organized in blocks of storage elements that are erasable together and in pages of storage elements within the blocks that are individually programmable as a unit, comprising:

as part of writing data into pages, (1) recording an indication of a time that data are written into individual pages and (2) writing the data in individual storage elements of the pages with more than two storage states, thereby storing more than one bit of data in the individual storage elements,

when updating data previously written into one or more initial pages, writing the updated data into one or more update pages and identify the initial and update data pages by common logical addresses, and

when reading data of two or more pages having the same logical addresses, read the indications of the times that data have been stored in the two or more pages and use the data in the two or more pages having more recent time indications without using data in the two or more pages having older time indications,

wherein the memory system being operated includes a controller connected with the array of charge storage elements and is positioned within an enclosed card having externally accessible electrical contacts connected with the controller.

20. (Previously Presented) The method of claim 19, wherein recording an indication of a time that data are written into individual pages includes recording a value of a clock within the memory system.

21. (Previously Presented) The method of claim 19, wherein recording an indication of a time that data are written into individual pages includes recording a different value of a sequence of numbers.

22. (Previously Presented) The method of claim 19, wherein recording the indication of the time that data are written into individual pages includes recording the indication within the pages wherein the data are written.

23. (Previously Presented) The method of claim 19, wherein updating data previously written into one or more initial pages is limited to updating data in a number of pages

less than all of the pages of a block while not updating data in a remaining one or more pages of the same block.

24. (Previously Presented) The method of claim 23, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

25. (Previously Presented) The method of claim 19, wherein as part of writing data into pages, logical addresses of the individual pages in which the data are written are also written in the individual pages.

REMARKS

Claims 4 and 7-25 remain pending in the present application without amendment.

Rejections of Claims 4, 7, 10-13, 16-22 and 25 Under 35 U.S.C. §103(a)

Claims 4, 7, 10-13, 16-22 and 25 have been rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,479,638 to Assar et al. (hereinafter "Assar") alone. All three independent claims 4, 13 and 19 presently in this application are included in this rejected group of claims. This rejection is respectfully traversed.

Independent Claims 4 and 19

The Office Action contends that Assar's "old/new flag" 116 (Figures 1 and 2, and step 210 of Figure 5) anticipates the limitations "recording an indication of a time from a clock source that data are written" (claim 4) and "recording an indication of a time that data are written" (claim 19). But Assar's old/new flag is submitted not to be such an indication and further not to have rendered the claimed technique obvious since Assar's expressed motivation for using the old/new flag is different than that of the time indication of the present application claims 4 and 19.

The Office Action (bottom paragraph on page 2, line 2) cites the CAM (content-addressable-memory) 106 (see Figures 1 and 2) of Assar as meeting the terms of the claimed memory array. By the Office Action's further reference (bottom paragraph of page 2, line 7) to the logical address 308 and physical address 408 of the CAM, it seems that the physical address is being taken to correspond to the data specified in the claims to be updated. The physical address 408 is indeed updated, when data in the memory 102 are updated, to designate the new block of the memory 102 where the updated data are stored. The process of updating data in the memory 102 (Assar, Figures 1 and 2) is apparently not considered by the Office Action to be pertinent to the claims. Even the rejection of dependent claims 11, 12, 17, 18, 19, 20 (Office Action, pages 3 and 4) references Assar's CAM as meeting their terms, not the data memory 102.

Assar describes the old/new flag 116 stored in the CAM 106 to be set for data written into a block of the memory 102 that contains data that is being rendered obsolete by new data being written into another block of the memory 102. The physical address 408 in block 110 of the CAM 106 in Assar's Figure 2 example designates a block of the memory 102 that is being updated, the initial data. The physical address 408 in block 114 designates a block of the memory 102 where the updated data are stored. Assar's purpose of the old/new flag 116 stored in the CAM block 110 is to identify blocks of the memory 102 containing obsolete data for later erasure when necessary to provide erased blocks for the storage of new data in the memory 102 (see Assar, col. 2, lns. 64-67; col. 4, lns. 59-65).

The old/new flag 116 is therefore set to provide an indication of the status of the block of the memory 102 that contains the *old* data, not the time that *new* data are being written into a page as is recited in claims 4 and 19. That is, in the example of Assar's Figure 2, the old/new flag 116 is set for the CAM block 110 containing the old data (*old* physical address 408) that is being replaced by new data (*new* physical address 408) written into the block 114. No indication of the time, or anything similar, that new data are written into the block 114 is recorded by Assar. This is not necessary for Assar since the purpose of the old/new flag 116 is to mark a block of the memory 102 containing obsolete data for subsequent erasure. Nothing is written into the CAM block 114 that provides an indication of a time in which the new data (physical address 408) are being written. Claims 4 and 19 specify that the indication of time that data are written is recorded "as part of writing [these] data into pages" (annotation added). In Assar, writing of the data and the old/new flag are separate steps (Assar, col. 4, lns. 51-55) since they are being written into different blocks 110 and 114.

Further, Assar's old/new flag 116 is not an indication of time, and particularly not time from a clock source, as specified by the present application claims 4 and 19. Assar's old/new flag is a single bit of data, and the same single bit is written for each block containing obsolete data. By the technique of claims 4 and 19, an indication of time is written as part of writing data into pages without being limited to cases where other stored data is being rendered obsolete by the current write operation. In Assar, on the other hand, the old/new flag is only written for a block containing data determined to have been rendered obsolete by writing new data into another block.

It may be noted that the present application describes the prior use of flags to identify pages containing obsolete data (see paragraphs [0007], page 3, and [0009]) and the problems with such use. Instead of having to write a flag into a previously programmed page, which can disturb the data stored in that page, an indication of time is recorded for data being written into a new page, which can be done without disturbing other data. A separate programming operation for the flag can also be avoided because the indication of time is recorded as part of the write operation for new data. In the example of Assar's Figures 1 and 2, the old/new flag 116 is written to the block 110 when that block already contains data of the used/free flag 112, logical address 308 and physical address 408. An application of what the present application discloses is that the later writing of the old/new flag 116, separately from writing the new data in the block 114, can disturb the data previously written in the block 112. The techniques claimed herein avoid this.

Another distinction of claims 4 and 19 over Assar is that the indication of time is claimed to be written for individual pages of data within erasable blocks. That is, the memory system in which the claimed method is practiced is formed of large blocks of memory cells that are erased together. Each block contains pages of data, and the indication of time is recorded for individual pages. Assar, on the other hand, has a capacity for storing only one old/new flag 116 for each of the blocks 110, 114, etc. of the CAM 106. Indeed, since the purpose of Assar's old/new flag is to mark blocks of the memory 102 that can be erased, this purpose would not be furthered by marking individual pages within a block. Assar marks an erasable unit, which is a block of memory cells. Pages within a block are not individually erasable so there would be no purpose for Assar to record an old/new flag for individual pages that are not individually erasable. Assar did not therefore make it obvious to record times that data are written into individual pages.

Finally, with respect to independent claims 4 and 19, the Office Action (page 3, lines 1-10) asserts that it would have been obvious to read the page having the more current time indication when there are two pages having the same logical address. As discussed above, Assar's old/new flag 116 is not the claimed indication of time, nor does Assar keep track of the status of the data on a page basis. Further, no support is found to exist in Assar for the obviousness conclusion expressed on page 3 of the Office Action. Other non-disclosed techniques could be have instead been recognized to distinguish the old and new data during

read. For example, it is noted that the process shown in the flowchart of Figure 5 ends with a step 212 that updates the address map to correlate the logical and physical addresses of new data just written. It is conceivable that the record of superseded data could be changed to point to only the block containing the new data, which would result in only the new data being accessed in response to the old logical address being used. The old/new flag 116 would in this case not be used to read data from the memory.

In view of these several distinctions over Assar, independent claims 4 and 19 are submitted to be patentable.

Claim 13

Independent claim 13 is somewhat different from claims 4 and 19 by specifying, instead of recording an indication of the time that the individual pages of data are written, that the pages of data with common logical addresses are read in an order that is the reverse of the order in which they were written. It is alleged in the Office Action (paragraph bridging pages 3 and 4) that it would have been obvious from Assar to practice the claimed technique, without providing any reasons. The conclusion is merely stated as a matter of fact. But nothing is found in Assar that even remotely suggests anything close to this. No mention of any particular order of reading data from two blocks can be found, let alone the particular order claimed. As mentioned above, other techniques could have occurred to one of ordinary skill of how to read the current pages of data and ignore the others. Assar is focused on setting the old/new flag for blocks storing superseded data so that those blocks may be erased when necessary to provide additional blocks for programming. How one of ordinary skill in the art would have realized from this that the blocks of data should be read in a reverse order is not understood. No basis for this large leap is given in the Office Action. It is respectfully submitted that to merely state that the differences of invention from Assar are obvious cannot be a valid rejection under 35 U.S.C. §103(a).

Further, claim 13 shares other novel features with claims 4 and 19 that are discussed above. Writing and reading data in units of pages within a block (the erase unit) are also specified in claim 13, for example, which is novel over Assar.

Dependent Claims

Each of the rejected claims 7, 10, 11, 12, 16, 17, 18, 20, 21, 22 and 25, which depend from one of the independent claims 4, 13 or 19, are submitted to be patentable for the same reasons discussed above with respect to their independent claims. In addition, many of these dependent claims add other novel elements not suggested by Assar.

Claims 7 and 22 additionally specify that the indication of time is stored in individual pages. Assar does not describe operation of a system with multiple pages within individual blocks. Assar's old/new flag is maintained on a block basis.

The multiple storage state technique of claims 11 and 17 is also not suggested by Assar. The portion of Assar cited in the Office Action (page 3, lines 14-16) deals with the use of the old/new flag to control which blocks are erased, quite different subject matter.

Claims 20 and 21 add details of the nature of the "indication of time" of independent claim 19. Claim 20 specifies that the time comes from a clock within the memory system, something quite different from Assar's old/new flag 116 that appears to be relied upon in the Office Action as disclosure of this feature. Assar's old/new flag is certainly not the claimed "value of a clock within the memory system."

Claim 21 calls for storing a unique sequence of numbers as the "indication of time," again very different from the single bit old/new flag 116 of Assar. The Office Action (page 4, lines 13-14) references a counter described by Assar for keeping track of the number of times that a block has been erased for the purpose of a second embodiment that uses this count to level the wear (usage) of the blocks. The number of times the individual blocks have been erased has nothing to do with the techniques of updating data of Assar's first embodiment that is referenced in rejecting independent claim 19.

Rejections of Claims 8, 9, 14, 15, 23 and 24 and 25 Under 35 U.S.C. §103(a)

Claims 8, 9, 14, 15, 23 and 24 have been rejected under 35 U.S.C. §103(a) as being unpatentable over Assar in view of United States Patent No. 6,223,308 to Estakhri et al. (hereinafter "Estakhri"). Since each of these claims is dependent on one of independent claims 4, 13 or 19, they are believed to be patentable over Assar for the reasons discussed above. It is

not seen that Estakhri suggests the differences between the independent claims and Assar that are identified above, and the Office Action does not contend otherwise.

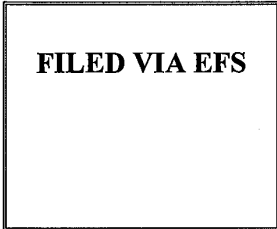
Estakhri does disclose, however, operation of a type of memory wherein there are multiple pages (of one sector each) within each erasable block of memory cells. Referring to Figure 23 referenced in the Office Action (paragraph bridging pages 4 and 5), each page stores a sector of user data 1052 and certain overhead data including an error correction code (ECC) 1014, logical addresses 1020 and 1030 of data stored in a block, a defect flag 1012, and so forth. But no suggestion can be found in Estakhri of storing an indication of the time of writing individual pages of data.

This group of dependent claims includes other limitations that would not have been obvious from a combination of Assar and Estakhri. Each of these claims add that fewer than all the pages in a block are updated in the manner recited in their respective independent claims 4, 13 and 19. This is a particularly useful application of the claimed technique since the updating of fewer than all the pages of the block previously required consolidation (garbage collection) of unchanged pages from the original block with the updated pages into another block that was erased. The claimed technique avoids this by retaining the obsolete data and then providing a way to distinguish those data pages from the pages with updated data. The amount of data copying, which is particularly time consuming, is significantly reduced.

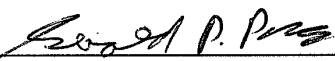
The basis of rejecting claims 8, 14 and 23 is not understood. Specifically, the relevance of the discussion (Estakhri, col. 19, ln. 63 – col. 20, ln. 7) of writing the LBA of a block into two locations within the block is not understood. That discussion is in the context of writing new data into an erased block. Other portions of Estakhri are instead primarily directed to the relocation of a number of pages of data less than a full block into another block having enough erased space to accommodate that number of pages. (see for example Estakhri, col. 11, lns. 1-27 and lns. 52-63.) It is not seen that Estakhri discusses updating data, at least in any manner relevant to the present claims. The rejection of claims 8, 14 and 23 is therefore respectfully traversed on this basis. Since claims 9, 15 and 24 depend from these, they are also submitted to be patentable for the same reason.

Conclusion

Accordingly, it is believed that this application is now in condition for allowance and an early indication of its allowance is solicited. However, if the Examiner has any further matters that need to be resolved, a telephone call to the undersigned attorney at 415-276-6534 would be appreciated.



Respectfully submitted,


Gerald P. Parsons
Reg. No. 24,486

January 23, 2008
Date

Davis Wright Tremaine LLP
505 Montgomery Street, Suite 800
San Francisco, CA 94111-6533
(415) 276-6500 (main)
(415) 276-6534 (direct)
(415) 276-6599 (fax)
Email: geraldparsons@dwt.com

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

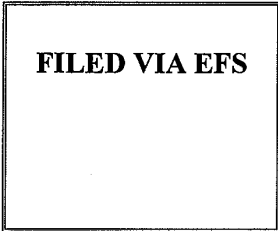
Applicant(s):	Kevin M. Conley		
Title:	Partial Block Data Programming And Reading Operations In A Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	October 13, 2005
Examiner:	Dinh, Ngoc V.	Group Art Unit:	2189
Docket No.:	SNDK.156US2	Conf. No.:	7727

Mail Stop
 Commissioner for Patents
 P.O. Box 1450
 Alexandria, VA 22313-1450


PETITION FOR EXTENSION OF TIME

Dear Sir:

Applicant respectfully petitions for a two-month extension of time within which to respond to the September 5, 2007 outstanding Office Action, such extension allowing the undersigned until January 23, 2008 to respond. The fee of \$460.00 has been authorized via EFS to Deposit Account 04-0258. The Commissioner is hereby authorized to charge any additional fees, which may be required, or credit any overpayment to Deposit Account 04-0258.



Respectfully submitted,

	January 23, 2008
Gerry P. Parsons Reg. No. 24, 486	Date

DAVIS WRIGHT TREMAINE LLP
 505 Montgomery Street, Suite 800
 San Francisco, California 94111-6533
 Telephone: (415) 276-6500
 Facsimile: (415) 276-6599
 Email: geraldparsons@dwt.com

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Kevin M. Conley		
Title:	Block Data Programming and Reading Operations in a Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	October 13, 2005
Examiner:	Ngoc V. Dinh	Group Art Unit:	2189
Docket No.:	SNDK.156US2	Conf. No.:	7727

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Dear Sir:

Pursuant to 37 C.F.R. §§ 1.56, 1.97 and 1.98, Applicant(s) call(s) the documents listed on the enclosed Form PTO-1449 to the Examiner's attention in this patent application.

Copies of the listed foreign patent documents and/or Other Art are enclosed.

Citation of this document shall not be construed as (1) an admission that the document is prior art with respect to the invention or inventions claimed in this application, (2) a representation that a search has been made (other than as indicated by any cited document), or (3) an admission that the cited information is, or is considered to be, material to patentability as defined in § 1.56(b).

Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

This information disclosure statement is submitted under 37 C.F.R. § 1.97(c). No fee should be required because, in accordance with § 1.97(e)(1), each item contained in this information disclosure statement was first cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this information disclosure statement. The Commissioner is authorized, however, to charge any fee that may be required, or to credit any overpayment, against Deposit Account No. 04-0258.

FILED VIA EFS

Respectfully submitted,

Gerald P. Parsons
Gerald P. Parsons
Reg. No. 24,486

Jan. 23, 2004
Date

Davis Wright Tremaine LLP
505 Montgomery Street, Suite 800
San Francisco, CA 94111-6533
(415) 276-6500 (main)
(415) 276-6534 (direct)
(415) 276-6599 (fax)
Email: geraldparsons@dwt.com

U.S. Department of Commerce, Patent and Trademark				Atty. Docket No.			Application No.	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT				SNDK.156US2			11/250,238	
				Applicants			Conf. No.	
(Use several sheets if necessary)				Conley			7727	
(Form PTO-1449)				Filing Date			Art Group	
				October 13, 2005			2189	
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
	1	Japanese Patent Office, "Decision of Refusal," corresponding Japanese Patent Application No. 2002-558275 on November 27, 2007, 3 pages (including translation).						
	2	The Patent Office of the People's Republic of China, "Notification of the First Office Action," corresponding Chinese Patent Application No. 200610142358.3 on December 14, 2007, 3 pages.						
Examiner			Date Considered					
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

FILED VIA EFS
Sheet 1 of 1

Electronic Patent Application Fee Transmittal

Application Number:	11250238			
Filing Date:	13-Oct-2005			
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY			
First Named Inventor/Applicant Name:	Kevin M. Conley			
Filer:	Gerald Paul Parsons/Mary Beth Stone (GPP)			
Attorney Docket Number:	SNDK.156US2			
Filed as Large Entity				
Utility Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Extension - 2 months with \$0 paid	1252	1	460	460

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Miscellaneous:				
Total in USD (\$)				460

Electronic Acknowledgement Receipt

EFS ID:	2756874
Application Number:	11250238
International Application Number:	
Confirmation Number:	7727
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY
First Named Inventor/Applicant Name:	Kevin M. Conley
Customer Number:	66785
Filer:	Gerald Paul Parsons/Mary Beth Stone (GPP)
Filer Authorized By:	Gerald Paul Parsons
Attorney Docket Number:	SNDK.156US2
Receipt Date:	23-JAN-2008
Filing Date:	13-OCT-2005
Time Stamp:	16:36:31
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$460
RAM confirmation Number	1684
Deposit Account	040258
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes) /Message Digest	Multi Part /.zip	Pages (if appl.)
-----------------	----------------------	-----------	----------------------------------	------------------	------------------

1	Foreign Reference	Reference_SNDK156CN1_O A.pdf	354441 <small>775f8689052927de19fccc9418c1d6452f900afd</small>	no	3
Warnings:					
Information:					
2	Foreign Reference	Reference_SNDK156JP0_D ecisionofRefusal.pdf	278362 <small>30ecb63b8fb1ccd1268b8fe2a50ebc509e08b9a8</small>	no	3
Warnings:					
Information:					
3		SNDK156US2_Trans_Resp OA_ExtTime_IDS.pdf	829407 <small>21cd30833d78245beb09c6cec8b3e12b8033bdfc</small>	yes	18
Multipart Description/PDF files in .zip description					
		Document Description	Start	End	
		Miscellaneous Incoming Letter	1	1	
		Amendment - After Non-Final Rejection	2	2	
		Claims	3	7	
		Applicant Arguments/Remarks Made in an Amendment	8	14	
		Extension of Time	15	15	
		Information Disclosure Statement (IDS) Filed	16	18	
Warnings:					
Information:					
4	Fee Worksheet (PTO-06)	fee-info.pdf	8212 <small>108fd90f177df876f66561758cfd412c8a752b9c</small>	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			1470422		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

LAWYERS

Davis Wright Tremaine LLP



ANCHORAGE BELLEVUE LOS ANGELES NEW YORK PORTLAND SAN FRANCISCO SEATTLE SHANGHAI WASHINGTON, D.C.

SUITE 800
505 MONTGOMERY STREET
SAN FRANCISCO, CA 94111-6533

TEL (415) 276-6500
FAX (415) 276-6599
www.dwt.com

January 23, 2008

Mail Stop

Commissioner For Patents

P.O. Box 1450

Alexandria, VA 22313-1450

Customer No. 66785

Re: Applicant(s): Kevin M. Conley
Title: Partial Block Data Programming And Reading Operations IN A Non-Volatile Memory
Application No.: 11/250,238 Filing Date: October 13, 2005
Examiner: Dinh, Ngoc V. Group Art Unit: 2189
Docket No.: SNDK.156US2 Conf. No.: 7727

Dear Sir:

Transmitted herewith are the following documents in the above-identified application:

- (1) This Transmittal Letter (1 pg);
 - (2) Response to Office Action (13 pgs.);
 - (3) Petition for Extension of Time (2 mos) (1 pg); and
 - (4) Information Disclosure Statement and 1449 Form (3 pgs.)(attached 2 references)
- The fee has been calculated as shown below:

CLAIMS AS AMENDED

	Claims Remaining <i>After</i> Amendment		Highest No. Previously Paid For	=	Present Extra	Rate		Additional Fee
Total Claims	20	Minus	20	=	0	X \$50.00	\$	0.00
Independent Claims	3	Minus	3	=	0	X \$210.00	\$	0.00
<input checked="" type="checkbox"/>	Fee for Petition of Extension of Time						\$	460.00
<u>Total additional fee for this Amendment:</u>							\$	<u>460.00</u>

- Conditional Petition for Extension of Time: If an extension of time is required for timely filing of the enclosed document(s) after all papers filed with this transmittal have been considered, an extension of time is hereby requested.
- The fee of \$460.00 has been authorized via EFS to Deposit Account 04-0258. The Commissioner is hereby authorized to charge any additional fees, which may be required, or credit any overpayment to Deposit Account 04-0258.

FILED VIA EFS

Respectfully submitted,

Gerald P. Parsons
Reg. No.24,486

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875					Application or Docket Number 11/250,238		Filing Date 10/13/2005		<input type="checkbox"/> To be Mailed							
APPLICATION AS FILED – PART I																
(Column 1)			(Column 2)			SMALL ENTITY <input type="checkbox"/>		OR			OTHER THAN SMALL ENTITY					
FOR		NUMBER FILED	NUMBER EXTRA		RATE (\$)	FEE (\$)	OR		RATE (\$)	FEE (\$)						
<input type="checkbox"/> BASIC FEE <small>(37 CFR 1.16(a), (b), or (c))</small>		N/A	N/A		N/A				N/A							
<input type="checkbox"/> SEARCH FEE <small>(37 CFR 1.16(k), (l), or (m))</small>		N/A	N/A		N/A		OR		N/A							
<input type="checkbox"/> EXAMINATION FEE <small>(37 CFR 1.16(o), (p), or (q))</small>		N/A	N/A		N/A		OR		N/A							
TOTAL CLAIMS <small>(37 CFR 1.16(i))</small>		minus 20 =	*		X \$ =		OR		X \$ =							
INDEPENDENT CLAIMS <small>(37 CFR 1.16(h))</small>		minus 3 =	*		X \$ =		OR		X \$ =							
<input type="checkbox"/> APPLICATION SIZE FEE <small>(37 CFR 1.16(s))</small>		If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$250 (\$125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).														
<input type="checkbox"/> MULTIPLE DEPENDENT CLAIM PRESENT <small>(37 CFR 1.16(j))</small>																
* If the difference in column 1 is less than zero, enter "0" in column 2.																
APPLICATION AS AMENDED – PART II										SMALL ENTITY		OR		OTHER THAN SMALL ENTITY		
(Column 1)			(Column 2)			(Column 3)			RATE (\$)		ADDITIONAL FEE (\$)		RATE (\$)		ADDITIONAL FEE (\$)	
AMENDMENT	01/23/2008		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	X \$ =		OR		X \$50=		0			
	Total <small>(37 CFR 1.16(o))</small>		* 20	Minus	** 20	= 0	X \$ =		OR		X \$210=		0			
	Independent <small>(37 CFR 1.16(h))</small>		* 3	Minus	***3	= 0			OR							
	<input type="checkbox"/> Application Size Fee <small>(37 CFR 1.16(s))</small>															
	<input type="checkbox"/> FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM <small>(37 CFR 1.16(j))</small>															
							TOTAL ADD'L FEE		OR		TOTAL ADD'L FEE		0			
AMENDMENT			CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	X \$ =		OR		X \$ =					
	Total <small>(37 CFR 1.16(o))</small>		*	Minus	**	=	X \$ =		OR		X \$ =					
	Independent <small>(37 CFR 1.16(h))</small>		*	Minus	***	=			OR							
	<input type="checkbox"/> Application Size Fee <small>(37 CFR 1.16(s))</small>															
	<input type="checkbox"/> FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM <small>(37 CFR 1.16(j))</small>															
							TOTAL ADD'L FEE		OR		TOTAL ADD'L FEE					
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.																
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".																
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".																
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.																

Legal Instrument Examiner:
/DENISE t. LILES/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Kevin M. Conley		
Title:	Block Data Programming and Reading Operations in a Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	October 13, 2005
Examiner:	Ngoc V. Dinh	Group Art Unit:	2189
Docket No.:	SNDK.156US2	Conf. No.:	7727

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Dear Sir:

Pursuant to 37 C.F.R. §§ 1.56, 1.97 and 1.98, Applicant(s) call(s) the documents listed on the enclosed Form PTO-1449 to the Examiner's attention in this patent application.

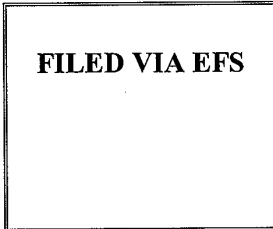
Copy of the listed foreign patent documents and/or Other Art is enclosed.

Citation of this document shall not be construed as (1) an admission that the document is prior art with respect to the invention or inventions claimed in this application, (2) a representation that a search has been made (other than as indicated by any cited document), or (3) an admission that the cited information is, or is considered to be, material to patentability as defined in § 1.56(b).

Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

This information disclosure statement is submitted under 37 C.F.R. § 1.97(c). No fee should be required because, in accordance with § 1.97(e)(1), each item contained in this information disclosure statement was first cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this information disclosure statement. The Commissioner is authorized, however, to charge any fee that may be required, or to credit any overpayment, against Deposit Account No. 04-0258.



Respectfully submitted,

Gerald P. Parsons March 13, 2008
Gerald P. Parsons Date
Reg. No. 24,486

Davis Wright Tremaine LLP
505 Montgomery Street, Suite 800
San Francisco, CA 94111-6533
(415) 276-6500 (main)
(415) 276-6534 (direct)
(415) 276-6599 (fax)
Email: geraldparsons@dwt.com

U.S. Department of Commerce, Patent and Trademark				Atty. Docket No.			Application No.	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT				SNDK.156US2			11/250,238	
				Applicants			Conf. No.	
(Use several sheets if necessary)				Conley			7727	
(Form PTO-1449)				Filing Date			Art Group	
				October 13, 2005			2189	
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
	1	The Patent Office of the People's Republic of China, "Notification of the First Office Action," corresponding Chinese Patent Application No. 200610142359.8 on December 14, 2007, 8 pages (including translation).						
Examiner			Date Considered					
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

Electronic Acknowledgement Receipt

EFS ID:	2996548
Application Number:	11250238
International Application Number:	
Confirmation Number:	7727
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY
First Named Inventor/Applicant Name:	Kevin M. Conley
Customer Number:	66785
Filer:	Gerald Paul Parsons/Svetlana Stellmach (GPP)
Filer Authorized By:	Gerald Paul Parsons
Attorney Docket Number:	SNDK.156US2
Receipt Date:	13-MAR-2008
Filing Date:	13-OCT-2005
Time Stamp:	17:21:38
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no
------------------------	----

File Listing:

Document Number	Document Description	File Name	File Size(Bytes) /Message Digest	Multi Part /.zip	Pages (if appl.)
1	Information Disclosure Statement (IDS) Filed	SNDK156US2_Suppl_IDS_3-13-08.pdf	111059 07cb6331bd712e546c5c885b8d331a4ce b2ce1d6	no	3

Warnings:

Information:

This is not an USPTO supplied IDS fillable form

2	NPL Documents	Reference_SNDK156CN2_O A_12-14-08.pdf	560507 <small>d094a31cab4c874befd38e1dc6b3f7333 7a94d2b5</small>	no	8
---	---------------	--	---	----	---

Warnings:

Information:

Total Files Size (in bytes): 671566

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.

11/250,238 10/13/2005 Kevin M. Conley SNDK.156US2 7727

66785 7590 04/11/2008
DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION
505 MONTGOMERY STREET
SUITE 800
SAN FRANCISCO, CA 94111

Table with 1 column: EXAMINER

DINH, NGOC V

Table with 2 columns: ART UNIT, PAPER NUMBER

2189

Table with 2 columns: MAIL DATE, DELIVERY MODE

04/11/2008 PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 11/250,238	Applicant(s) CONLEY, KEVIN M.	
	Examiner NGOC V. DINH	Art Unit 2189	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 03 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 23 January 2008.
- 2a) This action is **FINAL**. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 4 and 7-25 is/are pending in the application.
4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) 13-18 is/are allowed.
- 6) Claim(s) 4, 7, 10-12, 19-22, 25 is/are rejected.
- 7) Claim(s) 8, 9, 23 and 24 is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) All b) Some * c) None of:
1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. _____.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) Notice of References Cited (PTO-892)
- 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date 01/23/2008 and 3/13/08.
- 4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .
- 5) Notice of Informal Patent Application
- 6) Other: _____.

DETAILED ACTION

1. This Office Action is responsive to amendment filed 11/16/2007. Applicant's arguments filed 11/13/06 are moot in view of new ground rejection.

INFORMATION DISCLOSURE STATEMENT

2. The Applicant's submission of the IDS' filed 01/13/2008 and 3/13/2008 have been considered. As required by M.P.E.P. 609 C(2), a copy of the PTOL-1449s are attached to the instant office action.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 4, 7, 10-13, 16-22, 25 are rejected under 35 U.S.C 103(a) as being unpatentable over Assar PN. 5,479,638, and in view of Baron et al. PN. 6,288,862.

Claims 4, 19, 20-21, Assar teaches a method of operating a memory system having an array of reprogrammable non-volatile charge storage elements Flash EEPROM, col. 3/45-50; fig. 1] organized in blocks of storage elements that are erasable together and in pages of storage elements within the blocks that are individually programmable as a unit [fig. 1-2], comprising: when updating data [updated information files, col. 2/62-63] previously written into one or more initial pages, writing the updated data into one or more update pages and identify the initial and update data pages [old/new Flag, fig. 1-2; col. 9/20-23] by common logical addresses [correlating a logical address assigned to superseded data to a physical address of updated data,

col. 9/28-33], wherein the memory system being operated includes a controller connected with the array of charge storage elements and is positioned within an enclosed card having externally accessible electrical contacts connected with the controller [col. 10/claim 18].

Assar does not teach: as part of writing data into pages, recording an indication of a time from a clock source that data are written into individual pages, when reading data of two or more pages having the same logical addresses, read the indications of the times that data have been stored in the two or more pages and use the data in the two or more pages having more recent time indications without using data in the two or more pages having older time indications.

Baron teaches a method of distinguish new information from old information, wherein new recording blocks and old recording blocks have the same recording block identifications, using a time stamp Write Pass Count (WPC), wherein new and old information both have the same block identification number (block ID), [distinguishing new recording blocks from old recording block stored in a medium where the new recording blocks and old recording blocks have the same recording block identifications, col.2/41-4, Fig. 4,8; Write Pass Count read from the medium ... distinguishing new recording blocks from old recording block, col. 2/21-40; col. 1/59-67;clock, col. 4/35-58; col. 5/13-28]. Therefore, the indications of the times (claim 4) and sequence of numbers (claim 21) correspond to Write Pass Count.

Accordingly, the limitation: “recording an indication of a time from a clock source” corresponds to [sequence counter or time stamp called a write Pass Count, col. 1/59-67; col. 3/34-40]; “when reading data of two or more pages having the same logical addresses” corresponds to [when reading the information from the medium, col. 1/50-55; In situations when two or more recording blocks have the same recording block identification, col. 2/23-25]; and “read the indications of the times that data have been stored in the two or more pages and use the data in the two or more pages having more recent time indications without using data in the two or more pages having older time indications” corresponds to [to provide a unique identifier for each occurrence that new information is written into the medium, col. 1/60-67; col. 5/12-25].

It would have been obvious to one having ordinary skill in the art at the time the invention was made to combine Baron to Assar in order to “prevent old information accidentally left in the medium may be output during read process” [Baron, col. 1/50-55].

Claims 7, 22, recording the indication of the time that data are written into individual pages includes recording the indication within the pages wherein the data are written [Baron, col. 6/1-15; Fig. 4].

Claims 11, 17, as part of writing data into pages, data are written in individual storage elements of the pages with more than two storage states, thereby storing more than one bit of data in the individual storage elements [Assar, Flash, fig. 2].

Claims 10, 12, 16, 18, 25, the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements [Assar, Flash]. As part of writing data into pages, logical addresses of the individual pages in which the data are written are also written in the individual pages [Assar, Fig. 2].

Claim 13, Assar and Baron teach the claimed limitations as mentioned above and further teach: reading data from the pages of the blocks in an order that is a reverse of the sequence in which they were written and ignore data in any page having the same logical address as a page from which data have already been read [Baron, By storing the Master Write Pass Count 134 in the nonvolatile memory 208, each time the Master Write Pass Count 134 is advanced, the new value can be written directly over the old value, col. 4/8-11]. Accordingly, during read operation, Baron’s system reads data in reverse order (read the new value and ignores old value) in order to prevent "problem when reading the information from the medium because old information accidentally left in the medium may be output during a read process, Baron, col. 1/50-55].

4. Claims 8-9, 14-15, 23-24 are rejected under 35 U.S.C 103(a) as being unpatentable over Assar, in view of Baron and further in view of The Admitted Prior Art (APA).

Claims 8, 14, 23, Assar-Baron does not explicitly teaches updating data previously written into one or more initial pages is limited to updating data in a number of pages less than all of the pages of a block while not updating data in a remaining one or more pages of the same block.

The APA teaches updating data previously written into one or more initial pages is limited to updating data in a number of pages less than all of the pages of a block while not updating data in a remaining one or more pages of the same block [Specification, page 2/[0007]].

It would have been obvious to one having ordinary skill in the art at the time the invention was made to combine the APA to Assar-Baron, because the process of “updating data in a number of pages less than all of the pages of a block while not updating data in a remaining one or more pages of the same block” is known in the art as flash partial-write process for efficiently storing and managing mapping information as well as data in flash memory device.

Claims 9, 15, 24, the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements [Assar, EEPROM, col. 3/45-50]

CONCLUSION

5. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a) Abe PN. 6,715,068 discloses Flash with read/write based on time stamp.

Any response to this action should be mailed to:

Under Secretary of Commerce for intellectual Property and Director of the

United States Patent and Trademark Office

PO Box 1450

Alexandria, VA 22313-1450

or faxed to:

(571) 273-8300, (for Official communications intended for entry)

Application/Control Number: 11/250,238
Art Unit: 2189

Page 6

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PMR) system. Status information for published Applications may be obtained from either Private PMR or Public PMR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pak-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ngoc Dinh whose telephone number is (571) 272-4191. The examiner can normally be reached on Monday-Friday 8:30 AM-5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Reginald Bragdon, can be reached on (571) 272-4204.

/N. V. D./

Examiner, Art Unit 2189

April 01, 2008

/Reginald G. Bragdon/

Supervisory Patent Examiner, Art Unit 2189

Notice of References Cited	Application/Control No. 11/250,238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.	
	Examiner NGOC V. DINH	Art Unit 2189	Page 1 of 1

U.S. PATENT DOCUMENTS

*	Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A US-6,715,068	03-2004	Abe, Kazunori	713/1
	B US-			
	C US-			
	D US-			
	E US-			
	F US-			
	G US-			
	H US-			
	I US-			
	J US-			
	K US-			
	L US-			
	M US-			


FOREIGN PATENT DOCUMENTS

*	Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N				
	O				
	P				
	Q				
	R				
	S				
	T				

NON-PATENT DOCUMENTS

*	Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U
	V
	W
	X

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

<i>Index of Claims</i> 	Application/Control No. 11250238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.
	Examiner NGOC V DINH	Art Unit 2189

✓	Rejected
=	Allowed


-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

Claims renumbered in the same order as presented by applicant
 CPA
 T.D.
 R.1.47

CLAIM		DATE									
Final	Original	03/03/2008									
	1	-									
	2	-									
	3	-									
	4	✓									
	5	-									
	6	-									
	7	✓									
	8	O									
	9	O									
	10	✓									
	11	✓									
	12	✓									
	13	=									
	14	=									
	15	=									
	16	=									
	17	=									
	18	=									
	19	✓									
	20	✓									
	21	✓									
	22	✓									
	23	O									
	24	O									
	25	✓									

Search Notes 	Application/Control No. 11250238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.
	Examiner NGOC V DINH	Art Unit 2189

SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES		
Search Notes	Date	Examiner
Limited classified search of Class/subclass. East text search w/o classified/search. See printout.	2/19/08	ND
Consulted Kevin Verbrugge	02/04/08	
Consulted Kevin Ellis	02/19/08	

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner

U.S. Department of Commerce, Patent and Trademark				Atty. Docket No.			Application No.	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT				SNDK.156US2			11/250,238	
				Applicants			Conf. No.	
(Use several sheets if necessary)				Conley			7727	
(Form PTO-1449)				Filing Date			Art Group	
				October 13, 2005			2189	
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
/N.D./	1	Japanese Patent Office, "Decision of Refusal," corresponding Japanese Patent Application No. 2002-558275 on November 27, 2007, 3 pages (including translation).						
/N.D./	2	The Patent Office of the People's Republic of China, "Notification of the First Office Action," corresponding Chinese Patent Application No. 200610142358.3 on December 14, 2007, 3 pages.						
Examiner	/Ngoc Dinh/			Date Considered	02/19/2008			
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	518166	(eprom flash non \$volatile)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/14 09:33
L2	14568	(programming program\$4 programing update updat\$3) same (time \$stamp (time adj stamp))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/14 09:35
L3	353	1 same 2	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/14 09:35
L4	6	(most near2 recent near3 (page block sector)) same 3	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/14 09:36
L5	12	(most near2 recent near3 (page block sector)) and 3	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/14 09:37
L6	19748	(programming program\$4 programing update updat\$3 read\$3 write writing) same (time \$stamp (time adj stamp))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/14 11:47
L7	50866	(programming program\$4 programing update updat\$3 read\$3 write writing) same (time \$stamp age (time adj stamp))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/14 11:48
L8	1088	1 same 7	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/14 11:48
L9	397749	(old new superseded modified updated) with (version data)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/14 11:49
L10	148	8 same 9	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/14 11:49

2/14/08 12:39:03 PM

C:\Documents and Settings\ndinh\My Documents\EAST\workspaces\11250238.wsp

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	759744	prom eeprom flash rom non \$volatile raid	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 10:27
L2	17	((time adj stamp) time \$stamp) same ((page block sector) with ("same" common identical) near2 (logical adj address)))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 10:32
L3	17	1 and 2	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 10:32
L5	248	(journal journal \$3) same ((time adj stamp) time \$stamp)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 12:27
L6	157	1 and 5	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 12:27
L7	319073	(new old superseded updated) with (version data)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 12:27
L8	1728	((new old superseded updated original modified) with (version data) same (logical near2 address)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 12:29
L9	1299	1 and 7 and 8	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 12:29
L10	1365	1 and 8	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 12:30

L11	2	10 and 5	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 12:30
L12	879	((("same" common identical) near2 (logical adj address))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 12:32
L13	151	7 same 12	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 12:32
L14	0	5 and 13	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 12:33
L15	0	(journal journal \$3) same 13	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 12:33
L16	23	((time adj stamp) time \$stamp) and 13	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 12:34
L17	147	1 same 12	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 13:10
L18	0	17 same (swapp swapping)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 13:10
L19	18	17 and (swapp swapping)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 13:10
L20	2415	(read write writing written) same ((data version page block sector) with (most near3 recently))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 14:00
L21	182	1 same 20	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/02/19 14:00

2/19/08 3:03:28 PM

C:\Documents and Settings\ndinh\My Documents\EAST\workspaces\11250238.wsp

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	527809	(eeprom flash non \$volatile)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/27 08:49
L2	51865	(programming program\$4 programing update updat\$3 read\$3 write writing) same (time \$stamp age (time adj stamp))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/27 08:49
L3	1109	L1 same L2	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/27 08:49
L4	404443	(old new superseded modified updated) with (version data)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/27 08:49
L5	149	L3 same L4	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/27 08:49
L6	527809	eeprom flash non \$volatile	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/27 08:52
L7	708	(writing write programming program) with (partial near2 (block page))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/27 08:54
L8	806	(writing write programming program updat\$3) with (partial near2 (block page))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/27 08:54
L12	146	6 same 8	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/27 08:56

3/27/08 10:12:39 AM

C:\Documents and Settings\ndinh\My Documents\EAST\workspaces\11250238.wsp

U.S. Department of Commerce, Patent and Trademark				Atty. Docket No.			Application No.	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT				SNDK.156US2			11/250,238	
				Applicants			Conf. No.	
(Use several sheets if necessary)				Conley			7727	
(Form PTO-1449)				Filing Date			Art Group	
				October 13, 2005			2189	
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
/N.D./	1	The Patent Office of the People's Republic of China, "Notification of the First Office Action," corresponding Chinese Patent Application No. 200610142359.8 on December 14, 2007, 8 pages (including translation).						
Examiner	/Ngoc Dinh/			Date Considered	04/09/2008			
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Kevin M. Conley
Title: Partial Block Data Programming And Reading Operations In A Non-Volatile Memory
Application No.: 11/250,238 Filing Date: October 13, 2005
Examiner: Dinh, Ngoc V. Group Art Unit: 2189
Docket No.: SNDK.156US2 Conf. No.: 7727

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE TO OFFICE ACTION AND AMENDMENT

Sir:

This is in response to the non-final Office Action dated April 11, 2008,
Claim Amendments are reflected in the listing of claims, which begins on page 2 of this
paper.

Specification Amendments begin on page 16 of this paper.

Remarks begin on page 17 of this paper.

Reconsideration is kindly requested in light of the following amendments and remarks.

Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

CLAIM AMENDMENTS

Please amend the claims by canceling claims 19-25, amending claims 4, 8, 10, 13, 14 and 16, and adding new claims 26-70, all without prejudice, as indicated on the following listing of all the claims in the present application after this Amendment:

1 – 3. (Cancelled)

4. (Currently Amended) A method of operating a memory system ~~having an array~~ of reprogrammable non-volatile charge storage elements organized in blocks of a minimum number of storage elements that are erasable together as a unit and in pages of storage elements within the blocks that have specified offset positions within their respective blocks and which are individually programmable as a unit, comprising:

as part of writing data into pages, recording an indication of a time from a clock source that data are written into individual pages,

~~when updating data previously written into one or more initial pages,~~ pages of one of original data blocks by writing the updated data into one or more update pages of an update data block and identify logically linking data of the corresponding previously written pages initial and update data pages by common logical addresses, wherein the updated data are caused to be writable into pages of the update data block having different offset positions than the pages of the original data block into which the logically linked data being updated was previously written, and

when reading data of two or more logically linked data pages ~~having the same logical addresses,~~ read the indications of the times that the data have been stored in the two or more pages and use the data in the two or more pages having more recent time indications without using data in the two or more pages having older time indications.

5 – 6. (Cancelled)

7. (Previously Presented) The method of claim 4, wherein recording the indication of the time that data are written into individual pages includes recording the indication within the pages wherein the data are written.

8. (Currently Amended) The method of claim 4, wherein updating data previously written into one or more ~~initial~~ pages of one of the original data blocks includes ~~is limited to~~ updating data in a number of pages less than all of the pages of ~~[[a]]~~ the one of the original data blocks while not updating data in a remaining one or more pages of the same original data block.

9. (Previously Presented) The method of claim 8, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

10. (Currently Amended) The method of claim 4, wherein as part of ~~writing~~ updating data into pages, logically linking the corresponding previously written pages and update data pages comprises writing logical addresses of ~~the individual pages in which the updated data are written~~ are also written in the individual update pages in which the updated data are written.

11. (Previously Presented) The method of claim 4, wherein as part of writing data into pages, data are written in individual storage elements of the pages with more than two storage states, thereby storing more than one bit of data in the individual storage elements.

12. (Previously Presented) The method of claim 4, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

13. (Currently Amended) A method of operating a memory system ~~having an array~~ of reprogrammable non-volatile charge storage elements organized in blocks of a minimum number of storage elements that are erasable together as a unit and in pages of storage elements

within the blocks that have specified offset positions within their respective blocks and which are individually programmable as a unit, comprising:

as part of writing data into pages, data are written into the pages of the individual blocks in sequence,

updating data previously written into one or more initial pages of one of original data blocks by writing the updated data into one or more update pages of an update data block and identify the initial and logically linking data of previously written pages with corresponding update data pages by common logical addresses, wherein the updated data are caused to be writable into pages of the update data block having different offset positions as the pages of the original data block into which the logically linked data being updated was previously written, and

reading data from the pages of the original data block and update data block blocks in an order that is a reverse of the sequence in which they were written and ignore data in any page that is logically linked with having the same logical address as a page from which data have already been read.

14. (Currently Amended) The method of claim 13, wherein updating data previously written into one or more ~~initial~~ pages of one of the original data blocks ~~includes is limited to~~ updating data in a number of pages less than all of the pages of ~~[[a]]~~ the one of the original data blocks while not updating data in a remaining one or more pages of the same original data block.

15. (Previously Presented) The method of claim 14, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

16. (Currently Amended) The method of claim 13, wherein as part of ~~writing~~ updating data into pages, logically linking the corresponding previously written pages and update data pages comprises writing logical addresses of the ~~individual pages in which the~~ updated data are written are also written in the individual update pages in which the updated data are written.

17. (Previously Presented) The method of claim 13, wherein as part of writing data into pages, data are written in individual storage elements of the pages with more than two storage states, thereby storing more than one bit of data in the individual storage elements.

18. (Previously Presented) The method of claim 13, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

19 – 25. (Cancelled)

26. (New) The method of claim 4, wherein logically linking the corresponding previously written pages and update data pages comprises maintaining common logical addresses thereof.

27. (New) The method of claim 26, wherein maintaining common logical addresses comprises storing the common logical addresses in the pages of the original data block and in the pages of the update data block along with the respective previously written and updated data therein.

28. (New) The method of claim 26, wherein reading the data further comprises organizing the read data by the logical page addresses associated with the read data.

29. (New) The method of claim 4, wherein data are written into individual pages of the original and update data blocks in a specified sequence.

30. (New) The method of claim 4, additionally comprising positioning the one original data block and the update data block in different ones of a plurality of units of the memory system, wherein the units are physically separate groupings of blocks of charge storage elements in which programming operations may be performed independently, and linking the

first and second blocks to cause their charge storage elements to be erasable together and their pages to be programmable together in parallel.

31. (New) The method of claim 4, additionally comprising:
updating data previously written into one or more pages of a second of the original data blocks by writing the updated data into one or more update pages of the update data block, and
logically linking data of the corresponding previously written pages of the second of the original data blocks and the update data pages.

32. (New) The method of claim 31, wherein updating data previously written into one or more pages of the one and of the second of the original data blocks includes updating data in a number of pages less than all of the pages of the individual original data block while not updating data in a remaining one or more pages of the same individual original data block.

33. (New) The method of claim 4, additionally comprising:
subsequently updating for a second time at least some of the data previously written into one or more pages of one of the original data blocks that were previously updated and written into the update data block,
writing the data updated for a second time into the update data block as second updated data pages, and
logically linking data of the corresponding previously written pages and the second updated data pages.

34. (New) The method of claim 33, wherein updating data previously written into one or more pages of one of the original data blocks and subsequently updating the data for the second time each includes updating data in a number of pages less than all of the pages of the individual original data block while not updating data in a remaining one or more pages of the same individual original data block.

35. (New) The method of claim 4, wherein the memory system in which the method is practiced is contained within an enclosed card having an electrical connector along one edge thereof for operably connecting with a host system.

36. (New) The method of claim 13, wherein logically linking the corresponding previously written pages and update data pages comprises maintaining common logical addresses thereof.

37. (New) The method of claim 36, wherein maintaining common logical addresses comprises storing the common logical addresses in the pages of the original data block and in the pages of the update data block along with the respective previously written and updated data therein.

38. (New) The method of claim 36, wherein reading the data further comprise organizing the read data by the logical page addresses associated with the read data.

39. (New) The method of claim 13, wherein data are written into individual pages of the blocks in a specified sequence.

40. (New) The method of claim 13, additionally comprising positioning the one original data block and the update data block in different ones of a plurality of units of the memory system, wherein the units are physically separate groupings of blocks of charge storage elements in which programming operations may be performed independently, and linking the first and second blocks to cause their charge storage elements to be erasable together and their pages to be programmable together in parallel.

41. (New) The method of claim 13, additionally comprising:
updating data previously written into one or more pages of a second of the original data blocks by writing the updated data into one or more update pages of the update data block, and

logically linking data of the corresponding previously written pages of the second of the original data blocks and the update data pages.

42. (New) The method of claim 41, wherein updating data previously written into one or more pages of the one and of the second of the original data blocks includes updating data in a number of pages less than all of the pages of the individual original data block while not updating data in a remaining one or more pages of the same individual original data block.

43. (New) The method of claim 13, additionally comprising:
subsequently updating for a second time at least some of the data previously written into one or more pages of one of the original data blocks that were previously updated and written into the update data block,

writing the data updated for a second time into the update data block as second updated data pages, and

logically linking data of the corresponding previously written pages and the second updated data pages.

44. (New) The method of claim 43, wherein updating data previously written into one or more pages of one of the original data blocks and subsequently updating the data for the second time each includes updating data in a number of pages less than all of the pages of the individual original data block while not updating data in a remaining one or more pages of the same individual original data block.

45. (New) The method of claim 13, wherein the memory system in which the method is practiced is contained within an enclosed card having an electrical connector along one edge thereof for operably connecting with a host system.

46. (New) In a re-programmable non-volatile semiconductor memory system having a plurality of blocks of a minimum number of memory charge storage elements that are erasable together as a unit, the plurality of blocks individually being divided into a plurality of a given

number of pages of memory storage elements that are individually programmable as a unit and which have specified offset positions within their respective blocks, a method of operating the memory system, comprising:

programming original data into individual ones of a first plurality of pages in at least a first block, the pages of original data having logical addresses associated therewith,

thereafter programming, into individual ones of a second plurality of pages in a second block, an updated version of less than the given number of pages of the original data programmed into the first plurality of pages, the pages of the updated version of the original data having logical addresses associated therewith, wherein the logical addresses associated with the pages of the updated version of the original data are the same as the logical addresses associated with the pages of original data,

wherein programming the second plurality of pages additionally comprises causing the updated version of the original data to be programmable in those of the second plurality of pages that have different offset positions within the second block than the offset positions of the first plurality of pages within said at least the first block that contain pages of original data with the same associated logical addresses,

thereafter reading data from the first and second plurality of pages, and organizing pages of the read data by their associated logical addresses.

47. (New) The method of claim 46, wherein reading data and organizing pages of the read data by their associated logical addresses comprises, for the pages of read data having the same logical addresses associated therewith, utilizing the pages of the updated version of the original data and omitting use of the pages of original data.

48. (New) The method of claim 47, wherein reading data and organizing pages of the read data by their associated logical addresses additionally comprises utilizing the pages of original data that have not been updated.

49. (New) The method of claim 48, wherein

programming original data into individual ones of the first plurality of pages and programming an updated version of the original data into individual ones of the second plurality of pages additionally comprises programming the individual pages with an indication of a relative time of programming the data therein, and

reading data and organizing pages of the read data additionally comprises, for pages of data having the same logical addresses associated therewith, selecting the updated data from the read pages having the more recent time indication and omitting use of the original data from the pages having the older time indication.

50. (New) The method of claim 46, wherein programming data into the first plurality of pages in at least the first block and programming data into the second plurality of pages in the second block each comprise programming data into individual pages of the individual blocks in a specified sequence.

51. (New) The method of claim 50, wherein reading data and organizing pages of the read data comprise reading the first and second plurality of pages in an order that is reverse to said specified sequence, and ignoring data read from pages having logical addresses that are the same as logical addresses associated with other pages of data that have already been read.

52. (New) The method of claim 46, wherein no indication is programmed into individual ones of the first plurality of pages, after the original data previously programmed therein are updated, that indicates that the previously programmed original data were updated.

53. (New) The method of claim 46, wherein programming original data into the first plurality of pages and programming the updated version of the original data into the second plurality of pages each comprise programming the logical addresses in individual pages along with the data with which the programmed logical addresses are associated.

54. (New) The method of claim 46, additionally comprising positioning the first and second blocks of charge storage elements in different ones of a plurality of units of the memory

system, wherein the plurality of units are physically separate groupings of blocks of charge storage elements in which programming operations may be performed independently, and linking the first and second blocks to cause their charge storage elements to be erasable together and their pages to be programmable together in parallel.

55. (New) The method of claim 46, additionally comprising:

subsequently programming, into individual ones of the second plurality of pages, a further updated version of at least some of the original data programmed into the first plurality of pages that have previously been updated and the updated version programmed into the second plurality of pages,

wherein the logical addresses associated with the pages of the further updated version of the original data are the same as the logical addresses associated with the pages of original data and the pages of the previously updated version of the original data.

56. (New) The method of claim 46, wherein the charge storage elements of the memory system in which the method is carried out comprise electrically conductive floating gates.

57. (New) The method of claim 46, additionally comprising operating the individual memory system charge storage elements with more than two storage states, thereby storing more than one bit of data in each storage element, wherein programming the pages includes programming the individual memory storage elements into more than two storage states and reading data includes reading the more than two storage states from the individual memory storage elements.

58. (New) The method of claim 46, wherein the memory system in which the method is practiced is contained within an enclosed card having an electrical connector along one edge thereof for operably connecting the memory system with a host system.

59. (New) In a re-programmable non-volatile semiconductor memory system having a plurality of blocks of memory charge storage elements that are erasable together as a unit, the plurality of blocks individually being divided into a plurality of a given number of pages of memory storage elements that are programmable together, a method of operating the memory system, comprising:

programming individual ones of a first plurality of said given number of pages in at least a first block with original data and a logical address associated with the page of original data,

thereafter programming individual ones of a second plurality of a total number of pages less than said given number in a second block with updated data and a logical address associated with the page of updated data, wherein the logical addresses associated with the pages of updated data programmed into the second plurality of pages are the same as those associated with the pages of original data programmed into the first plurality of pages,

wherein programming the second plurality of pages additionally comprises causing the updated version of the original data to be programmable in those of the second plurality of pages that have different offset positions within the second block than the offset positions of the first plurality of pages within said at least the first block that contain pages of original data with the same logical addresses associated therewith,

thereafter reading data from the first and second plurality of pages, and

organizing pages of the read data by the logical addresses associated therewith.

60. (New) The method of claim 59, wherein reading data and organizing pages of the read data by the logical addresses associated therewith comprises, for a plurality of pages of read data having the same logical addresses associated therewith, utilizing the pages of the updated version of the original data and omitting use of the pages of original data.

61. (New) The method of claim 60, wherein reading data and organizing pages of the read data by the logical addresses associated therewith additionally comprises utilizing the pages of original data that have not been updated.

62. (New) The method of claim 59, wherein programming data into the first plurality of pages in at least the first block and programming data into the second plurality of pages in the second block each comprise programming data into individual pages of the individual blocks in a specified sequence.

63. (New) The method of claim 59, wherein no indication is programmed into individual ones of the first plurality of pages, after the original data previously programmed therein are updated, that indicates that the previously programmed original data were updated.

64. (New) The method of claim 59, wherein programming original data into the first plurality of pages and programming the updated version of the original data into the second plurality of pages each comprise additionally programming in the pages the logical addresses for associated with the data programmed therein.

65. (New) The method of claim 59, additionally comprising positioning the first and second blocks of charge storage elements in different ones of a plurality of units of the memory system, wherein the units are physically separate groupings of blocks of charge storage elements in which programming operations may be performed independently, and linking the first and second blocks to cause their charge storage elements to be erasable together and their pages to be programmable together in parallel.

66. (New) The method of claim 59, additionally comprising:
subsequently programming, into individual ones of the second plurality of pages, a further updated version of at least some of the original data programmed into the first plurality of pages that have previously been updated and the updated version programmed into the second plurality of pages,

wherein the logical addresses associated with the further updated version of the original data are the same as the logical addresses associated with the original data and the previously updated version of the original data.

67. (New) The method of claim 59, wherein the addresses associated with pages of original and updated data are individually expressed as a logical block number and a logical page offset.

68. (New) A method of operating a memory system having an array of reprogrammable non-volatile charge storage elements organized in blocks of storage elements that are erasable together and in pages of storage elements within the blocks that are individually programmable as a unit, comprising:

maintaining an indication of a time separately for individual blocks,
as part of writing data into any one or more of the pages of one of the blocks, updating the indication of a time maintained for the one block to a current time,
when updating data previously written into one or more initial pages, writing the updated data into one or more update pages and identifying the data written into the initial and update pages by common logical addresses, and

as part of reading data having the same logical addresses from two or more pages of two or more blocks, reading the indications of the times from the two or more blocks and using the data in the two or more blocks having more recent indications of time without using data in the two or more blocks having older indications of time, and reading data from pages within each of the two or more blocks according to a reverse order in physical address to that by which the pages were written, and ignoring data from any page having the same logical address as data of a page from which data have already been read.

69. (New) A method of operating a non-volatile memory system having a plurality of blocks of memory storage elements that are individually erasable together, wherein the individual blocks are divided into a plurality of pages of storage elements that are programmable together, the blocks being organized in at least two separate units in which programming may be performed independently, comprising:

linking at least one block from individual ones of said at least two units to form a metablock wherein the storage elements of its component blocks are erased together, and

updating pages of original data within any of the metablock component blocks less than all the pages within the block by programming replacement data into pages within another at least one block in only a designated one of the units regardless of which unit the data being updated are stored.

70. (New) The method of claim 69, wherein storing the original and replacement data comprises:

identifying the original and replacement data by the same logical address to the memory system, and

distinguishing the replacement data from the original data by determining the relative order in time in which the original and replacement data have been programmed in their respective pages of the memory.

SPECIFICATION AMENDMENTS

Please amend paragraph and 0030, as follows:

[0030] If desired, a plurality of arrays 400, together with related X decoders, Y decoders, program/verified circuitry, data registers, and the like are provided, for example as taught by U.S. Patent 5,890,192, issued March 30, 1999, and assigned to Sandisk Corporation, the assignee of this application, which is hereby incorporated by this reference. Related memory system features are described in co-pending patent application serial no. 09/505,555, filed February 17, 2000 by Kevin Conley et al., now U.S. patent no. 6,426,893, which application is expressly incorporated herein by this reference.

REMARKS

Claims 4, 7-18 and 26-70 remain pending after this Amendment. Although a number of claims are being added, only six independent claims are present, namely amended versions of original claims 4 and 13 and new claims 46, 59, 68 and 69. Also, many of the claims dependent from claims 4, 13, 46 and 59 are repetitive in adding similar subject matter to their respective independent claims. Claims 19-25 are being cancelled.

The existing claims are being amended, and the new claims are presented, in a manner that more specifically distinguishes the cited prior art, and, therefore, are submitted to be patentable.

Claim Rejections Under 35 U.S.C. §103

Remaining claims 4, 7 and 10-13 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. patent no. 5,479,638 to Assar et al. (hereinafter "Assar") in view of U.S. patent no. 6,288,862 to Baron et al. (hereinafter "Baron"). Dependent claims 8-9 and 14-15 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Assar in view of Baron and further in view of prior art alleged to have been admitted by paragraph 0007 of the present application (hereinafter "APA").

Rejected Independent Claims 4 and 13

Since claims 4 and 13 have been rejected on similar grounds and there are common arguments in response, these claims are being discussed together.

Initially, it should be noted that neither of the cited Assar nor Baron references describe the claimed non-volatile memory architecture where memory elements are organized into blocks of storage elements that are erasable together, with pages of storage elements within the blocks that are programmable as a unit. In Assar, the memory 100 is divided into sector length data blocks 102 (Assar, col. 3, lns. 51-53) without the blocks being further divided into pages. Rather than using a table that links logical addresses of data with physical addresses of the memory blocks 102 in which the addressed data are stored, an address map is stored in a content addressable memory (CAM) 106. Baron describes a structure of data stored on magnetic tape, so is not related to a block erasable and page programmable memory system.

This is important in the present application since it is in large block reprogrammable non-volatile memories to which the invention is primarily directed. That is, blocks of charge storage elements that are erasable together are divided into pages of charge storage elements that are programmable as a unit. The problem solved by the present invention is how to efficiently operate such a memory system to update previously stored data in a manner that overcomes the disadvantages of prior techniques. (See, for example, paragraphs 0004 – 0008 of the Background and Summary of the present application.)

The cited Assar patent describes one of those prior techniques. Old/new flags 116 (Assar, Figs. 1 and 2) are stored in a content-addressable-memory (CAM), and old/new flags 104 are stored in the memory 100 with the data of its blocks N. Blocks of the memory 100 are addressed by the CAM. The flag 116 is set when the data of its associated block N have become obsolete by the writing of updated data (Assar, col. 4, lns. 51-53; steps 208 and 210 of Fig. 5). The purpose of setting the old/new flag is to mark blocks whose data and CAM entries are to be erased (col. 4, lns. 62-65) and to prevent the superseded version of the data from being accessed (col. 6, lns. 1-4).

As stated in para. 0008 of the present application, an important aspect of the present invention is to avoid the use of the flags described in para. 0007 that are written to indicate obsolete data because this usually involves overwriting a portion of the memory already programmed and this can cause these programmed data to be disturbed (that is, the stored charge levels changed). Such overwriting may occur because setting the flag 116 causes the flag data already written in the CAM block 110, 114 of memory cells to be changed, adjacent other data being stored in the CAM blocks 110, 114 that can be disturbed. The same overwriting occurs with the flag 104 in the blocks N of data storage capacity. Each of Assar's memory blocks stores one sector of data (Assar, col. 3, lns. 51-54), not multiple sectors or pages.

The solution of claim 4 to the problem of overwriting flags is to instead keep track of when data are written into pages by writing an indication of time from a clock source as to when data are written. The updated data are then distinguishable from the previously written data to which the updated data are logically linked. The updated data have the most recent indication of time. Claim 4 recites this as occurring in pages of blocks, wherein the blocks are specified in the preamble to contain a minimum number of storage elements that are erasable together as a unit,

and the pages are individually programmable as a unit. The solution of claim 13 to the problem of overwriting flags is to instead read pages of data in an order that is a reverse of the sequence in which the pages were written, and ignore any data in any page that is logically linked with data of a page already read. Assar describes a much different type of memory system than that claimed, and uses the type of flags to mark updated obsolete data that the present application describes in paras. 0007 and 0008 to be undesirable. The present invention is an improvement over Assar technique by using a technique that does not need such flags in order to separately identify original data and their updates.

Claims 4 and 13 are each being amended to add a feature specific to its different type of memory system that further distinguishes Assar. They each now specify that the updated data are caused to be writable into pages of the update block that have different offsets within that block than the pages of the original data block in which the updated data were previously written. This overcomes problems with another prior art technique described in the Background of the present application (see para. 0007, lns. 7-11), in a manner summarized in the Summary of the present application (see para.0010). The update block is used more efficiently and the need for time consuming data copying from one block to another is reduced.

The cited Baron reference, describing the storage of data blocks on magnetic tape, certainly does not suggest a need to avoid the use of Assar's flags. Although Baron records the time of writing blocks of data on magnetic tape, relevant to claim 4, Baron does not mention anything about the problem with overwriting flags, such as Assar's old/new flag 116, in reprogrammable non-volatile memory. It is not understood why one of ordinary skill would have thought that use of Assar's old/new flag needed to be changed. Neither of the cited references refers to any problem with it. It is only the present application that describes the undesirability of marking data that have been updated with flags.

With regard to claim 13, the Office Action (p. 4) alleges that Baron's reading of its Master Write Pass Count 134 suggests reading data in a reverse order. When the Master Write Pass Count 134 is updated, the prior value is apparently written over and thus disappears. It is the current updated value of the Master Write Pass Count 134 that Baron describes to be added to the header of each of the recording blocks (see Baron, col. 3, lns. 42-46). The Office Action refers particularly to Baron's disclosure in col. 4, lns. 8-11 about updating the Master Write Pass

Count 134. The Office Action seems to contend that by reading the new value of the Master Write Pass Count 134 and ignoring the old value, this is reading data in a reverse order.

However, this ground of rejection is not understood, and so for that reason is respectfully traversed. If this ground of rejection is repeated in a future Office Action, a further explanation of the basis for the rejection would be appreciated. But there are other reasons given herein to support the patentability of claim 13 over Assar and Baron.

In the Office Action (p. 4, lns. 1-3) it is stated that the motivation of combining Baron with Assar was to “prevent old information accidentally left in the medium may be output during read process,” citing Baron, Col. 1, lns. 50-55. But that is what Assar’s old/new flag does. There is no suggestion in either reference that there is a problem with this. The re-writing of flags into programmed blocks are described in the present application as undesirable (see paras. 0007 and 0008). Without a perceived need to improve Assar, it is submitted that use of an indication of time from a clock source for blocks of data stored on a magnetic tape would not have suggested modifying Assar to replace the use of flags with indications of time in the type of memory recited in claims 4 and 13, which is different than that of Assar.

Regardless of whether it would have been obvious to combine the teachings of Baron with those of Assar in the different type of memory recited in claims 4 and 13, neither reference suggests the amended feature of causing updated data to be writable into pages of an update block that have different offset positions from those of the pages of the original data block which contained the logically linked previously written data that was updated. Even if it would have been obvious to modify Assar in view of Baron, this feature now part of each of claims 4 and 13 would not have been suggested.

Rejected Dependent Claims 7-12 and 14-18, and New Dependent Claims 26-45

In addition to independent claims 4 and 13 being patentable, their dependent claims add additional novel features, some of which are pointed out in this section.

Claims 8, 14, 32, 34, 42 and 44 specify that fewer than all the pages of data of a block are updated, and the Office Action (page 5) refers to the discussion of paragraph 0007 of the present application as APA. What the present application describes is the situation where the techniques of the present invention are particularly valuable, namely where only a small amount of data are being updated. In the large block, multiple page type of memories, a great deal of overhead used

to be expended in copying data and the like in order to update only one or a few pages of data. The prior art use of flags reduced the amount of data copying somewhat but updating data, as recognized as part of the present invention, was still not very efficient. The updating of a small number of pages of initial data, relative to the given number of pages in a block, is particularly benefited from the feature of claims 4 and 13 wherein the updated data are caused to be writable in pages of the update block with offsets different than those of the initial pages of data that have been updated.

Claims 30 and 40 recite placement of the original data blocks and update data blocks in different units of memory having defined characteristics. This is supported by the discussion in the present application (see paragraphs 0062-0063) of the use of metablocks.

Claims 31, 32, 41 and 42 contain the feature of writing of updates from first and second update blocks into a common update data block. This concept is described in paragraphs 0062-0064 of the present application.

Claims 33, 34, 43 and 44 add to the combination of claim 4 the updating of original data for a second time. An example of this is illustrated in Figure 11 of the present application and described in the present application with respect to that figure.

For the reasons given in this and the preceding section, all of the claims 4, 7-18 and 26-45 are submitted to be patentable.

New Claims 46-67

Independent claims 46 and 59 of this group of claims are similar to claim 24 of the grandparent patent no. 6,763,424 of the present application but with differences in their scope.

These claims define a method of operating a memory system having its charge storage elements organized into blocks that are erasable together, and the blocks divided into pages of memory storage elements that are individually programmable as a unit. It is in this type of reprogrammable non-volatile memory system where the methods of claims 46-67 provide significant improvements.

Claims 46 and 59 each include the feature, similar to one discussed above that is part of independent claims 4 and 13, wherein the updated version of the original data are caused to be programmable into pages of the second block that have different offset positions within that

block than the offset positions in the at least the first block that contain the original data with the same associated logical addresses.

Claims 46 and 59 also call for programming into the second plurality of pages an updated version of less than the given number of pages of original data, wherein the “given number” is the number of pages in an individual block. Similar limitations are discussed above with respect to dependent claims 8, 14, 32, 34, 42 and 44.

Further, independent claims 46 and 59 each include a limitation of “organizing pages of the read data by their associated logical addresses” in the last paragraph of each.

Dependent claims 47-58 and 60-67 contain many of the same limitations that are discussed above with respect to dependent claims 7-12, 14-18 and 26-45.

New Claim 68

Claim 68 is similar to the main claim allowed in Japanese patent application no. 2002-558275, based on international patent application no. PCT/US02/00366 that in turn claims priority from United States application no. 09/766,436, the grandparent to the present application. Claim 68 recites maintaining an indication of time for blocks and reading data from pages in individual blocks in a reverse order. This combination of the use of the indication of time and reverse reading is not suggested by the cited prior art. An example of this technique is described in the present application in the last sentence of paragraph 0008 (Summary), and in paragraphs 0053 and 0054 (a second specific implementation).

New Claim 69-70

Claims 69-70 are similar to, but with a different scope than, claims 17 and 18 of grandparent patent no. 6,763,424 and claims 16 and 17 of parent patent no. 6,968,421. The claimed subject matter is described in the present application primarily in paragraph 0011 (Summary) and paragraphs 0062 – 0064 with respect to Figures 15 and 16.

Terminal Disclaimers Already Filed

A Terminal Disclaimer over parent patent no. 6,968,421 was filed in the present application January 8, 2007, and another Terminal Disclaimer over grandparent patent no. 6,763,424 was filed February 21, 2007.

Information Disclosure Statements

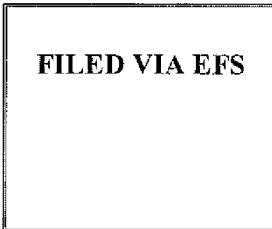
It is noted with appreciation that the Examiner has considered and made of record all of the references submitted in Information Disclosure Statements dated March 13, 2008 and January 23, 2008.

Another Information Disclosure Statement is being prepared for filing in this case, primarily to identify prior art that has been alleged to be relevant to claims 17, 18, 20, 24 and 30 of the grandparent patent no. 6,763,424 by Respondents to Investigation No. 337-TA-619 of the United States International Trade Commission (ITC). Claims 17, 18, 24 and 30 of patent no. 6,763,424 have been asserted by Complainant SanDisk Corporation in that ITC action to be infringed by "Certain Flash Memory Controllers, Drives, Memory Cards, and Media Players, and Products Containing Same" being imported by Respondents. A trial in this matter before the Administrative Law Judge is expected later this Fall.


In addition to prior art, a portion of an Order (dated July 15, 2008) of the Administrative Law Judge in the ITC action is being included. That portion construes certain terms of the asserted claims of grandparent patent no. 6,763,424. The portions of the Order construing claims of two other unrelated patents is being omitted in order to reduce the size of what is being filed.

Conclusion

It is believed that this application is now in condition for allowance and an early indication of its allowance is solicited. However, if the Examiner has any further matters that need to be resolved or any questions about this response, a telephone call to the undersigned attorney at 415-276-6534 (direct line) would be appreciated.



Respectfully submitted,


Gerald P. Parsons
Reg. No. 24,486

October 10, 2008
Date

Davis Wright Tremaine LLP
505 Montgomery Street, Suite 800
San Francisco, CA 94111-6533
(415) 276-6500 (main)
(415) 276-6534 (direct)
(415) 276-6599 (fax)
Email: geraldparsons@dwt.com

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

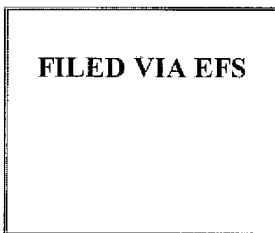
Applicant: Kevin M. Conley
 Title: Partial Block Data Programming And Reading Operations In A Non-Volatile Memory
 Application No.: 11/250,238 Filing Date: October 13, 2005
 Examiner: Dinh, Ngoc V. Group Art Unit: 2189
 Docket No.: SNDK.156US2 Conf. No.: 7727

Mail Stop Amendment
 Commissioner for Patents
 P.O. Box 1450
 Alexandria, VA 22313-1450

PETITION FOR EXTENSION OF TIME

Dear Sir:

Applicant respectfully petitions for a three-month extension of time within which to respond to the April 11, 2008, outstanding Office Action, such extension allowing the undersigned until October 11, 2008, to respond. The fee of \$1,110.00 has been authorized via EFS to Deposit Account 04-0258. The Commissioner is hereby authorized to charge any additional fees, which may be required, or credit any overpayment to Deposit Account 04-0258.



Respectfully submitted,

Gerald P. Parsons October 10, 2008
 Gerald P. Parsons Date
 Reg. No. 24,486

DAVIS WRIGHT TREMAINE LLP
 505 Montgomery Street, Suite 800
 San Francisco, California 94111-6533
 (415) 276-6500 (main)
 (415) 276-6534 (direct)
 (415) 276-6599 (fax)
 Email: geraldparsons@dwt.com

Electronic Patent Application Fee Transmittal

Application Number:	11250238			
Filing Date:	13-Oct-2005			
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY			
First Named Inventor/Applicant Name:	Kevin M. Conley			
Filer:	Gerald Paul Parsons/Svetlana Stellmach (GPP)			
Attorney Docket Number:	SNDK.156US2			
Filed as Large Entity				
Utility under 35 USC 111(a) Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Claims in excess of 20	1202	38	52	1976
Independent claims in excess of 3	1201	3	220	660
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Extension - 3 months with \$0 paid	1253	1	1110	1110
Miscellaneous:				
Total in USD (\$)				3746

Electronic Acknowledgement Receipt

EFS ID:	4094140
Application Number:	11250238
International Application Number:	
Confirmation Number:	7727
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY
First Named Inventor/Applicant Name:	Kevin M. Conley
Customer Number:	66785
Filer:	Gerald Paul Parsons/Svetlana Stellmach (GPP)
Filer Authorized By:	Gerald Paul Parsons
Attorney Docket Number:	SNDK.156US2
Receipt Date:	10-OCT-2008
Filing Date:	13-OCT-2005
Time Stamp:	12:50:50
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$3746
RAM confirmation Number	8554
Deposit Account	040258
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
-----------------	----------------------	-----------	-------------------------------------	------------------	------------------

1		SNDK156US2_Trans_Response- OA_Ext-Time_10-10-08.pdf	844878 ed31ddee9bc5aaef8e31fd79ef6aa3d90839 bff9	yes	26
Multipart Description/PDF files in .zip description					
Document Description		Start	End		
Miscellaneous Incoming Letter		1	1		
Amendment/Req. Reconsideration-After Non-Final Reject		2	2		
Claims		3	16		
Specification		17	17		
Applicant Arguments/Remarks Made in an Amendment		18	25		
Extension of Time		26	26		
Warnings:					
Information:					
2	Fee Worksheet (PTO-06)	fee-info.pdf	34029 6edf2583f872417564ba9ff304affdd304b0a f9b	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			878907		
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					

Davis Wright Tremaine LLP



ANCHORAGE BELLEVUE LOS ANGELES NEW YORK PORTLAND SAN FRANCISCO SEATTLE SHANGHAI WASHINGTON, D.C.

SUITE 800
505 MONTGOMERY STREET
SAN FRANCISCO, CA 94111-6533

TEL (415) 276-6500
FAX (415) 276-6599
www.dwt.com

October 10, 2008

Customer No. 66785

Commissioner For Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Re: Applicant: Kevin M. Conley
Title: Partial Block Data Programming And Reading Operations In A Non-Volatile Memory
Application No.: 11/250,238 Filing Date: October 13, 2005
Examiner: Dinh, Ngoc V. Group Art Unit: 2189
Docket No.: SNDK.156US2 Conf. No.: 7727

Dear Sir:

Transmitted herewith are the following documents in the above-identified application:

- (1) This Transmittal Letter;
- (2) Response to Office Action (24 pages);
- (3) Petition for Extension of Time (1 page).

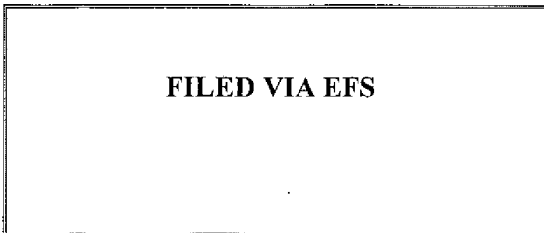
The fee has been calculated as shown below:

CLAIMS AS AMENDED

	Claims Remaining <u>After</u> <u>Amendment</u>		Highest No. Previously Paid For	=	Present Extra	x	Rate	\$	Additional Fee	
Total Claims	58	Minus	20	=	38	x	\$52.00	\$	1,976.00	
Independent Claims	6	Minus	3	=	3	x	\$220.00	\$	660.00	
<input type="checkbox"/>	Fee of _____ for the first filing of one or more multiple dependent claims per application								\$	0.00
<input checked="" type="checkbox"/>	Fee for three-month of Extension of Time								\$	1,110.00
<u>Total additional fee for this Amendment:</u>								\$	<u>3,746.00</u>	

Conditional Petition for Extension of Time: If an extension of time is required for timely filing of the enclosed document(s) after all papers filed with this transmittal have been considered, an extension of time is hereby requested.

The fee of \$3,746.00 has been authorized via EFS to Deposit Account 040258. The Commissioner is hereby authorized to charge any additional fees, which may be required, or credit any overpayment to Deposit Account 040258.



Respectfully submitted,

Gerald P. Parsons
Reg. No. 24,486

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875					Application or Docket Number 11/250,238		Filing Date 10/13/2005		<input type="checkbox"/> To be Mailed										
APPLICATION AS FILED – PART I																			
(Column 1)			(Column 2)			SMALL ENTITY <input type="checkbox"/>		OR			OTHER THAN SMALL ENTITY								
FOR		NUMBER FILED	NUMBER EXTRA		RATE (\$)	FEE (\$)	OR		RATE (\$)	FEE (\$)									
<input type="checkbox"/> BASIC FEE <small>(37 CFR 1.16(a), (b), or (c))</small>		N/A	N/A		N/A				N/A										
<input type="checkbox"/> SEARCH FEE <small>(37 CFR 1.16(k), (l), or (m))</small>		N/A	N/A		N/A		N/A												
<input type="checkbox"/> EXAMINATION FEE <small>(37 CFR 1.16(o), (p), or (q))</small>		N/A	N/A		N/A		N/A												
TOTAL CLAIMS <small>(37 CFR 1.16(i))</small>		minus 20 =	*		X \$ =		OR		X \$ =										
INDEPENDENT CLAIMS <small>(37 CFR 1.16(h))</small>		minus 3 =	*		X \$ =		OR		X \$ =										
<input type="checkbox"/> APPLICATION SIZE FEE <small>(37 CFR 1.16(s))</small>		If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$250 (\$125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).																	
<input type="checkbox"/> MULTIPLE DEPENDENT CLAIM PRESENT <small>(37 CFR 1.16(j))</small>																			
* If the difference in column 1 is less than zero, enter "0" in column 2.																			
APPLICATION AS AMENDED – PART II										SMALL ENTITY		OR		OTHER THAN SMALL ENTITY					
(Column 1)			(Column 2)			(Column 3)			RATE (\$)		ADDITIONAL FEE (\$)		RATE (\$)		ADDITIONAL FEE (\$)				
AMENDMENT	10/10/2008		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	OR		OR		OR		OR						
	Total <small>(37 CFR 1.16(o))</small>	* 58	Minus	** 20	= 38	X \$ =										X \$2=	1976		
	Independent <small>(37 CFR 1.16(h))</small>	* 6	Minus	***3	= 3	X \$ =		X \$220=	660										
	<input type="checkbox"/> Application Size Fee <small>(37 CFR 1.16(s))</small>																		
	<input type="checkbox"/> FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM <small>(37 CFR 1.16(j))</small>																		
												TOTAL ADD'L FEE		OR		TOTAL ADD'L FEE		2636	
AMENDMENT			CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	OR		OR		OR		OR						
	Total <small>(37 CFR 1.16(o))</small>	*	Minus	**	=	X \$ =										X \$ =			
	Independent <small>(37 CFR 1.16(h))</small>	*	Minus	***	=	X \$ =		X \$ =											
	<input type="checkbox"/> Application Size Fee <small>(37 CFR 1.16(s))</small>																		
	<input type="checkbox"/> FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM <small>(37 CFR 1.16(j))</small>																		
												TOTAL ADD'L FEE		OR		TOTAL ADD'L FEE			
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.																			
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".																			
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".																			
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.																			
Legal Instrument Examiner: /DALE A. HALL/																			

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Kevin M. Conley
Title: Block Data Programming and Reading Operations in a Non-Volatile
Memory
Application No.: 11/250,238 Filing Date: October 13, 2005
Examiner: Ngoc V. Dinh Group Art Unit: 2189
Docket No.: SNDK.156US2 Conf. No.: 7727

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Dear Sir:

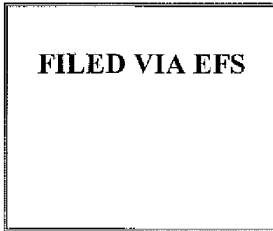
Pursuant to 37 C.F.R. §§ 1.56, 1.97 and 1.98, the documents listed on the enclosed Form PTO-1449 are being called to the Examiner's attention.

According to 37 C.F.R. 1.98(2)(ii), copies of the U.S. Patents and U.S. Published Patent Applications documents are not required and are therefore not enclosed. Copies of the listed Foreign Patent Documents and Other Art are enclosed.

Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

This information disclosure statement is submitted under 37 C.F.R. § 1.97(c). The fee of \$180.00 has been authorized via EFS to Deposit Account 04-0258. The Commissioner is hereby authorized to charge any additional fees, which may be required, or credit any overpayment to Deposit Account 04-0258.



Respectfully submitted,

Gerald P. Parsons October 15, 2008
Gerald P. Parsons Date
Reg. No. 24,486

Davis Wright Tremaine LLP
505 Montgomery Street, Suite 800
San Francisco, CA 94111-6533
(415) 276-6500 (main)
(415) 276-6534 (direct)
(415) 276-6599 (fax)
Email: geraldparsons@dwt.com

U.S. Department of Commerce, Patent and Trademark	Atty. Docket No.	Application No.
INFORMATION DISCLOSURE STATEMENT BY APPLICANT	SNDK.156US2	11/250,238
	Applicants	Conf. No.
(Use several sheets if necessary)	Conley	7727
(Form PTO-1449)	Filing Date	Art Group
	October 13, 2005	2189

U.S. Patent Documents

*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate
	1	5,388,248	2/7/1995	Robinson et al.			
	2	5,544,356	8/6/1996	Robinson et al.			
	3	5,627,783	5/6/1997	Miyauchi			
	4	5,682,499	10/28/1997	Bakke et al.			
	5	5,740,396	4/14/1998	Mason			
	6	5,822,781	10/13/1998	Wells et al.			
	7	5,867,417	2/2/1999	Wallace et al.			
	8	6,202,138 B1	3/13/2001	Estakhri et al.			
	9	6,219,752 B1	4/17/2001	Sekido			
	10	6,262,918 B1	6/17/2001	Estakhri et al.			
	11	6,288,862 B1	9/11/2001	Baron et al.			
	12	6,330,633 B1	12/11/2001	Kusakabe et al.			
	13	6,584,579 B1	6/24/2003	Komatsu et al.			
	14	6,725,321 B1	4/20/2004	Sinclair et al.			
	15	6,845,438 B1	1/18/2005	Tanaka et al.			
	16	6,925,012 B2	8/2/2005	Yamagami et al.			
	17	7,167,944 B1	1/23/2007	Estakhri			

Foreign Patent Documents

							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
	18	EP 1 352 394 B1	5/24/2006	Europe				
	19	JP 8-212019 A	8/20/1996	Japan			Abstract	
	20	JP 8-212019 A	8/20/1996	Japan			X	
	21	JP 11-110300 A	4/23/1999	Japan			Abstract	
	22	JP P3070539 B2	7/31/2000	Japan			X	
	23	JP 2000-163302 A	6/16/2000	Japan			Abstract	

FILED VIA EFS
Sheet 1 of 2

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.	Application No.
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		SNDK.156US2	11/250,238
		Applicants	Conf. No.
(Use several sheets if necessary)		Conley	7727
(Form PTO-1449)		Filing Date	Art Group
		October 13, 2005	2189
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)			
	24	Japanese Patent Office, "Notification of Reasons for Refusal," corresponding Japanese Patent Application No. 2002-558275, mailed on May 13, 2008, 3 pages.	
	25	Korean Patent Office, "Office Action," corresponding Korean Patent Application No. 2003-7009551, mailed on August 25, 2008, 14 pages (including translation.)	
	26	PCMCIA, PC Card Standard 6.0. PCMCIA March 1997, Volume 1, Overview and Glossary, 29 pages.	
	27	PCMCIA, PC Card Standard 6.0. PCMCIA March 1997, Volume 3, Physical Specification, 61 pages.	
	28	PCMCIA, PC Card Standard 6.0. PCMCIA March 1997, Volume 7, Media Formats Storage, Specification, 39 pages.	
	29	Barre, "Flash Memory Magnetic Disk Replacement," IEEE Transactions on Magnetics, Vol. 29, No. 6, November 1993, 4 pages.	
	30	Chan, "World's Smallest Solid State Storage Device Sets New Industry Standard," IEEE, November 4-6, 1997, pp. 484-487.	
	31	Wells et al. "Flash Solid State Drive with 6MB/s Read/Write Channel and Data Compression," ISSCC 93/Session 3/Non-Volatile, Dynamic, and Experimental Memories/Paper WP 3.6, 4 pages.	
	32	United States International Trade Commission, In the Matter of Certain Flash Memory Controllers, Drivers, Memory Cards, and Media Palyers and Products Containing same, Inv. No. 337-TA-619, Order No. 33: Order Construing the Terms of the Asserted Claims of the Patents at Issue, July 15, 2008, pages i-iii, 1-10 and 54-67.	
Examiner		Date Considered	
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.			

(19)



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) **EP 1 352 394 B1**

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
24.05.2006 Bulletin 2006/21

(51) Int Cl.:
G11C 16/00 (2006.01)

(21) Application number: **02703078.2**

(86) International application number:
PCT/US2002/000366

(22) Date of filing: **07.01.2002**

(87) International publication number:
WO 2002/058074 (25.07.2002 Gazette 2002/30)

(54) **PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY**

TEILDATENPROGRAMMIER- UND LESEOPERATIONEN IN EINEM NICHTFLÜCHTIGEN SPEICHER

OPERATIONS DE PROGRAMMATION ET DE LECTURE DE BLOCS DE DONNEES PARTIELLES DANS UNE MEMOIRE NON VOLATILE

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR

(72) Inventor: **CONLEY, Kevin, M. San Jose, CA 95120 (US)**

(30) Priority: **19.01.2001 US 766436**

(74) Representative: **Hitchcock, Esmond Antony Lloyd Wise Commonwealth House, 1-19 New Oxford Street London WC1A 1LW (GB)**

(43) Date of publication of application:
15.10.2003 Bulletin 2003/42

(60) Divisional application:
**05077929.7 / 1 645 964
06075106.2 / 1 653 323**

(56) References cited:
**WO-A-02/49039 FR-A- 2 742 893
US-A- 5 598 370**

(73) Proprietor: **SanDisk Corporation Sunnyvale, CA 94089 (US)**

EP 1 352 394 B1

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Description

[0001] This invention pertains to the field of semiconductor non-volatile data storage system architectures and their methods of operation, and has application to data storage systems based on flash electrically erasable and programmable read-only memories (EEPROMs).

[0002] A common application of flash EEPROM devices is as a mass data storage subsystem for electronic devices. Such subsystems are commonly implemented as either removable memory cards that can be inserted into multiple host systems or as non-removable embedded storage within the host system. In both implementations, the subsystem includes one or more flash devices and often a subsystem controller.

[0003] Flash EEPROM devices are composed of one or more arrays of transistor cells, each cell capable of non-volatile storage of one or more bits of data. Thus flash memory does not require power to retain the data programmed therein. Once programmed however, a cell must be erased before it can be reprogrammed with a new data value. These arrays of cells are partitioned into groups to provide for efficient implementation of read, program and erase functions. A typical flash memory architecture for mass storage arranges large groups of cells into erasable blocks, wherein a block contains the smallest number of cells (unit of erase) that are erasable at one time.

[0004] In one commercial form, each block contains enough cells to store one sector of user data plus some overhead data related to the user data and/or to the block in which it is stored. The amount of user data included in a sector is the standard 512 bytes in one class of such memory systems but can be of some other size. Because the isolation of individual blocks of cells from one another that is required to make them individually erasable takes space on the integrated circuit chip, another class of flash memories makes the blocks significantly larger so there is less space required for such isolation. But since it is also desired to handle user data in much smaller sectors, each large block is often further partitioned into individually addressable pages that are the basic unit for reading and programming user data (unit of programming and/or reading). Each page usually stores one sector of user data, but a page may store a partial sector or multiple sectors. A "sector" is used herein to refer to an amount of user data that is transferred to and from the host as a unit.

[0005] The subsystem controller in a large block system performs a number of functions including the translation between logical addresses (LBAs) received by the memory sub-system from a host, and physical block numbers (PBNs) and page addresses within the memory cell array. This translation often involves use of intermediate terms for a logical block number (LBN) and logical page. The controller also manages the low level flash circuit operation through a series of commands that it issues to the flash memory devices via an interface bus. Another function the controller performs is to maintain the integrity of data stored to the subsystem through various means, such as by using an error correction code (ECC).

[0006] In an ideal case, the data in all the pages of a block are usually updated together by writing the updated data to the pages within an unassigned, erased block, and a logical-to-physical block number table is updated with the new address. The original block is then available to be erased. However, it is more typical that the data stored in a number of pages less than all of the pages within a given block must be updated. The data stored in the remaining pages of the given block remains unchanged. The probability of this occurring is higher in systems where the number of sectors of data stored per block is higher. One technique now used to accomplish such a partial block update is to write the data of the pages to be updated into a corresponding number of the pages of an unused erased block and then copy the unchanged pages from the original block into pages of the new block. The original block may then be erased and added to an inventory of unused blocks in which data may later be programmed. Another technique similarly writes the updated pages to a new block but eliminates the need to copy the other pages of data into the new block by changing the flags of the pages in the original block which are being updated to indicate they contain obsolete data. Then when the data are read, the updated data read from pages of the new block are combined with the unchanged data read from pages of the original block that are not flagged as obsolete.

[0007] Reference is directed to US Patent No: 5,598,370 in which is disclosed a non-volatile memory with cluster-erase flash capability. This can be used to simultaneously store original and replacement data by identifying both data by the same logical address, and distinguishing replacement from original data with reference to the sequence numbers assigned thereto. The preamble of claim 1 is based on this document.

[0008] The present invention is directed at a method of substituting new data for superseded data in a non-volatile memory system having an array of non-volatile memory storage elements organised in blocks with the blocks organised in pages, each block containing the smallest group of storage elements that is erasable. The method comprises writing new data as updated pages into pages of a said block, said new data being less than all of the original pages of another said block; addressing both the original pages and the updated pages with the same logical address; and reading and distinguishing the updated pages from the original pages. According to the invention the reading is with reference to the relative time of writing into pages of said blocks by reading the pages from said block and then from said other block both in a reverse order from that in which the pages have been written and ignores any pages having the same logical address as pages it has already read.

[0009] A non-volatile memory system according to the invention comprises an array of non-volatile memory storage elements organised in blocks, said blocks organised in pages, wherein a block contains the smallest group of storage elements that is erasable; a programming mechanism that writes into pages of a said block an updated version of less than all of the original pages of another said block; an address mechanism that addresses both the original pages and the updated pages with the same logical address; and a reading mechanism that distinguishes the updated pages from the original pages. The reading mechanism reads with reference to the relative time of writing into pages of said blocks by reading the pages from said block and then from said other block in a reverse order from that in which the pages have been written and ignores any pages having the same logical address as pages it has already read.

[0010] Using the present invention, both the copying of unchanged data from the original to the new blocks and the need to update flags within the original block can be avoided when the data of fewer than all of the pages within a block are being updated. This is accomplished by maintaining both the superceded data pages and the updated pages of data with a common logical address. The original and updated pages of data are then distinguished by the relative order in which they were programmed. During reading the most recent data stored in the pages having the same logical address are combined with the unchanged pages of data while data in the original versions of the updated pages are ignored. The updated data can be written to either pages within a different block than the original data, or to available unused pages within the same block. In one specific implementation, a form of time stamp is stored with each page of data that allows determining the relative order that pages with the same logical address were written. In another specific implementation, in a system where pages are programmed in a particular order within the blocks, a form of time stamp is stored with each block of data, and the most recent copy of a page within a block is established by its physical location within the block.

[0011] These techniques avoid both the necessity for copying unchanged data from the original to new block and the need to change a flag or other data in the pages of the original block whose data have been updated. By not having to change a flag or other data in the superceded pages, a potential of disturbing the previously written data in adjacent pages of that same block that can occur from such a writing operation is eliminated. Also, a performance penalty of the additional program operation is avoided.

[0012] A further operational feature, which may be used in conjunction with the above summarized techniques, keeps track of the logical offset of individual pages of data within the individual memory cell blocks, so that the updated data need not be stored with the same physical page offset as the superceded data. This allows more efficient use of the pages of new blocks, and even allows the updated data to be stored in any erased pages of the same block as the superceded data.

[0013] Two or more blocks positioned in separate units of the memory array (also termed "sub-arrays") may be grouped together for programming and reading together as part of a single operation. Such a multiple block group is referenced herein as a "metablock." Its component blocks may be either all located on a single memory integrated circuit chip, or, in systems using more than one such chip, located on two or more different chips. When data in fewer than all of the pages of one of these blocks is updated, the use of another block in that same unit is normally required. Indeed, the techniques described above, may be employed separately with each block of the metablock. Therefore, when data within pages of more than one block of the metablock are updated, pages within more than one additional block are required to be used. If there are four blocks of four different memory units that form the metablock, for example, there is some probability that up to an additional four blocks, one in each of the units, will be used to store updated pages of the original blocks. One update block is potentially required in each unit for each block of the original metablock. Updated data from pages of more than one of the blocks in the metablock can be stored in pages of a common block in only one of the units. This significantly reduces the number of unused erased blocks that are needed to store updated data, thereby making more efficient use of the available memory cell blocks to store data. This technique is particularly useful when the memory system frequently updates single pages from a metablock.

[0014] Aspects, features and advantages of the present invention are included in the following description of exemplary embodiments, which description should be read in conjunction with the accompanying drawings, wherein:

Figure 1 is a block diagram of a typical prior art flash EEPROM memory array with memory control logic, data and address registers;

Figure 2 illustrates an architecture utilizing memories of Figure 1 with a system controller;

Figure 3 is a timing diagram showing a typical copy operation of the memory system of Figure 2;

Figure 4 illustrates an existing process of updating data in less than all of the pages of a multi-paged block;

Figures 5A and 5B are tables of corresponding logical and physical block addresses for each of the original and new blocks of Figure 4, respectively;

Figure 6 illustrates another existing process of updating data in less than all of the pages of a multi-paged block;

Figures 7A and 7B are tables of corresponding logical and physical page addresses for the original and new blocks of Figure 6, respectively;

Figure 8 illustrates an example of an improved process of updating data in less than all of the pages of a multi-

paged block;

Figure 9 is a table of corresponding logical and physical page numbers for the new block of Figure 8;

Figure 10 provides an example of a layout of the data in a page shown in Figure 8;

Figure 11 illustrates a further development of the example of Figure 8;

5 Figure 12 is a table of corresponding logical and physical page numbers for the new block of Figure 11;

Figure 13 illustrates one way to read the updated data in the blocks of Figure 11;

Figure 14 is a flow diagram of a process of programming data into a memory system organized as illustrated in Figures 8 and 9; Figure 15 illustrates an existing multi-unit memory with blocks from the individual units being linked together into a metablock and

10 Figure 16 illustrates an improved method of updating data of a metablock in the multi-unit memory of Figure 12 when the amount of updated data is much less than the data storage capacity of the metablock.

DESCRIPTION OF EXISTING LARGE BLOCK MANAGEMENT TECHNIQUES

15 **[0015]** Figure 1 shows a typical flash memory device internal architecture. The primary features include an input/output (I/O) bus 411 and control signals 412 to interface to an external controller, a memory control circuit 450 to control internal memory operations with registers for command, address and status signals. One or more arrays 400 of flash EEPROM cells are included, each array having its own row decoder (XDEC) 401 and column decoder (YDEC) 402, a group of
20 usually include one or more conductive floating gates as storage elements but other long term electron charge storage elements may be used instead. The memory cell array may be operated with two levels of charge defined for each storage element to therefore store one bit of data with each element. Alternatively, more than two storage states may be defined for each storage element, in which case more than one bit of data is stored in each element.

25 **[0016]** If desired, a plurality of arrays 400, together with related X decoders, Y decoders, program/verified circuitry, data registers, and the like are provided, for example as taught by U.S. Patent 5,890,192, issued March 30, 1999, and assigned to Sandisk Corporation, the assignee of this application, which is hereby incorporated by this reference. Related memory system features are described in co-pending patent application serial no. 09/505,555, filed February 17, 2000 by Kevin Conley et al., which application is expressly incorporated herein by this reference.

30 **[0017]** The external interface I/O bus 411 and control signals 412 can include the following:

CS - Chip Select.	Used to activate flash memory interface.
RS - Read Strobe.	Used to indicate the I/O bus is being used to transfer data from the memory array.
WS - Write Strobe.	Used to indicate the I/O bus is being used to transfer data to the memory array.
AS - Address Strobe.	Indicates that the I/O bus is being used to transfer address information.
35 AD[7:0] - Address/Data Bus	This I/O bus is used to transfer data between controller and the flash memory command, address and data registers of the memory control 450.

40 **[0018]** This interface is given only as an example as other signal configurations can be used to give the same functionality. Figure 1 shows only one flash memory array 400 with its related components, but a multiplicity of such arrays can exist on a single flash memory chip that share a common interface and memory control circuitry but have separate XDEC, YDEC, SA/PROG and DATA REG circuitry in order to allow parallel read and program operations.

45 **[0019]** Data is transferred from the memory array through the data register 404 to an external controller via the data registers' coupling to the I/O bus AD[7:0] 411. The data register 404 is also coupled the sense amplifier/programming circuit 454. The number of elements of the data register coupled to each sense amplifier/programming circuit element may depend on the number of bits stored in each storage element of the memory cells, flash EEPROM cells each containing one or more floating gates as the storage elements. Each storage element may store a plurality of bits, such as 2 or 4, if the memory cells are operated in a multi-state mode. Alternatively, the memory cells may be operated in a binary mode to store one bit of data per storage element.

50 **[0020]** The row decoder 401 decodes row addresses for the array 400 in order to select the physical page to be accessed. The row decoder 401 receives row addresses via internal row address lines 419 from the memory control logic 450. A column decoder 402 receives column addresses via internal column address lines 429 from the memory control logic 450.

55 **[0021]** Figure 2 shows an architecture of a typical non-volatile data storage system, in this case employing flash memory cells as the storage media. In one form, this system is encapsulated within a removable card having an electrical connector extending along one side to provide the host interface when inserted into a receptacle of a host. Alternatively, the system of Figure 2 may be embedded into a host system in the form of a permanently installed embedded circuit or otherwise. The system utilizes a single controller 301 that performs high level host and memory control functions. The

flash memory media is composed of one or more flash memory devices, each such device often formed on its own integrated circuit chip. The system controller and the flash memory are connected by a bus 302 that allows the controller 301 to load command, address, and transfer data to and from the flash memory array. The controller 301 interfaces with a host system (not shown) with which user data is transferred to and from the flash memory array. In the case where the system of Figure 2 is included in a card, the host interface includes a mating plug and socket assembly (not shown) on the card and host equipment.

[0022] The controller 301 receives a command from the host to read or write one or more sectors of user data starting at a particular logical address. This address may or may not align with a boundary of a physical block of memory cells.

[0023] In some prior art systems having large capacity memory cell blocks that are divided into multiple pages, as discussed above, the data from a block that is not being updated needs to be copied from the original block to a new block that also contains the new, updated data being written by the host. This technique is illustrated in Figure 4, wherein two of a large number of blocks of memory are included. One block 11 (PBN0) is illustrated to be divided into 8 pages for storing one sector of user data in each of its pages. Overhead data fields contained within each page include a field 13 containing the LBN of the block 11. The order of the logical pages within a logical block is fixed with respect to the corresponding physical pages within a physical block. A second similarly configured block 15 (PBN1) is selected from an inventory of unused, erased blocks. Data within pages 3-5 of the original block 11 are being updated by three pages of new data 17. The new data is written into the corresponding pages 3-5 of the new block 15, and user data from pages 0-2, 6 and 7 of the block 11 are copied into corresponding pages of the new block 15. All pages of the new block 15 are preferably programmed in a single sequence of programming operations. After the block 15 is programmed, the original block 11 can be erased and placed in inventory for later use. The copying of data between the blocks 11 and 15, which involves reading the data from one or more pages in the original block and subsequently programming the same data to pages in a newly assigned block, greatly reduces the write performance and usable lifetime of the storage system.

[0024] With reference to Figures 5A and 5B, partial tables show mapping of the logical blocks into the original and new physical blocks 11 and 15 before (Figure 5A) and after (Figure 5B) the updating of data described with respect to Figure 4. Before the data update, the original block 11, in this example, stores pages 0-7 of LBN0 into corresponding pages 0-7 of PBN0. After the data update, the new block 15 stores pages 0-7 of LBN0 in corresponding pages 0-7 of PBN1. Receipt of a request to read data from LBN0 is then directed to the physical block 15 instead of the physical block 11. In a typical controller operation, a table in the form of that shown in Figures 5A and 5B is built from the LBN field 13 read from a physical page and knowledge of the PBN that is addressed when reading the data field 13. The table is usually stored in a volatile memory of the controller for ease of access, although only a portion of a complete table for the entire system is typically stored at any one time. A portion of the table is usually formed immediately in advance of a read or programming operation that involves the blocks included in the table portion.

[0025] In other prior art systems, flags are recorded with the user data in pages and are used to indicate that pages of data in the original block that are being superceded by the newly written data are invalid. Only the new data is written to the newly assigned block. Thus the data in pages of the block not involved in the write operation but contained in the same physical block as the superceded data need not be copied into the new block. This operation is illustrated in Figure 6, where pages 3-5 of data within an original block 21 (PBN0) are again being updated. Updated pages 3-5 of data 23 are written into corresponding pages of a new block 25. As part of the same operation, an old/new flag 27 is written in each of the pages 3-5 to indicate the data of those pages is old, while the flag 27 for the remaining pages 0-2, 6 and 7 remains set at "new". Similarly, the new PBN1 is written into another overhead data field of each of the pages 3-5 in the block 21 to indicate where the updated data are located. The LBN and page are stored in a field 31 within each of the physical pages.

[0026] Figures 7A and 7B are tables of the correspondence between the data LBN/page and the PBN/page before (Figure 7A) and after (Figure 7B) the data update is complete. The unchanged pages 0-2, 6 and 7 of the LBN remain mapped into PBN0 while the updated pages 3-5 are shown to reside in PBN1. The table of Figure 7B is built by the memory controller by reading the overhead data fields 27, 29 and 31 of the pages within the block PBN0 after the data update. Since the flag 27 is set to "old" in each of pages 3-5 of the original block PBN0, that block will no longer appear in the table for those pages. Rather, the new block number PBN1 appears instead, having been read from the overhead fields 29' of the updated pages. When data are being read from LBN0, the user data stored in the pages listed in the right column of Figure 7B are read and then assembled in the order shown for transfer to the host.

[0027] Various flags are typically located in the same physical page as the other associated overhead data, such as the LBN and an ECC. Thus, to program the old/new flags 27, and others, in pages where the data has been superceded requires that a page support multiple programming cycles. That is, the memory array must have the capability that its pages can be programmed in at least at least two stages between erasures. Furthermore, the block must support the ability to program a page when other pages in the block with higher offsets or addresses have been already programmed. A limitation of some flash memories however prevents the usage of such flags by specifying that the pages in a block can only be programmed in a physically sequential manner. Furthermore, the pages support a finite number of program cycles and in some cases additional programming of programmed pages is not permitted.

[0028] What is needed is a mechanism by which data that partially supercedes data stored in an existing block can be written without either copying unchanged data from the existing block or programming flags to pages that have been previously programmed.

5 **DESCRIPTION OF EXEMPLARY EMBODIMENTS OF THE INVENTION**

[0029] There are many different types of flash EEPROM, each of which presents its own limitations that must be worked around to operate a high performance memory system formed on a small amount of integrated circuit area. Some do not provide for writing any data into a page that has already been programmed, so updating flags in a page that contains superceded data, as described above, is not possible. Others allow such flags to be written but doing so in pages whose data is being superceded can disturb data in other pages of the same block that remain current.

10 **[0030]** An example memory system where this has been found to be a problem is a NAND type, where a column of memory cells is formed as a series circuit string between a bit line and a common potential. Each word line extends across a row of memory cells formed of one cell in each such string. Such a memory is particularly susceptible to such memory state disturbs when being operated in a multi-state mode to store more than one bit of data in each such cell. Such operation divides an available window of a memory cell transistor threshold voltage range into narrow non-overlapping voltage level ranges, each range becoming narrower as the number of levels, and thus the number of bits being stored in each cell, are increased. For example, if four threshold ranges are used, two bits of data are stored in each cell's storage element. And since each of the four threshold voltage ranges is necessarily small, the chance of the state of a cell being disturbed by programming other cells in the same block is increased with multi-state operation. In this case, the writing of the old/new or other flags, as described with respect to Figures 6, 7A and 7B, cannot be tolerated.

20 **[0031]** A common feature of each of the existing memory management techniques described above with respect to Figures 4-7B is that a logical block number (LBN) and page offset is mapped within the system to at most two physical block numbers (PBNs). One block is the original block and the other contains the updated page data. Data are written to the page location in the block corresponding to the low order bits of its logical address (LBA). This mapping is typical in various types of memory systems. In the techniques described below, pages containing updated data are also assigned the same LBN and page offsets as the pages whose data has been superceded. But rather than tagging the pages containing original data as being superceded, the memory controller distinguishes the pages containing the superceded data from those containing the new, updated version either (1) by keeping track of the order in which the pages having the same logical addresses were written, such as by use of a counter, and/or (2) from the physical page addresses wherein, when pages are written in order within blocks from the lowest page address to the highest, the higher physical address contains the most recent copy of the data. When the data is accessed for reading, therefore, those in the most current pages are used in cases where there are pages containing superceded data that have the same logical addresses, while the superceded data are ignored.

35 **[0032]** A first specific implementation of this technique is described with respect to Figures 8 and 9. The situation is the same in this example as that in the prior art techniques described with respect to Figures 4-7B, namely the partial re-write of data within a block 35, although each block is now shown to contain 16 pages. New data 37 for each of the pages 3-5 of the block 35 (PBN 35) is written into three pages of a new block 39 (PBN1) that has previously been erased, similar to that described previously. A LBN and page offset overhead data field 41 written into the pages of PBN1 that contain the updated data is the same as that in the pages of the superceded data in the initial block PBN0. The table of Figure 9, formed from the data within the fields 41 and 41', shows this. The logical LBN and page offsets, in the first column, are mapped into both the first physical block (PBN0), in the second column, and, for the pages that have been updated, also into the second physical block (PBN1) in the third column. The LBN and logical page offsets 41' written into each of the three pages of updated data within the new block PBN1 are the same as those 41 written into each of a corresponding logical page of the original block PBN0.

45 **[0033]** In order to determine which of two pages having the same LBN and page offset contains the updated data, each page contains another overhead field 43 that provides an indication of its time of programming, at least relative to the time that other pages with the same logical address are programmed. This allows the controller to determine, when reading the data from the memory, the relative ages of the pages of data that are assigned the same logical address.

50 **[0034]** There are several ways in which the field 43, which contains a form of time stamp, may be written. The most straight forward way is to record in that field, when the data of its associated page is programmed, the output of a real-time clock in the system. Later programmed pages with the same logical address then have a later time recorded in the field 43. But when such a real-time clock is not available in the system, other techniques can be used. One specific technique is to store the output of a modulo-N counter as the value of the field 43. The range of the counter should be one more than the number of pages that are contemplated to be stored with the same logical page number. When updating the data of a particular page in the original block PBN0, for example; the controller first reads the count stored in the field 43 of the page whose data are being updated, increments the count by some amount, such as one, and then writes that incremented count in the new block PBN1 as the field 43'. The counter, upon reaching a count of N+1, rolls

EP 1 352 394 B1

over to 0. Since the number of blocks with the same LBN is less than N, there is always a point of discontinuity in the values of stored counts. It is easy then to handle the rollover with normalized to the point of discontinuity.

[0035] The controller, when called upon to read the data, easily distinguishes between the new and superceded pages' data by comparing the counts in the fields 43 and 43' of pages having the same LBA and page offset. In response to a need to read the most recent version of a data file, data from the identified new pages are then assembled, along with original pages that have not been updated, into the most recent version of the data file.

[0036] It will be noted that, in the example of Figure 8, the new data pages 37 are stored in the first three pages 0-2 of the new block PBN1, rather than in the same pages 3-5 which they replace in the original block PBN0. By keeping track of the individual logical page numbers, the updated data need not necessarily be stored in the same page offset of the new block as that of the old block where superceded data is contained. Page(s) of updated data can also be written to erased pages of the same block as the page of data being superceded.

[0037] As a result, there is no constraint presented by the techniques being described that limit which physical page new data can be written into. But the memory system in which these techniques are implemented may present some constraints. For example, one NAND system requires that the pages within the blocks be programmed in sequential order. That means that programming of the middle pages 3-5, as done in the new block 25 (Figure 6), wastes the pages 0-2, which cannot later be programmed. By storing the new data 37 in the first available pages of the new block 39 (Figure 8) in such a restrictive system, the remaining pages 3-7 are available for later use to store other data. Indeed, if the block 39 had other data stored in its pages 0-4 at the time the three pages of new data 37 were being stored, the new data could be stored in the remaining unused pages 5-7. This makes maximum use of the available storage capacity for such a system.

[0038] An example of the structure of data stored in an individual page of the blocks of Figure 8 is shown in Figure 10. The largest part is user data 45. An error correction code (ECC) 47 calculated from the user data is also stored in the page. Overhead data 49, including the LBN and page tag 41 (logical page offset), the time stamp 43 and an ECC 51 calculated from the overhead data are also stored in the page. By having an ECC 50 covering the overhead data that is separate from the user data ECC 47, the overhead 49 may be read separately from the user data and evaluated as valid without the need to transfer all of the data stored in the page. Alternatively, however, where the separate reading of the overhead data 49 is not a frequent event, all of the data in the page may be covered by a single ECC in order to reduce the total number of bits of ECC in a page.

[0039] A second specific implementation of the inventive technique can also be described with respect to Figure 8. In this example, the time stamp is used only to determine the relative age of the data stored in blocks, while the most recent pages among those that carry the same LBN and page number are determined by their relative physical locations. The time stamp 43 then does not need to be stored as part of each page. Rather, a single time stamp can be recorded for each block, either as part of the block or elsewhere within the non-volatile memory, and is updated each time a page of data is written into the block. Data is then read from pages in an order of descending physical address, starting from the last page of the most recently updated block containing data pages having the same LBN.

[0040] In Figure 8, for example, the pages are first read in the new block PBN1 from the last (page 15) to the first (page 0), followed by reading the pages of the original block PBN0 in the same reverse order. Once logical pages 3, 4 and 5 have been read from the new block PBN1, the superceded data in those pages of the original block PBN0 that are identified by the same logical page numbers can be skipped during the reading process. Specifically, physical pages 3, 4 and 5 of the old block PBN0 are skipped during reading, in this example, once the controller determines that their LBN/pages 41 are the same as those of the pages already read from the new block PBN1. This process can increase the speed of reading and reduce the number of overhead bits 49 that need to be stored for each page. Further, when this reverse page reading technique is employed, the table of Figure 9 used by the controller during a reading operation can be simplified into the form of Figures 5A and 5B. Only an identity of those physical blocks containing data of a common logical block and the relative times that the physical blocks were programmed need to be known in order to carry out this efficient reading process.

[0041] Figure 11 illustrates an extension of the example of Figure 8 by including a second update to the data originally written in the block PBN0. New data 51 for logical pages 5, 6, 7 and 8 is written to the respective physical pages 3, 4, 5 and 6 of the new block PBN1, along with their LBN and page number. Note, in this example, that the data of logical page 5 is being updated for the second time. During a reading operation that begins from the last page of the new block PBN1, the most recently written logical pages 8, 7, 6 and 5 of the data of interest are first read in that order. Thereafter, it will be noted that the LBN/page overhead field in physical page 2 of PBN1 is the same as that read from the physical page 3, so the user data of page 2 is not read. The physical pages 1 and 0 are then read. Next, the pages of the original block PBN0 are read, beginning with physical page 15. After reading physical pages 15-9, the controller will note that the LBN/page fields of each of pages 8-3 match those of pages whose data has already been read, so the old data need not be read from those pages. The efficiency of the reading process is thus improved. Finally, the original data of physical pages 2-0 are read since that data was not updated.

[0042] It will be noted that this example of reading pages in a reverse order efficiently sorts out the new data pages

from the superceded data pages because data are written in physical page locations of an erased block in order from page 0 on. This technique is not limited to use with a memory system having such a specific programming constraint, however. So long as the order in which pages are programmed within a given block is known, the data from those pages may be read in the reverse order from which they were written. What is desired is that the most recently programmed pages having a common LBN with others that were earlier programmed be read first, and these are the most recently programmed pages. The most recent versions of updated pages are read first so that the superceded versions may easily be identified thereafter.

[0043] A table showing the correspondence between the logical data and physical page addresses for the example of Figure 11 is given in Figure 12. Although there have been two data updates, both are represented by the single column for the second block PBN1. The physical page noted in PBN1 for the logical page 5 is simply changed upon the second update to that page occurring. If the updating involves a third block, then another column is added for that other block. The table of Figure 12, constructed by reading the overhead data from each of the pages in blocks to which data of a common LBN has been written, can be used by the first implementation when the reverse page reading technique is not used. When the reverse page reading technique described above is used, the table of Figure 12 need be built only to identify a correspondence between an LBN and all PBNs containing data of that LBN.

[0044] An efficient way to organize pages of data being read from a physical block, where one or more of the pages has been updated, is illustrated by Figure 13. Enough space is provided in a volatile memory of the controller to buffer at least several pages of data at a time, and preferably a full block of data. That is what is shown in Figure 13. Sixteen pages of data, equal to the amount stored in a non-volatile memory block, are stored in the controller memory. Since the pages are most commonly read out of order, each page of data is stored in its proper position with respect to the other pages. For example, in the reverse page read operation of Figure 11, logical page 8 if the first to be read, so it is stored in position 8 of the controller memory, as indicated by the "1" in a circle. The next is logical page 7, and so forth, until all pages of data desired by the host are read and stored in the controller memory. The entire set of page data is then transferred to the host without having to manipulate the order of the data in the buffer memory. The pages of data have already be organized by writing them to the proper location in the controller memory.

[0045] A method of programming a non-volatile memory system that utilizes the techniques described with respect to Figures 8 and 9 is illustrated in the flow chart of Figure 14. Data for pages of an existing file to be updated are received from a host system, as indicated by the block 52. It is first determined by a step 53 whether the number of pages of updated data to be stored is equal to or greater than the storage capacity of a block of the system, 16 pages being shown as the block capacity, for simplicity, in the above described example. If so, one or more unused, erased blocks are addressed, in a step 55, and the new data pages are written to the addressed block(s), in a step 57. Typically, the updating of one block or more of data will result in one or more blocks storing the data that have been superceded by the new data. If so, as indicated by a step 59, those blocks with superceded data are identified for erasure. For the purpose of increasing performance, it is preferable that erase operations occur in the background, or when host requested programming or reading operations are not taking place. After being erased, the blocks are returned to the inventory of unused, erased blocks for further use. Alternatively, erasure of the blocks can be deferred until they are needed for programming operations.

[0046] If, on the other hand, in the step 53, it is determined that there are fewer pages of new data than will utilize the full storage capacity of a block, a next step 61 determines whether there are enough unused pages in a block having some pages programmed with other data. If so, such a block is addressed, in a step 63. If not, a totally unused, erased block is addressed, in a step 65. In either case, in a step 67, the new data are programmed into unused pages of the addressed block. As part of this programming process, the LBN and page offset is written into the fields 41, and the time stamp into the fields 43 of each of the pages (Figure 8) of the updated data, in the manner described above.

[0047] A desirable feature of the programming process is to make available for future programming any blocks that store only superceded data. So the question is asked, in a step 69, whether the data updating process has resulted in an entire block remaining with only superceded data. If so, such a block is queued for erasure, in a step 71, and the process is then completed. If not, the step 71 is omitted and the data update is finished.

METABLOCK OPERATION

[0048] In order to improve performance by reducing programming time, a goal is to program as many cells in parallel as can reasonably be done without incurring other penalties. One implementation divides the memory array into largely independent sub-arrays or units, such as multiple units 80-83 of Figure 15, each unit in turn being divided into a large number of blocks, as shown. Pages of data are then programmed at the same time into more than one of the units. Another configuration further combines one or more of these units from multiple memory chips. These multiple chips may be connected to a single bus (as shown in Figure 2) or multiple independent busses for higher data throughput. An extension of this is to link blocks from different units for programming, reading and erasing together, an example being shown in Figure 15. Blocks 85-88 from respective ones of the units 80-83 can be operated together as a metablock, for

example. As with the memory embodiments described above, each block, the smallest erasable group of the memory array, is typically divided into multiple pages, a page containing the smallest number of cells that are programmable together within the block. Therefore, a programming operation of the metablock shown in Figure 15 will usually include the simultaneously programming of data into at least one page of each of the blocks 85-88 forming the metablock, which is repeated until the metablock is full or the incoming data has all been programmed. Other metablocks are formed of different blocks from the array units, one block from each unit.

[0049] In the course of operating such a memory, as with others, pages of data less than an entire block often need to be updated. This can be done for individual blocks of a metablock in the same manner as described above with respect to either of Figures 4 or 6, but preferably by use of the improved technique described with respect to Figure 8. When any of these three techniques are used to update data of one block of the metablock, an additional block of memory within the same unit is also used. Further, a data update may require writing new data for one or more pages of two or more of the blocks of a metablock. This can then require use of up to four additional blocks 90-93, one in each of the four units, to update a data file stored in the metablock, even though the data in only a few pages is being updated.

[0050] In order to reduce the number of blocks required for such partial block updates, according to another aspect of the present invention, updates to pages of data within any of the blocks of the illustrated metablock are made, as illustrated by Figure 16, to a single additional block 90 in the memory unit 80, so long as unused pages in the block 80 remain. If, for example, data in three pages of the block 86 and two pages of the block 88 are being updated at one time, all five pages of the new data are written into the block 90. This can save the use of one block of memory, thereby to effectively increase the number of available erased blocks by one block. This helps avoid, or at least postpone, the time when an inventory of erased blocks becomes exhausted. If one or more pages from each of the four blocks 85-88 are being updated, all of the new data pages are programmed in the single block 90, thereby avoiding tying up an additional three blocks of memory to make the update. If the number of pages of new data exceed the capacity of an unused block, pages that the block 90 cannot accept are written to another unused block which may be in the same unit 80 or one of the other units 81-83.

Claims

1. A method of substituting new data for superseded data in a non-volatile memory system having an array of non-volatile memory storage elements (400) organised in blocks (35,39) with the blocks organised in pages, each block containing the smallest group of storage elements that is erasable; which method comprises writing new data as updated pages into pages of a said block, said new data being less than all of the original pages of another said block; addressing both the original pages and the updated pages with the same logical address; and reading and distinguishing the updated pages from the original pages,

CHARACTERIZED IN THAT

said reading is with reference to the relative time of writing into pages of said blocks by reading the pages from said block and then from said other block both in a reverse order from that in which the pages have been written and ignores any pages having the same logical address as pages it has already read.

2. A method according to Claim 1 wherein the pages within the individual blocks of the memory system are programmed in a designated order.

3. A method according to Claim 1 or Claim 2 including the step of operating the individual memory storage elements (400) with more than two storage states, thereby storing more than one bit of data in each storage element, and wherein reading pages of data includes reading the more than two storage states from the individual memory storage elements.

4. A non-volatile memory system, comprising:

an array of non-volatile memory storage elements (400) organised in blocks (35,39), said blocks organised in pages, wherein a block contains the smallest group of storage elements that is erasable;

a programming mechanism that writes into pages of a said block an updated version of less than all of the original pages of another said block;

an address mechanism (401,402,450) that addresses both the original pages and the updated pages with the same logical address; and

a reading mechanism (404,450,454) that distinguishes the updated pages from the original pages,

CHARACTERIZED IN THAT

the reading mechanism reads with reference to the relative time of writing into pages of said blocks by reading the pages from said block and then from said other block in a reverse order from that in which the pages have been written and ignores any pages having the same logical address as pages it has already read.

- 5 5. A memory system according to Claim 4 wherein the memory storage elements include individual floating gates.
6. A memory system according to Claim 4 or Claim 5 formed within an enclosed card having electrical contacts for connecting with a host system.

10

Patentansprüche

- 15 1. Verfahren zum Ersetzen von Daten durch neue Daten in einem nichtflüchtigen Speichersystem mit einem Array aus nichtflüchtigen Speicherelementen (400), welches in Blöcken (35, 39) organisiert ist, wobei diese Blöcke in Seiten organisiert sind und jeder Block die kleinste Anzahl von löschbaren Speicher-Elementen enthält; umfassend Schreiben neuer Daten als aktualisierte Seiten in Seiten eines anderen Blocks, wobei diese neuen Daten weniger als alle der ursprünglichen Seiten dieses anderen Blocks umfassen; Adressierung sowohl der ursprünglichen Seiten als auch der aktualisierten Seiten mit derselben logischen Adresse; und
- 20 Lesen und Unterscheiden der aktualisierten Seiten von den ursprünglichen Seiten, **dadurch gekennzeichnet, dass** das Lesen, in Bezug auf die relative Zeit des Schreibens in Seiten dieser Blöcke, erfolgt durch Lesen der Seiten aus diesem Block und danach aus diesem anderen Block, beides in umgekehrter Reihenfolge als diejenige, in der die Seiten geschrieben wurden und alle Seiten ignoriert werden, die die gleiche logische Adresse haben wie Seiten, die bereits gelesen wurden.
- 25 2. Verfahren gemäß Anspruch 1, worin die Seiten innerhalb der einzelnen Blöcke des Speichersystems in einer bestimmten Reihenfolge programmiert werden.
- 30 3. Verfahren gemäß Anspruch 1 oder Anspruch 2, einschließlich den Schritt, die einzelnen Speicherelemente (400), die mehr als zwei Speicherzustände aufweisen, anzusteuern und dabei mehr als ein Bit Daten in jedem Speicherelement zu speichern, wobei Lesen von Seiten mit Daten das Lesen von mehr als zwei Speicherzuständen der einzelnen Speicherelemente umfasst.
- 35 4. Nichtflüchtiges Speichersystem, umfassend ein Array aus nichtflüchtigen Speicherelementen (400), welches in Blöcken (35, 39) organisiert ist, wobei diese Blöcke in Seiten organisiert sind und jeder Block die kleinste Anzahl von löschbaren Speicher-Elementen enthält; einen Programmier-Mechanismus, der in Seiten dieses Blocks eine aktualisierte Version mit weniger als alle der ursprünglichen Seiten dieses anderen Blocks schreibt; einen Adressierungs-Mechanismus (401, 402, 450), der sowohl die ursprünglichen Seiten als auch die aktualisierten Seiten mit derselben logischen Adresse adressiert; und einen Lese-Mechanismus (404, 450, 454), der zwischen den aktualisierten Seiten und den ursprünglichen Seiten unterscheidet, **dadurch gekennzeichnet, dass**
- 45 der Lese-Mechanismus, in Bezug auf die relative Zeit des Schreibens in Seiten dieser Blöcke, erfolgt durch Lesen der Seiten aus diesem Block und danach aus diesem anderen Block, beides in umgekehrter Reihenfolge als diejenige, in der die Seiten geschrieben wurden und alle Seiten ignoriert werden, die die gleiche logische Adresse haben wie Seiten, die bereits gelesen wurden.
- 50 5. Speichersystem gemäß Anspruch 4, in dem die Datenspeicherelemente einzelne Floating Gates umfassen.
6. Speichersystem gemäß Anspruch 4 oder Anspruch 5, das sich in einer geschlossenen Karte befindet, die elektrische Kontakte zum Verbinden mit einem Host-System aufweist.

55

Revendications

1. Procédé de substitution de données périmées par de nouvelles données dans un système de mémoire non volatile

EP 1 352 394 B1

ayant un réseau d'éléments de stockage en mémoire non volatile (400) organisés en blocs (35, 39) les blocs étant organisés en pages, chaque bloc contenant le plus petit groupe d'éléments de stockage qui est effaçable, ledit procédé comprenant les étapes consistant à :

- 5 écrire de nouvelles données comme pages mises à jour dans des pages dudit bloc, lesdites nouvelles données étant moins nombreuses que toutes les pages originales d'un autre dit bloc ; adresser à la fois les pages originales et les pages mises à jour avec la même adresse logique ; et lire et distinguer les pages mises à jour des pages originales,
caractérisé en ce que
- 10 ladite lecture s'effectue en référence au temps relatif d'écriture dans les pages desdits blocs en lisant les pages dudit bloc, puis dudit autre bloc dans un ordre inverse par rapport auquel les pages ont été écrites, et ignore toutes les pages ayant la même adresse logique que les pages ayant déjà été lues.
- 15 **2.** Procédé selon la revendication 1, dans lequel les pages situées dans les blocs individuels du système de mémoire sont programmées dans un ordre désigné.
- 3.** Procédé selon la revendication 1 ou la revendication 2, comprenant l'étape consistant à exploiter les éléments de stockage en mémoire individuels (400) avec plus de deux états de stockage, stockant de cette manière plus d'un bit de données dans chaque élément de stockage, et dans lequel la lecture de pages de données comprend la
- 20 lecture de plus que les deux états de stockage des éléments de stockage en mémoire individuels.
- 4.** Système de mémoire non volatile, comprenant :
- 25 un réseau d'éléments de stockage en mémoire non volatile (400) organisés en blocs (35, 39), lesdits blocs étant organisés en pages, dans lequel un bloc contient le plus petit groupe d'éléments de stockage qui est effaçable ; un mécanisme de programmation qui écrit, dans les pages d'un dit bloc, une version mise à jour moindre que toutes les pages originales d'un autre dit bloc ; un mécanisme d'adressage (401, 402, 450) qui adresse à la fois les pages originales et les pages mises à jour
- 30 avec la même adresse logique ; et un mécanisme de lecture (404, 450, 454) qui distingue les pages mises à jour des pages originales,
caractérisé en ce que le mécanisme de lecture lit en référence au temps relatif d'écriture dans les pages desdits blocs en lisant les pages dudit bloc, puis dudit autre bloc dans un ordre inverse par rapport auquel les pages ont été écrites, et ignore toutes les pages ayant la même adresse logique que les pages qu'il a déjà lues.
- 35 **5.** Système de mémoire selon la revendication 4, dans lequel les éléments de stockage en mémoire comprennent des grilles flottantes.
- 40 **6.** Système de mémoire selon la revendication 4 ou la revendication 5 formé à l'intérieur d'une carte encapsulée ayant des contacts électriques pour se connecter à un système hôte.
- 45
- 50
- 55

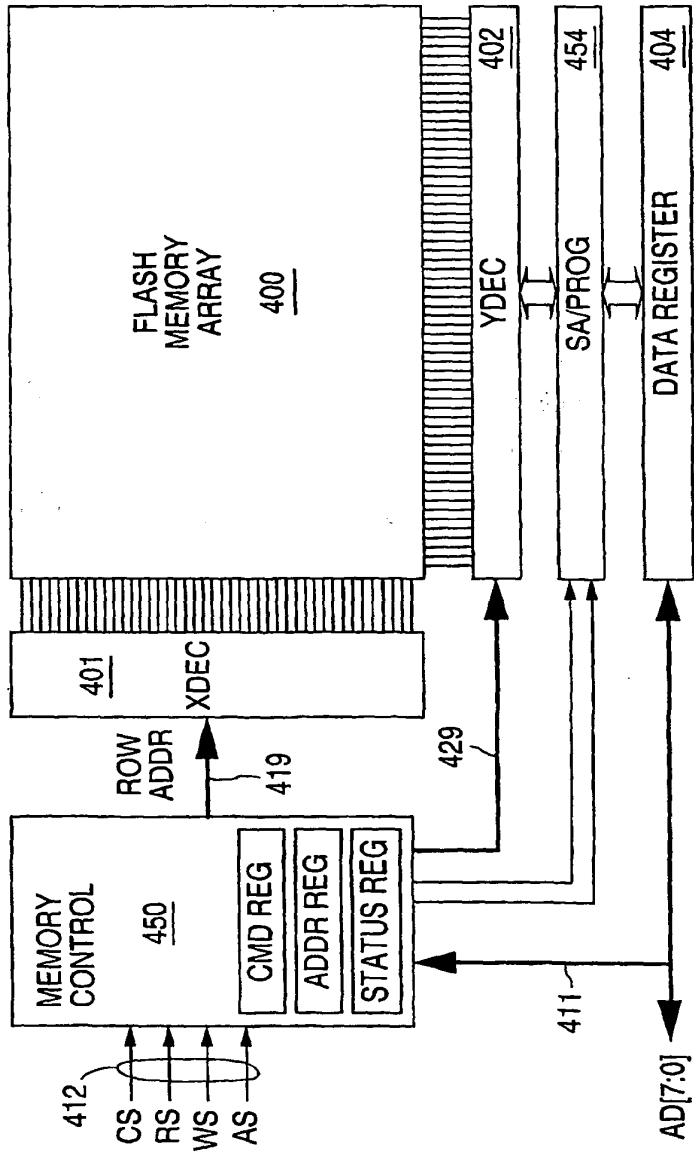


FIG. 1
(PRIOR ART)

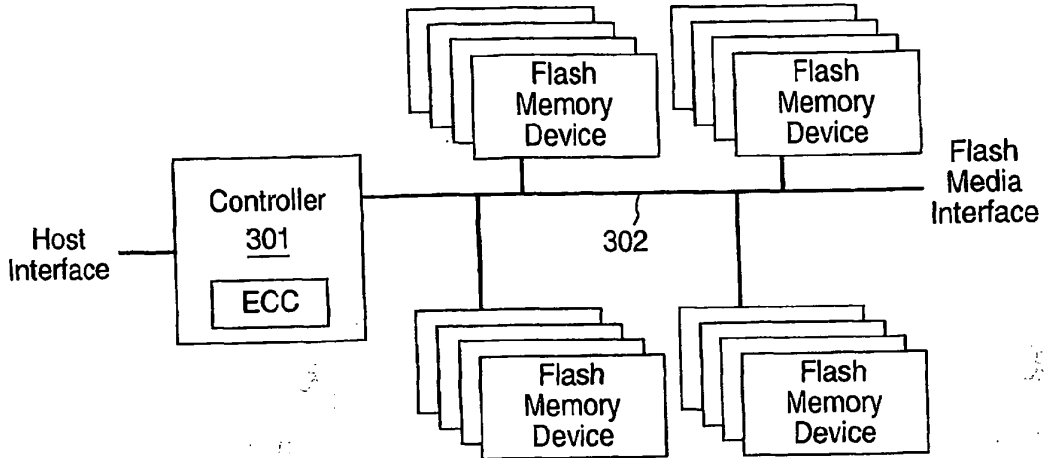


FIG. 2
(PRIOR ART)

LBN	PBN
0	0
⋮	⋮
⋮	⋮
⋮	⋮
⋮	⋮

Original Block 11

FIG. 5A

LBN	PBN
0	1
⋮	⋮
⋮	⋮
⋮	⋮
⋮	⋮

With New Block 15

FIG. 5B

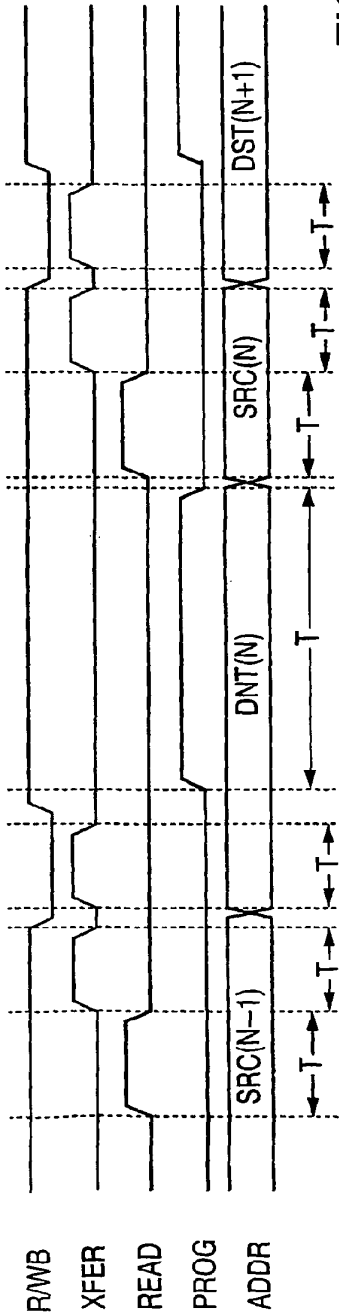


FIG. 3
(PRIOR ART)

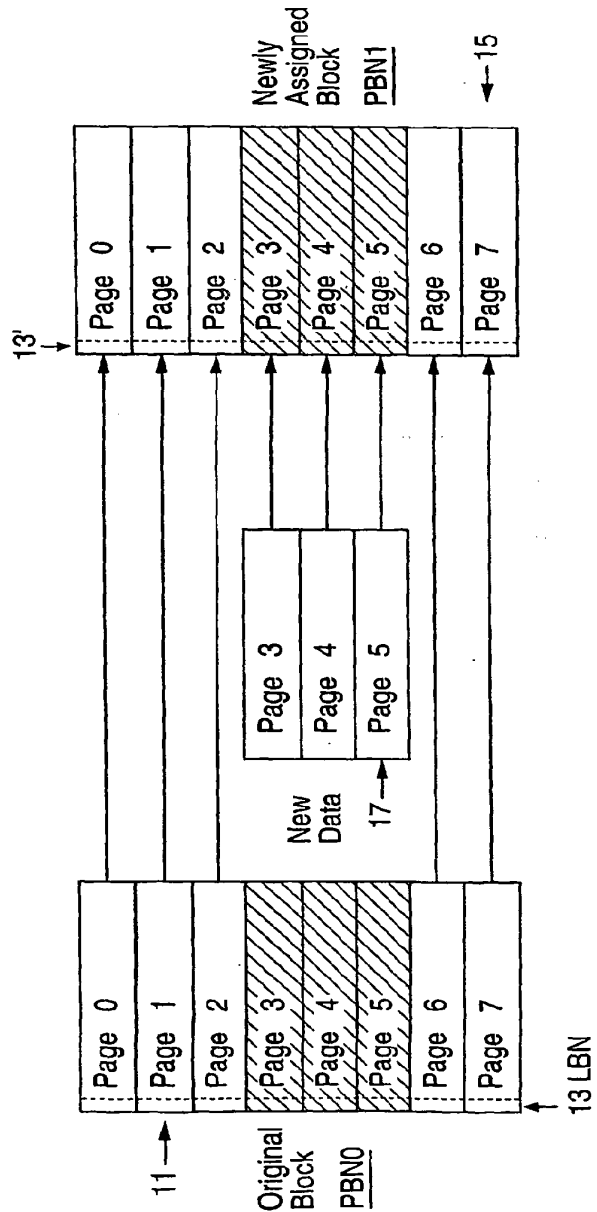


FIG. 4
(PRIOR ART)

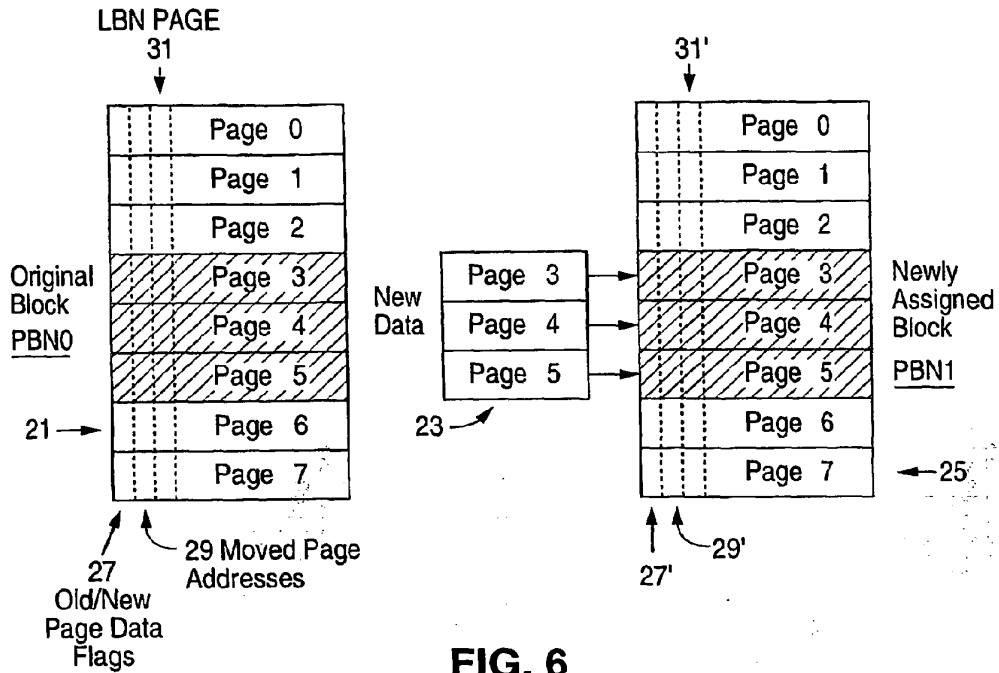


FIG. 6
(PRIOR ART)

LBN	Page	PBN	Page
0	0	0	0
0	1	0	1
0	2	0	2
0	3	0	3
0	4	0	4
0	5	0	5
0	6	0	6
0	7	0	7
:	:	:	:

Original Block

FIG. 7A

LBN	Page	PBN	Page
0	0	0	0
0	1	0	1
0	2	0	2
0	3	1	3
0	4	1	4
0	5	1	5
0	6	0	6
0	7	0	7
:	:	:	:

With New Block

FIG. 7B

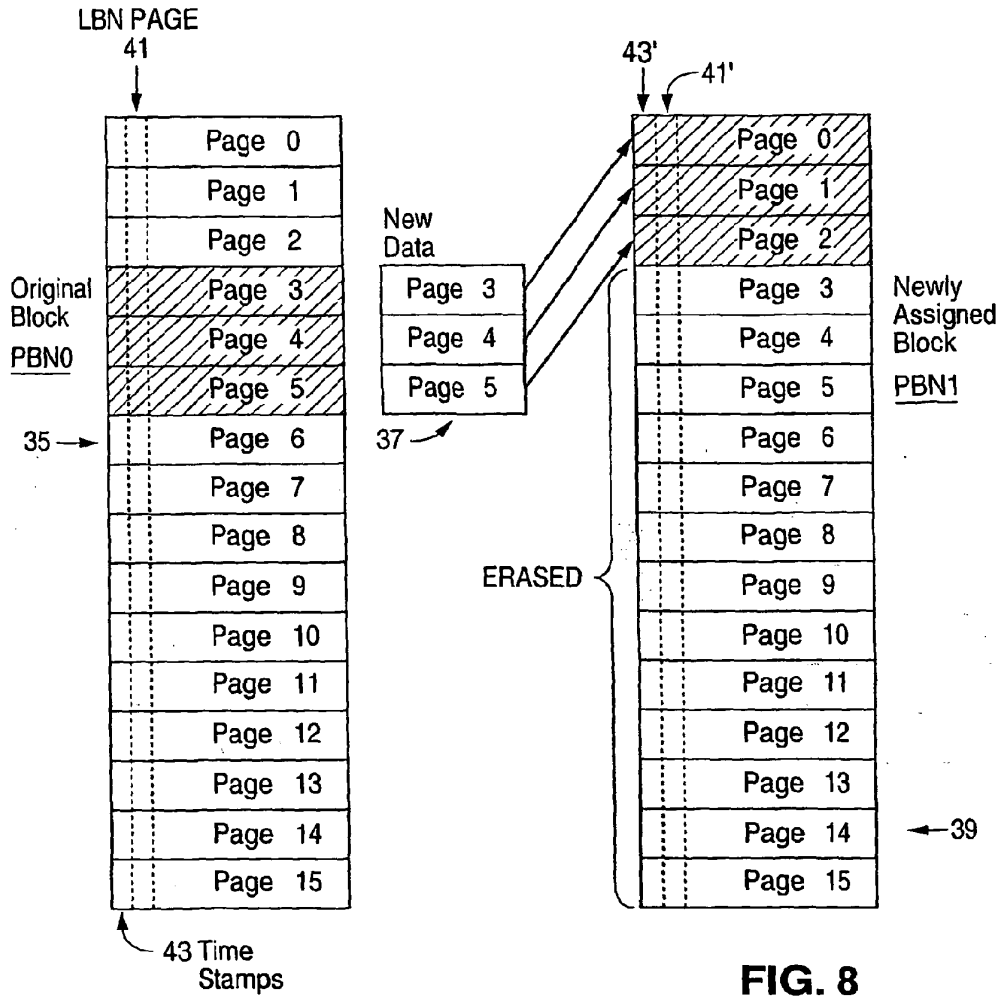


FIG. 8

LBN	Page	PBN0	Page	PBN1	Page
0	0	0	0		
0	1	0	1		
0	2	0	2		
0	3	0	3	1	0
0	4	0	4	1	1
0	5	0	5	1	2
0	6	0	6		
0	7	0	7		

FIG. 9

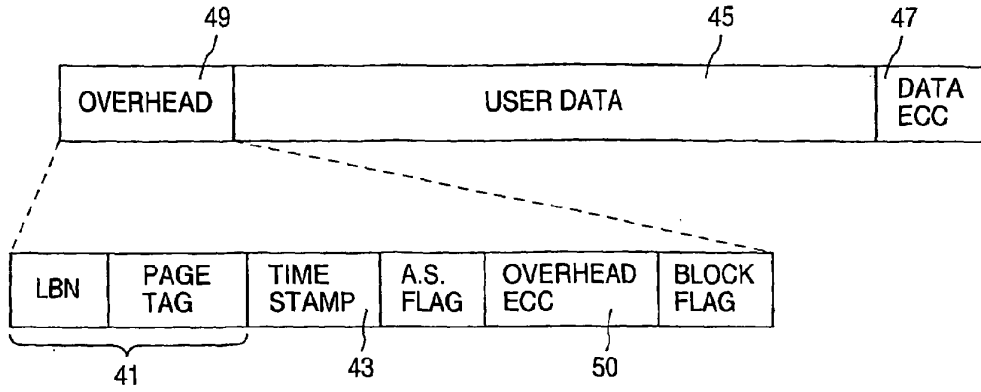


FIG. 10

LBN	Page	PBN0	Page	PBN1	Page
0	0	0	0		
0	1	0	1		
0	2	0	2		
0	3	0	3	1	0
0	4	0	4	1	1
0	5	0	5	1	3
0	6	0	6	1	4
0	7	0	7	1	5
0	8	0	8	1	6
0	9	0	9		
:	:	:	:		:

FIG. 12

CONTROLLER RAM

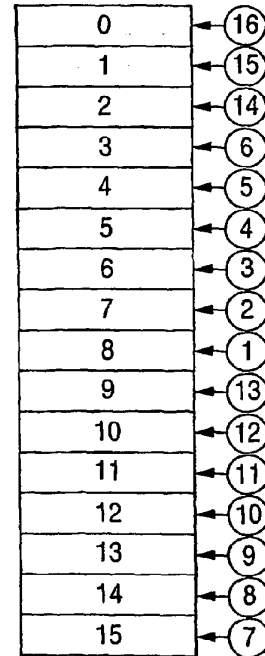


FIG. 13

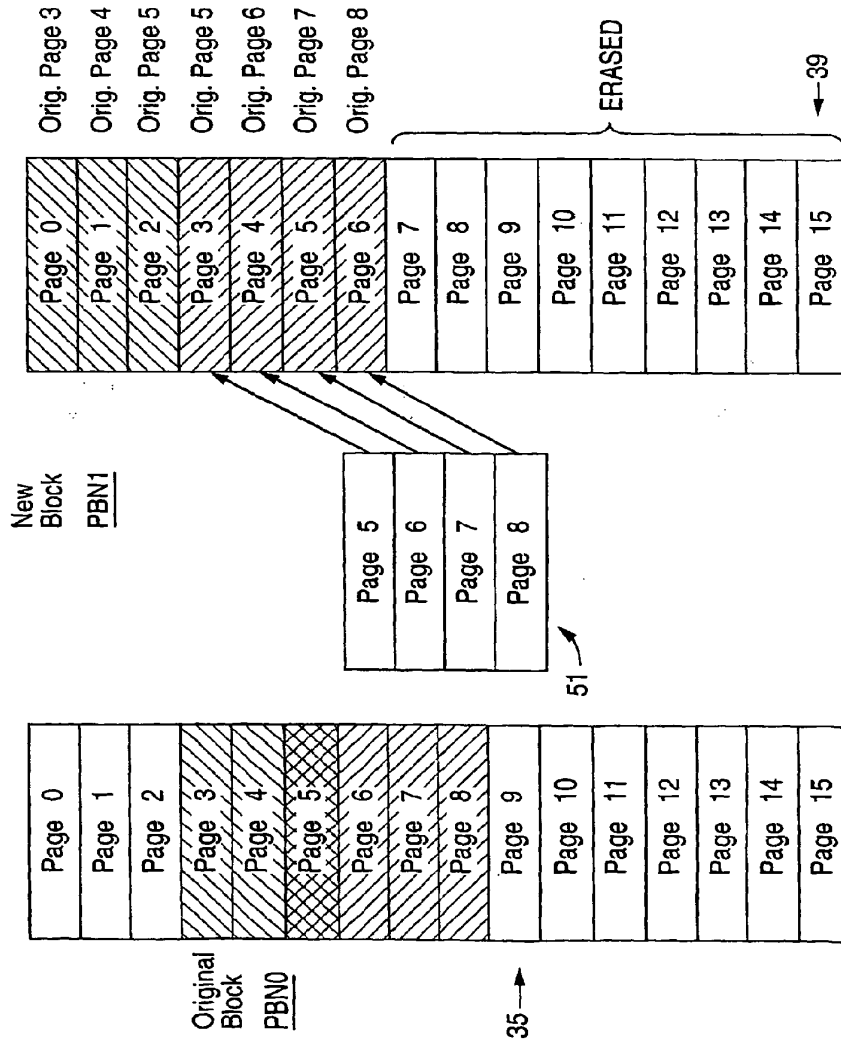


FIG. 11

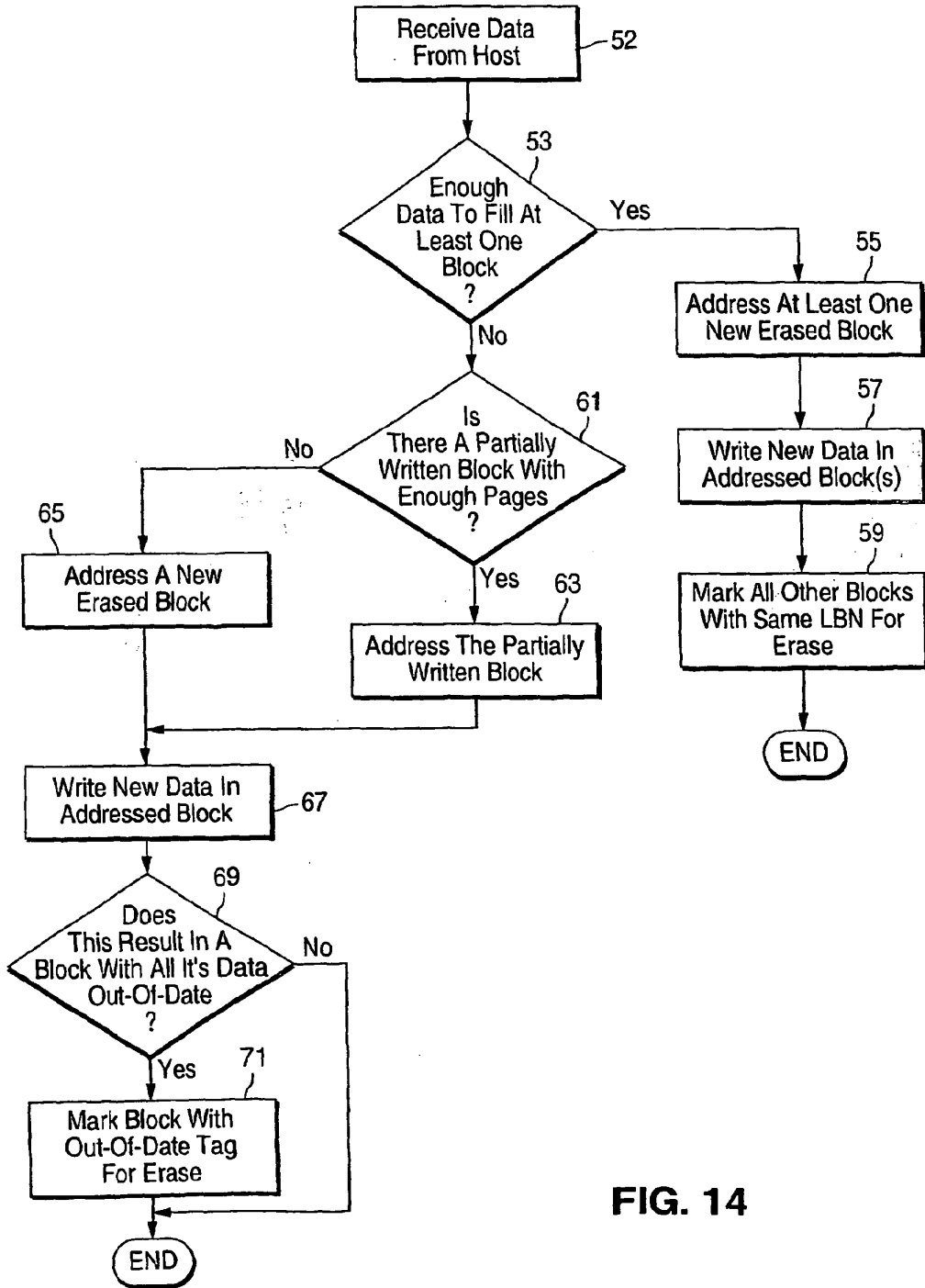


FIG. 14

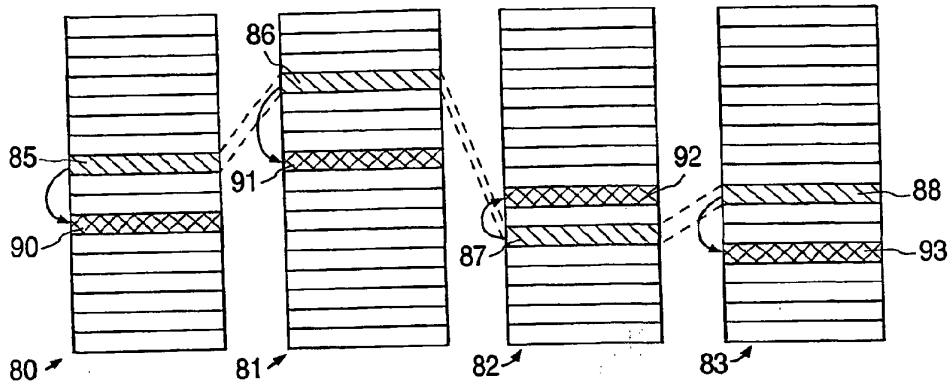


FIG. 15

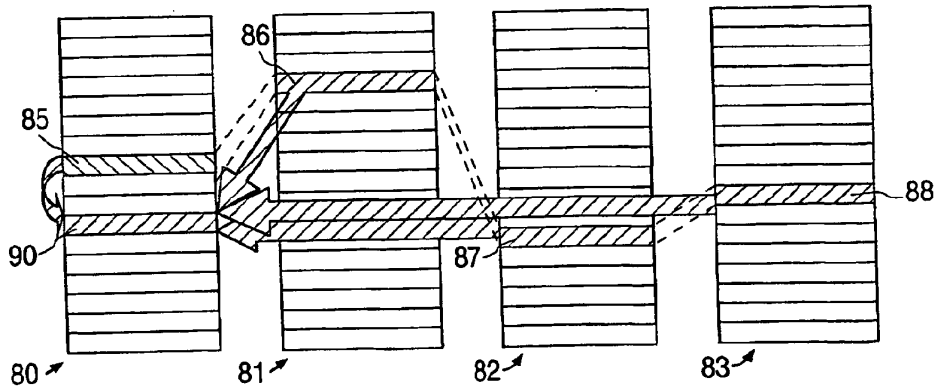


FIG. 16

SEMICONDUCTOR DISK DEVICE

Publication number: JP8212019
 Publication date: 1996-08-20
 Inventor: MIYAUCHI SHIGENORI
 Applicant: MITSUBISHI ELECTRIC CORP
 Classification:
 - international: G11C17/00; G06F3/06; G06F3/08; G06F12/00; G06F12/02; G06F12/16; G11C16/02; G11C17/00; G06F3/06; G06F3/08; G06F12/00; G06F12/02; G06F12/16; G11C16/02; (IPC1-7): G06F3/08; G11C16/06
 - European: G06F3/06E; G06F12/02D2E2
 Application number: JP19950014030 19950131
 Priority number(s): JP19950014030 19950131

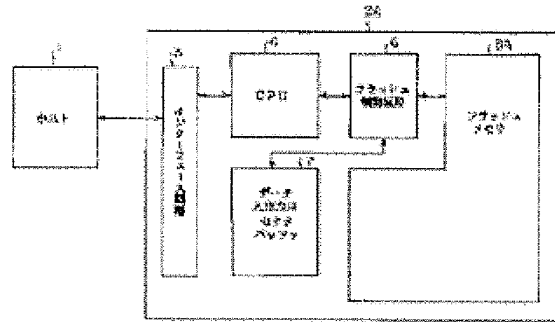
Also published as:

US 5627783 (A1)
 G B2297637 (A)

Report a data error here

Abstract of JP8212019

PURPOSE: To expand a data area by eliminating the necessity of an address conversion table for data management. CONSTITUTION: This semiconductor disk device is provided with a flash memory 8A including plural erasing blocks each of which is constituted of a logical erasing block number storing area, plural data storing areas for storing data, plural data state flag areas capable of storing a data state flag expressing whether data are stored in each data storing area or not in each data storing area, and plural updating data relative information storing areas for storing relative information expressing the address of an updated destination in each data storing area; a CPU 4 for converting a logical sector address into a logical erasing block number and its offset value, retrieving a corresponding erasing block and a corresponding data storing area in the flash memory 8, and when relative information does not exist in a corresponding updating data relative information storing area, reading out the contents of the corresponding data storing area.



Data supplied from the esp@cenet database - Worldwide

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-212019

(43)公開日 平成8年(1996)8月20日

(51)Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 3/08				
G 1 1 C 16/06			G 1 1 C 17/ 00	5 3 0 B

審査請求 未請求 請求項の数6 OL (全 15 頁)

(21)出願番号 特願平7-14030

(22)出願日 平成7年(1995)1月31日

(71)出願人 000006013
三菱電機株式会社
東京都千代田区丸の内二丁目2番3号

(72)発明者 宮内 成典
伊丹市瑞原4丁目1番地 三菱電機株式会
社北伊丹製作所内

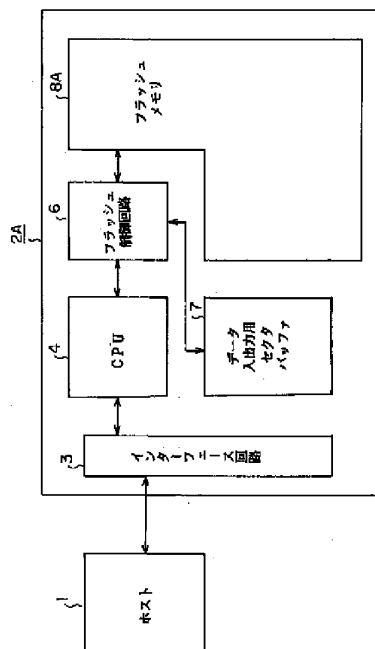
(74)代理人 弁理士 曾我 道照 (外6名)

(54)【発明の名称】 半導体ディスク装置

(57)【要約】

【構成】 論理消去ブロック番号格納領域と、データを格納する複数のデータ格納領域と、前記データ格納領域にデータが格納されているか否かを表すデータ状態フラグを格納する前記データ格納領域毎のデータ状態フラグ領域と、更新先の場所を表す連鎖情報を格納する前記データ格納領域毎の更新データ連鎖情報格納領域とから構成される消去ブロックを複数有するフラッシュメモリ8Aと、論理セクタアドレスを論理消去ブロック番号とそのオフセット値に変換し、この変換した論理消去ブロック番号とそのオフセット値に基づいて前記フラッシュメモリ8A上の該当消去ブロック及び該当データ格納領域を捜し出し、該当更新データ連鎖情報格納領域に連鎖情報が存在しなければ前記該当データ格納領域の内容を読み出すCPU4とを備える。

【効果】 データ管理用のアドレス変換テーブルが不要となり、その分データエリアを大きくできる。



【特許請求の範囲】

【請求項1】 論理消去ブロック番号を格納する論理消去ブロック番号格納領域と、データを格納する複数のデータ格納領域と、前記データ格納領域にデータが格納されているか否かを表すデータ状態フラグを格納する前記データ格納領域毎のデータ状態フラグ領域と、更新先の場所を表す連鎖情報を格納する前記データ格納領域毎の更新データ連鎖情報格納領域とから構成される消去ブロックを複数有するフラッシュメモリ、並びに前記論理消去ブロック番号、前記データ状態フラグ、及び前記連鎖情報に基づいて前記フラッシュメモリ上のデータを管理するCPUを備えたことを特徴とする半導体ディスク装置。

【請求項2】 前記CPUは、論理セクタアドレスを論理消去ブロック番号とそのオフセット値に変換し、この変換した論理消去ブロック番号とそのオフセット値に基づいて前記フラッシュメモリ上の該当消去ブロック及び該当データ格納領域を捜し出し、該当更新データ連鎖情報格納領域に連鎖情報が存在しなければ前記該当データ格納領域の内容を読み出し、前記該当更新データ連鎖情報格納領域に前記連鎖情報が存在すればその連鎖情報に基づいて該当データ格納領域の内容を読み出すことを特徴とする請求項1記載の半導体ディスク装置。

【請求項3】 前記CPUは、論理セクタアドレスを論理消去ブロック番号とそのオフセット値に変換し、この変換した論理消去ブロック番号とそのオフセット値に基づいて前記フラッシュメモリ上の該当消去ブロック及び該当データ格納領域を捜し出し、該当データ状態フラグが未使用を示すときは前記該当データ格納領域にデータを書き込み、前記該当データ状態フラグを使用中にし、前記該当データ状態フラグが使用中を示すときは空きのデータ格納領域を捜し出してそこにデータを書き込み、このデータ格納領域に対応するデータ状態フラグを使用中にするとともに、前の使用中のデータ格納領域に対応する更新データ連鎖情報格納領域に連鎖情報を書き込むことを特徴とする請求項1記載の半導体ディスク装置。

【請求項4】 論理消去ブロック番号を格納する論理消去ブロック番号格納領域と、データを格納する複数のデータ格納領域と、前記データ格納領域にデータが格納されているか否かを表すデータ状態フラグを格納する前記データ格納領域毎のデータ状態フラグ領域と、更新先の場所を表す連鎖情報を格納する前記データ格納領域毎の更新データ連鎖情報格納領域とから構成される消去ブロックを複数有するフラッシュメモリ、前記論理消去ブロック番号を物理消去ブロック番号へ変換するためのアドレス変換テーブル、並びに前記論理消去ブロック番号、前記データ状態フラグ、及び前記連鎖情報に基づいて前記フラッシュメモリ上のデータを管理するCPUを備えたことを特徴とする半導体ディスク装置。

【請求項5】 前記CPUは、論理セクタアドレスを論

理消去ブロック番号とそのオフセット値に変換し、前記アドレス変換テーブルに基づいて前記論理消去ブロック番号を物理消去ブロック番号へ変換し、この物理消去ブロック番号とそのオフセット値に基づいて前記フラッシュメモリ上の該当消去ブロック及び該当データ格納領域を捜し出し、該当更新データ連鎖情報格納領域に連鎖情報が存在しなければ前記該当データ格納領域の内容を読み出すことを特徴とする請求項4記載の半導体ディスク装置。

10 【請求項6】 前記CPUは、論理セクタアドレスを論理消去ブロック番号とそのオフセット値に変換し、前記アドレス変換テーブルに基づいて前記論理消去ブロック番号を物理消去ブロック番号へ変換し、この物理消去ブロック番号とそのオフセット値に基づいて前記フラッシュメモリ上の該当消去ブロック及び該当データ格納領域を捜し出し、該当データ状態フラグが未使用を示すときは前記該当データ格納領域にデータを書き込み、前記該当データ状態フラグを使用中にすることを特徴とする請求項4記載の半導体ディスク装置。

20 【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は、フラッシュメモリを記憶媒体として用いた半導体ディスクカード等の半導体ディスク装置に関するものである。

【0002】

【従来の技術】 今日、パーソナルコンピュータの分野において比較的大容量のデータを記憶させておく際には、ハードディスク装置などの磁気記憶媒体が用いられることが多い。これは、消費電力こそ大きいコストパフォーマンスが非常によいためである。

【0003】 一方、フラッシュメモリ等の半導体メモリを上記ハードディスク装置のように動作させる半導体ディスク装置が出現した。この半導体ディスク装置は上記ハードディスク装置と違いモーターなどのメカニカルな部分が存在しないため、コストパフォーマンスは磁気記憶媒体に遅れをとるものの低消費電力、高信頼性という面を生かし携帯情報端末などに普及しつつある。

【0004】 なお、フラッシュメモリの特徴は以下のとおりである。第1に、電氣的にデータの書き込み、消去が可能で不揮発メモリである。第2に、データが既に書き込まれているメモリセルにデータを上書きすることはできない（このため、常に消去動作がつきまとう）。第3に、データの消去単位は、数K～数十KByte単位である。第4に、書き込み、消去回数に制限がある。

【0005】 従来の半導体ディスク装置の構成について図10、図11、図12及び図13を参照しながら説明する。図10は、従来の半導体ディスク装置の全体構成を示すブロック図である。図11は、図10のアドレス変換テーブルの内部構成を示す図である。図12は、図10のフラッシュメモリの内部構成を示す図である。ま

た、図13は、図12の消去ブロックの内部構成を示す図である。

【0006】図10において、従来の半導体ディスク装置2は、インターフェイス回路3と、CPU4と、アドレス変換テーブル5と、フラッシュ制御回路6と、データ入出力用セクタバッファ7と、フラッシュメモリ8とを備える。

【0007】半導体ディスク装置2と接続するホスト1の代表的な例は、ノートパソコンや携帯情報端末である。半導体ディスク装置2は、現在の所、カード型のリムーバブルタイプが主流である。インターフェイス回路3は、ホスト1との情報をやりとりする。CPU4は、データの入出力及びフラッシュメモリ8への命令を出力する。

【0008】論理セクタ/物理セクタアドレス変換テーブル5は、論理セクタアドレスを物理セクタアドレスに変換するためのテーブルである。論理セクタアドレス(LSA: Logical Sector Address)とはホスト1が半導体ディスク装置2に指定するセクタアドレスのことである。また、物理セクタアドレス(PSA: Physical Sector Address)とは、半導体ディスク装置2内で使用されるフラッシュメモリ8のアドレスのことである。

【0009】フラッシュ制御回路6は、複雑でないフラッシュメモリ8のデータ処理を行う。単純なデータの受け渡し等はフラッシュ制御回路6で行い、他の処理はCPU4で行う。データ入出力用セクタバッファ7は、データをフラッシュメモリ8からインターフェイス回路3を通して出力、あるいはインターフェイス回路3を通してフラッシュメモリ8にデータを入力する際に用いられる。

【0010】図11において、アドレス変換テーブル5は、論理セクタアドレス(LSA)格納部と物理セクタアドレス(PSA)格納部とから構成される。

【0011】LSA格納部には論理セクタアドレスが保存されている。内容は固定されている。PSA格納部には任意の(図では1~n)フラッシュメモリ8のセクタ番号が保存される。このアドレス変換テーブル5を用いることで、ホスト1が指定する論理セクタアドレスに左右されることなく内部管理に都合のよい物理セクタアドレスにデータを保存することができる。このアドレス変換テーブル5は、頻繁に書き込み・消去されるのでSRAMで構成するのが一般的である。

【0012】このアドレス変換テーブル5の容量は、次のような条件のとき以下になる。20メガバイト(MByte)のフラッシュメモリ8を使用し、データの入出力単位(セクタ)を512バイト(Byte)とすると、半導体ディスク装置2内のセクタ数は次のようになる。半導体ディスク装置2内のセクタ数=20メガバイト÷512バイト=40960セクタ

【0013】次に、「40960」を2進数表現する際

に必要なビット数は、 $1n40960 \div 1n2 = 15.3$ となり、16桁必要となる。

【0014】これにより必要なアドレス変換テーブル5の容量は、 $40960 \times 16 = 655360$ ビットとなり、最終的に、80キロバイト(KByte)必要となる。

【0015】図12において、フラッシュメモリ8は、複数の消去ブロック9と、予備の複数の消去ブロック9とから構成される。

【0016】フラッシュメモリ8は電氣的に書き込み消去可能な不揮発メモリである。不揮発であるためDRAM・SRAMのように電池によるバックアップの必要もなく、また電氣的にデータの消去が可能なのでEPROMと違いボードから外すことなくデータを変更することができる。1セルで1ビットのデータを記憶することができるため、EEROMより安価にメモリを作製することができる。以上の点がフラッシュメモリ8の長所にあたる。短所としては、消去回数に1万回~10万回程度の上限があること、書き込みの際には必ず消去動作が必要なこと(このためデータがすでに書き込まれているセルに上書きすることは不可能)、消去単位は数K~数十KByteのブロック単位であること、等が上げられる。

【0017】図13において、1つの消去ブロック9は、先頭に消去ブロック情報格納領域10と、複数のデータ格納領域11と、LSA格納領域12とを有する。

【0018】消去ブロック情報格納領域10に現在のブロック消去回数を格納しておく。データ格納領域11は、通常512バイト(=1セクタ)の大きさである。LSA格納領域12はセクタごとに存在し、データを書き込む際にホスト1が指定したLSAを格納しておく。これは、論理セクタ/物理セクタアドレス変換テーブル5をSRAMで構成した際、電源オフと同時にデータが消えてしまうためである。電源をオンにしたときに全てのセクタのLSA格納領域12を検索しSRAMテーブル5を再構築する際に用いられる。

【0019】つぎに、従来の半導体ディスク装置の動作について図14、図15及び図16を参照しながら説明する。図14は、従来の半導体ディスク装置の読み出し動作を説明するための図である。また、図15及び図16は、従来の半導体ディスク装置の書き込み動作を説明するための図である。

【0020】フラッシュメモリ8を用いた半導体ディスク装置2はハードディスク装置とは異なり、データを上書きすることができない。従って、ホスト1から送られてくるデータの論理セクタアドレスとそのデータをフラッシュメモリ8のどの物理セクタアドレスに書き込むかを示すアドレス変換テーブル5をSRAM内に記憶させておくことが行われる。このテーブル5を用いることでLSAに左右されることなくフラッシュメモリ8の記憶領域を有効に使用することが可能となる。

【0021】まず、半導体ディスク装置2からのデータ

の読み出し動作を図14で説明する。ホスト1は読み出したデータのセクタアドレスを半導体ディスク装置2に送る。ホスト1から送られてくるアドレスデータには2種類ある。L S A形式とCHS形式である。L S A形式が1～nまでの通し番号でセクタを指定するのに対し、CHS形式はハードディスク装置で使用されるシリンダ・ヘッド・セクタという3つのデータの組み合わせでデータ領域を指定する。半導体ディスク装置2内ではL S A / P S Aアドレス変換テーブル5を用いるため、ホスト1からCHS形式のデータが入力された場合は、例えばインターフェース回路内でL S Aに変換し次の作業に移る。

【0022】CPU4は、アドレス変換テーブル5を用いてホスト1が指定したL S AをP S Aにアドレス変換する。最後にP S Aに対応したフラッシュメモリ8内からデータが読み出される。

【0023】例えば、ホスト1が指定したL S Aが「2」であった場合、アドレス変換テーブル5により「6」というP S Aに変換される。これにより、「A」というデータが読み出されることになる。L S A格納領域12にはL S Aである「2」が格納されている。

【0024】次に、半導体ディスク装置2へのデータの書き込み動作を図15及び図16で説明する。「A」、「B」、「C」というデータがP S Aの「1」、「3」、「7」に格納されている状態を初期状態とする。データを書き込む際に注意しなければならないのはフラッシュメモリ8はデータの再書き込みができないという点である。上記初期状態の場合、P S A「1」、「3」、「7」の領域である。

【0025】データが書き込まれていないL S Aをホスト1が指定してきた場合は、CPU4は、フラッシュメモリ8内の適当な空き領域(P S A「2」、「4」～「6」、「8」～「12」)にデータを書き込み、アドレス変換テーブル5内のデータを更新する。図15は、L S A「4」へ「D」というデータの書き込みをホスト1が指定した場合の例である。データ「D」とホスト1が指定したL S Aを空き領域P S A「4」に書き込み、アドレス変換テーブル5のL S A「4」に対応したP S Aの部分にP S Aの値「4」を書き込む。

【0026】ホスト1から、既にデータが書き込まれている領域への再書き込みが要求された場合(例えば、同名ファイルの上書き保存)であっても、再書き込みデータをフラッシュメモリ8の空き領域に書き込み、アドレス変換テーブル5を更新する。図16は、L S A「2」のデータを再書き込みした際の結果である。更新データ「B'」を空き領域P S A「5」に書き込み、アドレス変換テーブル5のL S A「2」に対応するP S Aを「5」と更新する。なお、P S A「3」が使用済みデータであることは、カード内のCPU4は認識しておかなければならない。

【0027】

【発明が解決しようとする課題】 上述したような従来の半導体ディスク装置では、アドレス変換テーブル5がセクタ(データ管理の最小単位)ごとに1つのP S Aを格納するメモリ領域が必要となるため、フラッシュメモリ8が大容量になるにつれアドレス変換テーブル5も大容量となるという問題点があった。

【0028】この発明は、前述した問題点を解決するためになされたもので、メモリ管理用アドレス変換テーブルを必要としない半導体ディスク装置を得ることを目的とする。

【0029】また、この発明は、アドレス変換テーブルの容量を小さくできる半導体ディスク装置を得ることを目的とする。

【0030】

【課題を解決するための手段】 この発明に係る半導体ディスク装置は、論理消去ブロック番号を格納する論理消去ブロック番号格納領域と、データを格納する複数のデータ格納領域と、前記データ格納領域にデータが格納されているか否かを表すデータ状態フラグを格納する前記データ格納領域毎のデータ状態フラグ領域と、更新先の場所を表す連鎖情報を格納する前記データ格納領域毎の更新データ連鎖情報格納領域とから構成される消去ブロックを複数有するフラッシュメモリと、前記論理消去ブロック番号、前記データ状態フラグ、及び前記連鎖情報に基づいて前記フラッシュメモリ上のデータを管理するCPUとを備えたものである。

【0031】また、この発明に係る半導体ディスク装置は、前記CPUが、論理セクタアドレスを論理消去ブロック番号とそのオフセット値に変換し、この変換した論理消去ブロック番号とそのオフセット値に基づいて前記フラッシュメモリ上の該当消去ブロック及び該当データ格納領域を捜し出し、該当更新データ連鎖情報格納領域に連鎖情報が存在しなければ前記該当データ格納領域の内容を読み出し、前記該当更新データ連鎖情報格納領域に前記連鎖情報が存在すればその連鎖情報に基づいて該当データ格納領域の内容を読み出すものである。

【0032】また、この発明に係る半導体ディスク装置は、前記CPUが、論理セクタアドレスを論理消去ブロック番号とそのオフセット値に変換し、この変換した論理消去ブロック番号とそのオフセット値に基づいて前記フラッシュメモリ上の該当消去ブロック及び該当データ格納領域を捜し出し、該当データ状態フラグが未使用を示すときは前記該当データ格納領域にデータを書き込み、前記該当データ状態フラグが使用中にし、前記該当データ状態フラグが使用中を示すときは空きのデータ格納領域を捜し出してそこにデータを書き込み、このデータ格納領域に対応するデータ状態フラグが使用中にするとともに、前の使用中のデータ格納領域に対応する更新データ連鎖情報格納領域に連鎖情報を書き込むものであ

る。

【0033】また、この発明に係る半導体ディスク装置は、論理消去ブロック番号を格納する論理消去ブロック番号格納領域と、データを格納する複数のデータ格納領域と、前記データ格納領域にデータが格納されているか否かを表すデータ状態フラグを格納する前記データ格納領域毎のデータ状態フラグ領域と、更新先の場所を表す連鎖情報を格納する前記データ格納領域毎の更新データ連鎖情報格納領域とから構成される消去ブロックを複数有するフラッシュメモリと、前記論理消去ブロック番号を物理消去ブロック番号へ変換するためのアドレス変換テーブルと、前記論理消去ブロック番号、前記データ状態フラグ、及び前記連鎖情報に基づいて前記フラッシュメモリ上のデータを管理するCPUとを備えたものである。

【0034】また、この発明に係る半導体ディスク装置は、前記CPUが、論理セクタアドレスを論理消去ブロック番号とそのオフセット値に変換し、前記アドレス変換テーブルに基づいて前記論理消去ブロック番号を物理消去ブロック番号へ変換し、この物理消去ブロック番号とそのオフセット値に基づいて前記フラッシュメモリ上の該当消去ブロック及び該当データ格納領域を捜し出し、該当更新データ連鎖情報格納領域に連鎖情報が存在しなければ前記該当データ格納領域の内容を読み出すものである。

【0035】さらに、この発明に係る半導体ディスク装置は、前記CPUが、論理セクタアドレスを論理消去ブロック番号とそのオフセット値に変換し、前記アドレス変換テーブルに基づいて前記論理消去ブロック番号を物理消去ブロック番号へ変換し、この物理消去ブロック番号とそのオフセット値に基づいて前記フラッシュメモリ上の該当消去ブロック及び該当データ格納領域を捜し出し、該当データ状態フラグが未使用を示すときは前記該当データ格納領域にデータを書き込み、前記該当データ状態フラグを使用中にするものである。

【0036】

【作用】この発明に係る半導体ディスク装置においては、論理消去ブロック番号を格納する論理消去ブロック番号格納領域と、データを格納する複数のデータ格納領域と、前記データ格納領域にデータが格納されているか否かを表すデータ状態フラグを格納する前記データ格納領域毎のデータ状態フラグ領域と、更新先の場所を表す連鎖情報を格納する前記データ格納領域毎の更新データ連鎖情報格納領域とから構成される消去ブロックを複数有するフラッシュメモリと、前記論理消去ブロック番号、前記データ状態フラグ、及び前記連鎖情報に基づいて前記フラッシュメモリ上のデータを管理するCPUとを備えたので、データ管理用のアドレス変換テーブルが不要となり、その分データエリアを大きくできる。

【0037】また、この発明に係る半導体ディスク装置

においては、前記CPUが、論理セクタアドレスを論理消去ブロック番号とそのオフセット値に変換し、この変換した論理消去ブロック番号とそのオフセット値に基づいて前記フラッシュメモリ上の該当消去ブロック及び該当データ格納領域を捜し出し、該当更新データ連鎖情報格納領域に連鎖情報が存在しなければ前記該当データ格納領域の内容を読み出し、前記該当更新データ連鎖情報格納領域に前記連鎖情報が存在すればその連鎖情報に基づいて該当データ格納領域の内容を読み出すので、データ管理用のアドレス変換テーブルが不要となり、その分データエリアを大きくできる。

【0038】また、この発明に係る半導体ディスク装置においては、前記CPUが、論理セクタアドレスを論理消去ブロック番号とそのオフセット値に変換し、この変換した論理消去ブロック番号とそのオフセット値に基づいて前記フラッシュメモリ上の該当消去ブロック及び該当データ格納領域を捜し出し、該当データ状態フラグが未使用を示すときは前記該当データ格納領域にデータを書き込み、前記該当データ状態フラグを使用中にし、前記該当データ状態フラグが使用中を示すときは空きのデータ格納領域を捜し出してそこにデータを書き込み、このデータ格納領域に対応するデータ状態フラグを使用中にするるとともに、前の使用中のデータ格納領域に対応する更新データ連鎖情報格納領域に連鎖情報を書き込むので、データ管理用のアドレス変換テーブルが不要となり、その分データエリアを大きくできる。

【0039】また、この発明に係る半導体ディスク装置においては、論理消去ブロック番号を格納する論理消去ブロック番号格納領域と、データを格納する複数のデータ格納領域と、前記データ格納領域にデータが格納されているか否かを表すデータ状態フラグを格納する前記データ格納領域毎のデータ状態フラグ領域と、更新先の場所を表す連鎖情報を格納する前記データ格納領域毎の更新データ連鎖情報格納領域とから構成される消去ブロックを複数有するフラッシュメモリと、前記論理消去ブロック番号を物理消去ブロック番号へ変換するためのアドレス変換テーブルと、前記論理消去ブロック番号、前記データ状態フラグ、及び前記連鎖情報に基づいて前記フラッシュメモリ上のデータを管理するCPUとを備えたので、データ管理用のアドレス変換テーブルを小さくでき、その分データエリアを大きくできる。

【0040】また、この発明に係る半導体ディスク装置においては、前記CPUが、論理セクタアドレスを論理消去ブロック番号とそのオフセット値に変換し、前記アドレス変換テーブルに基づいて前記論理消去ブロック番号を物理消去ブロック番号へ変換し、この物理消去ブロック番号とそのオフセット値に基づいて前記フラッシュメモリ上の該当消去ブロック及び該当データ格納領域を捜し出し、該当更新データ連鎖情報格納領域に連鎖情報が存在しなければ前記該当データ格納領域の内容を読み

出すので、データ管理用のアドレス変換テーブルを小さくでき、その分データエリアを大きくできる。

【0041】さらに、この発明に係る半導体ディスク装置においては、前記CPUが、論理セクタアドレスを論理消去ブロック番号とそのオフセット値に変換し、前記アドレス変換テーブルに基づいて前記論理消去ブロック番号を物理消去ブロック番号へ変換し、この物理消去ブロック番号とそのオフセット値に基づいて前記フラッシュメモリ上の該当消去ブロック及び該当データ格納領域を捜し出し、該当データ状態フラグが未使用を示すときは前記該当データ格納領域にデータを書き込み、前記該当データ状態フラグを使用中にするので、データ管理用のアドレス変換テーブルを小さくでき、その分データエリアを大きくできる。

【0042】

【実施例】

実施例1. 以下、この発明の実施例1の構成について図1、図2、図3及び図4を参照しながら説明する。図1は、この発明の実施例1の全体構成を示すブロック図である。図2は、図1のフラッシュメモリの内部構成を示す図である。図3は、図2の消去ブロックの内部構成を示す図である。図4は、図3の消去ブロック情報格納領域の内部構成を示す図である。なお、各図中、同一符号は同一又は相当部分を示す。

【0043】図1において、この実施例1に係る半導体ディスク装置2Aは、インターフェイス回路3と、CPU4と、フラッシュ制御回路6と、データ入出力用セクタバッファ7と、フラッシュメモリ8Aとを備える。

【0044】従来の半導体ディスク装置2との違いは、論理セクタ/物理セクタアドレス変換テーブルがない点である。その分、フラッシュメモリ8Aが大きくなっている。

【0045】図2において、フラッシュメモリ8Aは、複数の消去ブロック9Aと、予備の複数の消去ブロック9Aとから構成される。

【0046】メインメモリに使用するフラッシュメモリ8Aは、従来と同様ブロック消去型（消去ブロック単位は数K～数十Kバイト）のフラッシュメモリを用いる。このため、フラッシュメモリ内部ブロック構成は従来と同様である。

【0047】図3において、1つの消去ブロック9Aは、先頭に消去ブロック情報格納領域20と、複数のデータ格納領域21と、データ格納領域21毎のデータ状態フラグ22と更新データ連鎖情報（連鎖データ）格納領域23とを有する。なお、データ格納領域21は、512バイト（=1セクタ）程度の大きさが一般的である。

【0048】データ状態フラグ22には、データ格納領域21に格納されているデータがどのような状態のものを示すためのデータが格納される。データ格納領域2

1の状態は、未使用領域、使用領域（連鎖データなし）、使用領域（連鎖データあり）、不必要データ領域（他の領域に更新済みで消去待ち領域）の4状態が考えられる。

【0049】「000」は未使用領域、「001」は使用領域（連鎖データなし）、「011」は使用領域（連鎖データあり）、「111」は不必要データ領域というように、各状態はビットの組み合わせで表現する。フラッシュメモリ8Aは上書き不可能であるのでこのようなビット表現を用いることになる（「010」等の表現は使えない）。

【0050】連鎖データは、同一領域に上書き命令がホスト1から要求された場合に用いる。フラッシュメモリ8Aは上書き不可能なメモリであるため、データの上書きを要求された場合に他の空き領域にデータを書き込むことで対応する。この際、更新データ先を更新前のデータ領域から検索できる必要があるので更新先の場所を記憶しておくために更新データ連鎖情報格納領域23内の連鎖データが使われる。

【0051】図4において、消去ブロック情報格納領域20は、消去回数格納領域24と、論理消去ブロック番号格納領域25と、その他の情報の格納領域26とから構成される。

【0052】消去回数格納領域24には、今まで行われたブロック消去回数を格納する。フラッシュメモリ8Aは消去回数が10万～100万回程度であるから3バイトもあれば十分である。論理消去ブロック番号格納領域25には、論理消去ブロック番号が格納される。論理消去ブロック番号（PLBN:Physical-Logical Block Number）とは、物理的な（固定された）消去ブロック番号（PBN:Physical Block Number）とちがひ、全ての消去ブロックにつけられるものでどの論理消去ブロック番号も重ならない。このため、物理的な消去ブロック番号とメモリ管理に使われる論理消去ブロック番号は1対1で対応させることができる。仮に20MBの半導体ディスクカードを作製する場合、消去ブロック数は320ブロック程度で構成されるから論理消去ブロック番号格納領域には2バイト必要である。

【0053】この実施例1によるアドレスの算出方法について説明する。ホスト1から送られてくるアドレスはCHSもしくはLSAの形式である。もし、CHS形式で送られてきた場合は例えばインターフェース回路3でLSA形式に変換する。次にPSAと対応づける。具体的には、どの消去ブロックのどのデータ格納領域を扱うのかを決定する。

【0054】データの上書きが可能で消去回数を意識しなくてもよい記憶媒体を使用した装置の場合は、LSAとPSAを1対1（LSAが1ならPSAも1）で対応づけて管理しても問題は生じない。よって、LSAを管理ブロック（消去ブロックと同一）内のデータ領域の数

で割ったときの商を物理ブロック番号、余りを物理ブロック内のオフセット値として、計算を行えば容易に読み出し／書き込みデータのPSAを決定することができる。

【0055】この実施例1では、論理／物理アドレス変換テーブルを用いることなく消去ブロックごとの消去回数を均一化する方法を用いる。このため、消去ブロック9Aごとに論理消去ブロック番号(PLBN)をもたせる。また、ホスト1から送られてくるLSAを管理ブロック(消去ブロックと同一)内のデータ領域の数で割ったときの商を論理消去ブロック番号(PLBN)、余りを論理消去ブロック番号内のオフセット値とする。これにより、アドレス変換テーブルを用いることなく消去ブロック9A内の論理消去ブロック番号格納領域25を書き換えることで消去ブロック9Aごとの消去回数の均一化を図ることができる。

【0056】つぎに、この実施例1の動作について図5、図6及び図7を参照しながら説明する。図5は、この実施例1の読み出し動作を説明するための図である。図6は、この実施例1の読み出し動作を示すフローチャートである。図7は、この実施例1の書き込み動作を示すフローチャートである。

【0057】まず、半導体ディスク装置2Aからのデータ読み出し動作を図5及び図6で説明する。フラッシュメモリ8Aの消去ブロック9Aは説明しやすいよう消去ブロック9A内のデータ格納領域21は3つ、論理消去ブロック「4」、「5」はデータ退避用の予備ブロックとする。つまり、この半導体ディスク装置2Aの容量はホスト1から見た場合、512バイト×3×4=6キロバイトであり、ホスト1から送られてくるLSAは「0

～11」である。

【0058】初めに、ホスト1から読み出すべきデータのセクタ情報を受け取る(ステップ30)。これはLSAの形式かもしくはCHS形式で送られてくる。LSA形式に統一するためにCHSデータ形式で送られてきた場合はLSA形式に変換する(ステップ31～32)。この変換は、半導体ディスク装置2A内のCPU4を用いてもかまわないし、専用の回路を半導体ディスク装置内部に持たせてもかまわない。次にLSA形式のデータをPLBN形式に変換する(ステップ33)。変換に用いられる計算は以前述べた通りである。この変換も専用の回路もしくは内部CPUを用いて計算することができる。

【0059】これにより論理消去ブロック番号(PLBN)とそのオフセット値が決まる。最後に計算した論理消去ブロック番号がフラッシュメモリ8Aのどの消去ブロック9Aに対応しているかを求める(ステップ34～36)。まず、目的の論理消去ブロック番号がフラッシュメモリ8Aのどの消去ブロック9Aの消去ブロック情報格納領域20に格納されているかを調べる。

【0060】ホスト1からLSA=5というデータが送られてきた場合、PLBN変換によれば、論理消去ブロック番号=5÷3=1、オフセット値=5-1×3=2となり、以後(1, 2)と表現することとする。次に、更新データ連鎖情報格納領域23内の連鎖情報を読みとる(ステップ37～38)。この場合、連鎖情報がないため読み出すべきデータは「A」である(ステップ39)。

【0061】また、ホスト1からLSA=6というデータが送られてきた場合、PLBN変換で(2, 0)とわかる。アドレス(2, 0)の更新データ連鎖情報格納領域23には「40」が格納されている。これは、データが論理消去ブロック番号4のオフセット0にデータが更新されていることを表している。これにより読み出すべきデータが「B」であるとわかる(ステップ38, 34, 35, 37～39)。

【0062】20MBのフラッシュメモリ(消去ブロックサイズは64KB)を用いた半導体ディスク装置を例に考えると、検索する最大値は半導体ディスク装置内の消去ブロック数であるから、20MB/64KB=320ブロックとなる。最大値は320であるが、実際のファイルアクセスを考えた場合ファイルは連続したディスク領域に書き込まれるのが通例であるため次回の検索を現在の消去ブロック9Aから始めることでファイル検索数は大幅に減少すると考えられる。

【0063】また、この部分をテーブルとして持たせたものが次の実施例2でこの問題を回避することができる。この際、用いられるテーブルのサイズは後述するように従来の論理／物理アドレス変換テーブルの約227分の1ですむ。ただし、SRAMで構成した場合電源オフとともにデータが揮発してしまうため半導体ディスク立ち上げ時に論理消去ブロック番号と実際の物理消去ブロック番号との対応をCPUで確認しRAMテーブルを再構築する必要がある。

【0064】次に、半導体ディスク装置2Aへのデータ書き込み動作を図7で説明する。図7のステップ40～46は、図6のステップ30～36と同様であるので説明を省略する。読み出したデータ状態フラグ22が「000」であれば、該当データ格納領域21にデータを書き込み、データ状態フラグ22を「001」とする(ステップ47～50)。

【0065】また、ステップ48において該当データ状態フラグ22が「000」でない場合、つまり、ホスト1から既にデータが書き込まれているデータ格納領域21への書き込み命令を受けた場合は、同一消去ブロック9A内の適当な空きデータ格納領域21にデータを書き込み、対応するデータ状態フラグ22を「001」とする。そして、既にデータが書き込まれていたデータ格納領域21に対応するデータ状態フラグ22を「001」から「011」とし、更新データ連鎖情報格納領域23

に書き込みを行なった空きデータ格納領域21の論理消去ブロック番号とオフセット値を書き込む(ステップ48、51~54)。これによりこの連鎖をたどることで常に最新のデータを読み出すことが可能となる。

【0066】さらに、ステップ52において同一消去ブロック9A内に適当な空きデータ格納領域21がない場合は、データ状態フラグ「111」が多く存在する消去ブロック9Aを捜す。このデータ状態フラグ「111」はデリート(データ消去)命令で書き込まれる。この場合は、消去ブロック9Aの入れ換えという作業を用いて10 対応する(ステップ52、55~58)。なお、消去回数も考慮する。まずはじめに、上記条件で捜し出してきた転送元の消去ブロック9A内の有効データを何も書き込まれていない転送先の空き消去ブロック9A上に移動する。この際、上書きなどの過程で不要となったデータは移動しない。また、移動しなければならないデータのオフセット値と移動先のオフセット値は対応させて移動させる。連鎖データが存在する場合はそのデータももれなく書き込む。データの移動が終了した時点で転送元の論理消去ブロック番号を転送先の消去ブロック20 9Aの論理消去ブロック番号格納領域25に書き込む。その後、転送元の消去ブロック9Aのブロック消去を行う。

【0067】この実施例1は、従来の半導体ディスクカードのパフォーマンスを落とすことなく論理セクタ/物理セクタアドレス変換テーブルをなくしてしまえることができる。アドレス変換テーブルをなくすることで、無理なく半導体ディスクカードの大容量化を進めることができる。従来のアドレス変換テーブルを用いると20MBの半導体ディスク装置で80KBのアドレス変換テーブルが、40MBの半導体ディスク装置で160KB(1.25Mbit)のアドレス変換テーブルが必要となる。これがなくなればSRAMにかかっていたコストを削減することができ、またアドレス変換テーブル用SRAMメモリが搭載されていたスペースにフラッシュメモリを搭載できるため半導体ディスク装置の容量を増大させることができる。

【0068】実施例2. この発明の実施例2について図8及び図9を参照しながら説明する。図8は、この発明の実施例2の全体構成を示すブロック図である。図9は、図8のアドレス変換テーブルの内部構成を示す図である。

【0069】図8において、この実施例2に係る半導体ディスク装置2Bは、インターフェイス回路3と、CPU4と、アドレス変換テーブル5Aと、フラッシュ制御回路6と、データ入出力用セクタバッファ7と、フラッシュメモリ8Bとを備える。

【0070】図9において、アドレス変換テーブル5Aは、論理消去ブロック番号(PLBN)格納部と物理消去ブロック番号(PBN)格納部とから構成される。

【0071】このアドレス変換テーブル5Aの容量は、以下のようになる。20メガバイトのフラッシュメモリ8Bを使用し、1ブロック(消去ブロック)を64キロバイトとすると、半導体ディスク装置2B内のブロック数は次のようになる。半導体ディスク装置2B内のブロック数=20メガバイト÷64キロバイト=320ブロック

【0072】次に、「320」を2進数表現する際に必要なビット数は、 $1n320 \div 1n2 = 9$ となり、9桁必要となる。これにより必要なアドレス変換テーブル5Aの容量は、 $320 \times 9 = 2880$ ビットとなり、最終的に、360バイト必要となる。これは、従来の1/227である。

【0073】この実施例2の動作は、基本的には上記実施例1と同様であり、異なる点はアドレス変換テーブル5Aを用いてストレートにフラッシュメモリ8B上の消去ブロック9Aをアクセスすることができることである。

【0074】この実施例2は、従来の半導体ディスクカードのパフォーマンスを落とすことなくアドレス変換テーブルの容量を小さくしてしまえることができる。アドレス変換テーブルの容量を小さくすることで、無理なく半導体ディスクカードの大容量化を進めることができる。従来のアドレス変換テーブルを用いると20MBの半導体ディスクで80KBのアドレス変換テーブルが、40MBの半導体ディスクで160KB(1.25Mbit)のアドレス変換テーブルが必要となる。これが1/227に小さくなればSRAMにかかっていたコストを削減することができ、またアドレス変換テーブル用SRAMメモリが搭載されていたスペースにフラッシュメモリを搭載できるため半導体ディスク装置の容量を増大させることができる。

【0075】

【発明の効果】この発明に係る半導体ディスク装置は、以上説明したとおり、論理消去ブロック番号を格納する論理消去ブロック番号格納領域と、データを格納する複数のデータ格納領域と、前記データ格納領域にデータが格納されているか否かを表すデータ状態フラグを格納する前記データ格納領域毎のデータ状態フラグ領域と、更新先の場所を表す連鎖情報を格納する前記データ格納領域毎の更新データ連鎖情報格納領域とから構成される消去ブロックを複数有するフラッシュメモリと、前記論理消去ブロック番号、前記データ状態フラグ、及び前記連鎖情報に基づいて前記フラッシュメモリ上のデータを管理するCPUとを備えたので、データ管理用のアドレス変換テーブルが不要となり、その分データエリアを大きくできるという効果を奏する。

【0076】また、この発明に係る半導体ディスク装置は、以上説明したとおり、前記CPUが、論理セクタアドレスを論理消去ブロック番号とそのオフセット値に変換し、この変換した論理消去ブロック番号とそのオフセ

ット値に基づいて前記フラッシュメモリ上の該当消去ブロック及び該当データ格納領域を捜し出し、該当更新データ連鎖情報格納領域に連鎖情報が存在しなければ前記該当データ格納領域の内容を読み出し、前記該当更新データ連鎖情報格納領域に前記連鎖情報が存在すればその連鎖情報に基づいて該当データ格納領域の内容を読み出すので、データ管理用のアドレス変換テーブルが不要となり、その分データエリアを大きくできるという効果を奏する。

【0077】また、この発明に係る半導体ディスク装置は、以上説明したとおり、前記CPUが、論理セクタアドレスを論理消去ブロック番号とそのオフセット値に変換し、この変換した論理消去ブロック番号とそのオフセット値に基づいて前記フラッシュメモリ上の該当消去ブロック及び該当データ格納領域を捜し出し、該当データ状態フラグが未使用を示すときは前記該当データ格納領域にデータを書き込み、前記該当データ状態フラグが使用中を示すときは空きのデータ格納領域を捜し出してそこにデータを書き込み、このデータ格納領域に対応するデータ状態フラグを使用中にするとともに、前の使用中のデータ格納領域に対応する更新データ連鎖情報格納領域に連鎖情報を書き込むので、データ管理用のアドレス変換テーブルが不要となり、その分データエリアを大きくできるという効果を奏する。

【0078】また、この発明に係る半導体ディスク装置は、以上説明したとおり、論理消去ブロック番号を格納する論理消去ブロック番号格納領域と、データを格納する複数のデータ格納領域と、前記データ格納領域にデータが格納されているか否かを表すデータ状態フラグを格納する前記データ格納領域毎のデータ状態フラグ領域と、更新先の場所を表す連鎖情報を格納する前記データ格納領域毎の更新データ連鎖情報格納領域とから構成される消去ブロックを複数有するフラッシュメモリと、前記論理消去ブロック番号を物理消去ブロック番号へ変換するためのアドレス変換テーブルと、前記論理消去ブロック番号、前記データ状態フラグ、及び前記連鎖情報に基づいて前記フラッシュメモリ上のデータを管理するCPUとを備えたので、データ管理用のアドレス変換テーブルを小さくでき、その分データエリアを大きくできるという効果を奏する。

【0079】また、この発明に係る半導体ディスク装置は、以上説明したとおり、前記CPUが、論理セクタアドレスを論理消去ブロック番号とそのオフセット値に変換し、前記アドレス変換テーブルに基づいて前記論理消去ブロック番号を物理消去ブロック番号へ変換し、この物理消去ブロック番号とそのオフセット値に基づいて前記フラッシュメモリ上の該当消去ブロック及び該当データ格納領域を捜し出し、該当更新データ連鎖情報格納領域に連鎖情報が存在しなければ前記該当データ格納領域

の内容を読み出すので、データ管理用のアドレス変換テーブルを小さくでき、その分データエリアを大きくできるという効果を奏する。

【0080】さらに、この発明に係る半導体ディスク装置は、以上説明したとおり、前記CPUが、論理セクタアドレスを論理消去ブロック番号とそのオフセット値に変換し、前記アドレス変換テーブルに基づいて前記論理消去ブロック番号を物理消去ブロック番号へ変換し、この物理消去ブロック番号とそのオフセット値に基づいて前記フラッシュメモリ上の該当消去ブロック及び該当データ格納領域を捜し出し、該当データ状態フラグが未使用を示すときは前記該当データ格納領域にデータを書き込み、前記該当データ状態フラグが使用中にするので、データ管理用のアドレス変換テーブルを小さくでき、その分データエリアを大きくできるという効果を奏する。

【図面の簡単な説明】

【図1】 この発明の実施例1の全体構成を示すブロック図である。

【図2】 この発明の実施例1のフラッシュメモリ内の構成を示す図である。

【図3】 この発明の実施例1のフラッシュメモリ内の消去ブロックの構成を示す図である。

【図4】 この発明の実施例1の消去ブロック内の消去ブロック情報格納領域の構成を示す図である。

【図5】 この発明の実施例1のデータ読み出し動作を説明するための図である。

【図6】 この発明の実施例1のデータ読み出し動作を示すフローチャートである。

【図7】 この発明の実施例1のデータ書き込み動作を示すフローチャートである。

【図8】 この発明の実施例2の全体構成を示すブロック図である。

【図9】 この発明の実施例2のアドレス変換テーブルの構成を示す図である。

【図10】 従来の半導体ディスク装置の全体構成を示すブロック図である。

【図11】 従来の半導体ディスク装置のアドレス変換テーブルの構成を示す図である。

【図12】 従来の半導体ディスク装置のフラッシュメモリ内の構成を示す図である。

【図13】 従来の半導体ディスク装置のフラッシュメモリ内の消去ブロックの構成を示す図である。

【図14】 従来の半導体ディスク装置のデータ読み出し動作を説明するための図である。

【図15】 従来の半導体ディスク装置のデータ書き込み動作を説明するための図である。

【図16】 従来の半導体ディスク装置のデータ書き込み動作を説明するための図である。

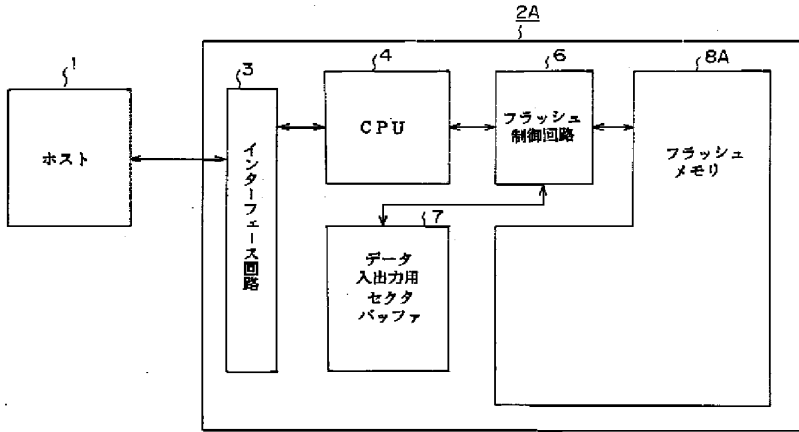
【符号の説明】

1 ホスト、2 A、2 B 半導体ディスク装置、3 イ

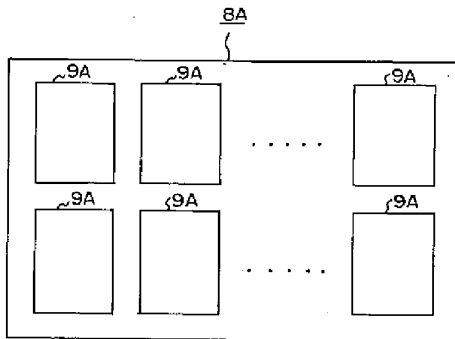
17
 インターフェース回路、4 CPU、5 A アドレス変換
 テーブル、6 フラッシュ制御回路、7 データ入出力

18
 用セクタバッファ、8 A、8 B フラッシュメモリ。

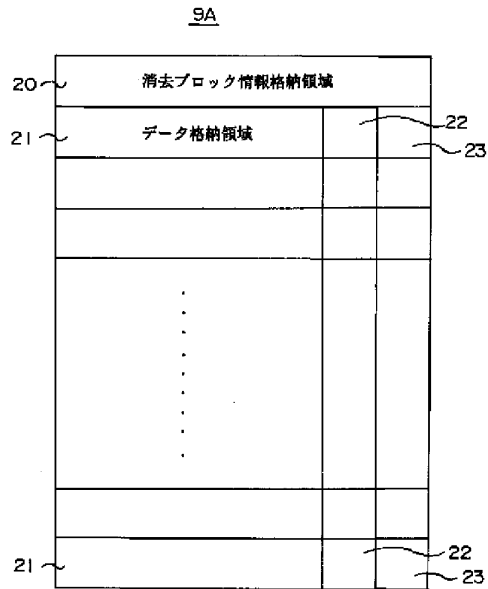
【図1】



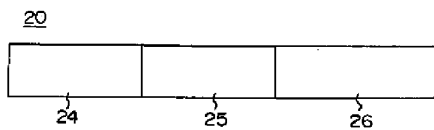
【図2】



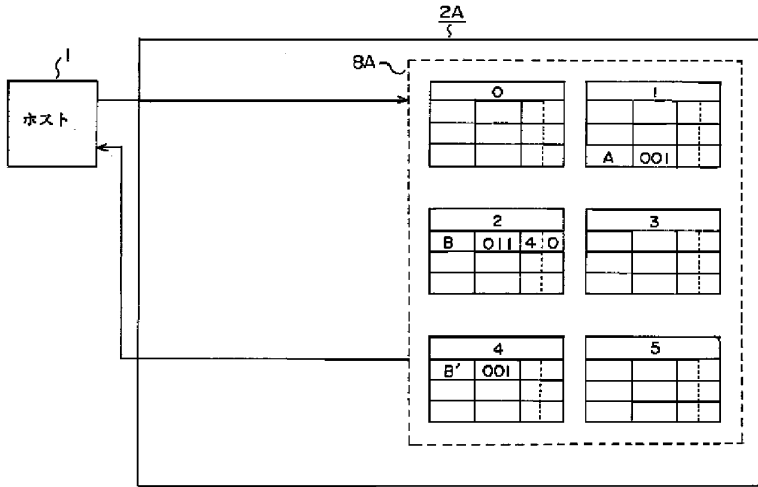
【図3】



【図4】



【図5】

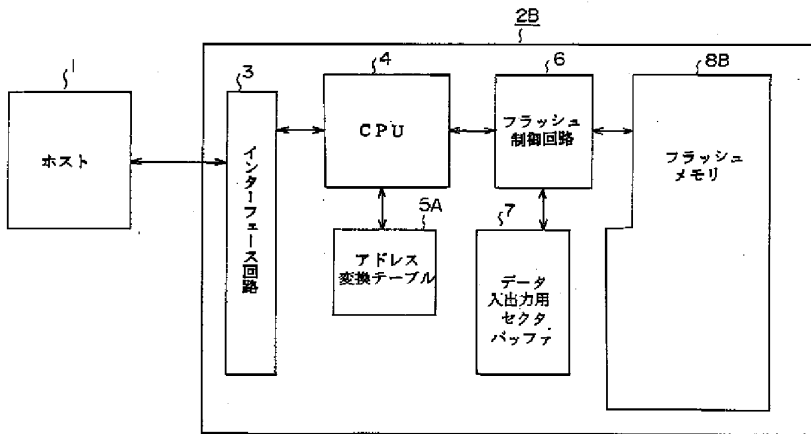


【図11】

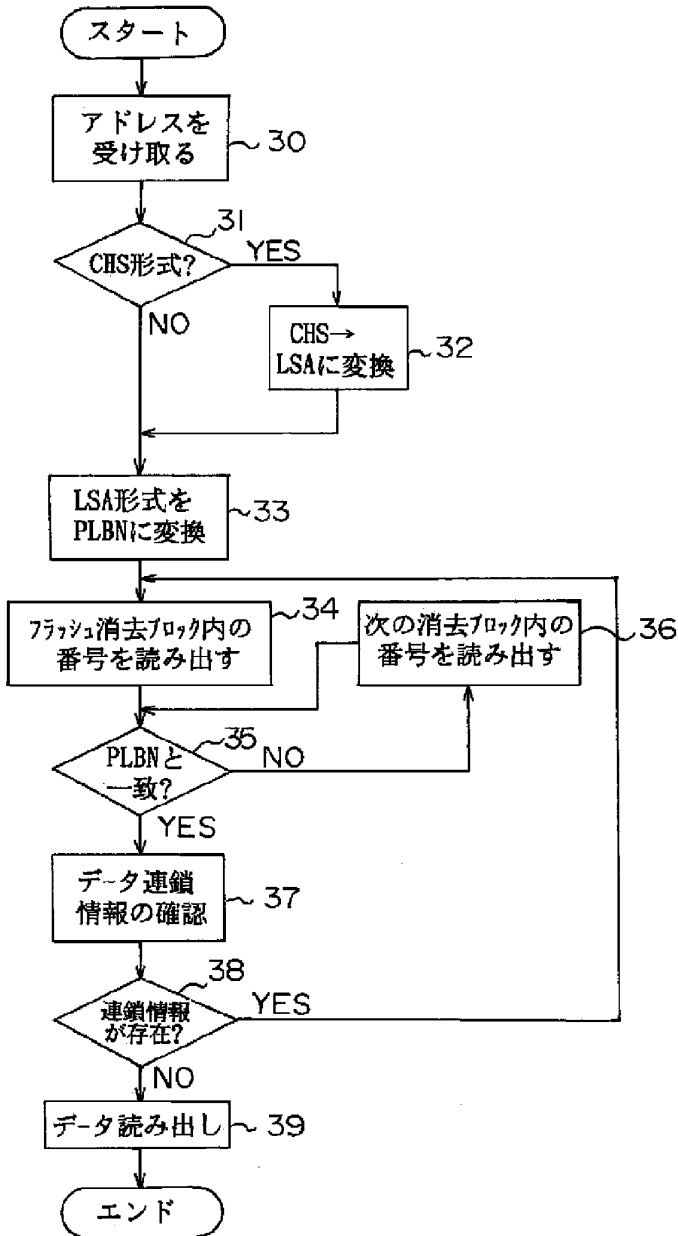
5

LSA	PSA
1	6
2	1
3	3
4	n
5	5
⋮	⋮
⋮	⋮
⋮	⋮
n-1	
n	

【図8】



【図6】

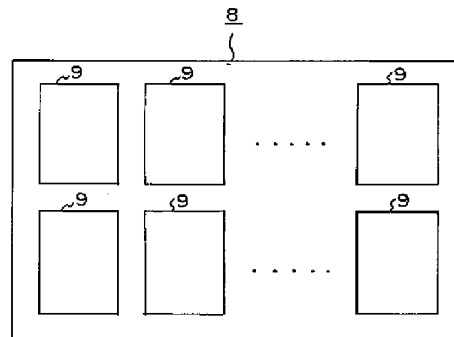


【図9】

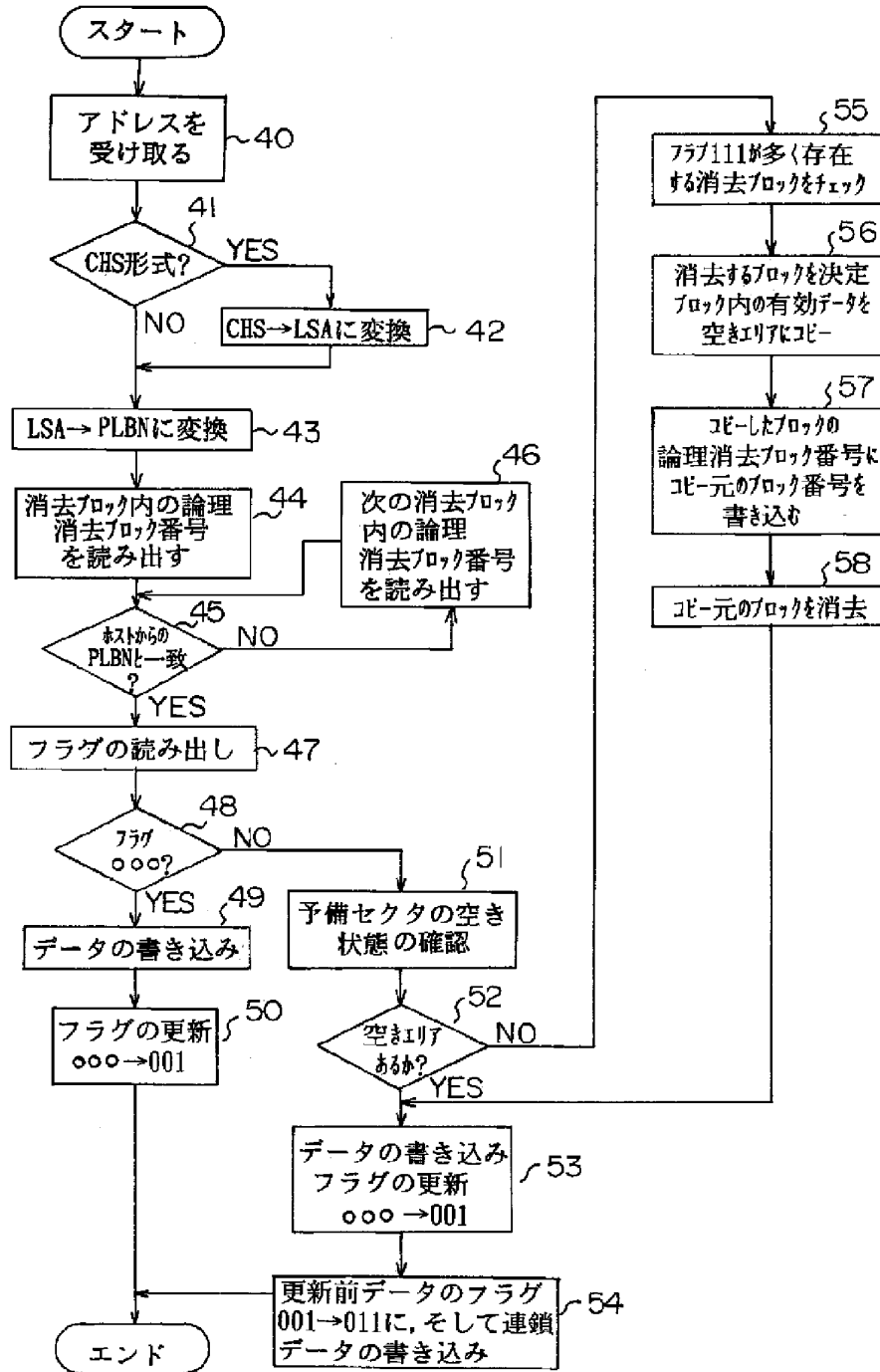
5A

PLBN	PBN
1	3
2	8
3	n
⋮	⋮
⋮	⋮
⋮	⋮
⋮	⋮
⋮	⋮
n-1	
n	

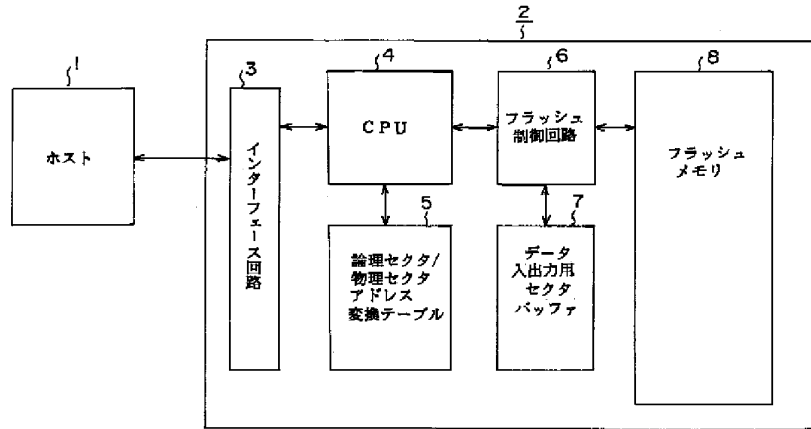
【図12】



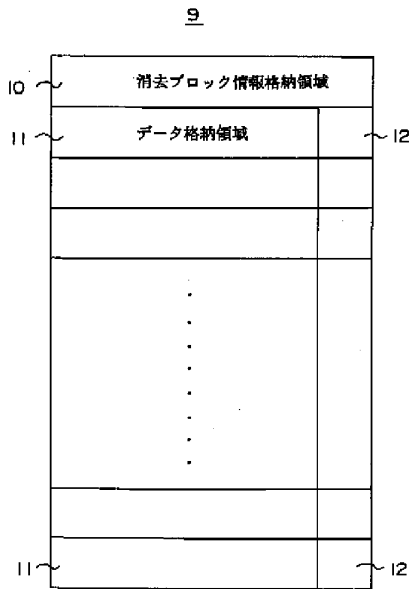
【図7】



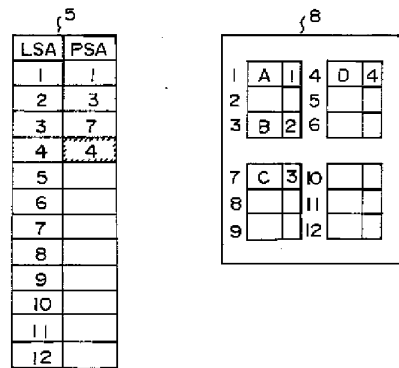
【図10】



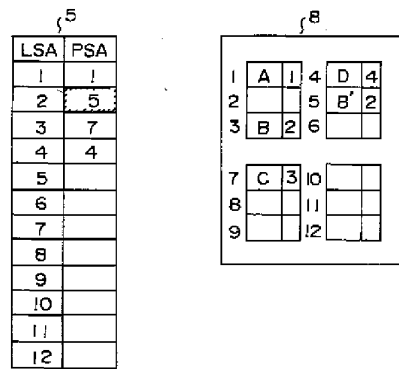
【図13】



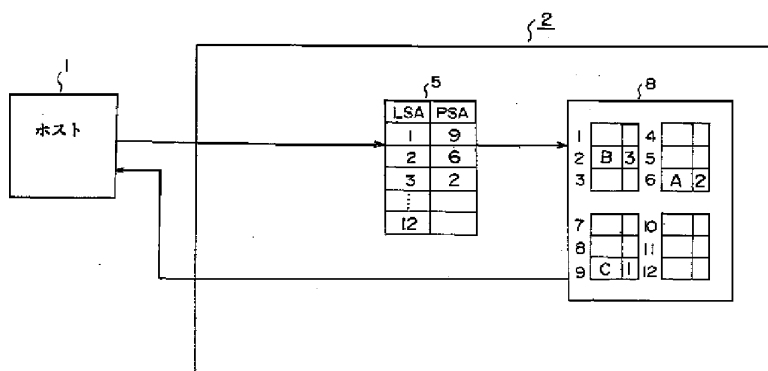
【図15】



【図16】



【図14】



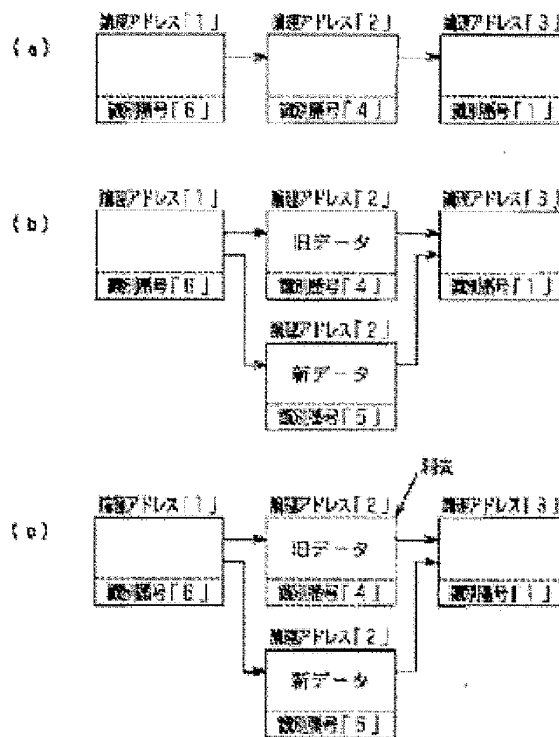
EXTERNAL STORAGE DEVICE AND DEVICE AND METHOD FOR DATA PROCESSING

Publication number: JP11110300
Publication date: 1999-04-23
Inventor: FUSE HIROAKI; SASA SATORU; ONOE ATSUSHI
Applicant: SONY CORP
Classification:
 - international: G06F12/16; G06F3/06; G06F3/08; G06F12/00; G06F12/16; G06F3/06; G06F3/08; G06F12/00; (IPC1-7): G06F12/16; G06F3/06; G06F3/08; G06F12/16
 - European:
Application number: JP19970267178 19970930
Priority number(s): JP19970267178 19970930

Report a data error here

Abstract of JP11110300

PROBLEM TO BE SOLVED: To make it possible to detect errors and to appropriately restore them even when there occur condition in which plural blocks having the same logical addresses simultaneously coexist (a logical address error) or condition in which a block indicated by a link address does not exist (a link address error) caused by forcibly removing an external storage device.
SOLUTION: In storing data in an external storage device in which a storage area is divided into plural blocks, the data are stored by allocating logical addresses to blocks and at the same time identification numbers for indicating new/old data concerned in the blocks in which the data are stored. Then, in reading the data out of the external storage device, when there exist plural blocks having the same logical address, the new/old of the data stored in those blocks are discriminated on the basis of the identification numbers, new data are recognized as invalid data and old ones valid data.



Data supplied from the esp@cenet database - Worldwide

Family list

4 family members for: **JP11110300**
Derived from 2 applications

[Back to JP11110300](#)

1 EXTERNAL STORAGE DEVICE AND DEVICE AND METHOD FOR DATA PROCESSING

Inventor: FUSE HIROAKI; SASA SATORU; (+1)

Applicant: SONY CORP

EC:

IPC: *G06F12/16; G06F3/06; G06F3/08* (+9)

Publication info: **JP3070539B2 B2** - 2000-07-31

JP11110300 A - 1999-04-23

2 EXTERNAL STORAGE DEVICE, DATA PROCESSOR, AND DATA PROCESSING METHOD

Inventor: FUSE HIROAKI; SASA SATORU; (+1)

Applicant: SONY CORP

EC:

IPC: *G06F12/16; G06F3/06; G06F3/08* (+9)

Publication info: **JP3640154B2 B2** - 2005-04-20

JP2000163302 A - 2000-06-16

Data supplied from the *esp@cenet* database - Worldwide

(51)Int.Cl. ⁶	識別記号	F I	
G 0 6 F 12/16	3 1 0	G 0 6 F 12/16	3 1 0 J
	3 4 0		3 4 0 P
	3 0 5	3/06	3 0 5 A
		3/08	H

審査請求 有 請求項の数 6 O L (全 24 頁)

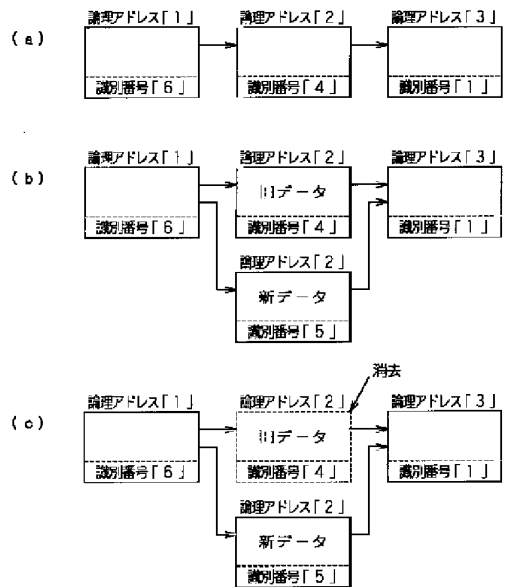
(21)出願番号	特願平9-267178	(71)出願人	000002185 ソニー株式会社 東京都品川区北品川6丁目7番35号
(22)出願日	平成9年(1997)9月30日	(72)発明者	布施 博明 東京都品川区北品川6丁目7番35号 ソニー株式会社内
		(72)発明者	佐々 哲 東京都品川区北品川6丁目7番35号 ソニー株式会社内
		(72)発明者	尾上 淳 東京都品川区北品川6丁目7番35号 ソニー株式会社内
		(74)代理人	弁理士 小池 晃 (外2名)

(54)【発明の名称】 外部記憶装置、データ処理装置及びデータ処理方法

(57)【要約】

【課題】 外部記憶装置が強制的に取り外されたりして、同じ論理アドレスを持つ複数のブロックが同時に存在するような状態（論理アドレスエラー）や、連結アドレスで指し示されたブロックが存在しないような状態（連結アドレスエラー）になったとしても、それらのエラーを検出し適切に修復できるようにする。

【解決手段】 記憶領域が複数のブロックに分割される外部記憶装置にデータを格納する際に、ブロックに論理アドレスを割り当ててデータを格納するとともに、データが格納されるブロックに当該データの新旧を示す識別番号を格納する。そして、外部記憶装置からデータを読み出す際に、同じ論理アドレスを持つ複数のブロックが存在する場合には、識別番号に基づいて、それらのブロックに格納されているデータの新旧を判別し、新しい方のデータは無効なデータとし、古い方のデータを有効なデータとする。



ファイル更新の具体例

【特許請求の範囲】

【請求項1】 記憶領域が複数のブロックに分割されてなり、各ブロックに対して論理アドレスが割り当てられる外部記憶装置であって、各ブロックには、データと共に当該データの新旧を示す識別番号が格納され、

同じ論理アドレスを持つ複数のブロックが存在する場合には、上記識別番号に基づいて、それらのブロックに格納されているデータの新旧が判別され、新しい方のデータは無効なデータとされ、古い方のデータが有効なデータとされることを特徴とする外部記憶装置。

【請求項2】 記憶領域が複数のブロックに分割されてなり、各ブロックに対して論理アドレスが割り当てられるとともに、ブロックに格納するデータをブロック毎に管理するための情報である分散管理情報が各ブロックにそれぞれ格納され、全ブロックを管理するための情報である集合管理情報が各ブロックの分散管理情報から生成され、ブロック全体が上記集合管理情報に基づいて管理される外部記憶装置において、

データがファイル単位で格納されるとともに、一つのファイルが複数のブロックにわたる場合には、当該ファイルを格納しているブロックのそれぞれに次のブロックの論理アドレスが連結アドレスとして格納され、

上記集合管理情報が生成されるときに、連結アドレスが指し示す論理アドレスを持つブロックが存在するか否かが調べられることを特徴とする外部記憶装置。

【請求項3】 記憶領域が複数のブロックに分割されてなる外部記憶装置にデータを格納する際に、ブロックに論理アドレスを割り当ててデータを格納するとともに、データが格納されるブロックに当該データの新旧を示す識別番号を格納し、

上記外部記憶装置からデータを読み出す際に、同じ論理アドレスを持つ複数のブロックが存在する場合には、上記識別番号に基づいて、それらのブロックに格納されているデータの新旧を判別し、新しい方のデータは無効なデータとし、古い方のデータを有効なデータとすることを特徴とするデータ処理装置。

【請求項4】 記憶領域が複数のブロックに分割されてなる外部記憶装置に対して、各ブロックに論理アドレスを割り当ててデータを格納するデータ処理装置であって、

上記外部記憶装置に格納するデータをブロック毎に管理するための情報である分散管理情報を各ブロックにそれぞれ格納するとともに、全ブロックを管理するための情報である集合管理情報を各ブロックの分散管理情報から生成して当該集合管理情報に基づいてブロック全体を管理し、

上記外部記憶装置にデータをファイル単位で格納する際に、一つのファイルが複数のブロックにわたる場合には、当該ファイルが格納されるブロックのそれぞれに次

のブロックの論理アドレスを連結アドレスとして格納し、

上記集合管理情報を生成するときに、連結アドレスが指し示す論理アドレスを持つブロックが存在するか否かを調べられることを特徴とするデータ処理装置。

【請求項5】 記憶領域が複数のブロックに分割されてなる外部記憶装置にデータを格納する際に、ブロックに論理アドレスを割り当ててデータを格納するとともに、データが格納されるブロックに当該データの新旧を示す識別番号を格納し、

上記外部記憶装置からデータを読み出す際に、同じ論理アドレスを持つ複数のブロックが存在する場合には、上記識別番号に基づいて、それらのブロックに格納されているデータの新旧を判別し、新しい方のデータは無効なデータとし、古い方のデータを有効なデータとすることを特徴とするデータ処理方法。

【請求項6】 記憶領域が複数のブロックに分割されてなる外部記憶装置に対して、各ブロックに論理アドレスを割り当ててデータを格納する際に、

上記外部記憶装置に格納するデータをブロック毎に管理するための情報である分散管理情報を各ブロックにそれぞれ格納するとともに、全ブロックを管理するための情報である集合管理情報を各ブロックの分散管理情報から生成して当該集合管理情報に基づいてブロック全体を管理し、

上記外部記憶装置にデータをファイル単位で格納する際に、一つのファイルが複数のブロックにわたる場合には、当該ファイルが格納されるブロックのそれぞれに次のブロックの論理アドレスを連結アドレスとして格納し、

上記集合管理情報を生成するときに、連結アドレスが指し示す論理アドレスを持つブロックが存在するか否かを調べられることを特徴とするデータ処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、記憶領域が複数のブロックに分割されてなる外部記憶装置に関する。また、本発明は、記憶領域が複数のブロックに分割されてなる外部記憶装置にデータを格納するデータ処理装置に関する。また、本発明は、記憶領域が複数のブロックに分割されてなる外部記憶装置にデータを格納する際のデータ処理方法に関する。

【0002】

【従来の技術】パーソナルコンピュータやデジタルスチルカメラ等のようなデータ処理装置に用いられる外部記憶装置として、フラッシュメモリを備えた外部記憶装置がある。

【0003】フラッシュメモリを備えた外部記憶装置は、記憶領域を複数のブロックに分割し、データ領域の管理をブロック単位で行う。ここで、各ブロックはデー

タ消去の単位となる。すなわち、データを消去する際は、当該データを含むブロック全体に対して初期化処理を施す。これにより、当該ブロックに格納されているデータが一括して消去される。

【0004】このような外部記憶装置では、データをブロックに格納するときに、それらのブロックに対してユニークな論理アドレスが設定される。そして、各ブロックは、この論理アドレスを用いて管理される。

【0005】また、外部記憶装置に格納されるデータは、通常、ファイル単位で外部記憶装置に格納されるが、一つのファイルが複数のブロックにわたる場合には、それらのブロックの接続情報が必要となる。そこで、一つのファイルが複数のブロックにわたる場合には、当該ファイルを格納しているブロックのそれぞれに、次のブロックの論理アドレス（以下、連結アドレスと称する。）が格納される。

【0006】

【発明が解決しようとする課題】従来、このような外部記憶装置では、記憶領域内にエラーがあるか否かを検査する処理や、エラーがあった場合に当該エラーの修復を試みる処理を、外部記憶装置の起動時に毎回実行するようにしていた。なお、以下の説明では、このような処理のことを、エラー検出訂正処理と称する。通常、このようなエラー検出訂正処理は、比較的負荷が大きく処理に時間を要する処理である。したがって、従来の外部記憶装置は、エラー検出訂正処理のために、速やかに起動することができないという問題があった。

【0007】また、データ領域の管理をブロック単位で行うような外部記憶装置では、ブロックにデータを新規に書き込んでいるときや、ブロックに格納されているデータを更新しているときなどに、いきなり電源が遮断されたり、データ処理装置から外部記憶装置が強制的に取り外されたりしたような場合に、同じ論理アドレスを持つ複数のブロックが同時に存在するような状態（以下、論理アドレスエラーと称する。）となったり、連結アドレスで指し示されたブロックが存在しないような状態（以下、連結アドレスエラーと称する。）となったりする可能性がある。当然の事ながら、このような状態になると、ファイルが予期せぬブロックに連結されてしまったりして、外部記憶装置を正常に使用することができなくなってしまう。

【0008】しかしながら、従来の外部記憶装置は、論理アドレスエラーや連結アドレスエラーを検出して適切に修復するような機能を備えていなかった。そのため、従来は、いきなり電源が遮断されたり、データ処理装置から外部記憶装置が強制的に取り外されたりしたような場合に、その後、外部記憶装置を正常に使用することができなくなってしまうことがあった。

【0009】本発明は、以上のような従来の実情に鑑みて提案されたものであり、外部記憶装置に論理アドレス

エラーや連結アドレスエラーが生じて、それらのエラーを検出し適切に修復できるようにすることを目的としている。

【0010】

【課題を解決するための手段】本発明に係る第1の外部記憶装置は、記憶領域が複数のブロックに分割されてなり、各ブロックに対して論理アドレスが割り当てられる外部記憶装置である。各ブロックには、データと共に当該データの新旧を示す識別番号が格納される。そして、同じ論理アドレスを持つ複数のブロックが存在する場合には、識別番号に基づいて、それらのブロックに格納されているデータの新旧が判別され、新しい方のデータは無効なデータとされ、古い方のデータが有効なデータとされる。

【0011】また、本発明に係る第2の外部記憶装置は、記憶領域が複数のブロックに分割されてなり、各ブロックに対して論理アドレスが割り当てられる。また、ブロックに格納するデータをブロック毎に管理するための情報である分散管理情報が各ブロックにそれぞれ格納され、全ブロックを管理するための情報である集合管理情報が各ブロックの分散管理情報から生成され、ブロック全体が集合管理情報に基づいて管理される。この外部記憶装置は、データがファイル単位で格納されるとともに、一つのファイルが複数のブロックにわたる場合には、当該ファイルを格納しているブロックのそれぞれに次のブロックの論理アドレスが連結アドレスとして格納される。そして、集合管理情報が生成されるときに、連結アドレスが指し示す論理アドレスを持つブロックが存在するか否かが調べられる。

【0012】本発明に係る第1のデータ処理装置は、記憶領域が複数のブロックに分割されてなる外部記憶装置にデータを格納する際に、ブロックに論理アドレスを割り当ててデータを格納するとともに、データが格納されるブロックに当該データの新旧を示す識別番号を格納する。そして、外部記憶装置からデータを読み出す際に、同じ論理アドレスを持つ複数のブロックが存在する場合には、識別番号に基づいて、それらのブロックに格納されているデータの新旧を判別し、新しい方のデータは無効なデータとし、古い方のデータを有効なデータとする。

【0013】また、本発明に係る第2のデータ処理装置は、記憶領域が複数のブロックに分割されてなる外部記憶装置に対して、各ブロックに論理アドレスを割り当ててデータを格納するデータ処理装置である。そして、外部記憶装置に格納するデータをブロック毎に管理するための情報である分散管理情報を各ブロックにそれぞれ格納するとともに、全ブロックを管理するための情報である集合管理情報を各ブロックの分散管理情報から生成して当該集合管理情報に基づいてブロック全体を管理する。また、外部記憶装置にデータをファイル単位で格納

する際に、一つのファイルが複数のブロックにわたる場合には、当該ファイルが格納されるブロックのそれぞれに次のブロックの論理アドレスを連結アドレスとして格納する。そして、集合管理情報を生成するときに、連結アドレスが指し示す論理アドレスを持つブロックが存在するか否かを調べる。

【0014】本発明に係る第1のデータ処理方法では、記憶領域が複数のブロックに分割されてなる外部記憶装置にデータを格納する際に、ブロックに論理アドレスを割り当ててデータを格納するとともに、データが格納されるブロックに当該データの新旧を示す識別番号を格納する。そして、外部記憶装置からデータを読み出す際に、同じ論理アドレスを持つ複数のブロックが存在する場合には、識別番号に基づいて、それらのブロックに格納されているデータの新旧を判別し、新しい方のデータは無効なデータとし、古い方のデータを有効なデータとする。

【0015】また、本発明に係る第2のデータ処理方法では、記憶領域が複数のブロックに分割されてなる外部記憶装置に対して、各ブロックに論理アドレスを割り当ててデータを格納する際に、外部記憶装置に格納するデータをブロック毎に管理するための情報である分散管理情報を各ブロックにそれぞれ格納する。また、全ブロックを管理するための情報である集合管理情報を各ブロックの分散管理情報から生成して当該集合管理情報に基づいてブロック全体を管理する。また、外部記憶装置にデータをファイル単位で格納する際に、一つのファイルが複数のブロックにわたる場合には、当該ファイルが格納されるブロックのそれぞれに次のブロックの論理アドレスを連結アドレスとして格納する。そして、上記集合管理情報を生成するときに、連結アドレスが指し示す論理アドレスを持つブロックが存在するか否かを調べる。

【0016】

【発明の実施の形態】以下、本発明の実施の形態について、図面を参照しながら詳細に説明する。

【0017】1. システムの全体構成

本発明が適用されるシステムの一例について、その全体構成を図1に示す。このシステムは、ホスト側システムとなるデータ処理装置1と、シリアルインターフェースを介してデータ処理装置1に接続される外部記憶装置であるメモリカード2とから構成される。

【0018】なお、ここでは、データ処理装置1とメモリカード2との間でのデータのやり取りをシリアルインターフェースによって行うシステムを例に挙げるが、本発明は、データのやり取りをパラレルインターフェースによって行うシステムに対しても適用可能である。

【0019】データ処理装置1は、演算処理装置(CPU)3と、内部メモリ4と、補助記憶装置5と、シリアルインターフェース回路6とを備え、これらがバス7によって相互に接続されてなる。このデータ処理装置1

は、例えば、補助記憶装置5に格納されているプログラムを読み出して、当該プログラムを、内部メモリ4をワークエリアとして使用して、CPU3により実行する。このとき、必要に応じて、シリアルインターフェース回路6を介してメモリカード2との間でデータのやり取りを行う。

【0020】なお、本発明が適用されるシステムに使用されるデータ処理装置1は、外部記憶装置との間でデータのやり取りが可能なものであるならば特に限定されるものではなく、本発明は、パーソナルコンピュータ、デジタルスチルカメラ、デジタルビデオカメラ等、種々のデータ処理装置に適用可能である。

【0021】データ処理装置1とメモリカード2とは、シリアルインターフェースによって接続されており、具体的には、少なくとも3本のデータ線SCLK、State、DIOによって接続される。すなわち、データ処理装置1とメモリカード2とは、少なくとも、データ伝送時にクロック信号を伝送する第1のデータ線SCLKと、データ伝送時に必要なステータス信号を伝送する第2のデータ線Stateと、メモリカード2に書き込むデータ又はメモリカード2から読み出すデータ等をシリアルに伝送する第3のデータ線DIOとによって接続され、これらを介して、データ処理装置1とメモリカード2との間でのデータのやり取りを行う。

【0022】データ処理装置1とメモリカード2との間でのデータのやり取りは、通常、ヘッダーと実データとから構成されるファイル単位で行われる。なお、ファイルのヘッダーには、例えば、ファイルにアクセスするための情報や、データ処理装置1で実行されるプログラムで必要とされる情報等が格納される。

【0023】2. メモリカードの構成

メモリカード2は、図2に示すように、いわゆるコントロールICからなるコントローラ11と、コントローラ11によって管理されるフラッシュメモリ12とを備えている。

【0024】コントローラ11は、シリアル/パラレル変換やパラレル/シリアル変換等を行うシリアル/パラレル・パラレル/シリアル・インターフェース・シーケンサ13(以下、S/P&P/S・インターフェース・シーケンサ13と称する。)と、フラッシュメモリ12へのインターフェースを司るフラッシュメモリ・インターフェース・シーケンサ14と、S/P&P/S・インターフェース・シーケンサ13とフラッシュメモリ・インターフェース・シーケンサ14との間でやり取りされるデータを一時的に記憶するページバッファ15と、エラー訂正の処理を行うエラー訂正回路16と、フラッシュメモリ12へのアクセスを制御する制御コマンドの生成等を行うコマンドジェネレータ17と、このメモリカード2のバージョン情報や各種属性情報等が格納されているコンフィグレーションROM18と、各回路に対し

てそれらの動作に必要なクロック信号を供給する発振器19とを備えている。

【0025】S/P&P/S・インターフェース・シーケンサ13は、少なくとも上述した3本のデータ線SCLK, State, DIOを介して、データ処理装置1のシリアルインターフェース回路6に接続され、これらのデータ線SCLK, State, DIOを介して、データ処理装置1との間でデータのやり取りを行う。すなわち、S/P&P/S・インターフェース・シーケンサ13は、ページバッファ15から送られてきたパラレルデータをシリアルデータに変換して、データ処理装置1のシリアルインターフェース回路6へ送出する。また、S/P&P/S・インターフェース・シーケンサ13は、データ処理装置1のシリアルインターフェース回路6から送られてきたシリアルデータをパラレルデータに変換して、ページバッファ15へ送出する。

【0026】このS/P&P/S・インターフェース・シーケンサ13とデータ処理装置1との間のシリアルデータの伝送は、第1のデータ線SCLKによってデータ処理装置1から送られてくるクロック信号によって同期を取りながら、第3のデータ線DIOによって行われる。このとき、第3のデータ線DIOによってやり取りされるシリアルデータのデータ種別は、第2のデータ線Stateによって伝送されるステータス信号によって判別される。ここで、シリアルデータの種別には、例えば、フラッシュメモリ12に書き込むべきデータ、フラッシュメモリ12から読み出されたデータ、又はこのメモリカード2の動作を制御するための制御データ等がある。なお、ステータス信号は、メモリカード2の状態を示すためにも使用される。ステータス信号によって示されるメモリカード2の状態には、例えば、メモリカード2が何らかの処理の最中でデータ処理装置1からのデータ入力を受け付けない状態や、メモリカード2の側での処理が終了してデータ処理装置1からのデータ入力を待っている状態等がある。

【0027】また、S/P&P/S・インターフェース・シーケンサ13は、データ処理装置1から送られてきたデータがメモリカード2の動作を制御するための制御データである場合には、当該制御データをコマンドジェネレータ17に送出する。

【0028】コマンドジェネレータ17は、データ処理装置1からS/P&P/S・インターフェース・シーケンサ13を介して送られてきた制御データに基づいて、フラッシュメモリ12へのアクセスを制御する制御コマンドを生成し、当該制御コマンドをフラッシュメモリ・インターフェース・シーケンサ14へ送出する。フラッシュメモリ・インターフェース・シーケンサ14は、後述するように、この制御コマンドに基づいて、フラッシュメモリ12にデータを書き込んだり、フラッシュメモリ12からデータを読み出したりする。

【0029】なお、このコマンドジェネレータ17には、誤消去防止スイッチ20が接続されている。そして、この誤消去防止スイッチ20がオンになっているときには、フラッシュメモリ12に書かれているデータを消去するように指示する制御データがデータ処理装置1から送られてきたとしても、コマンドジェネレータ17は、フラッシュメモリ12に書かれているデータを消去するような制御コマンドを生成しない。すなわち、このメモリカード2は、誤消去防止スイッチ20によって、フラッシュメモリ12に保存されているデータの消去が行えない状態と、フラッシュメモリ12に保存されているデータの消去が行える状態とを切り換えることが可能となっている。

【0030】S/P&P/S・インターフェース・シーケンサ13とフラッシュメモリ・インターフェース・シーケンサ14との間に配されたページバッファ15は、いわゆるバッファメモリであり、S/P&P/S・インターフェース・シーケンサ13とフラッシュメモリ・インターフェース・シーケンサ14との間でやり取りされるデータを一時的に記憶する。

【0031】すなわち、S/P&P/S・インターフェース・シーケンサ13からフラッシュメモリ・インターフェース・シーケンサ14へ送られるデータは、先ず、S/P&P/S・インターフェース・シーケンサ13からページバッファ15に送られて、このページバッファ15によって一時的に記憶される。このとき、ページバッファ15に記憶されたデータは、エラー訂正回路16によってエラー訂正符号が付けられる。そして、エラー訂正符号が付けられたデータは、ページバッファ15から所定のページ単位毎（例えば1ページ=512バイトとされる。）に、フラッシュメモリ・インターフェース・シーケンサ14へと送られる。

【0032】或いは、フラッシュメモリ・インターフェース・シーケンサ14からS/P&P/S・インターフェース・シーケンサ13へ送られるデータは、先ず、フラッシュメモリ・インターフェース・シーケンサ14からページバッファ15に送られて、このページバッファ15によって一時的に記憶される。このとき、ページバッファ15に記憶されたデータは、エラー訂正回路16によってエラー訂正処理が施される。そして、エラー訂正処理が施されたデータは、ページバッファ15から所定のページ単位毎に、S/P&P/S・インターフェース・シーケンサ13へと送られる。

【0033】フラッシュメモリ・インターフェース・シーケンサ14は、コマンドジェネレータ17からの制御コマンドに基づいて、フラッシュメモリ12へのデータの書き込みや、フラッシュメモリ12からのデータの読み出し等を行う。すなわち、フラッシュメモリ・インターフェース・シーケンサ14は、コマンドジェネレータ17からの制御コマンドに基づいて、フラッシュメモリ

12からデータを読み出して、当該データを上述のようにページバッファ15を介して、S/P&P/S・インターフェース・シーケンサ13へと送出する。或いは、フラッシュメモリ・インターフェース・シーケンサ14は、コマンドジェネレータ17からの制御コマンドに基づいて、S/P&P/S・インターフェース・シーケンサ13からのデータを、上述のようにページバッファ15を介して受け取り、当該データをフラッシュメモリ12に書き込む。

【0034】コンフィグレーションROM18には、このメモ리카ード2のバージョン情報や各種属性情報等が格納されている。コンフィグレーションROM18に格納された情報は、必要に応じて、S/P&P/S・インターフェース・シーケンサ13を介してコマンドジェネレータ17によって読み出されて使用される。すなわち、コマンドジェネレータ17は、必要に応じて、コンフィグレーションROM18に格納されている情報を読み出し、この情報に基づいてメモ리카ード2に関する各種設定を行う。

【0035】以上のようなメモ리카ード2に対して、フラッシュメモリ12に書き込まれるデータが、上述した3本のデータ線SCLK, State, DIOを介して、データ処理装置1からシリアルデータとして送られてくると、まず、S/P&P/S・インターフェース・シーケンサ13は、当該シリアルデータをパラレルデータに変換し、当該パラレルデータをページバッファ15へ送出する。ページバッファ15は、S/P&P/S・インターフェース・シーケンサ13から送られてきたデータを一時的に記憶する。このとき、ページバッファ15に記憶されたデータには、エラー訂正回路16によってエラー訂正符号が付けられる。そして、エラー訂正符号が付けられたデータは、所定のページ単位毎にフラッシュメモリ・インターフェース・シーケンサ14に送出される。そして、フラッシュメモリ・インターフェース・シーケンサ14は、ページバッファ15から送られてきたデータを、コマンドジェネレータ17からの制御コマンドに基づいて、フラッシュメモリ12に書き込む。以上の処理により、データ処理装置1から送られてきたデータが、フラッシュメモリ12に書き込まれる。

【0036】また、以上のようなメモ리카ード2からデータを読み出す際は、まず、コマンドジェネレータ17からの制御コマンドに基づいて、フラッシュメモリ・インターフェース・シーケンサ14によって、フラッシュメモリ12からデータが読み出される。そして、フラッシュメモリ・インターフェース・シーケンサ14は、フラッシュメモリ12から読み出したデータをページバッファ15に送出する。ページバッファ15は、フラッシュメモリ・インターフェース・シーケンサ14から送られてきたデータを一時的に記憶する。このとき、ページバッファ15に記憶されたデータには、エラー訂正回路

16によってエラー訂正処理が施される。そして、エラー訂正処理が施されたデータは、所定のページ単位毎にS/P&P/S・インターフェース・シーケンサ13に送出される。そして、S/P&P/S・インターフェース・シーケンサ13は、ページバッファ15から送られてきたデータを、シリアルデータに変換した上で、上述した3本のデータ線SCLK, State, DIOを介して、データ処理装置1へと送出する。以上の処理により、フラッシュメモリ12から読み出されたデータが、データ処理装置1へと送出される。

【0037】なお、データの書き込みや読み出しを行う際は、フラッシュメモリ12に書き込まれるデータやフラッシュメモリ12から読み出されたデータのやり取りが行われるだけでなく、そのやり取りを制御するための制御データも、データ処理装置1からメモ리카ード2のS/P&P/S・インターフェース・シーケンサ13へ送られる。この制御データは、S/P&P/S・インターフェース・シーケンサ13からコマンドジェネレータ17に送られる。そして、コマンドジェネレータ17は、S/P&P/S・インターフェース・シーケンサ13から送られてきた制御データに基づいて、フラッシュメモリ12へのアクセスを制御する制御コマンドを生成する。そして、この制御コマンドは、フラッシュメモリ・インターフェース・シーケンサ14に送られ、フラッシュメモリ・インターフェース・シーケンサ14は、この制御コマンドに基づいてフラッシュメモリ12にアクセスして、データの書き込みやデータの読み出しを行う。

【0038】なお、メモ리카ード2は、上述した3本のデータ線SCLK, State, DIOを備えるだけでなく、その他に、電圧供給用の配線や、通常は使用しないリザーブの配線等を備えていてもよい。例えば、図2並びに後掲する図3では、上述した3本のデータ線SCLK, State, DIOの他に、4本の電源用の配線VSS1, VSS2, VCC, INTと、3本のリザーブの配線RSV1, RSV2, RSV3とをメモ리카ード2に設けた例を挙げている。

【0039】3. メモ리카ードの外観

つぎに、以上のようなメモ리카ード2の具体的な外形について、図3を参照して説明する。

【0040】メモ리카ード2は、合成樹脂等からなり平面形状が長方形とされる薄肉のカード状のケース21に、上述したコントローラ11やフラッシュメモリ12等が内蔵されてなる。そして、このメモ리카ード2は、当該メモ리카ード2を装着する装着機構を備えたデータ処理装置1に装着されて使用される。

【0041】このメモ리카ード2のケース21の前端部には、斜めに切り欠かれた切り欠き部22が形成されており、更に当該切り欠き部22が形成された部分に、10個の凹状部23が形成されている。そして、これらの

凹状部23の内部には、メモリカード2がデータ処理装置1の装着装置に装着されたときに、データ処理装置1の接続端子に接続される外部接続用端子が、それぞれ配されている。すなわち、このメモリカード2は、外部接続用端子として10本の端子24a、24b、24c、24d、24e、24f、24g、24h、24i、24jを備えている。これらの外部接続用端子の内訳は、3本のデータ線用の端子24b、24d、24h、4本の電源用端子24a、24f、24i、24j、及び3本のリザーブ端子24c、24e、24gである。

【0042】また、このメモリカード2のケース21の上面には、誤消去防止部材25が取り付けられている。誤消去防止部材25は、ケース21の内部に収納された上記誤消去防止スイッチ20に係合されており、この誤消去防止部材25をスライド操作することにより、誤消去防止スイッチ20のオン/オフの切り換えを行えるようになっている。

【0043】このメモリカード2には、データ処理装置1の装着装置に装着された際にメモリカード2がデータ処理装置1から脱落しないようにするため、ケース20の側面の一方に円弧状の第1のロック用切欠部26が形成され、ケース20の側面の他方に矩形状の第2のロック用切欠部27が形成されている。そして、このメモリカード2がデータ処理装置1の装着装置に装着されると、メモリカード2が脱落しないように、これらのロック用切欠部26、27が、データ処理装置1の装着装置に係合される。

【0044】なお、図3に示したメモリカード2は、本発明が適用される外部記憶装置の一例に過ぎない。すなわち、本発明は、外部記憶装置の外形に依存することなく、どんな外形の外部記憶装置にも適用可能である。

【0045】4. 記憶領域の構造

つぎに、以上のようなメモリカード2に搭載されるフラッシュメモリ12の記憶領域の構造について説明する。

【0046】このフラッシュメモリ12の記憶領域は、図4(a)に示すように、データ消去の単位となる複数のブロックに分割されてなる。なお、これらのブロックには、このメモリカード2が起動されたときにデータ処理装置1によって最初に読み込まれるデータであるブートデータが格納されるブートブロックと、任意のデータが書き込まれるデータブロックとがある。各ブロックには、それぞれ固有の物理アドレスが付けられている。これらのブロックは、データ消去の単位であると同時に、ファイル管理上の最小単位でもある。すなわち、ファイルは1つ又は複数のブロックに格納され、1つのブロックを複数のファイルで利用することはできない。

【0047】そして、各ブロックは、「1」又は「0」を示す2つの状態を取りうる複数のビットからなり、初期状態では、全てのビットが「1」とされており、ビット単位での変更は「1」から「0」へだけが可能となっ

ている。すなわち、「1」及び「0」からなるデータを書き込む際、「1」については該当するビットをそのまま保持し、「0」については該当するビットを「1」から「0」に変更する。

【0048】そして、一度書き込んだデータを消去する際は、ブロック単位で一括して初期化処理を行い、当該ブロックの全ビットを「1」とする。これにより、当該ブロックに書き込まれたデータが一括して消去され、そのブロックは再びデータの書き込みが可能な状態となる。

【0049】なお、「0」から「1」への変更を行うには、ブロック単位で一括して初期化処理を行い、当該ブロックの全ビットを「1」にする必要があるが、「1」から「0」への変更は、ブロック単位で一括して初期化処理を行わなくて可能である。以下の説明では、ブロック単位で一括して初期化処理を行うことなく「1」から「0」へ変更することを、オーバーライトと称する。

【0050】なお、本発明は、上述のように各ビットが2つの状態だけを取りうるフラッシュメモリ（いわゆる2値型のフラッシュメモリ）だけでなく、各ビットが3つ以上の状態を取りうるフラッシュメモリ（いわゆる多値型のフラッシュメモリ）にも適用可能である。

【0051】上記フラッシュメモリ12の各ブロックは、図4(b)に示すように、データの書き込みや読み出しの単位となる複数のページから構成される。すなわち、このフラッシュメモリ12にデータを書き込む際は、上述したように、ページ単位にてページバッファ15から送られてきたデータが、フラッシュメモリ・インターフェース・シーケンサ14によってページ単位にてフラッシュメモリ12に書き込まれる。また、このフラッシュメモリ12からデータを読み出す際は、フラッシュメモリ・インターフェース・シーケンサ14によってページ単位毎にデータが読み出されて、ページバッファ15へと送られる。

【0052】各ページは、データエリアと、冗長エリアとを有している。データエリアは、任意のデータが書き込まれる領域である。冗長エリアは、データエリアに書き込まれるデータの管理に必要な情報が格納される領域である。

【0053】具体的には、図4(c)に示すように、ブロックの先頭ページの冗長エリアには、当該ブロックを管理するために必要な情報として、いわゆる分散管理情報が格納される。また、ブロックの2ページ目以降の各ページの冗長エリアにも、予備の分散管理情報として、先頭ページの冗長エリアに格納された分散管理情報と同じものが格納される。ただし、最終ページの冗長エリアには、分散管理情報ではなく、分散管理情報だけでは管理しきれない追加情報として、いわゆる追加管理情報が格納される。

【0054】このように、このフラッシュメモリ12で

は、各ブロック内の冗長エリアに分散管理情報が格納される。分散管理情報は、当該分散管理情報が格納されたブロックを管理するための情報である。この分散管理情報により、例えば、当該ブロックがファイルの先頭となるブロックであるか否かについての情報や、複数のブロックからファイルが構成される場合にはそれらのブロックの繋がりを示す情報等を得ることができる。なお、この分散管理情報については、後で詳細に説明する。

【0055】そして、このメモ리카ード2では、各ブロックの分散管理情報を集めることにより、フラッシュメモリ全体を管理するための情報として、いわゆる集合管理情報を作成して、この集合管理情報をファイルとしてフラッシュメモリ12に格納しておくようにする。

【0056】そして、通常は、集合管理情報によって、各ブロックにアクセスするために必要な情報を得るようにする。すなわち、データ処理装置1とメモ리카ード2との間でデータのやり取りを行う際、データ処理装置1は、集合管理情報をメモ리카ード2から読み出して内部メモリ4に管理テーブルを作成し、この管理テーブルに基づいてメモ리카ード2にアクセスする。これにより、データアクセスの都度、個々のブロックに格納された分散管理情報にアクセスするような必要がなくなり、より高速なデータアクセスが可能となる。

【0057】5. 分散管理情報

つぎに、分散管理情報について詳細に説明する。

【0058】分散管理情報は、当該分散管理情報が格納されたブロックを管理するための情報であり、16バイトの冗長エリアに書き込まれてなる。具体的には、図5に示すように、1バイトの可/不可フラグと、1バイトのブロックフラグと、4ビットの最終フラグと、4ビットの参照フラグと、1バイトの管理フラグと、2バイトの論理アドレスと、2バイトの連結アドレスと、3バイトのリザーブ領域と、2バイトの分散管理情報用エラー訂正符号と、3バイトのデータ用エラー訂正符号とからなる。

【0059】可/不可フラグは、ブロックが使用可能状態か使用不可能状態かを示すフラグであり、具体的には、「使用可」と「使用不可」の2つの状態を示す。

「使用可」は、当該ブロックが使用可能な状態を示し、「使用不可」は、当該ブロックが使用不可能な状態であることを示す。例えば、ブロック内に回復不能なエラーが生じたようなときに、この可/不可フラグが「使用不可」に設定され、当該ブロックが使用不可とされる。

【0060】ブロックフラグは、ブロックの状態を示すフラグであり、具体的には、「未使用」「先頭使用」「使用」「未消去」の4つの状態を示す。「未使用」は、当該ブロックが未使用又は消去済みで、初期状態（全ビットが「1」の状態）とされており、直ぐにデータの書き込みが可能な状態を示す。「先頭使用」は、当該ブロックがファイルの先頭で使用されている状態を示

す。なお、ブートデータが格納されたブートブロックにおいて、ブロックフラグは「先頭使用」とされる。「使用」は、当該ブロックがファイルの先頭以外で使用されている状態を示す。ブロックフラグが「使用」のとき、当該ブロックは、他のブロックから連結されていることとなる。「未消去」は、当該ブロックに書かれていたデータが無効となった状態を示す。例えば、データの消去を行うときに、取りあえずブロックフラグを「未消去」にしておき、処理時間に余裕があるときに、ブロックフラグが「未消去」になっているブロックを消去するようにする。これにより、消去処理をより効率良く行うことが可能となる。

【0061】最終フラグは、ファイルが終わっているか否かを示すフラグであり、具体的には、「ブロック連続」「ブロック最終」の2つの状態を示す。「ブロック連続」は、次のブロックへの連結があることを示す。すなわち、「ブロック連続」は、当該ブロックに格納されたファイルにはまだ続きがあり、当該ファイルが他のブロックに続いていることを示す。「ブロック最終」は、最終ブロックであることを示す。すなわち、「ブロック最終」は、当該ブロックに格納されたファイルが、このブロックで終了していることを示す。

【0062】参照フラグは、追加管理情報の参照を指定するためのフラグであり、具体的には、「参照情報なし」「参照情報あり」の2つの状態を示す。「参照情報なし」は、ブロックの最終ページの冗長領域に、有効な追加管理情報が存在しないことを示す。「参照情報あり」は、ブロックの最終ページの冗長領域に、有効な追加管理情報が存在していることを示す。

【0063】管理フラグは、ブロックの属性等を示すフラグである。例えば、この管理フラグによって、当該ブロックが読み出し専用ブロックか、或いは書き込みも可能なブロックであるかが示される。また、例えば、この管理フラグによって、当該ブロックがブートブロックであるか、或いはデータブロックであるかが示される。

【0064】論理アドレスは、文字通りそのブロックの論理アドレスを示す。この論理アドレスの値は、データの書き換えを行うときなどに必要に応じて更新される。なお、論理アドレスの値は、正常に処理が行われている限り、同じ論理アドレスの値を同時に複数のブロックが持つことがないように設定される。

【0065】ところで、フラッシュメモリの場合、同一ブロック内でデータを書き換えるには、上述したように、先ずブロック消去を行う必要がある。しかしながら、保証されている消去可能回数には上限があり、ブロック消去の回数は出来るだけ少なくすることが要求される。そこで、ブロックのデータを更新する際は、同一のブロックを使って新たなデータに書き換えるのではなく、他のブロックに新たなデータを書き込むようにする。このとき、先にデータが格納されていたブロック

は、当該ブロックに格納されていたデータが無効になったことを示すように、ブロックフラグを「未消去」にする。そして、このメモ리카ード2では、このようにデータを更新した場合でも、当該データが格納されているブロックを示すアドレスが同じとなるように、各ブロックに対して予め設定されている物理アドレスとは別に、動的に変更が可能な論理アドレスを各ブロックに割り当て、この論理アドレスでデータが格納されているブロックを表すようにする。

【0066】連結アドレスは、当該ブロックに連結するブロックの論理アドレスを示す。すなわち、ブロックに格納されたファイルにはまだ続きがあり、当該ファイルが他のブロックに続いている場合、連結アドレスには、そのファイルの続きが格納された次のブロックの論理アドレスの値が設定される。

【0067】分散管理情報用エラー訂正符号は、分散管理情報のうち、管理フラグ、論理アドレス、連結アドレス及びリザーブ領域に書き込まれデータを対象としたエラー訂正符号である。なお、可/不可フラグ、ブロックフラグ、最終フラグ及び参照フラグは、分散管理情報用エラー訂正符号によるエラー訂正の対象となっていない。したがって、可/不可フラグ、ブロックフラグ、最終フラグ及び参照フラグは、分散管理情報用エラー訂正符号を更新することなく書き換えることが可能となっている。

【0068】データ用エラー訂正符号は、当該データ用エラー訂正符号が格納されているページのデータエリアに書き込まれたデータを対象としたエラー訂正符号である。

【0069】なお、分散管理情報用エラー訂正符号やデータ用エラー訂正符号は、メモ리카ード2の内部に配されたエラー訂正回路16によって使用される。したがって、これらのエラー訂正符号を用いてのエラー訂正は、データ処理装置1に依存することなく、メモ리카ード2に依存した任意の手法を使用することができる。

【0070】6. 追加管理情報
つぎに、追加管理情報について詳細に説明する。

【0071】追加管理情報は、ブロックの最終ページの16バイトの冗長エリアに格納される情報であり、分散管理情報だけでは管理しきれない追加情報を含んでいる。

【0072】具体的には、追加管理情報は、図6に示すように、1バイトの可/不可フラグと、1バイトのブロックフラグと、4ビットの最終フラグと、4ビットの参照フラグと、1バイトの識別番号と、2バイトの有効データサイズと、5バイトのリザーブ領域と、2バイトの追加管理情報用エラー訂正符号と、3バイトのデータ用エラー訂正符号とからなる。

【0073】ここで、可/不可フラグ、ブロックフラグ、最終フラグ、参照フラグ、リザーブ領域及びデータ

用エラー訂正符号については、分散管理情報の場合と同様である。また、追加管理情報用エラー訂正符号は、分散管理情報における分散管理情報用エラー訂正符号に相当するものであり、追加管理情報のうち、識別番号、有効データサイズ及びリザーブ領域に書き込まれデータを対象としたエラー訂正符号である。

【0074】そして、識別番号及び有効データサイズとが、分散管理情報だけでは管理しきれない追加情報として、追加管理情報に含まれている。

【0075】識別番号は、エラー処理用の情報であり、ブロックのデータを書き換える度に、この識別番号の値がインクリメントされる。この識別番号は、何らかのエラーが発生して、同じ論理アドレスを持つブロックが複数存在するようになってしまった場合に、それらのブロックに書き込まれたデータの新旧を識別するために使用される。なお、識別番号には1バイトの領域が使用され、その値の範囲は「0」から「255」までであり、その初期値は「0」とされる。なお、識別番号が「255」を越えたときには「0」に戻される。そして、同じ論理アドレスを持つデータブロックが複数存在する場合には、この識別番号の値が小さい方のデータブロックを有効とする。ただし、ブートブロックについては、ブートブロックの予備がある場合、正常時にはそれらのブートブロックの識別番号は同じ値とされる。何らかの異常により、それらのブートブロックの識別番号が異なるような状態となった場合には、識別番号の値が大きい方のブートブロックを有効とする。

【0076】また、有効データサイズは、ブロック内の有効なデータのサイズを示す。すなわち、当該ブロックのデータエリアに空きがある場合、有効データサイズには、当該データエリアに書き込まれたデータのサイズを示す値が設定される。このとき、分散管理情報の参照フラグは「参照情報あり」に設定される。なお、ブロックのデータエリアに空きがない場合、有効データサイズには、当該データエリアに空きがないことを示す値として、「0xffff」が設定される。

【0077】なお、以上のような分散管理情報及び追加管理情報は、ブロック内のデータが更新される毎に、常に最新情報となるように更新される。

【0078】7. 集合管理情報
つぎに、集合管理情報について詳細に説明する。

【0079】集合管理情報は、上述したように、各ブロックの分散管理情報を集めて作成されてなる情報であり、ファイルとしてフラッシュメモリ12に格納される。すなわち、図7に示すように、各ブロックの分散管理情報から、全ブロックをまとめて管理するための情報である集合管理情報のファイルが作成され、この集合管理情報が所定のブロックのデータエリアに格納される。なお、集合管理情報は、1つのブロックに格納されるものであっても、複数のブロックにわたって格納されるも

のであってもよい。そして、データ処理装置1は、通常は、この集合管理情報によって、各ブロックにアクセスするために必要な情報を得るようにする。

【0080】すなわち、メモ리카ード2に有効な集合管理情報がファイルとして格納されている場合、データ処理装置1は、その集合管理情報のファイルを読み出して内部メモリ4に展開し、メモ리카ード2を管理するための管理テーブルを作成する。なお、集合管理情報のファイルの先頭が格納されているブロックの物理アドレスは、ブートデータに含まれており、データ処理装置1は、この物理アドレスに基づいて集合管理情報のファイルにアクセスする。

【0081】この集合管理情報は、図8に示すように、この集合管理情報のヘッダーと、各ブロックの状態を示すビットマップテーブルと、ブロックにアクセスするときに、指定された論理アドレスから物理アドレスへの変換を行うための変換テーブルと、あるブロックの次のブロックを示す連結テーブルとを有する。

【0082】ビットマップテーブルには、各ブロックの分散管理情報から抽出された、可/不可フラグ、ブロックフラグ、最終フラグ、参照フラグ及び管理フラグ等の情報が格納される。

【0083】変換テーブルは、図9に示すように、論理アドレスに対応する物理アドレスが記述されたテーブルであり、物理アドレスが格納される領域は、1エントリあたり2バイトとされる。この変換テーブルを分散管理情報から作成するときは、対象となるブロックの分散管理情報に書かれている論理アドレスを調べ、テーブルの対応位置にそのブロックの物理アドレスを登録する。なお、論理アドレスを使用していない場合、対応する物理アドレスには「0xffff」を設定しておく。

【0084】連結テーブルは、図10に示すように、論理アドレスに対応する連結アドレスが記述されたテーブルであり、連結アドレスが格納される領域は、1エントリあたり2バイトとされる。この連結テーブルを分散管理情報から作成するときは、対象となるブロックの分散管理情報に書かれている連結アドレスを調べ、テーブルの対応位置にそのブロックの連結アドレスを登録する。

【0085】8.メモ리카ード起動時の手順
つぎに、メモ리카ード2の起動時の手順について、図11のフローチャートを参照して説明する。

【0086】このメモ리카ード2を起動する際は、図11に示すように、まず、ステップS1において、データ処理装置1は、メモ리카ード2のブートブロックからブートデータを読み込む。次に、ステップS2へ進む。

【0087】ステップS2において、データ処理装置1は、ブートブロックからのブートデータの読み込みが正常に行われたかを確認する。正常に読み込まれたならば、ステップS3へ進む、正常に読み込まれなかったならば、ステップS8へ進む。

【0088】ステップ3において、データ処理装置1は、読み込んだブートデータに基づいて、メモ리카ード2が当該データ処理装置1に対応しているか否かを判別する。対応したメモ리카ードであるならば、ステップS4へ進む、対応していないメモ리카ードならば、ステップS8へ進む。

【0089】ステップS4において、データ処理装置1は、メモ리카ード2から集合管理情報を読み込む。なお、集合管理情報が格納されているブロックの物理アドレスは、ブートデータの中で指定されている。次に、ステップS5へ進む。

【0090】ステップS5において、データ処理装置1は、有効な集合管理情報の読み込みが正常に行われたかを確認する。正常に読み込んだならば、ステップS6へ進む、正常に読み込めなかったならば、ステップS7へ進む。

【0091】ステップS6において、データ処理装置1は、読み込んだ集合管理情報を内部メモリ4に展開し、メモ리카ード2を管理するための管理テーブルを作成する。以上の処理で、メモ리카ード2の起動時の初期処理が完了し、メモ리카ード2の使用が可能となる。

【0092】また、ステップS5で有効な集合管理情報が正常に読み込めなかったと判断された場合は、上述したようにステップS7へ進む。ステップS7において、データ処理装置1は、各ブロックの分散管理情報を読み出して、集合管理情報を再構築する。そして、その集合管理情報を内部メモリ4に展開し、メモ리카ード2を管理するための管理テーブルを作成する。以上の処理で、メモ리카ード2の起動時の初期処理が完了し、メモ리카ード2の使用が可能となる。

【0093】一方、ステップS2でブートデータ読み込み時にエラーが生じたと判断された場合、及び、ステップS3でメモ리카ード2がデータ処理装置1に対応していないと判断された場合は、上述したようにステップS8へ進む。

【0094】ステップS8に進むのは、メモ리카ード2を使用できないときである。そこで、ステップS8において、データ処理装置1は、例えば、利用不可のメッセージを表示するなどの所定のエラー処理を行い、メモ리카ード2の起動処理を終了する。

【0095】9.データ更新処理時の集合管理情報の取り扱い

データ処理装置1は、メモ리카ード2へデータを書き込む処理や、メモ리카ード2からデータを消去する処理

(以下、これらの処理をまとめて「データ更新処理」と称する。)を行う毎に、内部メモリ4に保持している管理テーブルを、メモ리카ード2の実際の状態と整合するように(すなわち、分散管理情報の内容と整合するように)、随時更新していく。一方、メモ리카ード2にファイルとして格納されている集合管理情報は、データ更新

処理毎に更新されるのではなく、適当なタイミングにて、その変更内容が一括して更新される。

【0096】一般にフラッシュメモリ2の書き換え可能回数には上限があるが、集合管理情報の書き換えをある程度まとめて一括して行うようにすることで、集合管理情報が格納されているブロックの書き換え回数を削減することができ、メモリカード2の長寿命化を図ることができる。

【0097】ただし、データ更新処理を行う際は、メモリカード2に格納されている集合管理情報のファイルを無効にしてから行う。これは、分散管理情報と集合管理情報の一貫性を損なわないためである。データ更新処理時には、処理の対象となるブロックの分散管理情報は同時に更新されるが、集合管理情報の内容は同時には更新されないため、分散管理情報と集合管理情報が一致しない状態となる。そこで、このような状態のときには、メモリカード2に格納されている集合管理情報のファイルを無効にしておく。

【0098】具体的には、データ更新処理時には、図12に示すように、まず、ステップS11において、データ処理装置1は、メモリカード2に格納されている集合管理情報が有効であるか無効であるかを判別する。そして、集合管理情報が既に無効となっていたならば、そのままデータ更新処理に移行する。一方、集合管理情報が有効であれば、ステップS12へ進む。

【0099】ステップS12において、データ処理装置1は、集合管理情報を無効にする。具体的には、集合管理情報のファイルが格納されているブロックのブロックフラグを「未消去」にするか、或いは当該ブロックに対して消去処理を施してデータを消去する。そして、このように集合管理情報を無効にした上で、データ更新処理に移行する。

【0100】なお、メモリカード2に格納されている集合管理情報のファイルは、分散管理情報と集合管理情報の一貫性を損なわないために、データ更新処理時に無効とされるが、データ処理装置1の内部メモリ4に保持されている管理テーブルの内容は、常に最新の状態となるように随時更新される。そして、データ処理装置1は、通常は、この管理テーブルに基づいて各ブロックを管理する。

【0101】また、データ更新処理時に無効とされた集合管理情報は、適当なタイミングで、改めてメモリカード2に書き込まれて、再び有効とされる。なお、ここでの適当なタイミングとは、例えば、メモリカード2の使用を終了して電源を落とすときや、メモリカード2へのアクセスが所定時間以上なされなかったときや、データの書き換えが所定回数以上行われたときなどである。

【0102】具体的には、例えば、メモリカード2の使用を終了して電源を落とす前に、図13に示すような終了処理を行い、集合管理情報を有効なものとする。

【0103】この終了処理では、まず、ステップS21において、データ処理装置1は、メモリカード2に格納されている集合管理情報が有効であるか無効であるかを判別する。そして、集合管理情報が有効であれば、そのまま処理を終了する。一方、集合管理情報が無効であれば、ステップS22へ進む。

【0104】ステップS22において、データ処理装置1は、集合管理情報のファイルが格納されているブロックに対して、消去処理が施されているか否かを判別する。消去処理が施されていないならば、ステップS23へ進む。消去処理が施されていれば、ステップS24へ進む。

【0105】ステップS23において、データ処理装置1は、集合管理情報のファイルが格納されているブロックに対して消去処理を施す。その後、ステップS24へ進む。

【0106】ステップS24において、データ処理装置1は、メモリカード2に集合管理情報を書き込む。このとき、データ処理装置1は、内部メモリ4に保持している管理テーブルの内容に基づいて新しい集合管理情報のファイルを作成し、その新しい集合管理情報のファイルをメモリカード2に書き込む。これにより、メモリカード2の最新の状態を示す有効な集合管理情報が、メモリカード2に格納されたこととなる。

【0107】以上で終了処理が完了し、メモリカード2に有効な集合管理情報が格納された状態となる。

【0108】10. 新規ファイルの書き込み
つぎに、メモリカード2に新規なファイルを書き込む際の処理手順について説明する。メモリカード2にファイルを書き込む際の処理手順は、ファイルサイズが予め分かっている場合と、分かっている場合とで異なる。

【0109】10-1 ファイルサイズが予め分かっている場合

予めファイルのサイズが分かっている場合は、当該ファイルのデータを新規なブロックに書き込む毎に、データが当該ブロックに納まるかどうかを判断する。そして、データがブロック内に納まりきらない場合には、次に続くブロックの論理アドレスを確保しておき、データをデータエリアに書き込むとともに、次に続くブロックの論理アドレスを連結アドレスとして分散管理情報を書き込む。このとき、最終フラグは「ブロック連続」に設定しておく。一方、データがブロック内に納まる場合には、データの端数部分、すなわちデータエリアの空き領域は「0xffff」としておく。このとき、最終フラグは「ブロック最終」に設定しておく。追加管理情報に有効データサイズを書き込んでおく。

【0110】つぎに、以上のように予めサイズが分かっているファイルをメモリカード2に書き込む際の手順について、図14に示すフローチャートを参照して、詳細に説明する。なお、図14に示すフローチャート、並び

に後掲する図15及び図17に示すフローチャートでは、メモリカード2の誤消去防止スイッチ20のチェックや、何らかのエラーが発生したときの処理等については省略している。

【0111】予めサイズが分かっているファイルをメモリカード2に書き込む際は、先ず、ステップS31において、データ処理装置1は、メモリカード2に書き込む実データと、当該実データのヘッダーとを準備する。換言すれば、ステップS31において、データ処理装置1は、メモリカード2のデータエリアに書き込むファイルを準備する。なお、当該ファイルのヘッダーには、ファイルのサイズの情報が含まれる。次に、ステップS32へ進む。

【0112】ステップS32において、データ処理装置1は、最初にファイルが格納されるブロックのブロックフラグを「先頭使用」に設定するとともに、空いている論理アドレスを確保する。次に、ステップS33へ進む。

【0113】ステップS33において、データ処理装置1は、空いている物理アドレスを検索する。次に、ステップS34へ進む。

【0114】ステップS34において、データ処理装置1は、処理の対象となっているブロックにファイルが納まりきるか否かを判別する。ファイルがブロック内に納まりきらず、ファイルに続きがある場合には、ステップ35へ進み、一方、ファイルがブロック内に納まり、ファイルに続きがない場合には、ステップ36へ進む。

【0115】ステップS35において、データ処理装置1は、次に続くブロックの論理アドレスを確保して、この論理アドレスを連結アドレスとして設定する。次に、ステップS37へ進む。

【0116】一方、ステップS36において、データ処理装置1は、最終フラグを「ブロック最終」に設定するとともに、連結アドレスを「0xffff」に設定する。次に、ステップS37へ進む。

【0117】ステップS37において、データ処理装置1は、これまでのステップで設定された情報等に基づいて、処理の対象となっているブロックについての分散管理情報を作成する。次に、ステップS38へ進む。

【0118】ステップS38において、データ処理装置1は、処理の対象となっているブロックに、ページ単位でデータを順次書き込む。ここで、処理の対象となっているブロックにファイルが納まりきらない場合には、1ブロック分のデータがページ単位で書き込まれる。また、処理の対象となっているブロックにファイルが納まりきる場合には、必要なページ分だけページ単位でデータが書き込まれる。なお、このステップS38で、ブロックに書き込まれるのは、新規に書き込むファイルのデータと、ステップS37で作成された分散管理情報とである。次に、ステップS39へ進む。

【0119】ステップS39において、データ処理装置1は、ファイルの全データについて、メモリカード2への書き込みが終了したか否かを判別する。書き込みが終了しておらず、まだデータが残っていれば、ステップS33へ戻って処理を繰り返す。一方、書き込みが終了していれば、ステップ40へ進む。

【0120】ステップS40において、データ処理装置1は、書き込んだデータが、ブロックの途中で終わっているか否かを判別する。そして、データがブロックの途中で終わっていれば、ステップS41へ進む。一方、ブロックの最後までデータが格納されていれば、これで処理を終了する。

【0121】ステップS41において、データ処理装置1は、最終ページの冗長エリアに格納される追加管理情報に有効データサイズを書き込む。すなわち、データ処理装置1は、ファイルの最後の部分が格納されたブロックのデータエリアに書き込まれたデータのサイズを示す値を、当該ブロックの追加管理情報に有効データサイズとして書き込む。

【0122】以上で、予めサイズが分かっているファイルのメモリカード2への書き込みの処理が完了する。

【0123】10-2 ファイルサイズが分かっていない場合

ファイルのサイズが予め分からない場合には、次に続くブロックの論理アドレスを常に確保しておき、データが終了した時点で、最終ブロックの最終フラグをオーバーライトにより設定する。なお、その他の分散管理情報及び追加管理情報に関しては、予めファイルサイズが分かっているときと同様に設定される。

【0124】サイズが予め分かっているファイルをメモリカード2に書き込む際の手順について、図15に示すフローチャートを参照して、詳細に説明する。

【0125】サイズが予め分かっているファイルをメモリカード2に書き込む際は、先ず、ステップS51において、データ処理装置1は、メモリカード2に書き込むファイルの仮のヘッダーを作成する。この段階では、ファイルサイズが不明なので、この仮のヘッダーには、ファイルサイズの情報が含まれていない。次に、ステップS52へ進む。

【0126】ステップS52において、データ処理装置1は、最初にファイルが格納されるブロックのブロックフラグを「先頭使用」に設定するとともに、空いている論理アドレスを確保する。次に、ステップS53へ進む。

【0127】ステップS53において、データ処理装置1は、メモリカード2に書き込むデータを準備する。次に、ステップS54へ進む。

【0128】ステップS54において、データ処理装置1は、メモリカード2に書き込むデータが残っているか否かを判別する。データが終了しておらず、まだデータ

が残っていれば、ステップS55へ進む。一方、データが終了しており、データが残っていなければ、ステップS61へ進む。

【0129】ステップS55において、データ処理装置1は、空いている物理アドレスを検索する。次に、ステップS56へ進む。

【0130】ステップS56において、データ処理装置1は、次に続くブロックの論理アドレスを確保して、この論理アドレスを連結アドレスとして設定する。次に、ステップS57へ進む。

【0131】ステップS57において、データ処理装置1は、これまでのステップで設定された情報等に基づいて、処理の対象となっているブロックについての分散管理情報を作成する。次に、ステップS58へ進む。

【0132】ステップS58において、データ処理装置1は、処理の対象となっているブロックに、ページ単位でデータを順次書き込む。ここで、処理の対象となっているブロックにファイルが納まりきらない場合には、1ブロック分のデータがページ単位で書き込まれる。また、処理の対象となっているブロックにファイルが納まりきる場合には、必要なページ分だけページ単位でデータが書き込まれる。なお、このステップS58でブロックに書き込まれるのは、新規に書き込むファイルのデータと、ステップS57で作成された分散管理情報とである。次に、ステップS59へ進む。

【0133】ステップS59において、データ処理装置1は、書き込んだデータが、ブロックのデータエリアの途中で終わっているか否かを判別する。そして、データエリアの最後に至るまでデータが格納されていれば、ステップS53へ戻って処理を繰り返す。一方、データがデータエリアの途中で終わっていれば、ステップS60へ進む。

【0134】ステップS60において、データ処理装置1は、処理の対象となっているブロックの最終ページの冗長エリアに格納される追加管理情報に、有効データサイズを書き込む。すなわち、データ処理装置1は、ファイルの最後の部分が格納されたブロックのデータエリアに書き込まれたデータのサイズを示す値を、当該ブロックの追加管理情報に有効データサイズとして書き込む。次に、ステップS61へ進む。

【0135】ステップS61において、データ処理装置1は、処理の対象となっているブロックの最終フラグを「ブロック最終」にオーバーライトにより設定する。次に、ステップS62へ進む。

【0136】ステップS62において、データ処理装置1は、ファイルのヘッダーを更新する。すなわち、この段階では、ファイルサイズが明らかとなっているので、ファイルサイズの情報を含むヘッダーを新たに作成して、上述した仮のヘッダーを、ファイルサイズの情報を含む新たなヘッダーに書き換える。

【0137】以上で、予めサイズが分かっていたファイルのメモ리카ード2への書き込みの処理が完了する。

【0138】11. ファイルの更新
つぎに、メモ리카ード2に格納されているファイルを更新する際の処理手順について説明する。

【0139】ファイルの更新時には、データの書き換えの対象となるブロックと同じ論理アドレスを別のブロックに付して、当該ブロックに対して新しいデータを書き込む。このとき、古いデータが書かれているブロックは、ファイルの更新が終了するまで開放せずに保持しておく。これにより、ファイル更新中に障害が発生したとしても、ファイル更新前の状態に復旧することが可能となる。

【0140】このようなファイル更新の手順の具体的な例を図16を参照して説明する。

【0141】図16(a)に示すように、ファイルの先頭が、論理アドレス「1」のブロックに格納され、ファイルの次の部分が、論理アドレス「2」のブロックに格納され、ファイルの次の部分が、論理アドレス「3」のブロックに格納されていたとする。また、論理アドレス「1」のブロックの識別番号は「6」、論理アドレス「2」のブロックの識別番号は「4」、論理アドレス「3」のブロックの識別番号は「1」であったとする。

【0142】そして、このような状態のときに、論理アドレス「2」のブロックのデータを書き換えるとする。このときは、まず、図16(b)に示すように、空いている別にブロックに論理アドレス「2」を割り当て、このブロックに新しいデータを書き込む。ここで、新しいデータを書き込むブロックの識別番号には、古いデータが書き込まれているブロックの識別番号を1インクリメントした値、すなわち「5」を設定する。

【0143】この段階では、同じ論理アドレスを持つブロックが2つ存在していることとなる。そして、これらの2つのブロックのうち、識別番号が大きい方のブロックに格納されているデータが新しい方のデータであり、識別番号が小さい方のブロックに格納されているデータが古い方のデータとなる。

【0144】そして、新しいデータの書き込みが正常に完了したら、次に、図16(c)に示すように、古いデータが書き込まれていたブロックを消去する。なお、このときは、古いデータが書き込まれていたブロックに対して消去処理を施すのではなく、該当するブロックのブロックフラグを「未消去」にするだけにしておき、後から適当なタイミングで、このブロックに対して消去処理を施すようにしてもよい。

【0145】以上のような処理の手順について、図17に示すフローチャートを参照して、詳細に説明する。

【0146】ファイルを更新する際は、まず、ステップS71において、データ処理装置1は、更新の対象とな

るブロックを選択する。次に、ステップS72へ進む。

【0147】ステップS72において、データ処理装置1は、更新の対象となるブロックの識別番号を読み出し、その値を1インクリメントした値を、新しいデータを書き込むブロックの識別番号として設定する。また、新しいデータを書き込むブロックの論理アドレスとして、更新の対象となるブロックの論理アドレスと同じ値を設定する。次に、ステップS73へ進む。

【0148】ステップS73において、データ処理装置1は、ブロックに書き込む新しいデータを準備する。次に、ステップS74に進む。

【0149】ステップS74において、データ処理装置1は、空いている物理アドレスを検索する。次に、ステップS75に進む。

【0150】ステップS75において、データ処理装置1は、データの変更が全て完了しているか否かを判別する。完了していなければ、ステップS76へ進み、完了していれば、ステップS79へ進む。

【0151】ステップS76において、データ処理装置1は、次に続くブロックの論理アドレスを確保して、この論理アドレスを連結アドレスとして設定する。次に、ステップS77へ進む。

【0152】ステップS77において、データ処理装置1は、これまでのステップで設定された情報等に基づいて、新しいデータを書き込むブロックについての分散管理情報を作成する。次に、ステップS78へ進む。

【0153】ステップS78において、データ処理装置1は、ステップS74で検索された物理アドレスのブロックに、ページ単位で新しいデータを順次書き込む。ここで、処理の対象となっているブロックにファイルが納まりきらない場合には、1ブロック分のデータがページ単位で書き込まれる。また、処理の対象となっているブロックにファイルが納まりきる場合には、必要なページ分だけページ単位でデータが書き込まれる。なお、このステップS78で、ブロックに書き込まれるのは、新しいファイルのデータと、ステップS77で作成された分散管理情報とである。そして、このステップS78の後には、ステップS73へ戻って処理を繰り返す。

【0154】一方、ステップS79において、データ処理装置1は、最後に更新の対象となっていたブロックに、連結アドレスが設定されているか否かを判別する。連結アドレスが設定されていない場合は、ステップS80へ進み、連結アドレスが設定されていたならば、ステップS81へ進む。

【0155】ステップS80において、データ処理装置1は、最後に新しいデータを書き込んだブロックの最終フラグを「ブロック最終」に設定する。次に、ステップS82へ進む。

【0156】一方、ステップS81において、データ処理装置1は、最後に新しいデータを書き込んだブロック

の連結アドレスに、最後に更新の対象となっていたブロックに設定されていた連結アドレスの値を設定する。次に、ステップS82へ進む。

【0157】ステップS82において、データ処理装置1は、ファイルのヘッダーを更新する。すなわち、ファイルの更新により、ファイルサイズが変更となっている可能性があるため、新しいファイルサイズの情報を含むヘッダーを新たに作成して、ファイルのヘッダーを更新する。次に、ステップS83へ進む。

【0158】ステップS83において、データ処理装置1は、古いデータが書き込まれていたブロックを消去する。なお、このときは、古いデータが書き込まれていたブロックに対して消去処理を施すのではなく、該当するブロックのブロックフラグを「未消去」にするだけにしておき、後から適当なタイミングで、これらのブロックに対して消去処理を施すようにしてもよい。

【0159】以上でファイルの更新処理が完了する。

【0160】12. エラー検出訂正処理

以上のようなシステムでは、メモ리카ード2に新規ファイルを書き込んでいるときや、メモ리카ード2に格納されているファイルを更新しているときなどに、いきなり電源が遮断されたり、メモ리카ード2がデータ処理装置1から強制的に取り外されたりすると、同じ論理アドレスを持つ複数のブロックが同時に存在するような状態（すなわち論理アドレスエラー）となってしまうたり、或いは、連結アドレスで指し示されたブロックが存在しないような状態（すなわち連結アドレスエラー）となってしまうたりする可能性がある。

【0161】そこで、本発明を適用したシステムでは、集管理情報を構築する際に、論理アドレスエラーや連結アドレスエラーを検出して訂正するエラー検出訂正処理を行うようにする。以下、このエラー検出訂正処理について、詳細に説明する。

【0162】12-1 エラー検出テーブル

本発明を適用したシステムでは、集管理情報を構築する際に、論理アドレスエラーや連結アドレスエラーの検出を行う。そして、連結アドレスエラーを検出するために使用するテーブルとして、エラー検出テーブルを使用する。エラー検出テーブルは、連結アドレスエラーの検出にだけ使用されるテーブルであり、エラー検出訂正処理を行う際に、データ処理装置1の内部メモリ4に一時的に確保される。エラー検出テーブルのために確保されていた領域は、エラー検出訂正処理が終了した後、開放される。

【0163】このエラー検出テーブルは、図18に示すように、1論理アドレスに対して、各ブロックの連結状態を示す1ビットの領域を備えたテーブルである。換言すれば、エラー検出テーブルは、1エントリあたり1ビットとなっており、各エントリは論理アドレスの連結状態を「0」又は「1」で示す。処理の対象となるブロッ

クがN個ある場合、このエラー検出テーブルが占める領域は、N/8バイトとなる。

【0164】このエラー検出テーブルは、集合管理情報を構築しているときと、集合管理情報の構築が終了したときとで、各ブロックの連結状態を示す値の意味合いが異なる。

【0165】集合管理情報を構築している最中において、連結状態を示す値が「0」のとき、そのエントリは、正常な状態であることを示しているか、或いは、当該エントリに対応する論理アドレスが、現在処理の対象となっているブロックまでの間に他のブロックの連結アドレスで指定されていないことを示している。この状態のときは、今後の処理が進むに従って、値が「1」になる可能性があり、連結アドレスエラーであるかどうか不確定な状態である。

【0166】また、集合管理情報を構築している最中において、連結状態を示す値が「1」のとき、そのエントリは、当該エントリに対応する論理アドレスが、現在処理の対象となっているブロックまでの間に他のブロックの連結アドレスとして指定されているが、物理アドレスに対応していない状態を示す。この状態のときは、今後の処理が進むに従って、値が「0」になる可能性があり、連結アドレスエラーであるかどうか不確定な状態である。

【0167】一方、集合管理情報の構築が終了した段階で、連結状態を示す値が「1」のとき、そのエントリは、当該エントリに対応する論理アドレスが連結アドレスとして指定されているのに、対応する物理アドレスが存在しないことを示している。したがって、この状態のときは、連結アドレスエラーである。

【0168】また、集合管理情報の構築が終了した段階で、連結状態を示す値が「0」のとき、そのエントリは、当該エントリに対応する論理アドレスに関する連結が正常な状態となっていることを示している。

【0169】12-2 連結アドレスエラーの検出
つぎに、以上のようなエラー検出テーブルを用いて行われる、連結アドレスエラーの検出について、具体的な例を挙げて説明する。

【0170】例えば、図19に示すように、物理アドレス「10」のブロックについて、その論理アドレスが「1」、その連結アドレスが「3」であり、また、物理アドレス「17」のブロックについて、その論理アドレスが「3」、その連結アドレスが「2」であったとする。また、論理アドレスが「2」のブロックは存在しないとす。

【0171】このとき、集合管理情報の再構築の処理が行われると、集合管理情報の再構築に伴って、図20に示すように、連結アドレスエラーの検出が行われる。

【0172】先ず、初期状態では、図20(a)に示すように、論理アドレス「1」「2」「3」のそれぞれに

ついて、エラー検出テーブルの値は、すべて初期値「0」とする。また、集合管理情報の交換テーブルも、全て初期値「0xffff」とする。このとき、集合管理情報の連結テーブルは、全く値が入っていない状態とする。

【0173】次に、論理アドレス「1」のブロックについての情報を読み込む。これにより、図20(b)に示すように、論理アドレス「1」に対応する交換テーブルの値は、論理アドレス「1」のブロックの物理アドレスの値、すなわち「10」とされる。また、論理アドレス「1」に対応する連結テーブルの値は、論理アドレス「1」のブロックの連結アドレスの値、すなわち「3」とされる。

【0174】次に、論理アドレス「1」に対応する連結テーブルの値が指し示す論理アドレス「3」のエントリを確認する。このとき、論理アドレス「3」のエントリには、物理ブロックが割り当てられていないので、図20(c)に示すように、論理アドレス「3」に対応するエラー検出テーブルの値を「1」とする。また、論理アドレス「3」に対応する交換テーブルの値は、連結元の論理アドレスの値、すなわち「1」とする。

【0175】次に、エラー検出テーブルの値が「1」となっている論理アドレス「3」のブロックについて、その情報を読み込む。このとき、論理アドレス「3」のブロックは存在しており、当該ブロックの情報は正常に読み込むことができる。したがって、図20(d)に示すように、論理アドレス「3」に対応するエラー検出テーブルの値を「0」とする。また、このとき、論理アドレス「3」に対応する交換テーブルの値は、論理アドレス「3」のブロックの物理アドレスの値、すなわち「17」とする。また、論理アドレス「3」に対応する連結テーブルの値は、論理アドレス「3」のブロックの連結アドレスの値、すなわち「2」とする。

【0176】次に、論理アドレス「3」に対応する連結テーブルの値が指し示す論理アドレス「2」のエントリを確認する。このとき、論理アドレス「2」のエントリには、物理ブロックが割り当てられていないので、図20(e)に示すように、論理アドレス「2」に対応するエラー検出テーブルの値を「1」とする。また、論理アドレス「2」に対応する交換テーブルの値は、連結元の論理アドレスの値、すなわち「3」とする。

【0177】次に、エラー検出テーブルの値が「1」となっている論理アドレス「2」のブロックについて、その情報の読み込みを試みる。しかしながら、論理アドレス「2」のブロックは存在していないので、この段階で、連結アドレスエラーであることが判明する。

【0178】12-3 エラー訂正処理
本発明を適用したシステムにおいて、論理アドレスエラーに対するエラー訂正処理は、以下のように行う。

【0179】論理アドレスエラーが生じているときには、同じ論理アドレスを持つブロックをそれぞれ調べ

る。そして、ブロックとして完全なものが1つしかない場合には、完全なブロックを生かして、残りのブロックは無効とする。

【0180】また、同じ論理アドレスを持つ完全なブロック（ブートブロックを除く。）が複数ある場合には、識別番号を比較して値が小さい方のブロックを生かす。なお、一方のブロックの識別番号が「255」であり、他方のブロックの識別番号が「0」の場合には、識別番号が「255」のブロックを生かすようにする。

【0181】なお、通常は、同じ論理アドレスを持つブロックが複数あったとしても、それらの識別番号の差は1である。この条件に当てはまらないような場合には、システムの側で自動的にエラー訂正処理を行うのではなく、手動復旧モードとする。

【0182】また、本発明を適用したシステムにおいて、連結アドレスエラーが生じた場合には、メモ리카ード2を使用するアプリケーションソフトウェアやメモ리카ード2に格納するデータ等に応じた適切なエラー訂正処理を行うようにする。具体的には、例えば、以下に挙げるようなエラー訂正処理を行うようにすればよい。

【0183】すなわち、例えば、最後の連結アドレスが指し示すブロックとして新規ブロックを割り当てる。そして、最後のブロックのデータを正しく読めるページまで読み込んで、そのページまでのデータを新規ブロックにコピーする。このとき、新規ブロックの最終フラグは「ブロック最終」としておく。このようなエラー訂正処理は、対象となるデータが音楽データ等のようにデータの途中で意味があるデータの場合に、特に好適である。

【0184】或いは、例えば、連結アドレスエラーが生じたブロックを含むファイル全体を削除する。このようなエラー訂正処理は、対象となるデータがプログラムや画像データ等のようにデータの途中では意味をなさないデータである場合に、特に好適である。

【0185】12-4 集合管理情報の構築とエラー検出訂正処理

本発明を適用したシステムにおいて、物理アドレスエラーや連結アドレスエラーが発生するのは、データ更新処理の途中に何らかの障害が発生した場合である。そして、本発明を適用したシステムでは、上述したように、データ更新処理に先だってメモ리카ード2に格納されている集合管理情報を無効にするようにしている。したがって、物理アドレスエラーや連結アドレスエラーが発生するときには、集合管理情報が無効となっている。そして、集合管理情報が無効となっているときには、次にメモ리카ード2を起動するときに、全てのブロックの分散管理情報を調べ直して集合管理情報を再構築する処理が必ず行われる。

【0186】そこで、このシステムでは、集合管理情報を再構築する際に、全てのブロックを調べ直すというこ

とに着目し、このときにエラー検出訂正処理を同時に行う。換言すれば、集合管理情報が有効となっているときには、物理アドレスエラーや連結アドレスエラーが発生している可能性はないので、このときにはエラー検出訂正処理を行わない。すなわち、この方法では、集合管理情報の再構築時にだけ、エラー検出訂正処理を行う。これにより、エラー検出訂正処理のためにメモ리카ード2に余分にアクセスする必要がなくなる。その結果、例えば、メモ리카ード2の速やかな起動が可能となる。

【0187】このエラー検出訂正処理は、以下のような手順によって行われる。

【0188】（1）データ処理装置1の内部メモリ4上の変換テーブルを全て「0xffff」で初期化する。また、内部メモリ4上にエラー検出テーブルの領域を確保し、当該エラー検出テーブルを全て「0」で初期化する。

【0189】（2）ブロックの先頭に移動する。

【0190】（3）ブロックから分散管理情報を読み込み、当該分散管理情報を用いてビットマップテーブルを構築する。このとき、可/不可フラグが「使用不可」であるか、或いはブロックフラグが「未使用」又は「未消去」である場合には、ビットマップテーブルの作成が終了したら、次のブロックへ移動して処理を繰り返す。

【0191】（4）ブロックの論理アドレス（以下、論理アドレスAとする。）と、連結アドレス（以下、論理アドレスBとする。）とを調べる。

【0192】（5）論理アドレスAが「0xffff」の場合は、次のブロックへ移動し、（3）へ戻って処理を繰り返す。

【0193】（6）エラー検出テーブルの論理アドレスAの欄を調べる。エラー検出テーブルの論理アドレスAの欄が「1」となっている場合は、「0」に書き換えるとともに、変換テーブルの論理アドレスAの欄に、論理アドレス「A」のブロックの物理アドレスを書き込む。また、エラー検出テーブルの論理アドレスAの欄が「0」となっている場合は、変換テーブルの論理アドレスAの欄を調べる。そして、変換テーブルの論理アドレスAの値が「0xffff」のときは、そこに、論理アドレスAのブロックの物理アドレスを書き込む。

【0194】なお、変換テーブルの論理アドレスAの値として、「0xffff」以外の値が既に入っている場合は、論理アドレスエラーが発生している場合であるので、論理アドレスエラーに対するエラー訂正処理を行う。

【0195】（7）連結テーブルの論理アドレスAの欄に、論理アドレスBの値を記入する。

【0196】（8）最終フラグが「ブロック最終」となっているかを調べる。「ブロック最終」となっていれば、連結アドレスは無効なので、次のブロックに移動し、（3）へ戻って処理を繰り返す。

【0197】（9）論理アドレスBに物理アドレスが対応しているかどうかを、変換テーブルを用いて確認す

る。変換テーブルの論理アドレスBの値が「0xffff」以外の場合は、論理アドレスBに物理アドレスが対応している。一方、変換テーブルの論理アドレスBの値が「0xffff」の場合は、現在のブロックまでの段階では、論理アドレスBに物理アドレスが対応していない。このときは、エラー検出テーブルの論理アドレスBの欄に「1」を書き込むとともに、変換テーブルの論理アドレスBの欄に論理アドレスAの値を記入する。その後、次のブロックに移動し、(3)へ戻って処理を繰り返す。

【0198】なお、変換テーブルの論理アドレスBの値が「0xffff」となっても、ブロックの途中までしか処理を行っていない段階では、論理アドレスBに物理アドレスが本当に対応していないのかは明らかではない。すなわち、論理アドレスBに物理アドレスが本当に対応していない場合と、今後処理を進めて行くに従って、対応するブロックが現れる場合との2通りがあり得る。

【0199】(10)全てのブロックに対して処理を行った後、エラー検出テーブルを参照する。エラー検出テーブルの値が「1」となっている論理アドレスは、物理アドレスが対応していない。すなわち、連結アドレスエラーが発生している。このときは、変換テーブル内に連結元のブロックの論理アドレスが格納されているので、これを用いて元ブロックを特定し、適切なエラー訂正処理を行う。なお、エラー訂正処理を行った後は、エラー検出テーブルの値を「0」にし、該当する変換テーブルの値を「0xffff」にしておく。

【0200】本発明を適用したシステムでは、以上のように、集合管理情報を構築するときにエラー検出訂正処理を行う。以下、このような集合管理情報の構築及びエラー検出訂正処理の具体的な方法について、図21乃至図23に示すフローチャートを参照して、更に詳細に説明する。

【0201】なお、ここでは、変数としてI, A, B, C, D, T (I)を使用し、定数としてNを使用する。変数Iは、物理アドレスが入力される変数であり、変数Aは、論理アドレスが入力される変数であり、変数Bは、連結アドレスが入力される変数であり、変数C, Dは、識別番号の値が入力される変数であり、変数T (I)は、論理アドレス「I」に対応するエラー検出テーブルの値を示す変数である。また、定数Nは、総ブロック数を示す定数である。

【0202】集合管理情報を構築する際は、図21のステップS101において、データ処理装置1は、変換テーブルを初期化し、全ての値を「0xffff」とする。次にステップS102へ進む。

【0203】ステップS102において、データ処理装置1は、エラー検出テーブルを初期化し、全ての値を「0」とする。次にステップS103へ進む。

【0204】ステップS103において、データ処理装置1は、変数Iに「0」を代入する。次にステップS1

04へ進む。

【0205】ステップS104において、データ処理装置1は、物理アドレス「I」のブロックの分散管理情報をメモリカード2から読み込む。次にステップS105へ進む。

【0206】ステップS105において、データ処理装置1は、ステップS104で読み込んだ分散管理情報の可/不可フラグを参照して、物理アドレス「I」のブロックが使用可能か否かを判別する。使用可能であれば、ステップS106へ進み、使用不可能であれば、ステップS129へ進む。

【0207】ステップS106において、データ処理装置1は、ステップS104で読み込んだ分散管理情報のブロックフラグを参照して、物理アドレス「I」のブロックが使用中であるか否かを判別する。具体的には、ブロックフラグが「先頭使用」又は「使用」になっているか否かを判別する。ブロックフラグが「先頭使用」又は「使用」になっており、当該ブロックが使用中であれば、ステップS107へ進む。また、当該ブロックが使用中でなければ、ステップS129へ進む。

【0208】ステップS107において、データ処理装置1は、物理アドレス「I」のブロックについての情報をビットマップテーブルに加える。次に、ステップS108へ進む。

【0209】ステップS108において、データ処理装置1は、ステップS104で読み出した分散管理情報に基づいて、物理アドレス「I」のブロックの論理アドレスを変数Aに代入し、物理アドレス「I」のブロックの連結アドレスを変数Bに代入する。次にステップS109へ進む。

【0210】ステップS109において、データ処理装置1は、Aの値が「0xffff」であるか否かを判別する。「0xffff」でなければ、ステップS110へ進み、「0xffff」であれば、図22のステップS120へ進む。

【0211】ステップS110において、データ処理装置1は、エラー検出テーブルの論理アドレス「A」の値を調べる。次にステップS111へ進む。

【0212】ステップS111において、データ処理装置1は、エラー検出テーブルの論理アドレス「A」の値が「1」であるかを判別する。「1」であれば、ステップS112へ進み、「1」でなければステップS130へ進む。

【0213】ステップS112において、データ処理装置1は、エラー検出テーブルの論理アドレス「A」の値を「0」に書き換える。次に、ステップS113へ進む。

【0214】ステップS113において、データ処理装置1は、変換テーブルの論理アドレス「A」の欄に、変数「I」（すなわち物理ブロック「I」）を書き込む。次に、ステップS114へ進む。

【0215】ステップS114において、データ処理装置1は、連結テーブルの論理アドレス「A」の欄に、変数「B」（すなわち連結アドレス「B」）を書き込む。次に、ステップS115へ進む。

【0216】ステップS115において、データ処理装置1は、最終フラグが「ブロック最終」になっているかを判別する。「ブロック最終」になっていなければ、ステップS116へ進み、「ブロック最終」になっていれば、図22のステップS120へ進む。

【0217】ステップS116において、データ処理装置1は、論理アドレス「B」に対応する変換テーブルの値を調べる。次に、ステップS117へ進む。

【0218】ステップS117において、データ処理装置1は、論理アドレス「B」に対応する変換テーブルの値が「0xffff」であるかを判別する。「0xffff」であれば、ステップS118へ進み、「0xffff」でなければ、図22のステップS120へ進む。

【0219】ステップS118において、データ処理装置1は、エラー検出テーブルの論理アドレス「B」の欄を「1」に書き換える。次に、ステップS119へ進む。

【0220】ステップS119において、データ処理装置1は、エラー検出テーブルの論理アドレス「B」の欄に、論理アドレス「A」を書き込む。次に、図22のステップS120へ進む。

【0221】図22のステップS120において、データ処理装置1は、変数Iの値と、総ブロック数を示す定数Nの値とを比較する。 $I < N$ でなければ、ステップS121へ進み、 $I < N$ であれば、ステップS128へ進む。

【0222】ステップS121において、データ処理装置1は、変数Iに「0」を代入する。次にステップS122へ進む。

【0223】ステップS122において、データ処理装置1は、エラー検出テーブルの論理アドレス「I」の値を示す変数T(I)が「1」であるかを判別する。

「1」でなければステップS123へ進み、「1」であればステップS125へ進む。

【0224】ステップS123において、データ処理装置1は、変数Iの値と、総ブロック数を示す定数Nの値とを比較する。 $I = N$ であれば、これで処理を終了する。 $I = N$ でなければ、ステップS124へ進む。

【0225】ステップS124において、データ処理装置1は、変数Iの値を1インクリメントする。その後、ステップS122へ戻って処理を繰り返す。

【0226】また、ステップS122で変数T(I)が「1」であったときは、上述したようにステップS125へ進む。このステップS125に進むのは、連結アドレスエラーが生じていたときである。そこで、ステップS125において、データ処理装置1は、連結アドレス

エラーに対する所定のエラー訂正処理を行う。ここでは、上述したように、メモリカード2を使用するアプリケーションソフトウェアやメモリカード2に格納するデータ等に応じた適切なエラー訂正処理を行うようにする。そして、エラー訂正処理が完了したら、ステップS126へ進む。

【0227】ステップS126において、データ処理装置1は、エラー検出テーブルの論理アドレス「I」の値を「0」にする。次に、ステップS127へ進む。

【0228】ステップS127において、データ処理装置1は、変換テーブルの論理アドレス「I」の欄を「0xffff」に書き換える。その後、ステップS123へ進み、上述した処理を行う。

【0229】また、ステップS120で $I < N$ であったときは、上述したようにステップS128へ進む。このステップS128へ進むのは、全ブロックについての分散管理情報の読み出しが完了していないときである。そこで、このステップS128において、データ処理装置1は、変数Iの値を1インクリメントし、その後、図21のステップS104へ戻って処理を繰り返す。

【0230】また、図21のステップS105で物理アドレス「I」のブロックが使用不可能であったとき、及びステップS106で物理アドレス「I」のブロックが使用中でなかったときは、上述したようにステップS129へ進む。

【0231】ステップS129において、データ処理装置1は、物理アドレス「I」のブロックについての情報をビットマップテーブルに加える。その後、図22のステップS120へ進み、上述した処理を行う。

【0232】また、図21のステップS111でエラー検出テーブルの論理アドレス「A」の値が「1」でなかったときは、上述したようにステップS130へ進む。このステップS130において、データ処理装置1は、変換テーブルの論理アドレス「A」の値を調べる。次に、ステップS131へ進む。

【0233】ステップS131において、データ処理装置1は、変換テーブルの論理アドレス「A」の値が「0xffff」であるかを判別する。「0xffff」であれば、ステップS113へ進んで上述した処理を行い、「0xffff」でなければ、図23のステップS132へ進む。

【0234】図23のステップS132に進むのは、論理アドレスエラーが生じて、論理アドレス「A」を持つブロックが2つ存在しているときである。そこで、ステップS132において、データ処理装置1は、論理アドレス「A」を持つ2つのブロックの識別番号を読み出す。そして、一方のブロックの識別番号を読み出せたら、その値を変数Cに代入する。また、他方のブロックの識別番号を読み出せたら、その値を変数Dに代入する。次にステップS133へ進む。

【0235】ステップS133において、データ処理装

置1は、ステップS132での識別番号の読み出しが正常に行えたかを判別する。変数Cに代入される識別番号だけが読み出せたときは、ステップS134へ進み、それ以外のときは、ステップS137へ進む。

【0236】ステップS134において、データ処理装置1は、変数Dに対応するブロックのブロックフラグを「未消去」にする。次にステップS135へ進む。

【0237】ステップS135において、データ処理装置1は、ステップS134での処理、すなわち変数Dに対応するブロックのブロックフラグを「未消去」にする処理が成功したかを判別する。成功していれば、図21のステップS113へ戻って上述した処理を行い、成功していなければ、ステップS136へ進む。

【0238】ステップS136において、データ処理装置1は、変数Dに対応するブロックの可/不可フラグを「使用不可」にする。その後、図21のステップS113へ戻って上述した処理を行う。

【0239】また、ステップS137において、データ処理装置1は、ステップS132での識別番号の読み出しが正常に行えたかを判別する。変数Dに代入される識別番号だけが読み出せたときは、ステップS138へ進み、それ以外のときは、ステップS141へ進む。

【0240】ステップS138において、データ処理装置1は、変数Cに対応するブロックのブロックフラグを「未消去」にする。次にステップS139へ進む。

【0241】ステップS139において、データ処理装置1は、ステップS138での処理、すなわち変数Cに対応するブロックのブロックフラグを「未消去」にする処理が成功したかを判別する。成功していれば、図21のステップS113へ戻って上述した処理を行い、成功していなければ、ステップS140へ進む。

【0242】ステップS140において、データ処理装置1は、変数Cに対応するブロックの可/不可フラグを「使用不可」にする。その後、図21のステップS113へ戻って上述した処理を行う。

【0243】また、ステップS141において、データ処理装置1は、ステップS132での識別番号の読み出しが正常に行えたかを判別する。変数Cに代入される識別番号と、変数Dに代入される識別番号との両方が正常に読み出せたときは、ステップS142へ進む。一方、両方とも読み出せなかったときには、手動復旧モードに移行し、手動にてエラーの訂正に必要な適切な処理を行うようにする。

【0244】ステップS142において、データ処理装置1は、変数Cの値と、変数Dの値に「1」を加えた値とを比較する。これらの値が等しければ、ステップS134へ進み上述した処理を行い、これらの値が等しくなければ、ステップS143へ進む。

【0245】ステップS143において、データ処理装置1は、変数Dの値と、変数Cの値に「1」を加えた値

とを比較する。これらの値が等しければ、ステップS138へ進み上述した処理を行う。一方、これらの値が等しくないときは、手動復旧モードに移行し、手動にてエラーの訂正に必要な適切な処理を行うようにする。

【0246】以上のような処理により、集合管理情報の再構築時に、エラー検出訂正処理が同時に行われる。これにより、上述したように、エラー検出訂正処理のためにメモ리카ード2に余分にアクセスする必要がなくなる。すなわち、本発明を適用したシステムでは、データの書き込みが正常に終了し有効な集合管理情報がメモ리카ード2に書き戻されている場合には、エラー検出訂正処理は実行されない。

【0247】このように、本発明を適用したシステムでは、不要なエラー検出訂正処理を行わないので、メモ리카ード2へのアクセスの高効率化を図ることができる。特に、上記エラー検出訂正処理は、分散管理情報から集中管理情報を構築する作業と同時に進行するため、メモ리카ード2へのアクセスが最小限にとどめられる。

【0248】また、本発明を適用したシステムでは、メモ리카ード2の各ブロックに識別番号を付し、この識別番号を用いて論理アドレスエラーに対処できるようにしているので、データ更新処理を安全に行うことができる。すなわち、データ更新処理時に何らかのエラーが発生して、同じ論理アドレスを持つブロックが複数存在するようになってしまっても、識別番号を用いることで、データ更新処理前の状態のデータを復旧することができる。更に、本発明を適用したシステムでは、エラー検出テーブルを用いることで、連結アドレスエラーを検出することもでき、しかも連結先の存在しないブロックを検出することもできる。

【0249】

【発明の効果】以上詳細に説明したように、本発明によれば、外部記憶装置に論理アドレスエラーや連結アドレスエラーが生じて、それらのエラーを検出し適切に修復することができる。

【図面の簡単な説明】

【図1】本発明が適用されるシステムの全体構成を示す図である。

【図2】本発明を適用したメモ리카ードの構成を示すブロック図である。

【図3】本発明を適用したメモ리카ードの外観を示す斜視図である。

【図4】本発明を適用したメモ리카ードの記憶領域の構造を示す図である。

【図5】分散管理情報の構成を示す図である。

【図6】追加管理情報の構成を示す図である。

【図7】各ブロックの分散管理情報から集合管理情報を構築する様子を示す図である。

【図8】集合管理情報の構成を示す図である。

【図9】変換テーブルを示す図である。

- 【図10】連結テーブルを示す図である。
- 【図11】メモリカード起動時の手順を示すフローチャートである。
- 【図12】データ更新処理時の手順を示すフローチャートである。
- 【図13】終了処理の手順を示すフローチャートである。
- 【図14】サイズが分かっているファイルをメモリカードに書き込むときの手順を示すフローチャートである。
- 【図15】サイズが分かっていないファイルをメモリカードに書き込むときの手順を示すフローチャートである。
- 【図16】ファイル更新の手順について、具体的な例を挙げて示す概念図である。
- 【図17】ファイル更新の手順を示すフローチャートである。
- 【図18】エラー検出テーブルを示す図である。
- 【図19】ブロック間の連結状態の具体的な例を示す図である。
- 【図20】連結アドレスエラー検出の例として、ブロッ

ク間の連結状態が図19に示した状態のときの処理の流れを示す図である。

【図21】集合管理情報の構築とエラー検出訂正処理の手順を示すフローチャートである。

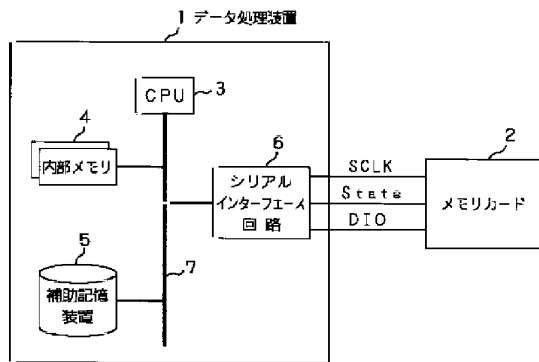
【図22】集合管理情報の構築とエラー検出訂正処理の手順を示すフローチャートである。

【図23】集合管理情報の構築とエラー検出訂正処理の手順を示すフローチャートである。

【符号の説明】

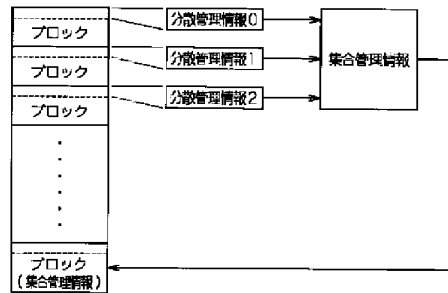
- 1 データ処理装置、 2 メモリカード、 3 演算処理装置、 4 内部メモリ、 5 補助記憶装置、 6 シリアルインターフェース回路、 7 バス、 11 コントローラ、 12 フラッシュメモリ、 13 シリアル/パラレル・パラレル/シリアル・インターフェース・シーケンサ、 14 フラッシュメモリ・インターフェース・シーケンサ、 15 ページバッファ、 16 エラー訂正回路、 17 コマンドジェネレータ、 18 コンフィグレーションROM、 19 発振器、 20 誤消去防止スイッチ

【図1】



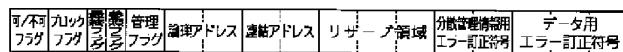
システムの全体構成

【図7】



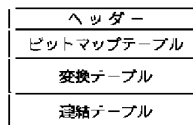
分散管理情報から集合管理情報の構築

【図5】



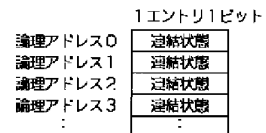
分散管理情報

【図8】



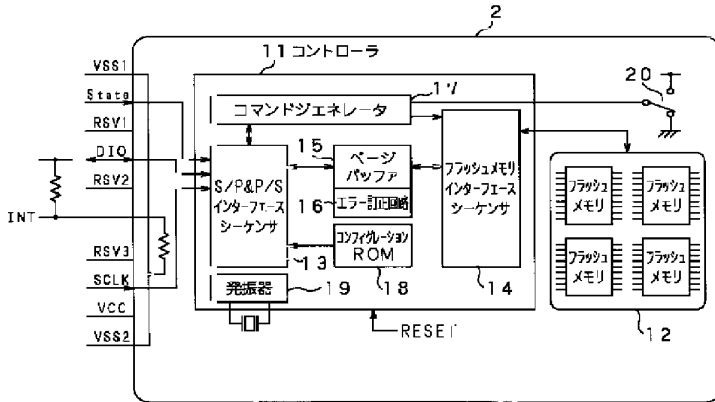
集合管理情報

【図18】



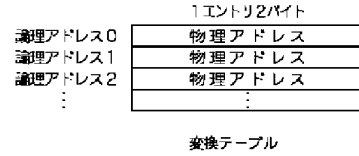
エラー検出テーブル

【図2】

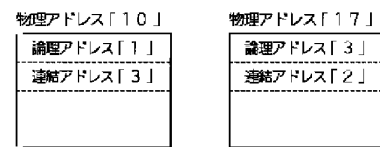


メモリカードの構成

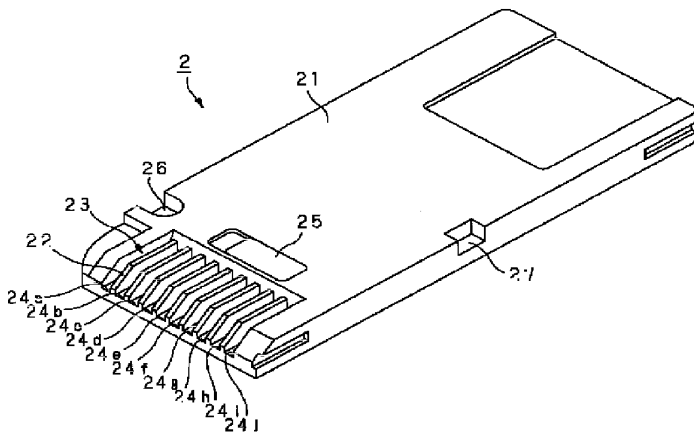
【図9】



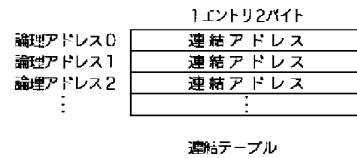
【図19】



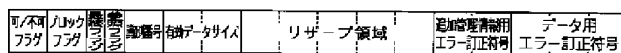
【図3】



【図10】

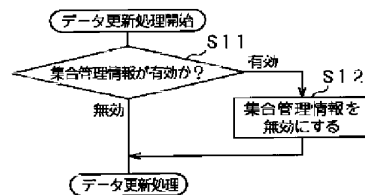


【図6】



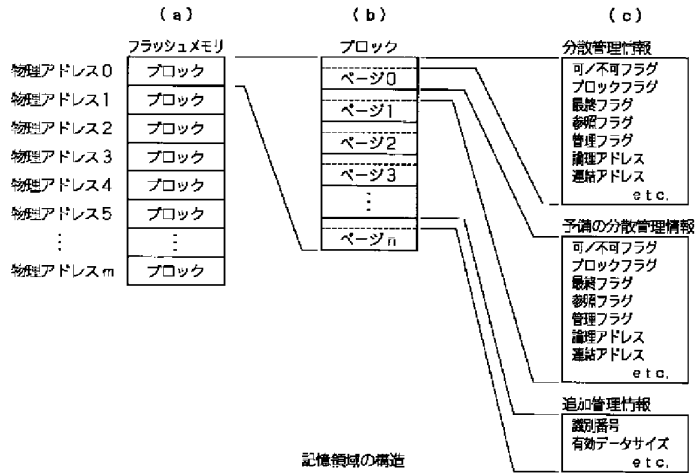
追加管理情報

【図12】

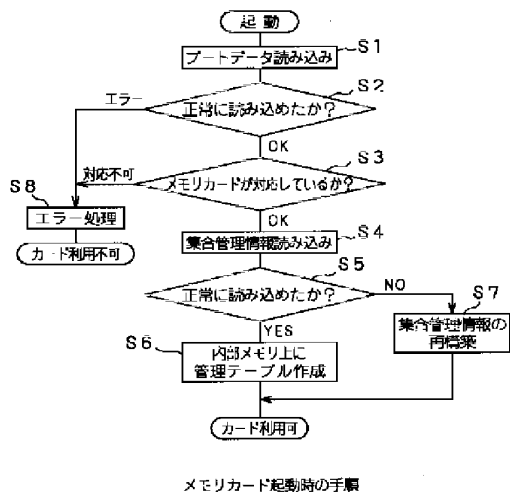


データ更新処理時の手順

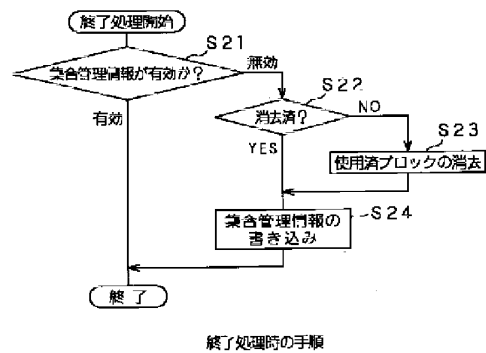
【図4】



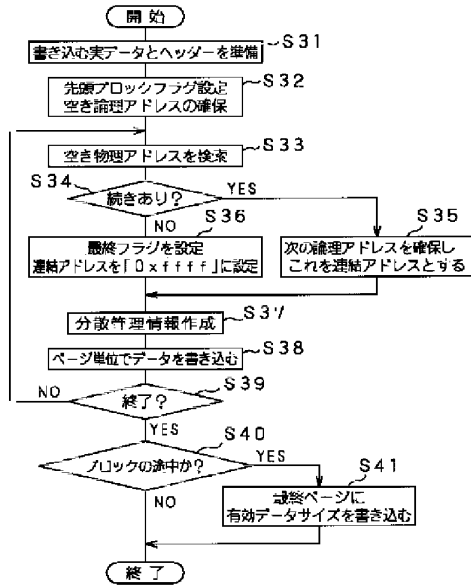
【図11】



【図13】

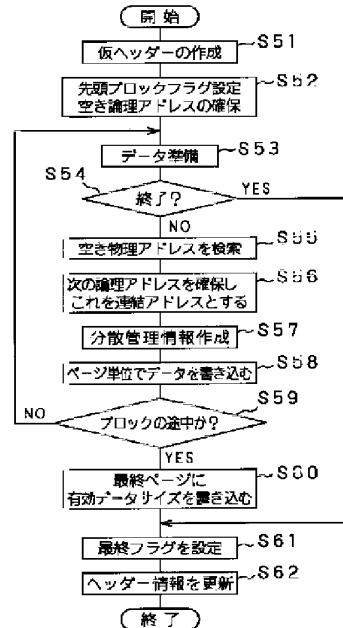


【図14】



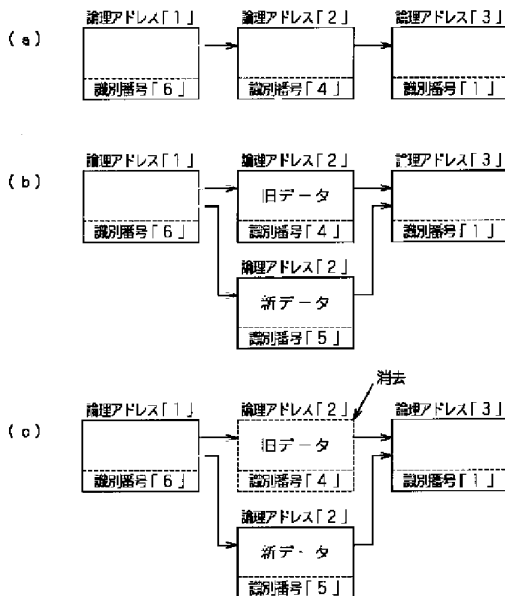
サイズが分かっているファイルの書き込み手順

【図15】



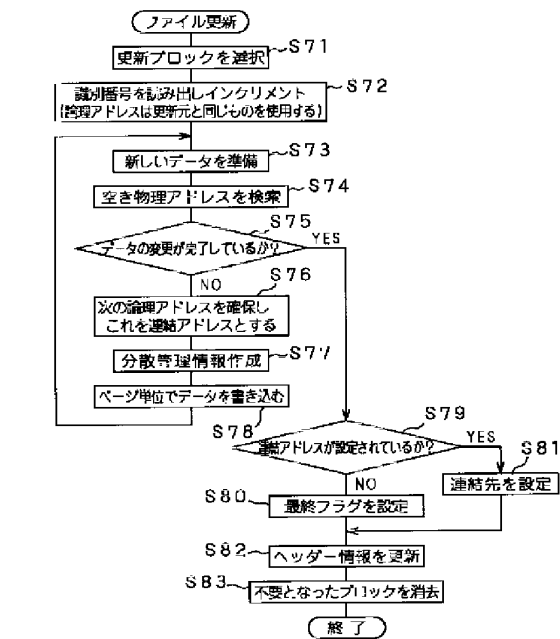
サイズが分かっているファイルの書き込み手順

【図16】



ファイル更新の具体例

【図17】



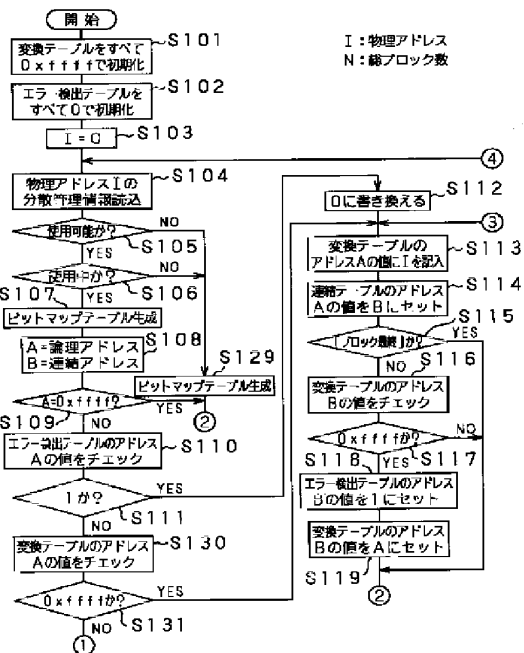
ファイル更新の手順

【図20】

	エラー検出 テーブル	変換 テーブル	連結 テーブル
(a)	論理アドレス1 論理アドレス2 論理アドレス3	C C C	0xffff 0xffff 0xffff
(b)	論理アドレス1 論理アドレス2 論理アドレス3	0 0 0	10 0xffff 0xffff
(c)	論理アドレス1 論理アドレス2 論理アドレス3	0 0 1	10 0xffff 1
(d)	論理アドレス1 論理アドレス2 論理アドレス3	0 1 0	10 0xffff 17
(e)	論理アドレス1 論理アドレス2 論理アドレス3	0 1 0	10 3 17

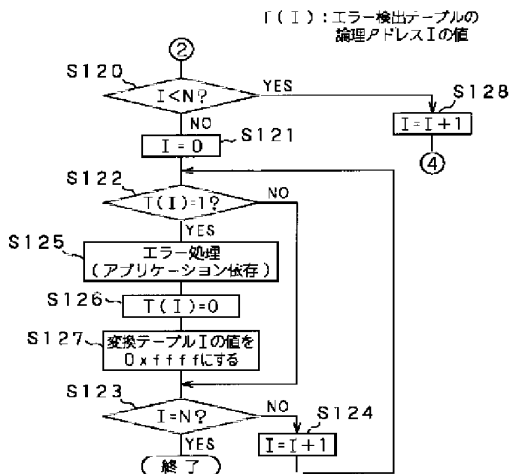
連結アドレスエラー検出の例

【図21】



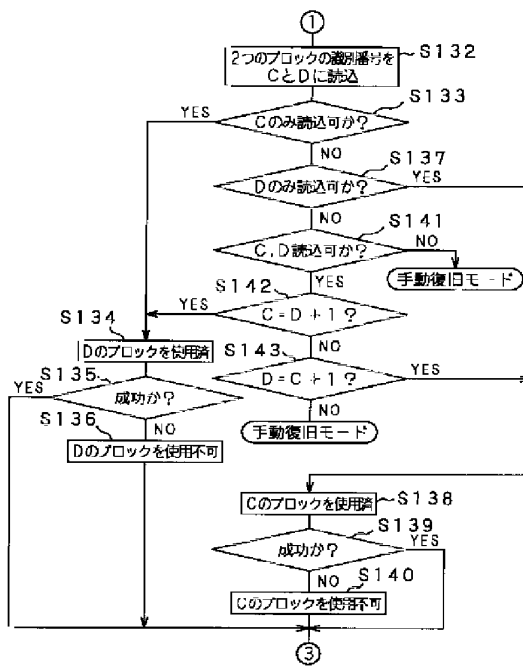
集合管理情報の構築とエラー検出訂正処理の手順

【図22】



集合管理情報の構築とエラー検出訂正処理の手順

【図23】



集合管理情報の構築とエラー検出訂正処理の手順

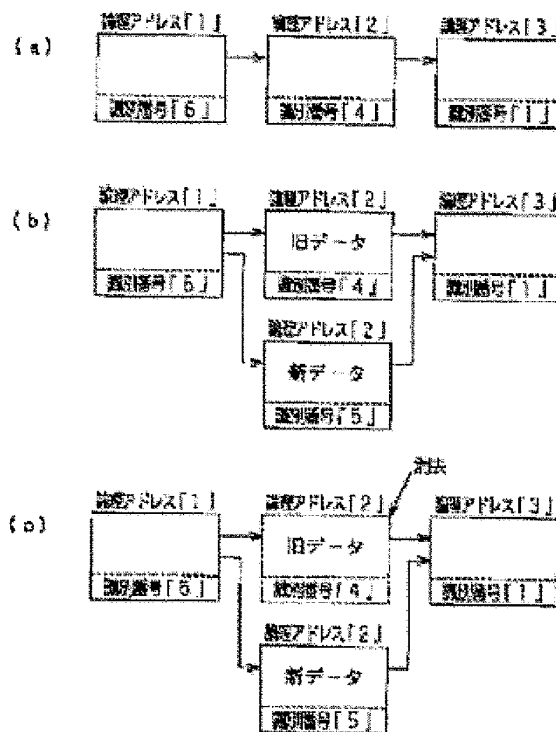
EXTERNAL STORAGE DEVICE, DATA PROCESSOR, AND DATA PROCESSING METHOD

Publication number: JP2000163302
 Publication date: 2000-06-16
 Inventor: FUSE HIROAKI; SASA SATORU; ONOE ATSUSHI
 Applicant: SONY CORP
 Classification:
 - international: G06F12/16; G06F3/06; G06F3/08; G06F12/00; G06F12/16; G06F3/06; G06F3/08; G06F12/00; (IPC1-7): G06F12/00; G06F3/06; G06F3/08; G06F12/16
 - European:
 Application number: JP19990346826 19970930
 Priority number(s): JP19990346826 19970930; JP19970267178 19970930

Report a data error here

Abstract of JP2000163302

PROBLEM TO BE SOLVED: To detect and properly repair an error occurring to the external storage device even in such a case by giving an identification number to each block of a memory card and handling an address error by using identification numbers. **SOLUTION:** For example, data divided from the head of a file are stored by blocks having logical addresses '1', '2', and '3' when the file is updated, and identification numbers '6', '4', and '1' are given to the respective blocks '1', '2', and '3'. When the data of the block having the logical address '2' is rewritten, the logical address '2' is assigned to another block, new data are written, and the value '5' obtained by increasing the identification number of the original block by one is set as its identification number. In this stage, the two blocks have the same address, but a final flag is stored in the block having the larger identification number by regarding the data stored in the block as new data and the original block is erased.



Data supplied from the esp@cenet database - Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-163302
(P2000-163302A)

(43) 公開日 平成12年6月16日 (2000.6.16)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
G 0 6 F 12/00	5 4 2	G 0 6 F 12/00	5 4 2 D
			5 4 2 M
3/06	3 0 5	3/06	3 0 5 A
3/08		3/08	H
12/16	3 4 0	12/16	3 4 0 P
審査請求 有 請求項の数 3 O L (全 23 頁)			

(21) 出願番号 特願平11-346826
 (62) 分割の表示 特願平9-267178の分割
 (22) 出願日 平成9年9月30日 (1997.9.30)

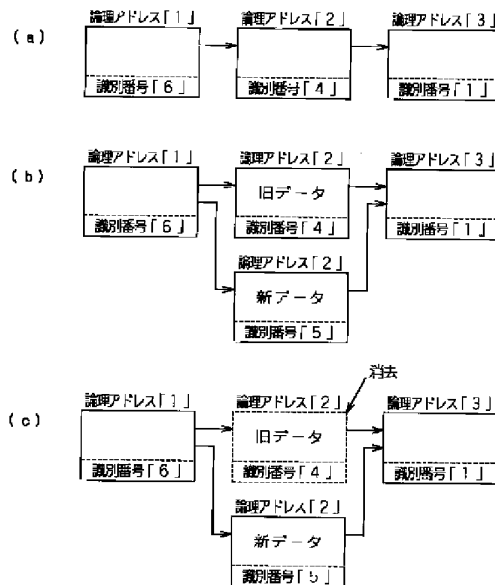
(71) 出願人 000002185
 ソニー株式会社
 東京都品川区北品川6丁目7番35号
 (72) 発明者 布施 博明
 東京都品川区北品川6丁目7番35号 ソニ
 ー株式会社内
 (72) 発明者 佐々 哲
 東京都品川区北品川6丁目7番35号 ソニ
 ー株式会社内
 (72) 発明者 尾上 淳
 東京都品川区北品川6丁目7番35号 ソニ
 ー株式会社内
 (74) 代理人 10006/736
 弁理士 小池 晃 (外2名)

(54) 【発明の名称】 外部記憶装置、データ処理装置及びデータ処理方法

(57) 【要約】

【課題】 外部記憶装置に論理アドレスエラーや連結アドレスエラーが生じて、それらのエラーを検出し適切に修復できるようにする。

【解決手段】 不揮発性メモリは、一括消去可能な複数のブロックに分割され、ブロックには、ブロックの論理アドレスが格納され、ファイルが複数のブロックに亘る場合には次のブロックの論理アドレスである連結アドレスが格納され、ファイルがブロックで終了している場合には最終ブロックであることを示すフラグが格納され、連結アドレスが指し示す論理アドレスが存在しないことが検出されたとき、ブロックの連結アドレスが指し示すブロックとして新規ブロックが割り当てられ、新規ブロックに正しく読めるページまでのデータがコピーされ、新規ブロックに最終ブロックであることを示すフラグが格納される。



ファイル更新の具体例

【特許請求の範囲】

【請求項1】 データ処理装置に着脱可能に接続され、上記データ処理装置からのデータを記憶する外部記憶装置において、

上記データ処理装置とシリアルデータをやり取りするためのシリアルインターフェースと、

上記データをファイル単位で格納する不揮発性メモリとを備え、

上記不揮発性メモリは、一括消去可能な複数のブロックに分割され、上記ブロックには、上記ブロックの論理アドレスが格納され、上記ファイルが複数のブロックに亘る場合には次のブロックの論理アドレスである連結アドレスが格納され、上記ファイルが上記ブロックで終了している場合には最終ブロックであることを示すフラグが格納され、

上記連結アドレスが指し示す論理アドレスが存在しないことが検出されたとき、上記ブロックの連結アドレスが指し示すブロックとして新規ブロックが割り当てられ、上記新規ブロックに正しく読めるページまでのデータがコピーされ、上記新規ブロックに最終ブロックであることを示すフラグが格納されることを特徴とする外部記憶装置。

【請求項2】 シリアルデータをやり取りするためのシリアルインターフェースと、一括消去可能な複数のブロックに分割され、上記ブロックにデータをファイル単位で格納する不揮発性メモリとを備える外部記憶装置が着脱されるデータ処理装置において、

上記ブロックに、上記ブロックの論理アドレスを格納し、上記ファイルが複数のブロックに亘る場合には次のブロックの論理アドレスである連結アドレスを格納し、上記ファイルが上記ブロックで終了している場合には最終ブロックであることを示すフラグを格納し、

上記連結アドレスが指し示す論理アドレスが存在しないことが検出されたとき、上記ブロックの連結アドレスが指し示すブロックとして新規ブロックを割り当て、上記新規ブロックに正しく読めるページまでのデータをコピーし、上記新規ブロックに最終ブロックであることを示すフラグを格納することを特徴とするデータ処理装置。

【請求項3】 一括消去可能な複数のブロックに分割された不揮発性メモリを有する外部記憶装置にファイル単位でデータを格納する際に、上記ブロックに、上記ブロックの論理アドレスを格納し、上記ファイルが複数のブロックに亘る場合には次のブロックの論理アドレスである連結アドレスを格納し、上記ファイルが上記ブロックで終了している場合には最終ブロックであることを示すフラグを格納し、

上記連結アドレスが指し示す論理アドレスが存在しないことが検出されたとき、上記ブロックの連結アドレスが指し示すブロックとして新規ブロックを割り当て、上記新規ブロックに正しく読めるページまでのデータをコピ

ーし、上記新規ブロックに最終ブロックであることを示すフラグを格納することを特徴とするデータ処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、記憶領域が複数のブロックに分割されてなる外部記憶装置に関する。また、本発明は、記憶領域が複数のブロックに分割されてなる外部記憶装置にデータを格納するデータ処理装置に関する。また、本発明は、記憶領域が複数のブロックに分割されてなる外部記憶装置にデータを格納する際のデータ処理方法に関する。

【0002】

【従来の技術】パーソナルコンピュータやデジタルスチルカメラ等のようなデータ処理装置に用いられる外部記憶装置として、フラッシュメモリを備えた外部記憶装置がある。

【0003】フラッシュメモリを備えた外部記憶装置は、記憶領域を複数のブロックに分割し、データ領域の管理をブロック単位で行う。ここで、各ブロックはデータ消去の単位となる。すなわち、データを消去する際は、当該データを含むブロック全体に対して初期化処理を施す。これにより、当該ブロックに格納されているデータが一括して消去される。

【0004】このような外部記憶装置では、データをブロックに格納するときに、それらのブロックに対してユニークな論理アドレスが設定される。そして、各ブロックは、この論理アドレスを用いて管理される。

【0005】また、外部記憶装置に格納されるデータは、通常、ファイル単位で外部記憶装置に格納されるが、一つのファイルが複数のブロックにわたる場合には、それらのブロックの連結情報が必要となる。そこで、一つのファイルが複数のブロックにわたる場合には、当該ファイルを格納しているブロックのそれぞれに、次のブロックの論理アドレス（以下、連結アドレスと称する。）が格納される。

【0006】

【発明が解決しようとする課題】従来、このような外部記憶装置では、記憶領域内にエラーがあるか否かを検査する処理や、エラーがあった場合に当該エラーの修復を試みる処理を、外部記憶装置の起動時に毎回実行するようにしていた。なお、以下の説明では、このような処理のことを、エラー検出訂正処理と称する。通常、このようなエラー検出訂正処理は、比較的負荷が大きく処理に時間を要する処理である。したがって、従来の外部記憶装置は、エラー検出訂正処理のために、速やかに起動することができないという問題があった。

【0007】また、データ領域の管理をブロック単位で行うような外部記憶装置では、ブロックにデータを新規に書き込んでいるときや、ブロックに格納されているデータを更新しているときなどに、いきなり電源が遮断さ

れたり、データ処理装置から外部記憶装置が強制的に取り外されたりしたような場合に、同じ論理アドレスを持つ複数のブロックが同時に存在するような状態（以下、論理アドレスエラーと称する。）となったり、連結アドレスで指し示されたブロックが存在しないような状態（以下、連結アドレスエラーと称する。）となったりする可能性がある。当然の事ながら、このような状態になると、ファイルが予期せぬブロックに連結されてしまったりして、外部記憶装置を正常に使用することができなくなってしまう。

【0008】しかしながら、従来の外部記憶装置は、論理アドレスエラーや連結アドレスエラーを検出して適切に修復するような機能を備えていなかった。そのため、従来は、いきなり電源が遮断されたり、データ処理装置から外部記憶装置が強制的に取り外されたりしたような場合に、その後、外部記憶装置を正常に使用することができなくなってしまうことがあった。

【0009】本発明は、以上のような従来の実情に鑑みて提案されたものであり、外部記憶装置に論理アドレスエラーや連結アドレスエラーが生じて、それらのエラーを検出し適切に修復できるようにすることを目的としている。

【0010】

【課題を解決するための手段】本発明に係る外部記憶装置は、データ処理装置に着脱可能に接続され、データ処理装置からのデータを記憶する外部記憶装置であり、データ処理装置とシリアルデータをやり取りするためのシリアルインターフェースと、データをファイル単位で格納する不揮発性メモリとを備える。そして、不揮発性メモリは、一括消去可能な複数のブロックに分割され、ブロックには、ブロックの論理アドレスが格納され、ファイルが複数のブロックに亘る場合には次のブロックの論理アドレスである連結アドレスが格納され、ファイルがブロックで終了している場合には最終ブロックであることを示すフラグが格納され、連結アドレスが指し示す論理アドレスが存在しないことが検出されたとき、ブロックの連結アドレスが指し示すブロックとして新規ブロックが割り当てられ、新規ブロックに正しく読めるページまでのデータがコピーされ、新規ブロックに最終ブロックであることを示すフラグが格納される。

【0011】また、本発明に係るデータ処理装置は、シリアルデータをやり取りするためのシリアルインターフェースと、一括消去可能な複数のブロックに分割され、ブロックにデータをファイル単位で格納する不揮発性メモリとを備える外部記憶装置が着脱されるデータ処理装置であって、ブロックに、ブロックの論理アドレスを格納し、ファイルが複数のブロックに亘る場合には次のブロックの論理アドレスである連結アドレスを格納し、ファイルがブロックで終了している場合には最終ブロックであることを示すフラグを格納し、連結アドレスが指し

示す論理アドレスが存在しないことが検出されたとき、ブロックの連結アドレスが指し示すブロックとして新規ブロックを割り当て、新規ブロックに正しく読めるページまでのデータをコピーし、新規ブロックに最終ブロックであることを示すフラグを格納するものである。

【0012】更に、本発明に係るデータ処理方法は、一括消去可能な複数のブロックに分割された不揮発性メモリを有する外部記憶装置にファイル単位でデータを格納する際に、ブロックに、ブロックの論理アドレスを格納し、ファイルが複数のブロックに亘る場合には次のブロックの論理アドレスである連結アドレスを格納し、ファイルがブロックで終了している場合には最終ブロックであることを示すフラグを格納し、連結アドレスが指し示す論理アドレスが存在しないことが検出されたとき、ブロックの連結アドレスが指し示すブロックとして新規ブロックを割り当て、新規ブロックに正しく読めるページまでのデータをコピーし、新規ブロックに最終ブロックであることを示すフラグを格納するものである。

【0013】

【発明の実施の形態】以下、本発明の実施の形態について、図面を参照しながら詳細に説明する。

【0014】1. システムの全体構成

本発明が適用されるシステムの一例について、その全体構成を図1に示す。このシステムは、ホスト側システムとなるデータ処理装置1と、シリアルインターフェースを介してデータ処理装置1に接続される外部記憶装置であるメモリカード2とから構成される。

【0015】データ処理装置1は、演算処理装置（CPU）3と、内部メモリ4と、補助記憶装置5と、シリアルインターフェース回路6とを備え、これらがバス7によって相互に接続されてなる。このデータ処理装置1は、例えば、補助記憶装置5に格納されているプログラムを読み出して、当該プログラムを、内部メモリ4をワークエリアとして使用して、CPU3により実行する。このとき、必要に応じて、シリアルインターフェース回路6を介してメモリカード2との間でデータのやり取りを行う。

【0016】なお、本発明が適用されるシステムに使用されるデータ処理装置1は、外部記憶装置との間でデータのやり取りが可能なものであるならば特に限定されるものではなく、本発明は、パーソナルコンピュータ、デジタルスチルカメラ、デジタルビデオカメラ等、種々のデータ処理装置に適用可能である。

【0017】データ処理装置1とメモリカード2とは、シリアルインターフェースによって接続されており、具体的には、少なくとも3本のデータ線SCLK、State、DIOによって接続される。すなわち、データ処理装置1とメモリカード2とは、少なくとも、データ伝送時にクロック信号を伝送する第1のデータ線SCLKと、データ伝送時に必要なステータス信号を伝送する第

2のデータ線Stateと、メモ리카ード2に書き込むデータ又はメモ리카ード2から読み出すデータ等をシリアルに伝送する第3のデータ線DIOとによって接続され、これらを介して、データ処理装置1とメモ리카ード2との間でのデータのやり取りを行う。

【0018】データ処理装置1とメモ리카ード2との間でのデータのやり取りは、通常、ヘッダーと実データとから構成されるファイル単位で行われる。なお、ファイルのヘッダーには、例えば、ファイルにアクセスするための情報や、データ処理装置1で実行されるプログラムが必要とされる情報等が格納される。

【0019】2.メモ리카ードの構成

メモ리카ード2は、図2に示すように、いわゆるコントローラICからなるコントローラ11と、コントローラ11によって管理されるフラッシュメモリ12とを備えている。

【0020】コントローラ11は、シリアル/パラレル変換やパラレル/シリアル変換等を行うシリアル/パラレル・パラレル/シリアル・インターフェース・シーケンサ13（以下、S/P&P/S・インターフェース・シーケンサ13と称する。）と、フラッシュメモリ12へのインターフェースを司るフラッシュメモリ・インターフェース・シーケンサ14と、S/P&P/S・インターフェース・シーケンサ13とフラッシュメモリ・インターフェース・シーケンサ14との間でやり取りされるデータを一時的に記憶するページバッファ15と、エラー訂正の処理を行うエラー訂正回路16と、フラッシュメモリ12へのアクセスを制御する制御コマンドの生成等を行うコマンドジェネレータ17と、このメモ리카ード2のバージョン情報や各種属性情報等が格納されているコンフィグレーションROM18と、各回路に対してそれらの動作に必要なクロック信号を供給する発振器19とを備えている。

【0021】S/P&P/S・インターフェース・シーケンサ13は、少なくとも上述した3本のデータ線SCLK, State, DIOを介して、データ処理装置1のシリアルインターフェース回路6に接続され、これらのデータ線SCLK, State, DIOを介して、データ処理装置1との間でデータのやり取りを行う。すなわち、S/P&P/S・インターフェース・シーケンサ13は、ページバッファ15から送られてきたパラレルデータをシリアルデータに変換して、データ処理装置1のシリアルインターフェース回路6へ送出する。また、S/P&P/S・インターフェース・シーケンサ13は、データ処理装置1のシリアルインターフェース回路6から送られてきたシリアルデータをパラレルデータに変換して、ページバッファ15へ送出する。

【0022】このS/P&P/S・インターフェース・シーケンサ13とデータ処理装置1との間でのシリアルデータの伝送は、第1のデータ線SCLKによってデー

タ処理装置1から送られてくるクロック信号によって同期を取りながら、第3のデータ線DIOによって行われる。このとき、第3のデータ線DIOによってやり取りされるシリアルデータのデータ種別は、第2のデータ線Stateによって伝送されるステータス信号によって判別される。ここで、シリアルデータの種別には、例えば、フラッシュメモリ12に書き込むべきデータ、フラッシュメモリ12から読み出されたデータ、又はこのメモ리카ード2の動作を制御するための制御データ等がある。なお、ステータス信号は、メモ리카ード2の状態を示すためにも使用される。ステータス信号によって示されるメモ리카ード2の状態には、例えば、メモ리카ード2が何らかの処理の最中でデータ処理装置1からのデータ入力を受け付けない状態や、メモ리카ード2の側での処理が終了してデータ処理装置1からのデータ入力を待っている状態等がある。

【0023】また、S/P&P/S・インターフェース・シーケンサ13は、データ処理装置1から送られてきたデータがメモ리카ード2の動作を制御するための制御データである場合には、当該制御データをコマンドジェネレータ17に送出する。

【0024】コマンドジェネレータ17は、データ処理装置1からS/P&P/S・インターフェース・シーケンサ13を介して送られてきた制御データに基づいて、フラッシュメモリ12へのアクセスを制御する制御コマンドを生成し、当該制御コマンドをフラッシュメモリ・インターフェース・シーケンサ14へ送出する。フラッシュメモリ・インターフェース・シーケンサ14は、後述するように、この制御コマンドに基づいて、フラッシュメモリ12にデータを書き込んだり、フラッシュメモリ12からデータを読み出したりする。

【0025】なお、このコマンドジェネレータ17には、誤消去防止スイッチ20が接続されている。そして、この誤消去防止スイッチ20がオンになっているときには、フラッシュメモリ12に書かれているデータを消去するように指示する制御データがデータ処理装置1から送られてきたとしても、コマンドジェネレータ17は、フラッシュメモリ12に書かれているデータを消去するような制御コマンドを生成しない。すなわち、このメモ리카ード2は、誤消去防止スイッチ20によって、フラッシュメモリ12に保存されているデータの消去が行えない状態と、フラッシュメモリ12に保存されているデータの消去が行える状態とを切り換えることが可能となっている。

【0026】S/P&P/S・インターフェース・シーケンサ13とフラッシュメモリ・インターフェース・シーケンサ14との間に配されたページバッファ15は、いわゆるバッファメモリであり、S/P&P/S・インターフェース・シーケンサ13とフラッシュメモリ・インターフェース・シーケンサ14との間でやり取りされ

るデータを一時的に記憶する。

【0027】すなわち、S/P&P/S・インターフェース・シーケンサ13からフラッシュメモリ・インターフェース・シーケンサ14へ送られるデータは、先ず、S/P&P/S・インターフェース・シーケンサ13からページバッファ15に送られて、このページバッファ15によって一時的に記憶される。このとき、ページバッファ15に記憶されたデータは、エラー訂正回路16によってエラー訂正符号が付けられる。そして、エラー訂正符号が付けられたデータは、ページバッファ15から所定のページ単位毎（例えば1ページ=512バイトとされる。）に、フラッシュメモリ・インターフェース・シーケンサ14へと送られる。

【0028】或いは、フラッシュメモリ・インターフェース・シーケンサ14からS/P&P/S・インターフェース・シーケンサ13へ送られるデータは、先ず、フラッシュメモリ・インターフェース・シーケンサ14からページバッファ15に送られて、このページバッファ15によって一時的に記憶される。このとき、ページバッファ15に記憶されたデータは、エラー訂正回路16によってエラー訂正処理が施される。そして、エラー訂正処理が施されたデータは、ページバッファ15から所定のページ単位毎に、S/P&P/S・インターフェース・シーケンサ13へと送られる。

【0029】フラッシュメモリ・インターフェース・シーケンサ14は、コマンドジェネレータ17からの制御コマンドに基づいて、フラッシュメモリ12へのデータの書き込みや、フラッシュメモリ12からのデータの読み出し等を行う。すなわち、フラッシュメモリ・インターフェース・シーケンサ14は、コマンドジェネレータ17からの制御コマンドに基づいて、フラッシュメモリ12からデータを読み出して、当該データを上述のようにページバッファ15を介して、S/P&P/S・インターフェース・シーケンサ13へと送出する。或いは、フラッシュメモリ・インターフェース・シーケンサ14は、コマンドジェネレータ17からの制御コマンドに基づいて、S/P&P/S・インターフェース・シーケンサ13からのデータを、上述のようにページバッファ15を介して受け取り、当該データをフラッシュメモリ12に書き込む。

【0030】コンフィグレーションROM18には、このメモリカード2のバージョン情報や各種属性情報等が格納されている。コンフィグレーションROM18に格納された情報は、必要に応じて、S/P&P/S・インターフェース・シーケンサ13を介してコマンドジェネレータ17によって読み出されて使用される。すなわち、コマンドジェネレータ17は、必要に応じて、コンフィグレーションROM18に格納されている情報を読み出し、この情報に基づいてメモリカード2に関する各種設定を行う。

【0031】以上のようなメモリカード2に対して、フラッシュメモリ12に書き込まれるデータが、上述した3本のデータ線SCLK, State, DIOを介して、データ処理装置1からシリアルデータとして送られてくると、先ず、S/P&P/S・インターフェース・シーケンサ13は、当該シリアルデータをパラレルデータに変換し、当該パラレルデータをページバッファ15へ送出する。ページバッファ15は、S/P&P/S・インターフェース・シーケンサ13から送られてきたデータを一時的に記憶する。このとき、ページバッファ15に記憶されたデータには、エラー訂正回路16によってエラー訂正符号が付けられる。そして、エラー訂正符号が付けられたデータは、所定のページ単位毎にフラッシュメモリ・インターフェース・シーケンサ14に送出される。そして、フラッシュメモリ・インターフェース・シーケンサ14は、ページバッファ15から送られてきたデータを、コマンドジェネレータ17からの制御コマンドに基づいて、フラッシュメモリ12に書き込む。以上の処理により、データ処理装置1から送られてきたデータが、フラッシュメモリ12に書き込まれる。

【0032】また、以上のようなメモリカード2からデータを読み出す際は、先ず、コマンドジェネレータ17からの制御コマンドに基づいて、フラッシュメモリ・インターフェース・シーケンサ14によって、フラッシュメモリ12からデータが読み出される。そして、フラッシュメモリ・インターフェース・シーケンサ14は、フラッシュメモリ12から読み出したデータをページバッファ15に送出する。ページバッファ15は、フラッシュメモリ・インターフェース・シーケンサ14から送られてきたデータを一時的に記憶する。このとき、ページバッファ15に記憶されたデータには、エラー訂正回路16によってエラー訂正処理が施される。そして、エラー訂正処理が施されたデータは、所定のページ単位毎にS/P&P/S・インターフェース・シーケンサ13に送出される。そして、S/P&P/S・インターフェース・シーケンサ13は、ページバッファ15から送られてきたデータを、シリアルデータに変換した上で、上述した3本のデータ線SCLK, State, DIOを介して、データ処理装置1へと送出する。以上の処理により、フラッシュメモリ12から読み出されたデータが、データ処理装置1へと送出される。

【0033】なお、データの書き込みや読み出しを行う際は、フラッシュメモリ12に書き込まれるデータやフラッシュメモリ12から読み出されたデータのやり取りが行われるだけでなく、そのやり取りを制御するための制御データも、データ処理装置1からメモリカード2のS/P&P/S・インターフェース・シーケンサ13へ送られる。この制御データは、S/P&P/S・インターフェース・シーケンサ13からコマンドジェネレータ17に送られる。そして、コマンドジェネレータ17

は、S/P&P/S・インターフェース・シーケンサ13から送られてきた制御データに基づいて、フラッシュメモリ12へのアクセスを制御する制御コマンドを生成する。そして、この制御コマンドは、フラッシュメモリ・インターフェース・シーケンサ14に送られ、フラッシュメモリ・インターフェース・シーケンサ14は、この制御コマンドに基づいてフラッシュメモリ12にアクセスして、データの書き込みやデータの読み出しを行う。

【0034】なお、メモリカード2は、上述した3本のデータ線SCLK, State, DIOを備えるだけでなく、その他に、電圧供給用の配線や、通常は使用しないリザーブの配線等を備えていてもよい。例えば、図2並びに後掲する図3では、上述した3本のデータ線SCLK, State, DIOの他に、4本の電源用の配線VSS1, VSS2, VCC, INTと、3本のリザーブの配線RSV1, RSV2, RSV3とをメモリカード2に設けた例を挙げている。

【0035】3. メモリカードの外観

つぎに、以上のようなメモリカード2の具体的な外形について、図3を参照して説明する。

【0036】メモリカード2は、合成樹脂等からなり平面形状が長方形とされる薄肉のカード状のケース21に、上述したコントローラ11やフラッシュメモリ12等が内蔵されてなる。そして、このメモリカード2は、当該メモリカード2を装着する装着機構を備えたデータ処理装置1に装着されて使用される。

【0037】このメモリカード2のケース21の前端部には、斜めに切り欠かれた切り欠き部22が形成されており、更に当該切り欠き部22が形成された部分に、10個の凹状部23が形成されている。そして、これらの凹状部23の内部には、メモリカード2がデータ処理装置1の装着装置に装着されたときに、データ処理装置1の接続端子に接続される外部接続用端子が、それぞれ配されている。すなわち、このメモリカード2は、外部接続用端子として10本の端子24a, 24b, 24c, 24d, 24e, 24f, 24g, 24h, 24i, 24jを備えている。これらの外部接続用端子の内訳は、3本のデータ線用の端子24b, 24d, 24h、4本の電源用端子24a, 24f, 24i, 24j、及び3本のリザーブ端子24c, 24e, 24gである。

【0038】また、このメモリカード2のケース21の上面には、誤消去防止部材25が取り付けられている。誤消去防止部材25は、ケース21の内部に収納された上記誤消去防止スイッチ20に係合されており、この誤消去防止部材25をスライド操作することにより、誤消去防止スイッチ20のオン/オフの切り換えを行えるようになっている。

【0039】このメモリカード2には、データ処理装置1の装着装置に装着された際にメモリカード2がデータ

処理装置1から脱落しないようにするため、ケース20の側面の一方に円弧状の第1のロック用切欠部26が形成され、ケース20の側面の他方に矩形状の第2のロック用切欠部27が形成されている。そして、このメモリカード2がデータ処理装置1の装着装置に装着されると、メモリカード2が脱落しないように、これらのロック用切欠部26, 27が、データ処理装置1の装着装置に係合される。

【0040】なお、図3に示したメモリカード2は、本発明が適用される外部記憶装置の一例に過ぎない。すなわち、本発明は、外部記憶装置の外形に依存することなく、どんな外形の外部記憶装置にも適用可能である。

【0041】4. 記憶領域の構造

つぎに、以上のようなメモリカード2に搭載されるフラッシュメモリ12の記憶領域の構造について説明する。

【0042】このフラッシュメモリ12の記憶領域は、図4(a)に示すように、データ消去の単位となる複数のブロックに分割されてなる。なお、これらのブロックには、このメモリカード2が起動されたときにデータ処理装置1によって最初に読み込まれるデータであるブートデータが格納されるブートブロックと、任意のデータが書き込まれるデータブロックとがある。各ブロックには、それぞれ固有の物理アドレスが付けられている。これらのブロックは、データ消去の単位であると同時に、ファイル管理上の最小単位でもある。すなわち、ファイルは1つ又は複数のブロックに格納され、1つのブロックを複数のファイルで利用することはできない。

【0043】そして、各ブロックは、「1」又は「0」を示す2つの状態を取りうる複数のビットからなり、初期状態では、全てのビットが「1」とされており、ビット単位での変更は「1」から「0」へだけが可能となっている。すなわち、「1」及び「0」からなるデータを書き込む際、「1」については該当するビットをそのまま保持し、「0」については該当するビットを「1」から「0」に変更する。

【0044】そして、一度書き込んだデータを消去する際は、ブロック単位で一括して初期化処理を行い、当該ブロックの全ビットを「1」とする。これにより、当該ブロックに書き込まれたデータが一括して消去され、そのブロックは再びデータの書き込みが可能な状態となる。

【0045】なお、「0」から「1」への変更を行うには、ブロック単位で一括して初期化処理を行い、当該ブロックの全ビットを「1」にする必要があるが、「1」から「0」への変更は、ブロック単位で一括して初期化処理を行わなくて可能である。以下の説明では、ブロック単位で一括して初期化処理を行うことなく「1」から「0」へ変更することを、オーバーライトと称する。

【0046】なお、本発明は、上述のように各ビットが2つの状態だけを取りうるフラッシュメモリ（いわゆる

2値型のフラッシュメモリ)だけでなく、各ビットが3つ以上の状態を取りうるフラッシュメモリ(いわゆる多値型のフラッシュメモリ)にも適用可能である。

【0047】上記フラッシュメモリ12の各ブロックは、図4(b)に示すように、データの書き込みや読み出しの単位となる複数のページから構成される。すなわち、このフラッシュメモリ12にデータを書き込む際は、上述したように、ページ単位にてページバッファ15から送られてきたデータが、フラッシュメモリ・インターフェース・シーケンサ14によってページ単位にてフラッシュメモリ12に書き込まれる。また、このフラッシュメモリ12からデータを読み出す際は、フラッシュメモリ・インターフェース・シーケンサ14によってページ単位毎にデータが読み出されて、ページバッファ15へと送られる。

【0048】各ページは、データエリアと、冗長エリアとを有している。データエリアは、任意のデータが書き込まれる領域である。冗長エリアは、データエリアに書き込まれるデータの管理に必要な情報が格納される領域である。

【0049】具体的には、図4(c)に示すように、ブロックの先頭ページの冗長エリアには、当該ブロックを管理するために必要な情報として、いわゆる分散管理情報が格納される。また、ブロックの2ページ目以降の各ページの冗長エリアにも、予備の分散管理情報として、先頭ページの冗長エリアに格納された分散管理情報と同じものが格納される。ただし、最終ページの冗長エリアには、分散管理情報ではなく、分散管理情報だけでは管理しきれない追加情報として、いわゆる追加管理情報が格納される。

【0050】このように、このフラッシュメモリ12では、各ブロック内の冗長エリアに分散管理情報が格納される。分散管理情報は、当該分散管理情報が格納されたブロックを管理するための情報である。この分散管理情報により、例えば、当該ブロックがファイルの先頭となるブロックであるか否かについての情報や、複数のブロックからファイルが構成される場合にはそれらのブロックの繋がりを示す情報等を得ることができる。なお、この分散管理情報については、後で詳細に説明する。

【0051】そして、このメモリカード2では、各ブロックの分散管理情報を集めることにより、フラッシュメモリ全体を管理するための情報として、いわゆる集合管理情報を作成して、この集合管理情報をファイルとしてフラッシュメモリ12に格納しておくようにする。

【0052】そして、通常は、集合管理情報によって、各ブロックにアクセスするために必要な情報を得るようにする。すなわち、データ処理装置1とメモリカード2との間でデータのやり取りを行う際、データ処理装置1は、集合管理情報をメモリカード2から読み出して内部メモリ4に管理テーブルを作成し、この管理テーブルに

基づいてメモリカード2にアクセスする。これにより、データアクセスの都度、個々のブロックに格納された分散管理情報にアクセスするよう必要がなくなり、より高速なデータアクセスが可能となる。

【0053】5. 分散管理情報

つぎに、分散管理情報について詳細に説明する。

【0054】分散管理情報は、当該分散管理情報が格納されたブロックを管理するための情報であり、16バイトの冗長エリアに書き込まれてなる。具体的には、図5に示すように、1バイトの可/不可フラグと、1バイトのブロックフラグと、4ビットの最終フラグと、4ビットの参照フラグと、1バイトの管理フラグと、2バイトの論理アドレスと、2バイトの連結アドレスと、3バイトのリザーブ領域と、2バイトの分散管理情報用エラー訂正符号と、3バイトのデータ用エラー訂正符号とからなる。

【0055】可/不可フラグは、ブロックが使用可能状態か使用不可能状態を示すフラグであり、具体的には、「使用可」と「使用不可」の2つの状態を示す。

「使用可」は、当該ブロックが使用可能な状態を示し、「使用不可」は、当該ブロックが使用不可能な状態であることを示す。例えば、ブロック内に回復不能なエラーが生じたようなときに、この可/不可フラグが「使用不可」に設定され、当該ブロックが使用不可とされる。

【0056】ブロックフラグは、ブロックの状態を示すフラグであり、具体的には、「未使用」「先頭使用」「使用」「未消去」の4つの状態を示す。「未使用」は、当該ブロックが未使用又は消去済みで、初期状態(全ビットが「1」の状態)とされており、直ぐにデータの書き込みが可能な状態を示す。「先頭使用」は、当該ブロックがファイルの先頭で使用されている状態を示す。なお、ブートデータが格納されたブートブロックにおいて、ブロックフラグは「先頭使用」とされる。「使用」は、当該ブロックがファイルの先頭以外で使用されている状態を示す。ブロックフラグが「使用」のとき、当該ブロックは、他のブロックから連結されていることとなる。「未消去」は、当該ブロックに書かれていたデータが無効となった状態を示す。例えば、データの消去を行うときに、取りあえずブロックフラグを「未消去」にしておき、処理時間に余裕があるときに、ブロックフラグが「未消去」になっているブロックを消去するようにする。これにより、消去処理をより効率良く行うことが可能となる。

【0057】最終フラグは、ファイルが終わっているか否かを示すフラグであり、具体的には、「ブロック連続」「ブロック最終」の2つの状態を示す。「ブロック連続」は、次のブロックへの連結があることを示す。すなわち、「ブロック連続」は、当該ブロックに格納されたファイルにはまだ続きがあり、当該ファイルが他のブロックに続いていることを示す。「ブロック最終」は、

最終ブロックであることを示す。すなわち、「ブロック最終」は、当該ブロックに格納されたファイルが、このブロックで終了していることを示す。

【0058】参照フラグは、追加管理情報の参照を指定するためのフラグであり、具体的には、「参照情報なし」「参照情報あり」の2つの状態を示す。「参照情報なし」は、ブロックの最終ページの冗長領域に、有効な追加管理情報が存在しないことを示す。「参照情報あり」は、ブロックの最終ページの冗長領域に、有効な追加管理情報が存在していることを示す。

【0059】管理フラグは、ブロックの属性等を示すフラグである。例えば、この管理フラグによって、当該ブロックが読み出し専用ブロックか、或いは書き込みも可能なブロックであるかが示される。また、例えば、この管理フラグによって、当該ブロックがブートブロックであるか、或いはデータブロックであるかが示される。

【0060】論理アドレスは、文字通りそのブロックの論理アドレスを示す。この論理アドレスの値は、データの書き換えを行うときなどに必要に応じて更新される。なお、論理アドレスの値は、正常に処理が行われている限り、同じ論理アドレスの値を同時に複数のブロックが持つことがないように設定される。

【0061】ところで、フラッシュメモリの場合、同一ブロック内でデータを書き換えるには、上述したように、先ずブロック消去を行う必要がある。しかしながら、保証されている消去可能回数には上限があり、ブロック消去の回数は出来るだけ少なくすることが要求される。そこで、ブロックのデータを更新する際は、同一のブロックを使って新たなデータに書き換えるのではなく、他のブロックに新たなデータを書き込むようにする。このとき、先にデータが格納されていたブロックは、当該ブロックに格納されていたデータが無効になったことを示すように、ブロックフラグを「未消去」にする。そして、このメモ리카ード2では、このようにデータを更新した場合でも、当該データが格納されているブロックを示すアドレスが同じとなるように、各ブロックに対して予め設定されている物理アドレスとは別に、動的に変更が可能な論理アドレスを各ブロックに割り当て、この論理アドレスでデータが格納されているブロックを表すようにする。

【0062】連結アドレスは、当該ブロックに連結するブロックの論理アドレスを示す。すなわち、ブロックに格納されたファイルにはまだ続きがあり、当該ファイルが他のブロックに続いている場合、連結アドレスには、そのファイルの続きが格納された次のブロックの論理アドレスの値が設定される。

【0063】分散管理情報用エラー訂正符号は、分散管理情報のうち、管理フラグ、論理アドレス、連結アドレス及びリザーブ領域に書き込まれたデータを対象としたエラー訂正符号である。なお、可/不可フラグ、ブロッ

クフラグ、最終フラグ及び参照フラグは、分散管理情報用エラー訂正符号によるエラー訂正の対象となっていない。したがって、可/不可フラグ、ブロックフラグ、最終フラグ及び参照フラグは、分散管理情報用エラー訂正符号を更新することなく書き換えることが可能となっている。

【0064】データ用エラー訂正符号は、当該データ用エラー訂正符号が格納されているページのデータエリアに書き込まれたデータを対象としたエラー訂正符号である。

【0065】なお、分散管理情報用エラー訂正符号やデータ用エラー訂正符号は、メモ리카ード2の内部に配されたエラー訂正回路16によって使用される。したがって、これらのエラー訂正符号を用いてのエラー訂正は、データ処理装置1に依存することなく、メモ리카ード2に依存した任意の手法を使用することができる。

【0066】6. 追加管理情報

つぎに、追加管理情報について詳細に説明する。

【0067】追加管理情報は、ブロックの最終ページの16バイトの冗長エリアに格納される情報であり、分散管理情報だけでは管理しきれない追加情報を含んでいる。

【0068】具体的には、追加管理情報は、図6に示すように、1バイトの可/不可フラグと、1バイトのブロックフラグと、4ビットの最終フラグと、4ビットの参照フラグと、1バイトの識別番号と、2バイトの有効データサイズと、5バイトのリザーブ領域と、2バイトの追加管理情報用エラー訂正符号と、3バイトのデータ用エラー訂正符号とからなる。

【0069】ここで、可/不可フラグ、ブロックフラグ、最終フラグ、参照フラグ、リザーブ領域及びデータ用エラー訂正符号については、分散管理情報の場合と同様である。また、追加管理情報用エラー訂正符号は、分散管理情報における分散管理情報用エラー訂正符号に相当するものであり、追加管理情報のうち、識別番号、有効データサイズ及びリザーブ領域に書き込まれたデータを対象としたエラー訂正符号である。

【0070】そして、識別番号及び有効データサイズとが、分散管理情報だけでは管理しきれない追加情報として、追加管理情報に含まれている。

【0071】識別番号は、エラー処理用の情報であり、ブロックのデータを書き換える度に、この識別番号の値がインクリメントされる。この識別番号は、何らかのエラーが発生して、同じ論理アドレスを持つブロックが複数存在するようになってしまった場合に、それらのブロックに書き込まれたデータの新旧を識別するために使用される。なお、識別番号には1バイトの領域が使用され、その値の範囲は「0」から「255」までであり、その初期値は「0」とされる。なお、識別番号が「255」を越えたときには「0」に戻される。そして、同じ

論理アドレスを持つデータブロックが複数存在する場合には、この識別番号の値が小さい方のデータブロックを有効とする。ただし、ブートブロックについては、ブートブロックの予備がある場合、正常時にはそれらのブートブロックの識別番号は同じ値とされる。何らかの異常により、それらのブートブロックの識別番号が異なるような状態となった場合には、識別番号の値が大きい方のブートブロックを有効とする。

【0072】また、有効データサイズは、ブロック内の有効なデータのサイズを示す。すなわち、当該ブロックのデータエリアに空きがある場合、有効データサイズには、当該データエリアに書き込まれたデータのサイズを示す値が設定される。このとき、分散管理情報の参照フラグは「参照情報あり」に設定される。なお、ブロックのデータエリアに空きがない場合、有効データサイズには、当該データエリアに空きがないことを示す値として、「0xffff」が設定される。

【0073】なお、以上のような分散管理情報及び追加管理情報は、ブロック内のデータが更新される毎に、常に最新情報となるように更新される。

【0074】7. 集合管理情報
つぎに、集合管理情報について詳細に説明する。

【0075】集合管理情報は、上述したように、各ブロックの分散管理情報を集めて作成されてなる情報であり、ファイルとしてフラッシュメモリ 12 に格納される。すなわち、図7に示すように、各ブロックの分散管理情報から、全ブロックをまとめて管理するための情報である集合管理情報のファイルが作成され、この集合管理情報が所定のブロックのデータエリアに格納される。なお、集合管理情報は、1つのブロックに格納されるものであっても、複数のブロックにわたって格納されるものであってもよい。そして、データ処理装置1は、通常は、この集合管理情報によって、各ブロックにアクセスするために必要な情報を得るようにする。

【0076】すなわち、メモリカード2に有効な集合管理情報がファイルとして格納されている場合、データ処理装置1は、その集合管理情報のファイルを読み出して内部メモリ4に展開し、メモリカード2を管理するための管理テーブルを作成する。なお、集合管理情報のファイルの先頭が格納されているブロックの物理アドレスは、ブートデータに含まれており、データ処理装置1は、この物理アドレスに基づいて集合管理情報のファイルにアクセスする。

【0077】この集合管理情報は、図8に示すように、この集合管理情報のヘッダーと、各ブロックの状態を示すビットマップテーブルと、ブロックにアクセスするときに、指定された論理アドレスから物理アドレスへの変換を行うための変換テーブルと、あるブロックの次のブロックを示す連結テーブルとを有する。

【0078】ビットマップテーブルには、各ブロックの

分散管理情報から抽出された、可/不可フラグ、ブロックフラグ、最終フラグ、参照フラグ及び管理フラグ等の情報が格納される。

【0079】変換テーブルは、図9に示すように、論理アドレスに対応する物理アドレスが記述されたテーブルであり、物理アドレスが格納される領域は、1エントリあたり2バイトとされる。この変換テーブルを分散管理情報から作成するときは、対象となるブロックの分散管理情報に書かれている論理アドレスを調べ、テーブルの対応位置にそのブロックの物理アドレスを登録する。なお、論理アドレスを使用していない場合、対応する物理アドレスには「0xffff」を設定しておく。

【0080】連結テーブルは、図10に示すように、論理アドレスに対応する連結アドレスが記述されたテーブルであり、連結アドレスが格納される領域は、1エントリあたり2バイトとされる。この連結テーブルを分散管理情報から作成するときは、対象となるブロックの分散管理情報に書かれている連結アドレスを調べ、テーブルの対応位置にそのブロックの連結アドレスを登録する。

【0081】8. メモリカード起動時の手順
つぎに、メモリカード2の起動時の手順について、図11のフローチャートを参照して説明する。

【0082】このメモリカード2を起動する際は、図11に示すように、先ず、ステップS1において、データ処理装置1は、メモリカード2のブートブロックからブートデータを読み込む。次に、ステップS2へ進む。

【0083】ステップS2において、データ処理装置1は、ブートブロックからのブートデータの読み込みが正常に行われたかを確認する。正常に読み込まれたならば、ステップS3へ進む、正常に読み込まれなかったならば、ステップS8へ進む。

【0084】ステップ3において、データ処理装置1は、読み込んだブートデータに基づいて、メモリカード2が当該データ処理装置1に対応しているか否かを判別する。対応したメモリカードであるならば、ステップS4へ進む、対応していないメモリカードならば、ステップS8へ進む。

【0085】ステップS4において、データ処理装置1は、メモリカード2から集合管理情報を読み込む。なお、集合管理情報が格納されているブロックの物理アドレスは、ブートデータの中で指定されている。次に、ステップS5へ進む。

【0086】ステップS5において、データ処理装置1は、有効な集合管理情報の読み込みが正常に行われたかを確認する。正常に読み込んだならば、ステップS6へ進む、正常に読み込まなかったならば、ステップS7へ進む。

【0087】ステップS6において、データ処理装置1は、読み込んだ集合管理情報を内部メモリ4に展開し、メモリカード2を管理するための管理テーブルを作成す

る。以上の処理で、メモ리카ード2の起動時の初期処理が完了し、メモ리카ード2の使用が可能となる。

【0088】また、ステップS5で有効な集合管理情報が正常に読み込めなかったと判断された場合は、上述したようにステップS7へ進む。ステップS7において、データ処理装置1は、各ブロックの分散管理情報を読み出して、集合管理情報を再構築する。そして、その集合管理情報を内部メモリ4に展開し、メモ리카ード2を管理するための管理テーブルを作成する。以上の処理で、メモ리카ード2の起動時の初期処理が完了し、メモ리카ード2の使用が可能となる。

【0089】一方、ステップS2でブートデータ読み込み時にエラーが生じたと判断された場合、及び、ステップS3でメモ리카ード2がデータ処理装置1に対応していないと判断された場合は、上述したようにステップS8へ進む。

【0090】ステップS8に進むのは、メモ리카ード2を使用できないときである。そこで、ステップS8において、データ処理装置1は、例えば、利用不可のメッセージを表示するなどの所定のエラー処理を行い、メモ리카ード2の起動処理を終了する。

【0091】9. データ更新処理時の集合管理情報の取り扱い

データ処理装置1は、メモ리카ード2へデータを書き込む処理や、メモ리카ード2からデータを消去する処理（以下、これらの処理をまとめて「データ更新処理」と称する。）を行う毎に、内部メモリ4に保持している管理テーブルを、メモ리카ード2の実際の状態と整合するように（すなわち、分散管理情報の内容と整合するように）、随時更新していく。一方、メモ리카ード2にファイルとして格納されている集合管理情報は、データ更新処理毎に更新されるのではなく、適当なタイミングにて、その変更内容が一括して更新される。

【0092】一般にフラッシュメモリ12の書き換え可能回数には上限があるが、集合管理情報の書き換えをある程度まとめて一括して行うようにすることで、集合管理情報が格納されているブロックの書き換え回数を削減することができ、メモ리카ード2の長寿命化を図ることができる。

【0093】ただし、データ更新処理を行う際は、メモ리카ード2に格納されている集合管理情報のファイルを無効にしてから行う。これは、分散管理情報と集合管理情報の一貫性を損なわないためである。データ更新処理時には、処理の対象となるブロックの分散管理情報は同時に更新されるが、集合管理情報の内容は同時には更新されないため、分散管理情報と集合管理情報が一致しない状態となる。そこで、このような状態のときには、メモ리카ード2に格納されている集合管理情報のファイルを無効しておく。

【0094】具体的には、データ更新処理時には、図1

2に示すように、まず、ステップS11において、データ処理装置1は、メモ리카ード2に格納されている集合管理情報が有効であるか無効であるかを判別する。そして、集合管理情報が既に無効となっていたならば、そのままデータ更新処理に移行する。一方、集合管理情報が有効であれば、ステップS12へ進む。

【0095】ステップS12において、データ処理装置1は、集合管理情報を無効にする。具体的には、集合管理情報のファイルが格納されているブロックのブロックフラグを「未消去」にするか、或いは当該ブロックに対して消去処理を施してデータを消去する。そして、このように集合管理情報を無効にした上で、データ更新処理に移行する。

【0096】なお、メモ리카ード2に格納されている集合管理情報のファイルは、分散管理情報と集合管理情報の一貫性を損なわないために、データ更新処理時に無効とされるが、データ処理装置1の内部メモリ4に保持されている管理テーブルの内容は、常に最新の状態となるように随時更新される。そして、データ処理装置1は、通常は、この管理テーブルに基づいて各ブロックを管理する。

【0097】また、データ更新処理時に無効とされた集合管理情報は、適当なタイミングで、改めてメモ리카ード2に書き込まれて、再び有効とされる。なお、ここでの適当なタイミングとは、例えば、メモ리카ード2の使用を終了して電源を落とすときや、メモ리카ード2へのアクセスが所定時間以上なされなかったときや、データの書き換えが所定回数以上行われたときなどである。

【0098】具体的には、例えば、メモ리카ード2の使用を終了して電源を落とす前に、図13に示すような終了処理を行い、集合管理情報を有効なものとする。

【0099】この終了処理では、まず、ステップS21において、データ処理装置1は、メモ리카ード2に格納されている集合管理情報が有効であるか無効であるかを判別する。そして、集合管理情報が有効であれば、そのまま処理を終了する。一方、集合管理情報が無効であれば、ステップS22へ進む。

【0100】ステップS22において、データ処理装置1は、集合管理情報のファイルが格納されているブロックに対して、消去処理が施されているか否かを判別する。消去処理が施されていないならば、ステップS23へ進み、消去処理が施されていれば、ステップS24へ進む。

【0101】ステップS23において、データ処理装置1は、集合管理情報のファイルが格納されているブロックに対して消去処理を施す。その後、ステップS24へ進む。

【0102】ステップS24において、データ処理装置1は、メモ리카ード2に集合管理情報を書き込む。このとき、データ処理装置1は、内部メモリ4に保持してい

る管理テーブルの内容に基づいて新しい集合管理情報のファイルを作成し、その新しい集合管理情報のファイルをメモリカード2に書き込む。これにより、メモリカード2の最新の状態を示す有効な集合管理情報が、メモリカード2に格納されたこととなる。

【0103】以上で終了処理が完了し、メモリカード2に有効な集合管理情報が格納された状態となる。

【0104】10. 新規ファイルの書き込み

つぎに、メモリカード2に新規なファイルを書き込む際の処理手順について説明する。メモリカード2にファイルを書き込む際の処理手順は、ファイルサイズが予め分かっている場合と、分かっている場合とで異なる。

【0105】10-1 ファイルサイズが予め分かっている場合

予めファイルのサイズが分かっている場合は、当該ファイルのデータを新規なブロックに書き込む毎に、データが当該ブロックに納まるかどうかを判断する。そして、データがブロック内に納まりきらない場合には、次に続くブロックの論理アドレスを確保しておき、データをデータエリアに書き込むとともに、次に続くブロックの論理アドレスを連結アドレスとして分散管理情報を書き込む。このとき、最終フラグは「ブロック連続」に設定しておく。一方、データがブロック内に納まる場合には、データの端数部分、すなわちデータエリアの空き領域は「0xffff」としておく。このとき、最終フラグは「ブロック最終」に設定しておき、追加管理情報に有効データサイズを書き込んでおく。

【0106】つぎに、以上のように予めサイズが分かっているファイルをメモリカード2に書き込む際の手順について、図14に示すフローチャートを参照して、詳細に説明する。なお、図14に示すフローチャート、並びに後掲する図15及び図17に示すフローチャートでは、メモリカード2の誤消去防止スイッチ20のチェックや、何らかのエラーが発生したときの処理等については省略している。

【0107】予めサイズが分かっているファイルをメモリカード2に書き込む際は、まず、ステップS31において、データ処理装置1は、メモリカード2に書き込む実データと、当該実データのヘッダーとを準備する。換言すれば、ステップS31において、データ処理装置1は、メモリカード2のデータエリアに書き込むファイルを準備する。なお、当該ファイルのヘッダーには、ファイルのサイズの情報が含まれる。次に、ステップS32へ進む。

【0108】ステップS32において、データ処理装置1は、最初にファイルが格納されるブロックのブロックフラグを「先頭使用」に設定するとともに、空いている論理アドレスを確保する。次に、ステップS33へ進む。

【0109】ステップS33において、データ処理装置

1は、空いている物理アドレスを検索する。次に、ステップS34へ進む。

【0110】ステップS34において、データ処理装置1は、処理の対象となっているブロックにファイルが納まりきるか否かを判別する。ファイルがブロック内に納まりきらず、ファイルに続きがある場合には、ステップ35へ進み、一方、ファイルがブロック内に納まり、ファイルに続きがない場合には、ステップ36へ進む。

【0111】ステップS35において、データ処理装置1は、次に続くブロックの論理アドレスを確保して、この論理アドレスを連結アドレスとして設定する。次に、ステップS37へ進む。

【0112】一方、ステップS36において、データ処理装置1は、最終フラグを「ブロック最終」に設定するとともに、連結アドレスを「0xffff」に設定する。次に、ステップS37へ進む。

【0113】ステップS37において、データ処理装置1は、これまでのステップで設定された情報等に基づいて、処理の対象となっているブロックについての分散管理情報を作成する。次に、ステップS38へ進む。

【0114】ステップS38において、データ処理装置1は、処理の対象となっているブロックに、ページ単位でデータを順次書き込む。ここで、処理の対象となっているブロックにファイルが納まりきらない場合には、1ブロック分のデータがページ単位で書き込まれる。また、処理の対象となっているブロックにファイルが納まりきる場合には、必要なページ分だけページ単位でデータが書き込まれる。なお、このステップS38で、ブロックに書き込まれるのは、新規に書き込むファイルのデータと、ステップS37で作成された分散管理情報とである。次に、ステップS39へ進む。

【0115】ステップS39において、データ処理装置1は、ファイルの全データについて、メモリカード2への書き込みが終了したか否かを判別する。書き込みが終了しておらず、まだデータが残っていれば、ステップS33へ戻って処理を繰り返す。一方、書き込みが終了していれば、ステップ40へ進む。

【0116】ステップS40において、データ処理装置1は、書き込んだデータが、ブロックの途中で終わっているか否かを判別する。そして、データがブロックの途中で終わっていれば、ステップS41へ進む。一方、ブロックの最後までデータが格納されていれば、これで処理を終了する。

【0117】ステップS41において、データ処理装置1は、最終ページの冗長エリアに格納される追加管理情報に有効データサイズを書き込む。すなわち、データ処理装置1は、ファイルの最後の部分が格納されたブロックのデータエリアに書き込まれたデータのサイズを示す値を、当該ブロックの追加管理情報に有効データサイズとして書き込む。

【0118】以上で、予めサイズが分かっているファイルのメモリカード2への書き込みの処理が完了する。

【0119】10-2 ファイルサイズが分かっている場合

ファイルのサイズが予め分からない場合には、次に続くブロックの論理アドレスを常に確保しておき、データが終了した時点で、最終ブロックの最終フラグをオーバーライトにより設定する。なお、その他の分散管理情報及び追加管理情報に関しては、予めファイルサイズが分かっているときと同様に設定される。

【0120】サイズが予め分かっているファイルメモリカード2に書き込む際の手順について、図15に示すフローチャートを参照して、詳細に説明する。

【0121】サイズが予め分かっているファイルメモリカード2に書き込む際は、先ず、ステップS51において、データ処理装置1は、メモリカード2に書き込むファイルの仮のヘッダーを作成する。この段階では、ファイルサイズが不明なので、この仮のヘッダーには、ファイルサイズの情報が含まれていない。次に、ステップS52へ進む。

【0122】ステップS52において、データ処理装置1は、最初にファイルが格納されるブロックのブロックフラグを「先頭使用」に設定するとともに、空いている論理アドレスを確保する。次に、ステップS53へ進む。

【0123】ステップS53において、データ処理装置1は、メモリカード2に書き込むデータを準備する。次に、ステップS54へ進む。

【0124】ステップS54において、データ処理装置1は、メモリカード2に書き込むデータが残っているか否かを判別する。データが終了しておらず、まだデータが残っていれば、ステップS55へ進む。一方、データが終了しており、データが残っていなければ、ステップS61へ進む。

【0125】ステップS55において、データ処理装置1は、空いている物理アドレスを検索する。次に、ステップS56へ進む。

【0126】ステップS56において、データ処理装置1は、次に続くブロックの論理アドレスを確保して、この論理アドレスを連結アドレスとして設定する。次に、ステップS57へ進む。

【0127】ステップS57において、データ処理装置1は、これまでのステップで設定された情報等に基づいて、処理の対象となっているブロックについての分散管理情報を作成する。次に、ステップS58へ進む。

【0128】ステップS58において、データ処理装置1は、処理の対象となっているブロックに、ページ単位でデータを順次書き込む。ここで、処理の対象となっているブロックにファイルが納まりきらない場合には、1ブロック分のデータがページ単位で書き込まれる。ま

た、処理の対象となっているブロックにファイルが納まりきる場合には、必要なページ分だけページ単位でデータが書き込まれる。なお、このステップS58でブロックに書き込まれるのは、新規に書き込むファイルのデータと、ステップS57で作成された分散管理情報とである。次に、ステップS59へ進む。

【0129】ステップS59において、データ処理装置1は、書き込んだデータが、ブロックのデータエリアの途中で終わっているか否かを判別する。そして、データエリアの最後に至るまでデータが格納されていれば、ステップS53へ戻って処理を繰り返す。一方、データがデータエリアの途中で終わってしまえば、ステップS60へ進む。

【0130】ステップS60において、データ処理装置1は、処理の対象となっているブロックの最終ページの冗長エリアに格納される追加管理情報に、有効データサイズを書き込む。すなわち、データ処理装置1は、ファイルの最後の部分が格納されたブロックのデータエリアに書き込まれたデータのサイズを示す値を、当該ブロックの追加管理情報に有効データサイズとして書き込む。次に、ステップS61へ進む。

【0131】ステップS61において、データ処理装置1は、処理の対象となっているブロックの最終フラグを「ブロック最終」にオーバーライトにより設定する。次に、ステップS62へ進む。

【0132】ステップS62において、データ処理装置1は、ファイルのヘッダーを更新する。すなわち、この段階では、ファイルサイズが明らかとなっているので、ファイルサイズの情報を含むヘッダーを新たに作成して、上述した仮のヘッダーを、ファイルサイズの情報を含む新たなヘッダーに書き換える。

【0133】以上で、予めサイズが分かっていたファイルのメモリカード2への書き込みの処理が完了する。

【0134】11. ファイルの更新
つぎに、メモリカード2に格納されているファイルを更新する際の処理手順について説明する。

【0135】ファイルの更新時には、データの書き換えの対象となるブロックと同じ論理アドレスを別のブロックに付して、当該ブロックに対して新しいデータを書き込む。このとき、古いデータが書かれているブロックは、ファイルの更新が終了するまで開放せずに保持しておく。これにより、ファイル更新中に障害が発生したとしても、ファイル更新前の状態に復旧することが可能となる。

【0136】このようなファイル更新の手順の具体的な例を図16を参照して説明する。

【0137】図16(a)に示すように、ファイルの先頭が、論理アドレス「1」のブロックに格納され、ファイルの次の部分が、論理アドレス「2」のブロックに格

納され、ファイルの次の部分が、論理アドレス「3」のブロックに格納されていたとする。また、論理アドレス「1」のブロックの識別番号は「6」、論理アドレス「2」のブロックの識別番号は「4」、論理アドレス「3」のブロックの識別番号は「1」であったとする。

【0138】そして、このような状態のときに、論理アドレス「2」のブロックのデータを書き換えるとする。このときは、先ず、図16(b)に示すように、空いている別にブロックに論理アドレス「2」を割り当て、このブロックに新しいデータを書き込む。ここで、新しいデータを書き込むブロックの識別番号には、古いデータが書き込まれているブロックの識別番号を1インクリメントした値、すなわち「5」を設定する。

【0139】この段階では、同じ論理アドレスを持つブロックが2つ存在していることとなる。そして、これらの2つのブロックのうち、識別番号が大きい方のブロックに格納されているデータが新しい方のデータであり、識別番号が小さい方のブロックに格納されているデータが古い方のデータとなる。

【0140】そして、新しいデータの書き込みが正常に完了したら、次に、図16(c)に示すように、古いデータが書き込まれていたブロックを消去する。なお、このときは、古いデータが書き込まれていたブロックに対して消去処理を施すのではなく、該当するブロックのブロックフラグを「未消去」にするだけにしておき、後から適当なタイミングで、このブロックに対して消去処理を施すようにしてもよい。

【0141】以上のような処理の手順について、図17に示すフローチャートを参照して、詳細に説明する。

【0142】ファイルを更新する際は、先ず、ステップS71において、データ処理装置1は、更新の対象となるブロックを選択する。次に、ステップS72へ進む。

【0143】ステップS72において、データ処理装置1は、更新の対象となるブロックの識別番号を読み出し、その値を1インクリメントした値を、新しいデータを書き込むブロックの識別番号として設定する。また、新しいデータを書き込むブロックの論理アドレスとして、更新の対象となるブロックの論理アドレスと同じ値を設定する。次に、ステップS73へ進む。

【0144】ステップS73において、データ処理装置1は、ブロックに書き込む新しいデータを準備する。次に、ステップS74に進む。

【0145】ステップS74において、データ処理装置1は、空いている物理アドレスを検索する。次に、ステップS75に進む。

【0146】ステップS75において、データ処理装置1は、データの変更が全て完了しているか否かを判別する。完了していなければ、ステップS76へ進み、完了していれば、ステップS79へ進む。

【0147】ステップS76において、データ処理装置

1は、次に続くブロックの論理アドレスを確保して、この論理アドレスを連結アドレスとして設定する。次に、ステップS77へ進む。

【0148】ステップS77において、データ処理装置1は、これまでのステップで設定された情報等に基づいて、新しいデータを書き込むブロックについての分散管理情報を作成する。次に、ステップS78へ進む。

【0149】ステップS78において、データ処理装置1は、ステップS74で検索された物理アドレスのブロックに、ページ単位で新しいデータを順次書き込む。ここで、処理の対象となっているブロックにファイルが納まりきらない場合には、1ブロック分のデータがページ単位で書き込まれる。また、処理の対象となっているブロックにファイルが納まりきる場合には、必要なページ分だけページ単位でデータが書き込まれる。なお、このステップS78で、ブロックに書き込まれるのは、新しいファイルのデータと、ステップS77で作成された分散管理情報とである。そして、このステップS78の後には、ステップS73へ戻って処理を繰り返す。

【0150】一方、ステップS79において、データ処理装置1は、最後に更新の対象となっていたブロックに、連結アドレスが設定されているか否かを判別する。連結アドレスが設定されていないならば、ステップS80へ進み、連結アドレスが設定されていたならば、ステップS81へ進む。

【0151】ステップS80において、データ処理装置1は、最後に新しいデータを書き込んだブロックの最終フラグを「ブロック最終」に設定する。次に、ステップS82へ進む。

【0152】一方、ステップS81において、データ処理装置1は、最後に新しいデータを書き込んだブロックの連結アドレスに、最後に更新の対象となっていたブロックに設定されていた連結アドレスの値を設定する。次に、ステップS82へ進む。

【0153】ステップS82において、データ処理装置1は、ファイルのヘッダーを更新する。すなわち、ファイルの更新により、ファイルサイズが変更となっている可能性があるため、新しいファイルサイズの情報を含むヘッダーを新たに作成して、ファイルのヘッダーを更新する。次に、ステップS83へ進む。

【0154】ステップS83において、データ処理装置1は、古いデータが書き込まれていたブロックを消去する。なお、このときは、古いデータが書き込まれていたブロックに対して消去処理を施すのではなく、該当するブロックのブロックフラグを「未消去」にするだけにしておき、後から適当なタイミングで、これらのブロックに対して消去処理を施すようにしてもよい。

【0155】以上でファイルの更新処理が完了する。

【0156】12. エラー検出訂正処理

以上のようなシステムでは、メモ리카ード2に新規ファ

イルを書き込んでいるときや、メモリカード2に格納されているファイルを更新しているときなどに、いきなり電源が遮断されたり、メモリカード2がデータ処理装置1から強制的に取り外されたりすると、同じ論理アドレスを持つ複数のブロックが同時に存在するような状態（すなわち論理アドレスエラー）となってしまうたり、或いは、連結アドレスで指し示されたブロックが存在しないような状態（すなわち連結アドレスエラー）となってしまうたりする可能性がある。

【0157】そこで、本発明を適用したシステムでは、集合管理情報を構築する際に、論理アドレスエラーや連結アドレスエラーを検出して訂正するエラー検出訂正処理を行うようにする。以下、このエラー検出訂正処理について、詳細に説明する。

【0158】12-1 エラー検出テーブル

本発明を適用したシステムでは、集合管理情報を構築する際に、論理アドレスエラーや連結アドレスエラーの検出を行う。そして、連結アドレスエラーを検出するために使用するテーブルとして、エラー検出テーブルを使用する。エラー検出テーブルは、連結アドレスエラーの検出にだけ使用されるテーブルであり、エラー検出訂正処理を行う際に、データ処理装置1の内部メモリ4に一時的に確保される。エラー検出テーブルのために確保されていた領域は、エラー検出訂正処理が終了した後、開放される。

【0159】このエラー検出テーブルは、図18に示すように、1論理アドレスに対して、各ブロックの連結状態を示す1ビットの領域を備えたテーブルである。換言すれば、エラー検出テーブルは、1エントリあたり1ビットとなっており、各エントリは論理アドレスの連結状態を「0」又は「1」で示す。処理の対象となるブロックがN個ある場合、このエラー検出テーブルが占める領域は、 $N/8$ バイトとなる。

【0160】このエラー検出テーブルは、集合管理情報を構築しているときと、集合管理情報の構築が終了したときとで、各ブロックの連結状態を示す値の意味合いが異なる。

【0161】集合管理情報を構築している最中において、連結状態を示す値が「0」のとき、そのエントリは、正常な状態であることを示しているか、或いは、当該エントリに対応する論理アドレスが、現在処理の対象となっているブロックまでの間に他のブロックの連結アドレスで指定されていないことを示している。この状態のときは、今後の処理が進むに従って、値が「1」になる可能性があり、連結アドレスエラーであるかどうか不確定な状態である。

【0162】また、集合管理情報を構築している最中において、連結状態を示す値が「1」のとき、そのエントリは、当該エントリに対応する論理アドレスが、現在処理の対象となっているブロックまでの間に他のブロック

の連結アドレスとして指定されているが、物理アドレスに対応していない状態を示す。この状態のときは、今後の処理が進むに従って、値が「0」になる可能性があり、連結アドレスエラーであるかどうか不確定な状態である。

【0163】一方、集合管理情報の構築が終了した段階で、連結状態を示す値が「1」のとき、そのエントリは、当該エントリに対応する論理アドレスが連結アドレスとして指定されているのに、対応する物理アドレスが存在しないことを示している。したがって、この状態のときは、連結アドレスエラーである。

【0164】また、集合管理情報の構築が終了した段階で、連結状態を示す値が「0」のとき、そのエントリは、当該エントリに対応する論理アドレスに関する連結が正常な状態となっていることを示している。

【0165】12-2 連結アドレスエラーの検出
つぎに、以上のようなエラー検出テーブルを用いて行われる、連結アドレスエラーの検出について、具体的な例を挙げて説明する。

【0166】例えば、図19に示すように、物理アドレス「10」のブロックについて、その論理アドレスが「1」、その連結アドレスが「3」であり、また、物理アドレス「17」のブロックについて、その論理アドレスが「3」、その連結アドレスが「2」であったとする。また、論理アドレスが「2」のブロックは存在しないとする。

【0167】このとき、集合管理情報の再構築の処理が行われると、集合管理情報の再構築に伴って、図20に示すように、連結アドレスエラーの検出が行われる。

【0168】先ず、初期状態では、図20(a)に示すように、論理アドレス「1」「2」「3」のそれぞれについて、エラー検出テーブルの値は、すべて初期値「0」とする。また、集合管理情報の変換テーブルも、全て初期値「0xffff」とする。このとき、集合管理情報の連結テーブルは、全く値が入っていない状態とする。

【0169】次に、論理アドレス「1」のブロックについての情報を読み込む。これにより、図20(b)に示すように、論理アドレス「1」に対応する変換テーブルの値は、論理アドレス「1」のブロックの物理アドレスの値、すなわち「10」とされる。また、論理アドレス「1」に対応する連結テーブルの値は、論理アドレス「1」のブロックの連結アドレスの値、すなわち「3」とされる。

【0170】次に、論理アドレス「1」に対応する連結テーブルの値が指し示す論理アドレス「3」のエントリを確認する。このとき、論理アドレス「3」のエントリには、物理ブロックが割り当てられていないので、図20(c)に示すように、論理アドレス「3」に対応するエラー検出テーブルの値を「1」とする。また、論理アドレス「3」に対応する変換テーブルの値は、連結元の

論理アドレスの値、すなわち「1」とする。

【0171】次に、エラー検出テーブルの値が「1」となっている論理アドレス「3」のブロックについて、その情報を読み込む。このとき、論理アドレス「3」のブロックは存在しており、当該ブロックの情報は正常に読み込むことができる。したがって、図20(d)に示すように、論理アドレス「3」に対応するエラー検出テーブルの値を「0」とする。また、このとき、論理アドレス「3」に対応する変換テーブルの値は、論理アドレス「3」のブロックの物理アドレスの値、すなわち「17」とする。また、論理アドレス「3」に対応する連結テーブルの値は、論理アドレス「3」のブロックの連結アドレスの値、すなわち「2」とする。

【0172】次に、論理アドレス「3」に対応する連結テーブルの値が指し示す論理アドレス「2」のエントリを確認する。このとき、論理アドレス「2」のエントリには、物理ブロックが割り当てられていないので、図20(e)に示すように、論理アドレス「2」に対応するエラー検出テーブルの値を「1」とする。また、論理アドレス「2」に対応する変換テーブルの値は、連結元の論理アドレスの値、すなわち「3」とする。

【0173】次に、エラー検出テーブルの値が「1」となっている論理アドレス「2」のブロックについて、その情報の読み込みを試みる。しかしながら、論理アドレスが「2」のブロックは存在していないので、この段階で、連結アドレスエラーであることが判明する。

【0174】12-3 エラー訂正処理

本発明を適用したシステムにおいて、論理アドレスエラーに対するエラー訂正処理は、以下のように行う。

【0175】論理アドレスエラーが生じているときには、同じ論理アドレスを持つブロックをそれぞれ調べる。そして、ブロックとして完全なものが1つしかない場合には、完全なブロックを生かして、残りのブロックは無効とする。

【0176】また、同じ論理アドレスを持つ完全なブロック（ブートブロックを除く。）が複数ある場合には、識別番号を比較して値が小さい方のブロックを生かす。なお、一方のブロックの識別番号が「255」であり、他方のブロックの識別番号が「0」の場合には、識別番号が「255」のブロックを生かすようにする。

【0177】なお、通常は、同じ論理アドレスを持つブロックが複数あったとしても、それらの識別番号の差は1である。この条件に当てはまらないような場合には、システムの側で自動的にエラー訂正処理を行うのではなく、手動復旧モードとする。

【0178】また、本発明を適用したシステムにおいて、連結アドレスエラーが生じた場合には、メモリアカード2を使用するアプリケーションソフトウェアやメモリアカード2に格納するデータ等に応じた適切なエラー訂正処理を行うようにする。具体的には、例えば、以下に挙

げるようなエラー訂正処理を行うようにすればよい。

【0179】すなわち、例えば、最後の連結アドレスが指し示すブロックとして新規ブロックを割り当てる。そして、最後のブロックのデータを正しく読めるページまで読み込んで、そのページまでのデータを新規ブロックにコピーする。このとき、新規ブロックの最終フラグは「ブロック最終」としておく。このようなエラー訂正処理は、対象となるデータが音楽データ等のようにデータの途中でも意味があるデータの場合に、特に好適である。

【0180】12-4 集合管理情報の構築とエラー検出訂正処理

本発明を適用したシステムにおいて、物理アドレスエラーや連結アドレスエラーが発生するのは、データ更新処理の途中に何らかの障害が発生した場合である。そして、本発明を適用したシステムでは、上述したように、データ更新処理に先だってメモリアカード2に格納されている集合管理情報を無効にするようにしている。したがって、物理アドレスエラーや連結アドレスエラーが発生するときには、集合管理情報が無効となっている。そして、集合管理情報が無効となっているときには、次にメモリアカード2を起動するとき、全てのブロックの分散管理情報を調べ直して集合管理情報を再構築する処理が必ず行われる。

【0181】そこで、このシステムでは、集合管理情報を再構築する際に、全てのブロックを調べ直すということに着目し、このときにエラー検出訂正処理を同時に行う。換言すれば、集合管理情報が有効となっているときには、物理アドレスエラーや連結アドレスエラーが発生している可能性はないので、このときにはエラー検出訂正処理を行わない。すなわち、この方法では、集合管理情報の再構築時にだけ、エラー検出訂正処理を行う。これにより、エラー検出訂正処理のためにメモリアカード2に余分にアクセスする必要がなくなる。その結果、例えば、メモリアカード2の速やかな起動が可能となる。

【0182】このエラー検出訂正処理は、以下のような手順によって行われる。

【0183】(1) データ処理装置1の内部メモリ4上の変換テーブルを全て「0xffff」で初期化する。また、内部メモリ4上にエラー検出テーブルの領域を確保し、当該エラー検出テーブルを全て「0」で初期化する。

【0184】(2) ブロックの先頭に移動する。

【0185】(3) ブロックから分散管理情報を読み込み、当該分散管理情報を用いてビットマップテーブルを構築する。このとき、可/不可フラグが「使用不可」であるか、或いはブロックフラグが「未使用」又は「未消去」である場合には、ビットマップテーブルの作成が終了したら、次のブロックへ移動して処理を繰り返す。

【0186】(4) ブロックの論理アドレス（以下、論理アドレスAとする。）と、連結アドレス（以下、論理

アドレスBとする。)とを調べる。

【0187】(5)論理アドレスAが「0xffff」の場合は、次のブロックへ移動し、(3)へ戻って処理を繰り返す。

【0188】(6)エラー検出テーブルの論理アドレスAの欄を調べる。エラー検出テーブルの論理アドレスAの欄が「1」となっている場合は、「0」に書き換えるとともに、変換テーブルの論理アドレスAの欄に、論理アドレス「A」のブロックの物理アドレスを書き込む。また、エラー検出テーブルの論理アドレスAの欄が「0」となっている場合は、変換テーブルの論理アドレスAの欄を調べる。そして、変換テーブルの論理アドレスAの値が「0xffff」のときは、そこに、論理アドレスAのブロックの物理アドレスを書き込む。

【0189】なお、変換テーブルの論理アドレスAの値として、「0xffff」以外の値が既に入っている場合は、論理アドレスエラーが発生している場合であるので、論理アドレスエラーに対するエラー訂正処理を行う。

【0190】(7)連結テーブルの論理アドレスAの欄に、論理アドレスBの値を記入する。

【0191】(8)最終フラグが「ブロック最終」となっているかを調べる。「ブロック最終」となっていれば、連結アドレスは無効なので、次のブロックに移動し、(3)へ戻って処理を繰り返す。

【0192】(9)論理アドレスBに物理アドレスが対応しているかどうかを、変換テーブルを用いて確認する。変換テーブルの論理アドレスBの値が「0xffff」以外の場合は、論理アドレスBに物理アドレスが対応している。一方、変換テーブルの論理アドレスBの値が「0xffff」の場合は、現在のブロックまでの段階では、論理アドレスBに物理アドレスが対応していない。このときは、エラー検出テーブルの論理アドレスBの欄に「1」を書き込むとともに、変換テーブルの論理アドレスBの欄に論理アドレスAの値を記入する。その後、次のブロックに移動し、(3)へ戻って処理を繰り返す。

【0193】なお、変換テーブルの論理アドレスBの値が「0xffff」となっても、ブロックの途中までしか処理を行っていない段階では、論理アドレスBに物理アドレスが本当に対応していないのかは明らかではない。すなわち、論理アドレスBに物理アドレスが本当に対応していない場合と、今後処理を進めて行くに従って、対応するブロックが現れる場合との2通りがあり得る。

【0194】(10)全てのブロックに対して処理を行った後、エラー検出テーブルを参照する。エラー検出テーブルの値が「1」となっている論理アドレスは、物理アドレスが対応していない。すなわち、連結アドレスエラーが発生している。このときは、変換テーブル内に連結元のブロックの論理アドレスが格納されているので、これを用いて元ブロックを特定し、適切なエラー訂正処理を行う。なお、エラー訂正処理を行った後は、エラー

検出テーブルの値を「0」にし、該当する変換テーブルの値を「0xffff」にしておく。

【0195】本発明を適用したシステムでは、以上のように、集合管理情報を構築するときにエラー検出訂正処理を行う。以下、このような集合管理情報の構築及びエラー検出訂正処理の具体的な方法について、図21乃至図23に示すフローチャートを参照して、更に詳細に説明する。

【0196】なお、ここでは、変数としてI, A, B, C, D, T (I)を使用し、定数としてNを使用する。変数Iは、物理アドレスが入力される変数であり、変数Aは、論理アドレスが入力される変数であり、変数Bは、連結アドレスが入力される変数であり、変数C, Dは、識別番号の値が入力される変数であり、変数T (I)は、論理アドレス「I」に対応するエラー検出テーブルの値を示す変数である。また、定数Nは、総ブロック数を示す定数である。

【0197】集合管理情報を構築する際は、図21のステップS101において、データ処理装置1は、変換テーブルを初期化し、全ての値を「0xffff」とする。次にステップS102へ進む。

【0198】ステップS102において、データ処理装置1は、エラー検出テーブルを初期化し、全ての値を「0」とする。次にステップS103へ進む。

【0199】ステップS103において、データ処理装置1は、変数Iに「0」を代入する。次にステップS104へ進む。

【0200】ステップS104において、データ処理装置1は、物理アドレス「I」のブロックの分散管理情報をメモ리카ード2から読み込む。次にステップS105へ進む。

【0201】ステップS105において、データ処理装置1は、ステップS104で読み込んだ分散管理情報の可/不可フラグを参照して、物理アドレス「I」のブロックが使用可能か否かを判別する。使用可能であれば、ステップS106へ進み、使用不可能であれば、ステップS129へ進む。

【0202】ステップS106において、データ処理装置1は、ステップS104で読み込んだ分散管理情報のブロックフラグを参照して、物理アドレス「I」のブロックが使用中であるか否かを判別する。具体的には、ブロックフラグが「先頭使用」又は「使用」になっているか否かを判別する。ブロックフラグが「先頭使用」又は「使用」になっており、当該ブロックが使用中であれば、ステップS107へ進む。また、当該ブロックが使用中でなければ、ステップS129へ進む。

【0203】ステップS107において、データ処理装置1は、物理アドレス「I」のブロックについての情報をビットマップテーブルに加える。次に、ステップS108へ進む。

【0204】ステップS108において、データ処理装置1は、ステップS104で読み出した分散管理情報に基づいて、物理アドレス「I」のブロックの論理アドレスを変数Aに代入し、物理アドレス「I」のブロックの連結アドレスを変数Bに代入する。次にステップS109へ進む。

【0205】ステップS109において、データ処理装置1は、Aの値が「0xffff」であるか否かを判別する。「0xffff」でなければ、ステップS110へ進み、「0xffff」であれば、図22のステップS120へ進む。

【0206】ステップS110において、データ処理装置1は、エラー検出テーブルの論理アドレス「A」の値を調べる。次にステップS111へ進む。

【0207】ステップS111において、データ処理装置1は、エラー検出テーブルの論理アドレス「A」の値が「1」であるかを判別する。「1」であれば、ステップS112へ進み、「1」でなければステップS130へ進む。

【0208】ステップS112において、データ処理装置1は、エラー検出テーブルの論理アドレス「A」の値を「0」に書き換える。次に、ステップS113へ進む。

【0209】ステップS113において、データ処理装置1は、変換テーブルの論理アドレス「A」の欄に、変数「I」（すなわち物理ブロック「I」）を書き込む。次に、ステップS114へ進む。

【0210】ステップS114において、データ処理装置1は、連結テーブルの論理アドレス「A」の欄に、変数「B」（すなわち連結アドレス「B」）を書き込む。次に、ステップS115へ進む。

【0211】ステップS115において、データ処理装置1は、最終フラグが「ブロック最終」になっているか否かを判別する。「ブロック最終」になっていなければ、ステップS116へ進み、「ブロック最終」になっていれば、図22のステップS120へ進む。

【0212】ステップS116において、データ処理装置1は、論理アドレス「B」に対応する変換テーブルの値を調べる。次に、ステップS117へ進む。

【0213】ステップS117において、データ処理装置1は、論理アドレス「B」に対応する変換テーブルの値が「0xffff」であるかを判別する。「0xffff」であれば、ステップS118へ進み、「0xffff」でなければ、図22のステップS120へ進む。

【0214】ステップS118において、データ処理装置1は、エラー検出テーブルの論理アドレス「B」の欄を「1」に書き換える。次に、ステップS119へ進む。

【0215】ステップS119において、データ処理装置1は、エラー検出テーブルの論理アドレス「B」の欄に、論理アドレス「A」を書き込む。次に、図22の

ステップS120へ進む。

【0216】図22のステップS120において、データ処理装置1は、変数Iの値と、総ブロック数を示す定数Nの値とを比較する。 $I < N$ でなければ、ステップS121へ進み、 $I < N$ であれば、ステップS128へ進む。

【0217】ステップS121において、データ処理装置1は、変数Iに「0」を代入する。次にステップS122へ進む。

【0218】ステップS122において、データ処理装置1は、エラー検出テーブルの論理アドレス「I」の値を示す変数T(I)が「1」であるかを判別する。

「1」でなければステップS123へ進み、「1」であればステップS125へ進む。

【0219】ステップS123において、データ処理装置1は、変数Iの値と、総ブロック数を示す定数Nの値とを比較する。 $I = N$ であれば、これで処理を終了する。 $I = N$ でなければ、ステップS124へ進む。

【0220】ステップS124において、データ処理装置1は、変数Iの値を1インクリメントする。その後、ステップS122へ戻って処理を繰り返す。

【0221】また、ステップS122で変数T(I)が「1」であったときは、上述したようにステップS125へ進む。このステップS125に進むのは、連結アドレスエラーが生じていたときである。そこで、ステップS125において、データ処理装置1は、連結アドレスエラーに対する所定のエラー訂正処理を行う。ここでは、上述したように、メモリカード2を使用するアプリケーションソフトウェアやメモリカード2に格納するデータ等に応じた適切なエラー訂正処理を行うようにする。そして、エラー訂正処理が完了したら、ステップS126へ進む。

【0222】ステップS126において、データ処理装置1は、エラー検出テーブルの論理アドレス「I」の値を「0」にする。次に、ステップS127へ進む。

【0223】ステップS127において、データ処理装置1は、変換テーブルの論理アドレス「I」の欄を「0xffff」に書き換える。その後、ステップS123へ進み、上述した処理を行う。

【0224】また、ステップS120で $I < N$ であったときは、上述したようにステップS128へ進む。このステップS128へ進むのは、全ブロックについての分散管理情報の読み出しが完了していないときである。そこで、このステップS128において、データ処理装置1は、変数Iの値を1インクリメントし、その後、図21のステップS104へ戻って処理を繰り返す。

【0225】また、図21のステップS105で物理アドレス「I」のブロックが使用不可能であったとき、及びステップS106で物理アドレス「I」のブロックが使用中でなかったときは、上述したようにステップS1

29へ進む。

【0226】ステップS129において、データ処理装置1は、物理アドレス「1」のブロックについての情報をビットマップテーブルに加える。その後、図22のステップS120へ進み、上述した処理を行う。

【0227】また、図21のステップS111でエラー検出テーブルの論理アドレス「A」の値が「1」でなかったときは、上述したようにステップS130へ進む。このステップS130において、データ処理装置1は、変換テーブルの論理アドレス「A」の値を調べる。次に、ステップS131へ進む。

【0228】ステップS131において、データ処理装置1は、変換テーブルの論理アドレス「A」の値が「0xffff」であるかを判別する。「0xffff」であれば、ステップS113へ進んで上述した処理を行い、「0xffff」でなければ、図23のステップS132へ進む。

【0229】図23のステップS132に進むのは、論理アドレスエラーが生じて、論理アドレス「A」を持つブロックが2つ存在しているときである。そこで、ステップS132において、データ処理装置1は、論理アドレス「A」を持つ2つのブロックの識別番号を読み出す。そして、一方のブロックの識別番号を読み出したら、その値を変数Cに代入する。また、他方のブロックの識別番号を読み出したら、その値を変数Dに代入する。次にステップS133へ進む。

【0230】ステップS133において、データ処理装置1は、ステップS132での識別番号の読み出しが正常に行えたかを判別する。変数Cに代入される識別番号だけが読み出せたときは、ステップS134へ進み、それ以外のときは、ステップS137へ進む。

【0231】ステップS134において、データ処理装置1は、変数Dに対応するブロックのブロックフラグを「未消去」にする。次にステップS135へ進む。

【0232】ステップS135において、データ処理装置1は、ステップS134での処理、すなわち変数Dに対応するブロックのブロックフラグを「未消去」にする処理が成功したかを判別する。成功していれば、図21のステップS113へ戻って上述した処理を行い、成功していなければ、ステップS136へ進む。

【0233】ステップS136において、データ処理装置1は、変数Dに対応するブロックの可/不可フラグを「使用不可」にする。その後、図21のステップS113へ戻って上述した処理を行う。

【0234】また、ステップS137において、データ処理装置1は、ステップS132での識別番号の読み出しが正常に行えたかを判別する。変数Dに代入される識別番号だけが読み出せたときは、ステップS138へ進み、それ以外のときは、ステップS141へ進む。

【0235】ステップS138において、データ処理装置1は、変数Cに対応するブロックのブロックフラグを

「未消去」にする。次にステップS139へ進む。

【0236】ステップS139において、データ処理装置1は、ステップS138での処理、すなわち変数Cに対応するブロックのブロックフラグを「未消去」にする処理が成功したかを判別する。成功していれば、図21のステップS113へ戻って上述した処理を行い、成功していなければ、ステップS140へ進む。

【0237】ステップS140において、データ処理装置1は、変数Cに対応するブロックの可/不可フラグを「使用不可」にする。その後、図21のステップS113へ戻って上述した処理を行う。

【0238】また、ステップS141において、データ処理装置1は、ステップS132での識別番号の読み出しが正常に行えたかを判別する。変数Cに代入される識別番号と、変数Dに代入される識別番号との両方が正常に読み出せたときは、ステップS142へ進む。一方、両方とも読み出せなかったときには、手動復旧モードに移行し、手動にてエラーの訂正に必要な適切な処理を行うようにする。

【0239】ステップS142において、データ処理装置1は、変数Cの値と、変数Dの値に「1」を加えた値とを比較する。これらの値が等しければ、ステップS134へ進み上述した処理を行い、これらの値が等しくなければ、ステップS143へ進む。

【0240】ステップS143において、データ処理装置1は、変数Dの値と、変数Cの値に「1」を加えた値とを比較する。これらの値が等しければ、ステップS138へ進み上述した処理を行う。一方、これらの値が等しくないときは、手動復旧モードに移行し、手動にてエラーの訂正に必要な適切な処理を行うようにする。

【0241】以上のような処理により、集合管理情報の再構築時に、エラー検出訂正処理が同時に行われる。これにより、上述したように、エラー検出訂正処理のためにメモ리카ード2に余分にアクセスする必要がなくなる。すなわち、本発明を適用したシステムでは、データの書き込みが正常に終了し有効な集合管理情報がメモ리카ード2に書き戻されている場合には、エラー検出訂正処理は実行されない。

【0242】このように、本発明を適用したシステムでは、不要なエラー検出訂正処理を行わないので、メモ리카ード2へのアクセスの高効率化を図ることができる。特に、上記エラー検出訂正処理は、分散管理情報から集中管理情報を構築する作業と同時に進行するため、メモ리카ード2へのアクセスが最小限にとどめられる。

【0243】また、本発明を適用したシステムでは、メモ리카ード2の各ブロックに識別番号を付し、この識別番号を用いて論理アドレスエラーに対処できるようにしているので、データ更新処理を安全に行うことができる。すなわち、データ更新処理時に何らかのエラーが発生して、同じ論理アドレスを持つブロックが複数存在するよ

うになってしまっても、識別番号を用いることで、データ更新処理前の状態のデータを復旧することができる。更に、本発明を適用したシステムでは、エラー検出テーブルを用いることで、連結アドレスエラーを検出することもでき、しかも連結先の存在しないブロックを検出することもできる。

【0244】

【発明の効果】以上詳細に説明したように、本発明によれば、最後の連結アドレスが指し示すブロックとして新規ブロックを割り当て、最後のブロックのデータを正しく読めるページまで読み込んで、そのページまでのデータを新規ブロックにコピーする。このとき、新規ブロックの最終フラグは「ブロック最終」としておく。したがって、外部記憶装置に論理アドレスエラーや連結アドレスエラーが生じて、それらのエラーを検出し適切に修復することができる。なお、このようなエラー訂正処理は、対象となるデータが音楽データ等のようにデータの途中でも意味があるデータの場合に、特に好適である。

【図面の簡単な説明】

【図1】本発明が適用されるシステムの全体構成を示す図である。

【図2】本発明を適用したメモ리카ードの構成を示すブロック図である。

【図3】本発明を適用したメモ리카ードの外観を示す斜視図である。

【図4】本発明を適用したメモ리카ードの記憶領域の構造を示す図である。

【図5】分散管理情報の構成を示す図である。

【図6】追加管理情報の構成を示す図である。

【図7】各ブロックの分散管理情報から集合管理情報を構築する様子を示す図である。

【図8】集合管理情報の構成を示す図である。

【図9】変換テーブルを示す図である。

【図10】連結テーブルを示す図である。

【図11】メモ리카ード起動時の手順を示すフローチャートである。

【図12】データ更新処理時の手順を示すフローチャートである。

【図13】終了処理の手順を示すフローチャートである。

【図14】サイズが分かっているファイルをメモ리카ードに書き込むときの手順を示すフローチャートである。

【図15】サイズが分かっていないファイルをメモ리카ードに書き込むときの手順を示すフローチャートである。

【図16】ファイル更新の手順について、具体的な例を挙げて示す概念図である。

【図17】ファイル更新の手順を示すフローチャートである。

【図18】エラー検出テーブルを示す図である。

【図19】ブロック間の連結状態の具体的な例を示す図である。

【図20】連結アドレスエラー検出の例として、ブロック間の連結状態が図19に示した状態のときの処理の流れを示す図である。

【図21】集合管理情報の構築とエラー検出訂正処理の手順を示すフローチャートである。

【図22】集合管理情報の構築とエラー検出訂正処理の手順を示すフローチャートである。

【図23】集合管理情報の構築とエラー検出訂正処理の手順を示すフローチャートである。

【符号の説明】

- 1 データ処理装置、 2 メモ리카ード、 3 演算処理装置、 4 内部メモリ、 5 補助記憶装置、 6 シリアルインターフェース回路、 7 バス、 11 コントローラ、 12 フラッシュメモリ、 13 シリアル/パラレル・パラレル/シリアル・インターフェース・シーケンサ、 14 フラッシュメモリ・インターフェース・シーケンサ、 15 ページバッファ、 16 エラー訂正回路、 17 コマンドジェネレータ、 18 コンフィグレーションROM、 19 発振器、 20 誤消去防止スイッチ

【図5】

可/不可 フラグ	ブロック 番号	分散 管理 フラグ	物理 フラグ	論理アドレス	連結アドレス	リザーブ領域	分散管理情報 エラー訂正符号	データ用 エラー訂正符号
-------------	------------	-----------------	-----------	--------	--------	--------	-------------------	-----------------

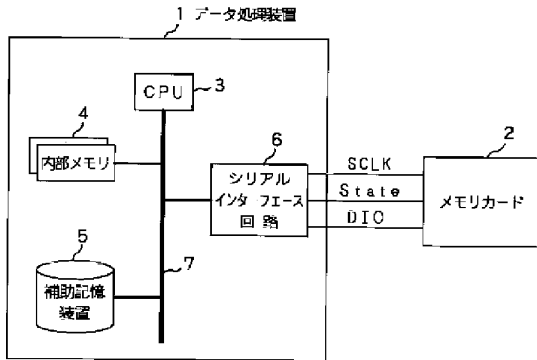
分散管理情報

【図8】

ヘッダー
ビットマップテーブル
変換テーブル
連結テーブル

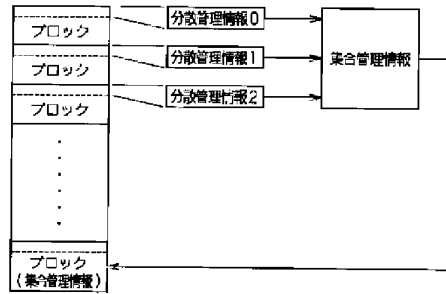
集合管理情報

【図1】



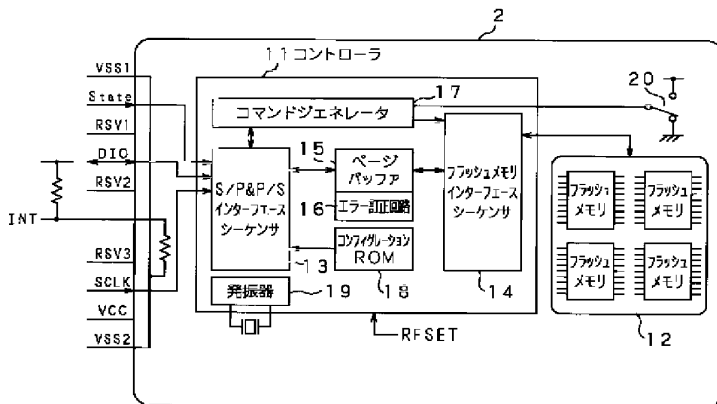
システムの全体構成

【図7】



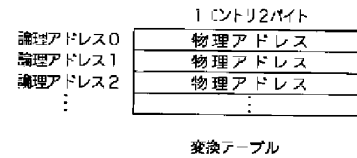
分散管理情報から集合管理情報の構築

【図2】

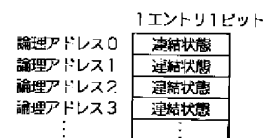


メモ리카 - ドの構成

【図9】



【図18】



エラー検出テーブル

【図6】

可/不可 フラグ	ブロック フラグ	隠 シ	隠 シ	隠 シ	有効データ サイズ	リザーブ領域	追加管理情報 エラー訂正符号	データ用 エラー訂正符号
-------------	-------------	--------	--------	--------	--------------	--------	-------------------	-----------------

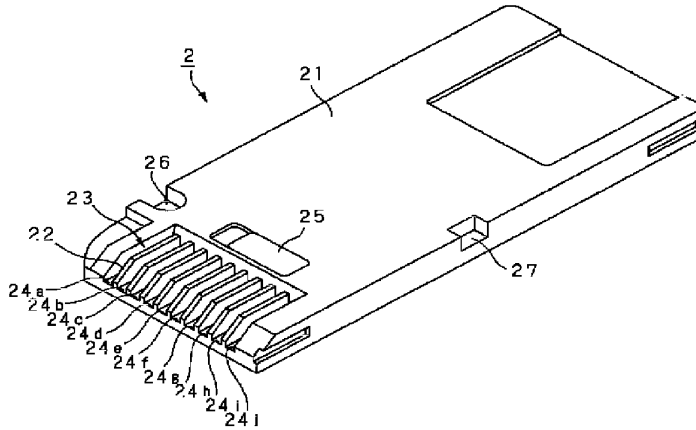
追加管理情報

【図10】

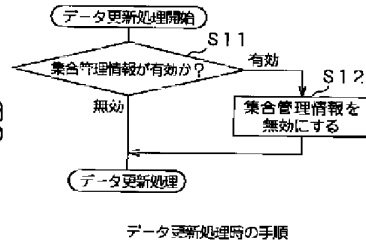
1 エントリ2バイト	
論理アドレス0	連結アドレス
論理アドレス1	連結アドレス
論理アドレス2	連結アドレス
...	...

連結テーブル

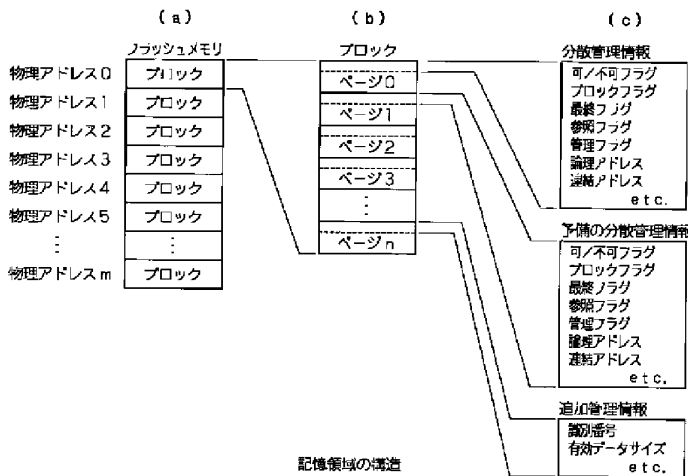
【図3】



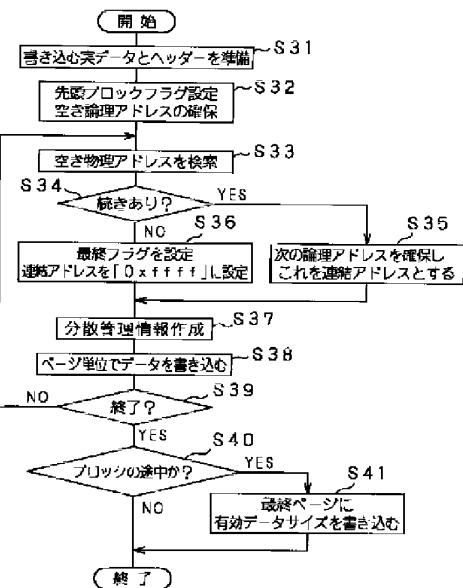
【図12】



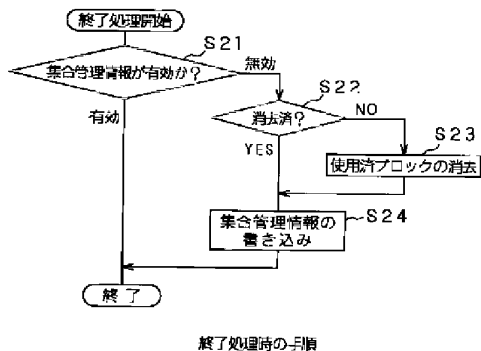
【図4】



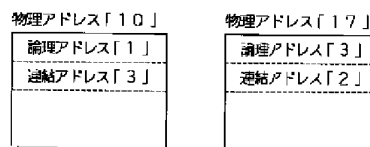
【図14】



【図13】



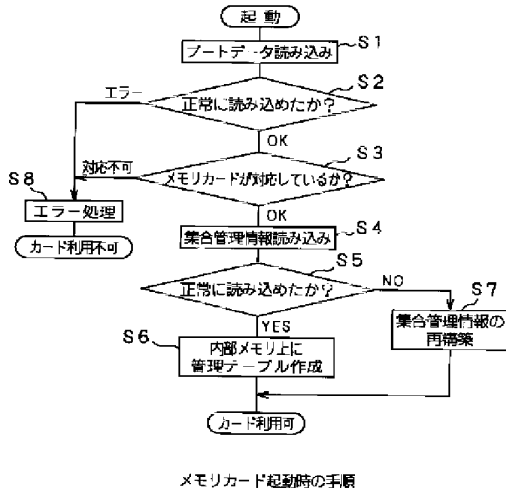
【図19】



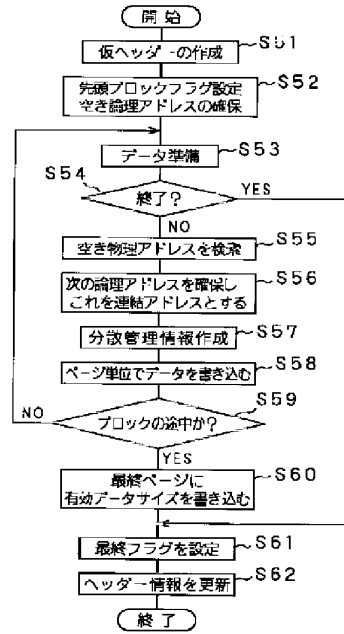
終了処理時の手順

サイズが分かっているファイルの書き込み手順

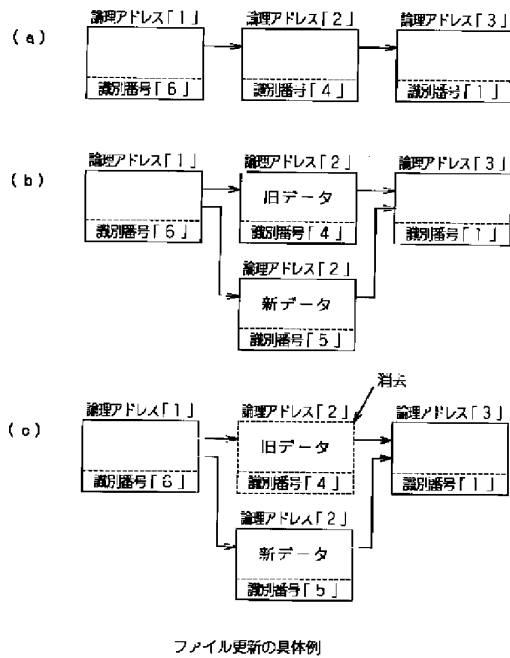
【図11】



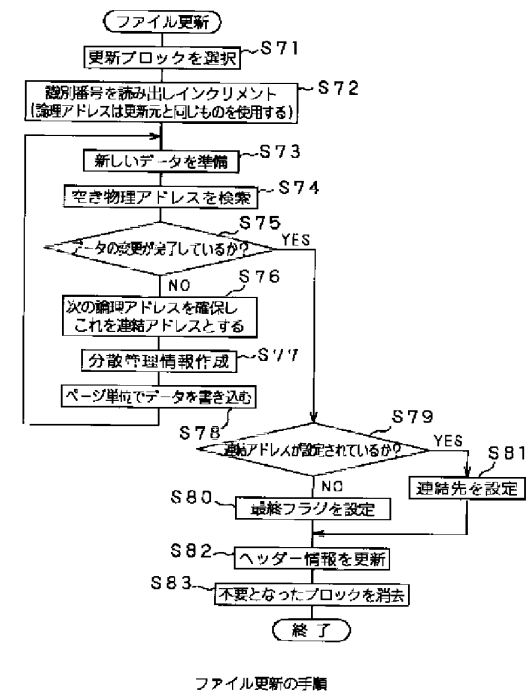
【図15】



【図16】



【図17】

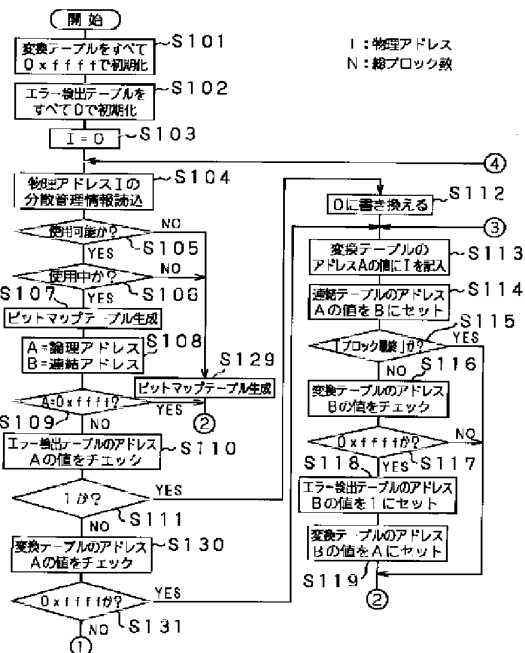


【図20】

	エラー検出 テーブル	変換 テーブル	連結 テーブル
(a)	論理アドレス1: 0 論理アドレス2: 0 論理アドレス3: 0	0xffff 0xffff 0xffff	
(b)	論理アドレス1: 0 論理アドレス2: 0 論理アドレス3: 0	10 0xffff 0xffff	3
(c)	論理アドレス1: C 論理アドレス2: 0 論理アドレス3: 1	10 0xffff 1	3
(d)	論理アドレス1: 0 論理アドレス2: 0 論理アドレス3: 0	10 0xffff 17	3 2
(e)	論理アドレス1: 0 論理アドレス2: 1 論理アドレス3: 0	10 3 17	3 2

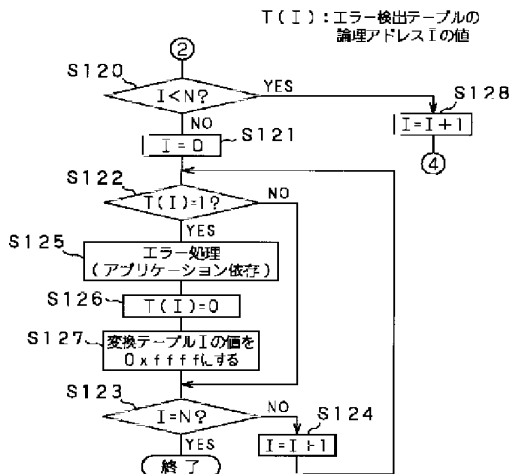
連結アドレスエラー検出の例

【図21】



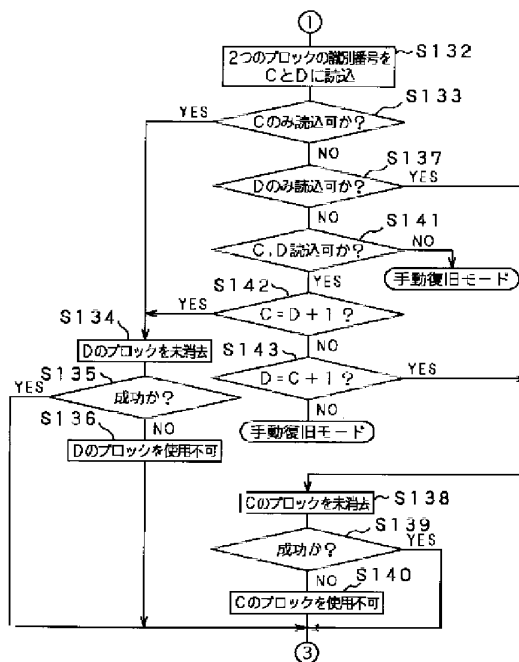
集合管理情報の構築とエラー検出訂正処理の手順

【図22】



集合管理情報の構築とエラー検出訂正処理の手順

【図23】



集合管理情報の構築とエラー検出訂正処理の手順

(19) Japan Patent Office (JP)
(12) Publication of Unexamined Patent Application (A)
(11) Patent Application Disclosure Number H8-212019

(51) Int. Cl. 6 ID No. Internal Administration Number FI
Technical Display Points
G06F 3/08
G11C 16/06
G11C 17/00 530 B

Examination requested? Not requested
Number of Claims OL (Total 15 pages)

(21) Application Number Application H7-14030
(22) Application Date January 31, 1995

(71) Applicant 000006013
Mitsubishi Electric Corp.
2-3, Marunouchi 2-chome, Chiyoda-ku, Tokyo
(72) Inventor MIYAUCHI, Shigenori
c/o Mitsuibishi Electric Corp. Kitaitami Plant, 1-banchi, Mizuhara 4-chome,
Itami-shi
(74) Agent Patent Attorney SOGA, Michiteru (and 6 others)
(54) [Title of the Invention] Semiconductor Disk Device

(54) [Title of the Invention] Semiconductor disk device

(57) [Abstract]

[Configuration] A Flash memory 8A having multiple erase blocks and comprising; a logical erase block number storage area, multiple data storage areas storing data, a data storage flag area for each of the aforementioned data storage areas storing a data status flag indicating whether or not data is stored in the aforementioned data storage area, and an updated data link information storage area for each of the aforementioned data storage areas storing link information indicating the location of the update destination, and a CPU 4 converting the logical sector address into a logical erase block number and its offset value, searching for the relevant erase block and the relevant data storage area in the aforementioned flash memory 8A based on this converted logical erase block number and its offset value, and reading the content of the aforementioned relevant data storage area if link information does not exist in the relevant update data link information storage area.

[Claim 5] A semiconductor disk device noted in Claim 4 and characterized by the aforementioned CPU converting the logical sector address to a logical erase block number and its offset value, converting the aforementioned logical erase block number to a physical erase block number based on the aforementioned address conversion table, searching for the relevant erase block and relevant data storage area in the aforementioned flash memory 8A based on this physical erase block and its offset value, and reading the content of the aforementioned data storage area when link information exists in the relevant new data link information storage area.

[Benefits of the Invention] Makes it unnecessary for a data management address conversion table and allows for a consequent increase in the data area.

57 Scope of Claims

[Claim 1] A semiconductor disk device characterized by being provided with a flash memory having a logical erase block number storage area storing logical erase block numbers, multiple data storage areas storing data, a data status flag area for each of the aforementioned data storage areas storing a data status flag indicating whether or not data is stored in the aforementioned data storage area, multiple erase blocks comprised of an update data link information storage area for each of the aforementioned data storage areas storing link information indicating the location of the update destination, as well as a CPU managing data in the aforementioned flash memory based on the aforementioned logical erase block number, the aforementioned data status flag, and the aforementioned link information.

[Claim 2] A semiconductor disk device noted in Claim 1 and characterized by the aforementioned CPU converting the logical sector address to a logical erase block number and its offset value, searching for the relevant erase block and relevant data storage area in the aforementioned flash memory based on this converted logical erase block and its offset value, reading the content of the aforementioned data storage area if link information does not exist in the relevant update data link information storage area, and reading the content of the relevant data storage area based on the link information if the aforementioned link information exists in the aforementioned relevant update data link information storage area.

[Claim 3] A semiconductor disk device noted in Claim 1 and characterized by the aforementioned CPU converting the logical sector address to a logical erase block number and its offset value, searching for the relevant erase block and relevant data storage area in the aforementioned flash memory based on this converted logical erase block and its offset value, writing data to the aforementioned relevant data storage area when the relevant data status flag indicates that it is 'unused', setting the aforementioned relevant data status flag to 'in use', searching for an empty data storage area when the aforementioned relevant data status flag is 'in use' and writing data to that [data storage area], setting the data status flag corresponding to this data storage area to 'in use',

and writing link information to the update data link information storage area corresponding to the previous 'in use' data storage area.

[Claim 4] A semiconductor disk device characterized by being provided with a flash memory having a logical erase block number storage area storing logical erase block numbers, multiple data storage areas storing data, a data status flag area for each of the aforementioned data storage areas storing a data status flag indicating whether or not data is stored in the aforementioned data storage area, multiple erase blocks comprised of an update data link information storage area for each of the aforementioned data storage areas storing link information indicating the location of the update destination, an address conversion table for converting the aforementioned logical erase block number to a physical erase block number, as well as a CPU managing data in the aforementioned flash memory based on the aforementioned logical erase block number, the aforementioned data status flag, and the aforementioned link information.

[Claim 5] A semiconductor disk device noted in Claim 4 and characterized by the aforementioned CPU converting the logical sector address to a logical erase block number and its offset value, converting the aforementioned logical erase block number to a physical erase block number based on the aforementioned address conversion table, searching for the relevant erase block and relevant data storage area in the aforementioned flash memory based on this physical erase block and its offset value, and reading the content of the aforementioned data storage area when link information exists in the relevant new data link information storage area.

[Claim 6] A semiconductor disk device noted in Claim 4 and characterized by the aforementioned CPU converting the logical sector address to a logical erase block number and its offset value, converting the aforementioned logical erase block number to a physical erase block number based on the aforementioned address conversion table, searching the relevant erase block and relevant data storage area in the aforementioned flash memory based on this physical erase block and its offset value, writing data to the aforementioned relevant data storage area when the relevant data status flag is 'unused', and setting the aforementioned relevant data status flag to 'in use'.

[Detailed Explanation of the Invention]

[0001]

[Field of the Invention] The present invention is related to a semiconductor disk such as a semiconductor card using flash memory as a storage medium.

[0002]

[Prior Art] Magnetic storage media such as hard disk drives are frequently used for storage of comparatively large amounts of data in the field of personal computers. This is due to excellent cost performance, despite high power consumption.

[0003] On the other hand, semiconductor disk devices in which semiconductor memory such as flash memory operates as a hard disk drive have appeared. Since this semiconductor disk device differs from the aforementioned hard disk drive in that it has no mechanical parts such as a motor, cost-performance is improved in comparison with magnetic media, and low power consumption and high reliability have ensured that its use is expanding in such devices as mobile data terminals.

[0004] Flash memory has the following characteristics. 1. Non-volatile memory permitting electrical writing and erasing of data. 2. Data cannot be overwritten to memory cells to which data has already been written (and the need for erase operation is therefore always present). 3. Data is erased in units of a few Kbytes to a few tens of Kbytes. 4. The number of write and erase cycles is limited.

[0005] The configuration of conventional semiconductor disk devices is shown in Fig. 10, Fig. 11, Fig. 12 and Fig. 13. Fig. 10 is a block diagram showing the overall configuration of a conventional semiconductor disk device. Fig. 11 is a diagram showing the internal configuration of the address

table in Fig. 10. Fig. 12 is a diagram showing the internal configuration of the flash memory in Fig. 10. Fig. 13 is a diagram showing the internal configuration of the erase block in Fig. 12.

[0006] In Fig. 10, the conventional semiconductor disk device 2 is provided with an interface circuit 3, CPU 4, address conversion table 5, flash control circuit 6, data I/O sector buffer 7, and flash memory 8.

[0007] Representative examples of host 1 connected to semiconductor disk device 2 are notebook PCs and mobile data terminals. The removable card-type is currently the primary form of semiconductor disk device 2. Interface circuit 3 sends and receives information to and from the host 1. CPU 4 outputs inputs and outputs data, and outputs commands to the flash memory 8.

[0008] The logical sector/physical sector address conversion table 5 is a table for the conversion of logical sector addresses to physical sector addresses. The logical sector address (LSA) is a sector address specified by the host 1 to the semiconductor disk device 2. Furthermore, the physical sector address (PSA) is an address in the flash memory 8 used in the semiconductor disk device 2.

[0009] The Flash control circuit 6 processes data in the non-complex flash memory 8. Sending and receiving of simple data is handled by the flash control circuit 6. Other processing is handled by the CPU 4. The data I/O sector buffer 7 is used when outputting data from the flash memory 8 via the interface circuit 3, or inputting data to the flash memory 8 via the interface circuit 3.

[0010] In Fig. 11, the address conversion table 5 is comprised of the LSA storage part and the PSA storage part.

[0011] The logical sector address is stored in the LSA storage part. Content is fixed. Any desired flash memory 8 sector number (1-n in the figure) is stored in the PSA storage part. Using this address conversion table 5, data may be stored at a physical sector address convenient for internal management without being affected by the logical sector address specified by the host 1. Since this address conversion table 5 is written and erased frequently, it is generally configured as SRAM.

[0012] The capacity of this address conversion table 5 is as follows under the following conditions. Using a 20MByte flash memory 8, with 512 byte data I/O units (sectors), the number of sectors in the semiconductor disk device 2 is as follows. Number of sectors in semiconductor disk device 2 = 20MByte / 512 byte = 40960 sectors

[0013] Next, the number of bits necessary to express 40960 in binary is $\ln 40960 / \ln 2 = 15.3$, therefore 16 bits are necessary.

[0014] Thus, the necessary capacity of the address conversion table 5 is $40960 \times 16 = 655360$ bits, and 80Kbyte is required.

[0015] In Fig. 12, flash memory 8 is comprised of multiple erase blocks 9 and spare multiple erase blocks 9.

[0016] Flash memory 8 is a non-volatile memory able to be electrically written and erased. Since it is non-volatile, battery backup as with DRAM and SRAM memory is unnecessary, and data can be erased electrically, so that data can be changed without removal from the board as with EPROM memory. Since one bit may be stored per cell, memory can be manufactured at a lower cost than EEPROM memory. The above are the advantages of flash memory 8. Disadvantages include a limit of between approximately 10,000 and 100,000 erase cycles, the need for erase operation when writing (and therefore data cannot be overwritten to memory cells to which data has already been written), and data is erased in units of a few Kbytes to a few tens of Kbytes.

[0017] In Fig. 13, a single erase block 9 has a leading erase block information storage area 10, multiple data storage areas 11, and an LSA storage area 12.

[0018] The current number of block erase cycles is stored in the erase block information storage area 10. The multiple data storage areas 11 are normally 512 bytes (1 sector) in size. An LSA storage area 12 exists for each sector, and stores the LSA specified by the host 1 when data is written. When the logical sector/physical sector address conversion table 5 is configured with

SRAM, data is lost when power is switched OFF. This is used when searching the LSA storage areas 12 for all sectors and rebuilding the SRAM table 5 when power is switched ON.

[0019] Next, operation of the conventional semiconductor disk device is explained in reference to Fig. 14, Fig. 15 and Fig. 16. Fig. 14 is used in explaining read operation of the conventional semiconductor disk device. Furthermore, Fig. 15 and Fig. 16 are used to explain the write operation of the conventional semiconductor disk device.

[0020] The semiconductor disk device 2 using the flash memory 8 differs from a hard disk drive in that data cannot be overwritten. The address conversion table 5 showing the logical sector address of the data sent from the host 1, and the physical address in the flash memory 8 to which the data is to be written, are therefore stored in SRAM. By using this table 5, the flash memory 8 storage area may be used efficiently without being affected by the LSA.

[0021] First, the operation of reading data from the semiconductor disk device 2 is explained with Fig. 14. Host 1 sends the sector address of the data to be read to the semiconductor disk device 2. The address data sent from the host 1 is of two types. LSA format and CHS format. The LSA format specifies the sector with a serial number between 1 and n, while the CHS format specifies the data area with a combination of three data items – the cylinder, the head, and the sector used in the hard disk drive. Since the LSA/PSA address conversion table 5 is used in the semiconductor disk device 2, when CHS format data from the host 1 is received, it is, for example, converted to LSA [format] in the interface circuit, and [processing] moves to the next operation.

[0022] CPU 4 uses address conversion table 5 to convert the LSA specified by the host 1 to a PSA. Data is then read from a flash memory 8 corresponding to the PSA.

[0023] For example, if the LSA specified by the host 1 is '2', it is converted to a PSA of '6' with the address conversion table 5. Thus data 'A' is read. The LSA '2' is stored in the LSA storage area 12.

[0024] Next, the operation of writing data to the semiconductor disk device 2 is explained with Fig. 15 and Fig. 16. It is assumed that the initial state is that of data 'A', 'B' and 'C' stored in PSAs '1', '2' and '7'. It must be noted that data cannot be rewritten to the flash memory 8 when writing data. PSAs '1', '3' and '7' in the aforementioned initial state.

[0025] When host 1 specifies an LSA for which data has not been written, the CPU 4 writes data to the appropriate empty areas (PSAs '2', '4' – '6', '8' – '12') in the flash memory 8, and updates data in the address conversion table 5. Fig. 15 shows an example in which the host 1 has specified writing of data 'D' to LSA '4'. Data 'D' and the LSA specified by the host 1 are written to the empty area PSA '4', and the PSA value '4' is written to the PSA part corresponding to the LSA '4' in the address conversion table 5.

[0026] Even if a request to rewrite to an area to which data has already been written (for example, save by overwriting a file of the same name) is received from the host 1, the data to be rewritten is written to an empty area of the flash memory 8, and the address conversion table 5 is updated. Fig. 16 shows the results of rewriting the LSA '2' data. The updated data 'B' is written to the empty area PSA '5', and the PSA corresponding to LSA '2' in the address conversion table 5 is updated to '5'. Furthermore, the CPU 4 in the card must be aware of the fact that PSA '3' is used data.

[0027]

[Problems to be Solved by the Invention] In the aforementioned conventional semiconductor disk device, a memory area to store a PSA is required in the address conversion table 5 for each sector (minimum unit for data management), resulting in a large increase in the size of the flash memory 8, and an associated increase in the size of the address conversion table 5.

[0028] The present invention is intended to resolve the aforementioned problem, and has as its objective a semiconductor disk device which does not require an address conversion table for memory management.

[0029] Furthermore, the present invention has as its objective a semiconductor disk device in which the size of the address conversion table can be reduced.

[0030]

[Means for Solving the Problem] The semiconductor disk device related to this invention is provided with a logical erase block number storage area storing logical erase block numbers, multiple data storage areas storing data, a data status flag area for each of the aforementioned data storage areas storing a data status flag indicating whether or not data is stored in the aforementioned data storage area, a flash memory having multiple erase blocks comprised of an update data link information storage area for each of the aforementioned data storage areas storing link information indicating the location of the update destination, and a CPU for management of data in the aforementioned flash memory based on the aforementioned logical erase block number, the aforementioned data status flag, and aforementioned link information.

[0031] Furthermore, the aforementioned CPU of the semiconductor disk device related to the present invention converts the logical sector address to a logical erase block number and its offset, searches for the relevant erase block and relevant storage area in the aforementioned flash memory based on this converted logical erase block and its offset value, reads the content of the aforementioned relevant data storage area if link information does not exist in the aforementioned relevant update data link information storage area, and reads the content of the relevant data storage area based on the link information if the aforementioned link information exists in the aforementioned relevant update data link information storage area.

[0032] Furthermore, the aforementioned CPU of the semiconductor disk device related to the present invention converts the logical sector address to a logical erase block number and its offset, searches for the relevant erase block and relevant storage area in the aforementioned flash memory based on this converted logical erase block and its offset value, writes data to the aforementioned relevant data storage area when the relevant data status flag indicates that it is 'unused', sets the aforementioned relevant data status flag to 'in use', searches for an empty data storage area when the aforementioned relevant data status flag indicates 'in use', and writes data to that [data storage area], setting the data status flag corresponding to this data storage area to 'in use', and writes link information to the update data link information storage area corresponding to the previous 'in use' data storage area.

[0033] Furthermore, the semiconductor disk device related to the present invention is provided with a logical erase block number storage area storing logical erase block numbers, multiple data storage areas storing data, a data status flag area for each of the aforementioned data storage areas storing a data status flag indicating whether or not data is stored in the aforementioned data storage area, a flash memory having multiple erase blocks comprised of an update data link information storage area for each of the aforementioned data storage areas storing link information indicating the location of the update destination, an address conversion table converting the aforementioned logical erase block number to a physical erase block number, and a CPU for management of data in the aforementioned flash memory based on the aforementioned logical erase block number, the aforementioned data status flag, and the aforementioned link information.

[0034] Furthermore, the aforementioned CPU of the semiconductor disk device related to the present invention converts the logical sector address to a logical deletion block number and its offset value, converts the aforementioned logical deletion block number to a physical deletion block number based on the aforementioned address conversion table, searches for the relevant deletion block and relevant data storage area in the aforementioned flash memory based on this physical deletion block and its offset value, and reads the content of the aforementioned relevant data storage area when link information does exist in the relevant new data link information storage area.

[0035] Furthermore, the aforementioned CPU of the semiconductor disk device related to the present invention converts the logical sector address to a logical deletion block number and its offset value, converts the aforementioned logical deletion block number to a physical deletion block number based on the aforementioned address conversion table, searches for the relevant deletion block and relevant data storage area in the aforementioned flash memory based on this physical deletion block and its offset value, and writes data to the aforementioned relevant data storage area when the relevant data status flag indicates that it is 'unused', and sets the aforementioned relevant data status flag to 'in use'.

[0036]

[Operation of the Invention] In the semiconductor disk device related to the present invention, the logical deletion block number storage area storing the logical deletion block number, multiple data storage areas storing data, a data status flag area for each of the aforementioned data storage areas storing a data status flag indicating whether or not data is stored in the aforementioned data storage area, a flash memory having multiple deletion blocks comprised of an update data link information storage area for each of the aforementioned data storage areas storing link information indicating the location of the update destination, and a CPU for management of data in the aforementioned flash memory based on the aforementioned logical deletion block number, the aforementioned data status flag, and the aforementioned link information, are provided so that an address conversion table for data management is unnecessary, and the size of the data area can consequently be increased.

[0037] Furthermore, in the semiconductor disk device related to the present invention, since the aforementioned CPU converts the logical sector address to a logical erase block number and its offset value, and searches for the relevant erase block and relevant data storage area based on this converted logical erase block number and its offset value, reads the content of the aforementioned relevant data storage area if link information does not exist in the relevant update data link information storage area, and reads the content of the relevant data storage area based on the link information if the aforementioned link information exists in the aforementioned relevant update data link information storage area, a data management address conversion table becomes unnecessary, permitting a consequent increase in the data area.

[0038] Furthermore, in the semiconductor disk device related to the present invention, since the aforementioned CPU converts the logical sector address to a logical erase block number and its offset value, searches for the relevant erase block and relevant data storage area based on this converted logical erase block number and its offset value, writes data to the aforementioned relevant data storage area when the relevant data status flag indicates that it is 'unused', sets the aforementioned relevant data status flag to 'in use', searches for an empty data storage area when the aforementioned relevant data status flag is 'in use' and writes data to that [data storage area], and sets the data status flag corresponding to this data storage area to 'in use', as well as writing link information to the update data link information storage area corresponding to the previous 'in use' data storage area, a data management address conversion table becomes unnecessary, permitting a consequent increase in the data area.

[0039] Furthermore, the semiconductor disk device related to the present invention is provided with a flash memory having a logical erase block number storage area storing logical erase block numbers, multiple data storage areas storing data, a data status flag area for each of the aforementioned data storage areas storing a data status flag indicating whether or not data is stored in the aforementioned data storage area, multiple erase blocks comprised of an update data link information storage area for each of the aforementioned data storage areas storing link information indicating the location of the update destination, an address conversion table for converting the aforementioned logical erase block number to a physical erase block number, as well as a CPU

managing data in the aforementioned flash memory based on the aforementioned logical erase block number, the aforementioned data status flag, and the aforementioned link information.

[0040] Furthermore, since the CPU of the semiconductor disk device related to the present invention converts the logical sector address to a logical erase block number and its offset value, converts the aforementioned logical erase block number to a physical erase block number based on the aforementioned address conversion table, searches for the relevant erase block and relevant storage area in the aforementioned flash memory based on this converted physical erase block and its offset value, reads the content of the aforementioned relevant data storage area if link information does not exist in the aforementioned relevant update data link information storage area, the data management address conversion is smaller, permitting a consequent increase in the data area.

[0041] Furthermore, in the semiconductor disk device related to the present invention, since the aforementioned CPU converts the logical sector address to a logical erase block number and its offset value, converts the aforementioned logical erase block number to a physical erase block number based on the aforementioned address conversion table, searches for the relevant erase block and relevant data storage area based on this converted physical erase block number and its offset value, writes data to the aforementioned relevant data storage area when the relevant data status flag indicates that it is 'unused', and sets the aforementioned relevant data status flag to 'in use', the data management address conversion is smaller, permitting a consequent increase in the data area.

[0042]

[Embodiments]

Embodiment 1. The configuration of Embodiment 1 of the present invention is explained below in reference to Fig. 1, Fig. 2, Fig. 3, and Fig. 4. Fig. 1 is a block diagram showing the overall configuration of Embodiment 1 of the present invention. Fig. 2 is a diagram showing the internal configuration of the flash memory in Fig. 1. Fig. 3 is a diagram showing the internal configuration of the erase block in Fig. 2. Fig. 4 is a diagram showing the internal configuration of the erase block information storage area in Fig. 3. The same symbols indicate the same or equivalent parts in all diagrams.

[0043] In Fig. 1, the semiconductor disk device 2A related to Embodiment 1 is provided with an interface circuit 3, a CPU 4, a flash control circuit 6, a data I/O sector buffer 7, and a flash memory 8A.

[0044] [This semiconductor disk device 2A] differs from the conventional semiconductor disk device 2 in that it has no logical sector/physical sector address conversion table. The size of the flash memory 8A is therefore increased by this amount.

[0045] In Fig. 2, flash memory 8A is comprised of multiple erase blocks 9A and multiple spare multiple deletion blocks 9A.

[0046] The flash memory 8A used in main memory is of the conventional block erase type (erase block units of a few Kbytes to a few tens of Kbytes). The internal block configuration of the flash memory is therefore of the conventional type.

[0047] In Fig. 3, an erase block 9A has a leading erase block information storage area 20, multiple data storage areas 21, a data status flag 22 for each data storage area 21, and an update data link information (link data) storage area 23. The data storage areas 21 are normally approximately 512 bytes (1 sector) in size.

[0048] The data status flag 22 stores data indicating the status of data stored in the data storage area 21. The status of the data storage area 21 can be one of four statuses – an unused area (no link data), a used area (with link data), or an unnecessary data area (updated in another area and awaiting erasure).

[0049] Each status is represented by a unique combination of bits (unused area - '000', used area (no link data) - '001', used area (with link data) - '011', unnecessary data area - '111'). These bit combinations are used since the flash memory 8A cannot be overwritten (combinations such as '010' cannot be used).

[0050] Link data is used when an overwrite instruction for the same area is requested from the host 1. Since the flash memory 8A is memory which cannot be overwritten, when overwriting of data is requested, the problem is resolved by writing data to an empty area. In this case, it is necessary to be able search for an update data destination from the data area before update, and link data in the update data link information storage area 23 is therefore used to save the location of the update destination.

[0051] In Fig. 4, the erase block information storage area 20 is comprised of the erase count storage area 24, the logical erase block number storage area 25, and a storage area 26 for other information.

[0052] The erase count storage area 24 stores the count of the number of blocks previously erased. Since the erase count for the flash memory 8A is between approximately 10,000 and 100,000 cycles, three bytes is sufficient. The logical erase block number storage area 25 stores the logical erase block number. The logical erase block number (PLBN: Physical-Logical Block Number) differs from the physical (fixed) erase block number (PBN: Physical Block Number) in that it is applied to all erase blocks, and is not duplicated in any logical erase block number. The physical erase block number and the logical erase block number used in memory management can be made to correspond directly. When manufacturing, for example, a 20MB semiconductor disk device card, it will comprise approximately 320 erase blocks, and a logical erase block number storage area of two bytes is therefore sufficient.

[0053] The following explains the method of address calculation in Embodiment 1. The address sent from the host 1 is in CHS or LHS format. If sent in CHS format, it is converted into LHS format by the interface circuit 3, and then assigned to a PSA. In practice, this determines which data storage area of which erase block is handled.

[0054] In devices using storage media for which the number of erase cycles when overwriting data is of no concern, directly corresponding LSAs and PSAs (LSA is 1, PSA also 1) present no problem in management. The quotient obtained by dividing the LSA storage area by the number of data areas in the management block (same as the erase block) is the physical block number, and the remainder is the offset value within the physical block, and thus a PSA for read and write data can be readily obtained by calculation.

[0055] Embodiment 1 uses a method of equalizing the number of erase cycles for each erase block without use of a logical/physical address conversion table. A logical erase block number (PLBN) is therefore used for each erase block 9A. Furthermore, the quotient obtained by dividing the LSA sent from the host 1 by the number of data areas in the management block (same as the erase block) is the logical erase block number (PLBN), and the remainder is the offset value within the logical erase block number. Thus, the number of erase cycles can be equalized for each erase block 9A by rewriting the logical erase block number storage area 25 within the erase block 9A without the use of an address conversion table.

[0056] Operation of Embodiment 1 is explained below in reference to Fig. 5, Fig. 6, and Fig. 7. Fig. 5 is a diagram used in explaining read operation in Embodiment 1. Fig. 6 is a flow chart showing read operation in Embodiment 1. Fig. 7 is a flow chart showing write operation in Embodiment 1.

[0057] Firstly, reading of data from the semiconductor disk device 2A is explained using Fig. 5 and Fig. 6. To simplify the explanation, the erase block 9A of flash memory 8A incorporates three data storage areas 21, and logical erase blocks '4' and '5' are spare blocks used to save data. In other words, as seen from the host 1, the capacity of this semiconductor disk device 2A is 512 byte x 3 x 4 = 6KByte, and the LSAs sent from the host 1 are '0 - 11'.

[0058] Firstly, the sector information for the data to be read from the host 1 is received (step 30). This [information] is sent in LSA or CHS format. Data sent in CHS format is converted to LSA format to ensure a uniform format (step 31 – 32). This conversion may be handled by the CPU 4 in the semiconductor disk device 2A, or by a dedicated circuit in the semiconductor disk device. Next, data in the LSA format is converted to PLBN format (step 33). The conversion calculation is as described above. The conversion calculation may be handled by a dedicated circuit or by an internal CPU.

[0059] Thus, the logical erase block number (PLBN) and its offset value is determined. Finally, the erase block 9A in the flash memory 8A corresponding to the calculated logical erase block number is found (step 34 – 36). Firstly, the flash memory 8A erase block 9A erase block information storage area 20 storing the target logical erase block number is determined.

[0060] When data LSA = 5 is sent from the host 1, PLBN conversion results in the logical erase block number $5 / 3 = 1$, and offset value $5 - 1 \times 3 = 2$, with subsequent representation as (1, 2). Next, link information in the update data link information storage area 23 is read (step 37 – 38). In this case, there is no link information and the data to be read is 'A' (step 39).

[0061] Furthermore, it will be apparent that, when data LSA = 6 is sent from the host 1, PLBN conversion results in (2, 0). '40' is stored in the address (2, 0) of the update data link information storage area 23. This indicates that data at offset 0 of the logical erase block number 4 is updated. It will be apparent that the data to be read in this manner is 'B' (step 38, 34, 35, 37 – 39).

[0062] Assuming a semiconductor disk device using a 20MByte flash memory (erase block size of 64KByte), the maximum searchable value is the erase block number of the semiconductor disk device, and is therefore $20\text{MByte} / 64\text{Kbyte} = 320$ blocks. While the maximum value is 320, consideration of actual file access shows that, since files are normally written to a contiguous disk area, the next search begins from the current erase block 9A, significantly reducing the number of file searches.

[0063] Furthermore, the following Embodiment 2 is able to avoid this problem by implementing this part as a table. In this case, the size of the table used is, as described later, approximately 1/227th of the conventional logical/physical address conversion table used. However, if SRAM is used, data is lost when power is switched OFF, and the correspondence between the logical erase block number and the actual physical erase block number

must be verified by the CPU at semiconductor disk device power ON, and the RAM table rebuilt.

[0064] Next, the operation of writing data to the semiconductor disk device 2A is explained with Fig. 7. Step 40 – 46 in Fig. 7 are the same as step 30 – 36 in Fig. 6, and an explanation is therefore omitted. If the read data status flag 22 is '000', data is written to the relevant data storage area 21, and the data status flag 22 is set to '001' (step 47 - 50).

[0065] Furthermore, if the relevant data status flag 22 is not '000' in step 48, in other words, when an instruction is received to write to data storage area 21 to which data has already been written from host 1, data is written to a suitable empty data storage area 21 within the same erase block 9A, and the corresponding data status flag 22 is set to '001'. The data status flag 22 corresponding to the data storage area 21 to which data has already been written is changed from '001' to '011', and the previously written logical erase block number and offset value of the empty data storage area 21 are written to the update data link information storage area 23 (step 48, 51 - 54). Thus, by tracking through this link, it is always possible to read the most recent data.

[0066] Furthermore, in step 52, if there is no suitable empty data storage area 21 in the same erase block 9A, a search is made for an erase block 9A with a large number of data status flags [set to] '111'. This '111' data status flag is written with the erase (erase data) instruction. In this case, this is accommodated by switching erase blocks 9A (step 52, 55 – 58). The number of erase cycles must also be considered. Firstly, the valid data within the erase block 9A at a transfer source found under

the aforementioned conditions is transferred to an empty erase block 9A at a destination to which no data has been written. In this case, data which has become unnecessary in the overwriting process is not moved. Furthermore, the offset value of data which must be moved, and the offset value of the destination the data is moved to, are made to correspond. When link data exists, that data is also written. Immediately after data is transferred, the logical erase block number of the transfer source is written to the logical erase block number storage area 25 of the erase block 9A. The transfer source erase block 9A block is then erased.

[0067] In Embodiment 1, the logical sector/physical sector address conversion table can be eliminated without deteriorating the performance of the conventional semiconductor disk card. By eliminating the address conversion table, it is reasonable that the capacity of the semiconductor disk card can be greatly increased. Use of a conventional address conversion table requires an address conversion table of 80Kbyte with a 20MByte semiconductor disk device, and a 160KByte (1.25Mbit) address conversion table with a 40MByte semiconductor disk device. If this is eliminated, the cost for SRAM can be reduced, flash memory can be mounted in the space for mounting of the address conversion table SRAM memory, and the capacity of the semiconductor disk device can therefore be increased.

[0068] Embodiment 2. Embodiment 2 of this invention is explained in reference to Fig. 8 and Fig. 9. Fig. 8 is a block diagram showing the overall configuration of Embodiment 2 of the present invention. Fig. 9 is a diagram showing the internal configuration of the address conversion table in Fig. 8.

[0069] In Fig. 8, the semiconductor disk device 2B related to this Embodiment 2 is provided with an interface circuit 3, a CPU 4, an address conversion table 5A, a flash memory control circuit 6, a data I/O sector buffer 7, and a flash memory 8B.

[0070] In Fig. 9, the address conversion table 5A is comprised of a logical erase block number (PLBN) storage part and a physical erase block number (PBN) storage part.

[0071] The capacity of this address conversion table 5A is as follows. Assuming use of a 20MByte flash memory 8B, with each block (erase block) of 64KByte, the number of blocks in the semiconductor disk device 2B is as follows. Number of blocks in semiconductor disk device 2B = 20MByte / 64KByte = 320 blocks

[0072] Next, the number of bits necessary to express 320 in binary is found as $\ln 320 / \ln 2 = 9$, therefore 16 bits are necessary. Thus, the necessary capacity of the address conversion table 5A is $320 \times 9 = 2880$ bits, and 360byte is required. This is 1/227th of that conventionally required.

[0073] Operation of Embodiment 2 is fundamentally the same as that of the aforementioned Embodiment 1, however it differs in that the erase block 9A in the flash memory 8B can be accessed directly using the address conversion table 5A.

[0074] In Embodiment 2, the size of the address conversion table can be reduced without deterioration in the performance of the conventional semiconductor disk card. By reducing the size of the address conversion table, it is reasonable that the capacity of the semiconductor disk card can be greatly increased. Use of a conventional address conversion table requires an address conversion table of 80KByte with a 20MByte semiconductor disk, and a 160KByte (1.25Mbit) address conversion table with a 40MByte semiconductor disk. This is 1/227th of that previously required, allowing a reduction in the cost for SRAM, and furthermore, flash memory can be mounted in the space for mounting of the address conversion table SRAM memory, and the capacity of the semiconductor disk device can therefore be increased.

[0075]

[Benefits of the Invention] The semiconductor disk device related to the present invention is, as explained above, provided with a flash memory having multiple erase blocks and comprised of a logical erase block number storage area storing logical erase block numbers, multiple data storage

areas storing data, a data status flag area for each of the aforementioned data storage areas storing a data status flag indicating whether or not data is stored in the aforementioned data storage area, an update data link information storage area for each of the aforementioned data storage areas storing link information indicating the location of the update destination, and a CPU for management of data in the aforementioned flash memory based on the aforementioned logical erase block number, the aforementioned data status flag, and the aforementioned link information, demonstrating the advantages of elimination of the address conversion table for management of data, and consequent increase in the size of the data area.

[0076] Furthermore, as explained above, the CPU of the semiconductor disk device related to the present invention converts the logical sector address to a logical erase block number and its offset value, searches for the relevant erase block and relevant data storage area in the aforementioned flash memory based on this converted logical erase block number and its offset value, reads the content of the relevant data storage based on link information if the link information exists in the aforementioned relevant update data link information storage area, demonstrating the advantages of elimination of the address conversion table for management of data, and consequent increase in the size of the data area.

[0077] Furthermore, as explained above, the CPU of the semiconductor disk device related to the present invention converts the logical sector address to a logical erase block number and its offset value, searches for the relevant erase block and relevant data storage area in the aforementioned flash memory based on this converted logical erase block number and its offset value, writes data to the aforementioned relevant data storage area when the relevant data status flag indicates that it is 'unused', sets the aforementioned relevant data status flag to 'in use', searches for an empty data storage area when the aforementioned relevant data status flag indicates 'in use' and writes data to that [data storage area], setting the data status flag corresponding to this data storage area to 'in use', and writes link information to the update data link information storage area corresponding to the previous 'in use' data storage area, demonstrating the advantages of elimination of the address conversion table for management of data, and consequent increase in the size of the data area.

[0078] The semiconductor disk device related to the present invention is, as explained above, provided with a flash memory having multiple erase blocks and comprised of a logical erase block number storage area storing logical erase block numbers, multiple data storage areas storing data, a data status flag area for each of the aforementioned data storage areas storing a data status flag indicating whether or not data is stored in the aforementioned data storage area, an update data link information storage area for each of the aforementioned data storage areas storing link information indicating the location of the update destination, and a CPU for management of data in the aforementioned flash memory based on an address conversion table for converting the aforementioned logical erase block number to a physical erase block number, the aforementioned logical erase block number, the aforementioned data status flag, and the aforementioned link information, demonstrating the advantages of elimination of the address conversion table for management of data, and consequent increase in the size of the data area.

[0079] Furthermore, as explained above, the CPU of the semiconductor disk device related to the present invention converts the logical sector address to a logical erase block number and its offset value, converts the aforementioned logical erase block number to a physical erase block number based on the aforementioned address conversion table, searches for the relevant erase block and relevant storage area in the aforementioned flash memory based on this converted physical erase block and its offset value, and reads the content of the aforementioned relevant data storage area if link information does not exist in the aforementioned relevant update data link information storage area, demonstrating the advantages of elimination of the address conversion table for management of data, and consequent increase in the size of the data area.

[0080] Furthermore, as explained above, the CPU of the semiconductor disk device related to the present invention converts the logical sector address to a logical erase block number and its offset value, converts the aforementioned logical erase block number to a physical erase block number based on the aforementioned address conversion table, searches for the relevant erase block and relevant data storage area in the aforementioned flash memory based on this physical erase block and its offset value, writes data to the aforementioned relevant data storage area when the relevant data status flag indicates that it is 'unused', and sets the aforementioned relevant data status flag to 'in use', demonstrating the advantages of elimination of the address conversion table for management of data, and consequent increase in the size of the data area.

Brief Description of the Drawings

[FIG. 1] A block diagram showing the overall configuration of Embodiment 1 of the present invention.

[FIG. 2] A diagram showing the internal configuration of the flash memory of Embodiment 1 of the present invention.

[FIG. 3] A diagram showing the configuration of the erase block within the flash memory of Embodiment 1 of the present invention.

[FIG. 4] A diagram showing the configuration of the erase block information storage area within the erase block of Embodiment 1 of the present invention.

[FIG. 5] A diagram used in explaining data read operation in Embodiment 1 of the present invention.

[FIG. 6] A flow chart showing data read operation in Embodiment 1 of the present invention.

[FIG. 7] A flow chart showing data write operation in Embodiment 1 of the present invention.

[FIG. 8] A block diagram showing the overall configuration of Embodiment 2 of the present invention.

[FIG. 9] A diagram showing the configuration of the address conversion table of Embodiment 2 of the present invention.

[FIG. 10] A block diagram showing the overall configuration of a conventional semiconductor disk device.

[FIG. 11] A diagram showing the configuration of the address conversion table of a conventional semiconductor disk device.

[FIG. 12] A diagram showing the configuration of the flash memory of a conventional semiconductor disk device.

[FIG. 13] A diagram showing the configuration of an erase block within the flash memory of a conventional semiconductor disk device.

[FIG. 14] A diagram used in explaining data read operation in a conventional semiconductor disk device.

[FIG. 15] A diagram used in explaining data write operation in a conventional semiconductor disk device.

[FIG. 16] A diagram used in explaining data write operation in a conventional semiconductor disk device.

Reference Numerals

1 Host, 2A, 2B semiconductor disk device, 3 Interface circuit, 4 CPU, 5A Address conversion table, 6 Flash control circuit, 7 Data I/O sector buffer, 8A, 8B Flash memory

FIG. 1

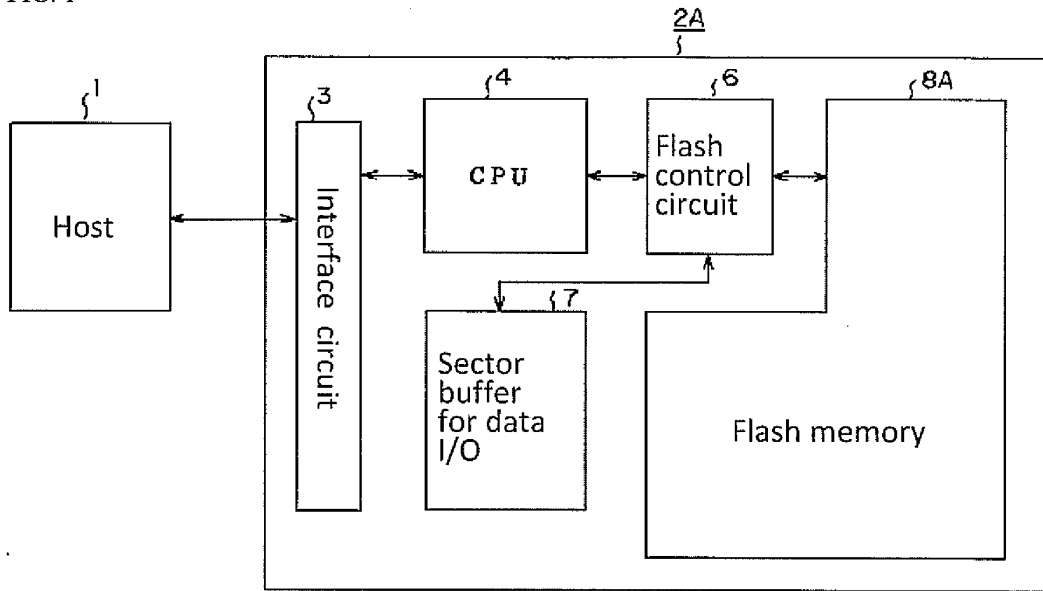


FIG. 2

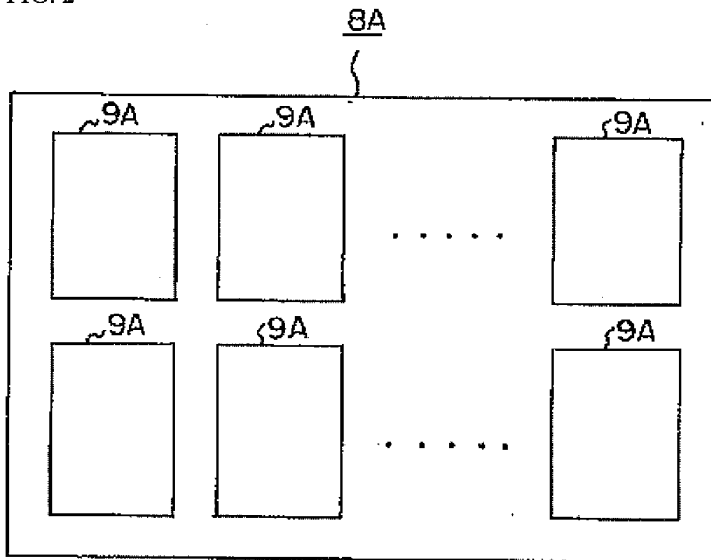


FIG. 3

9A

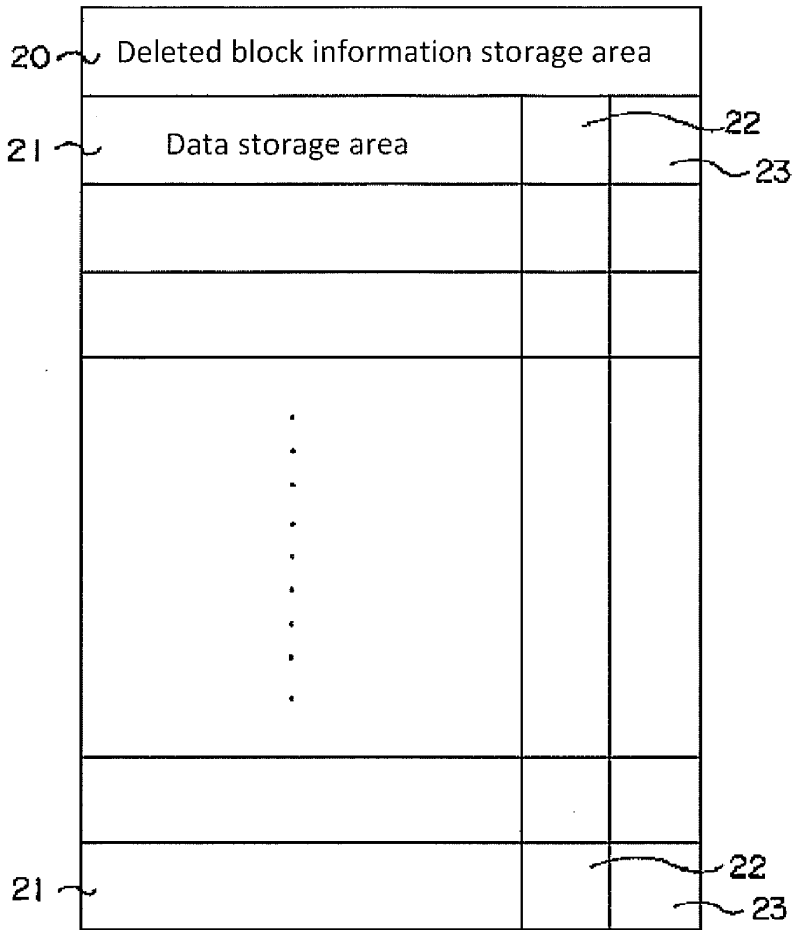


FIG. 4

20

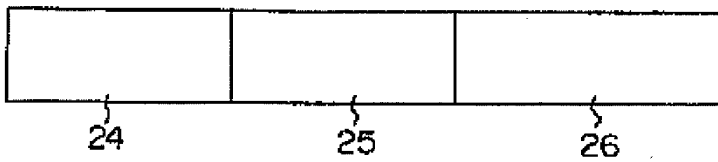


FIG. 5

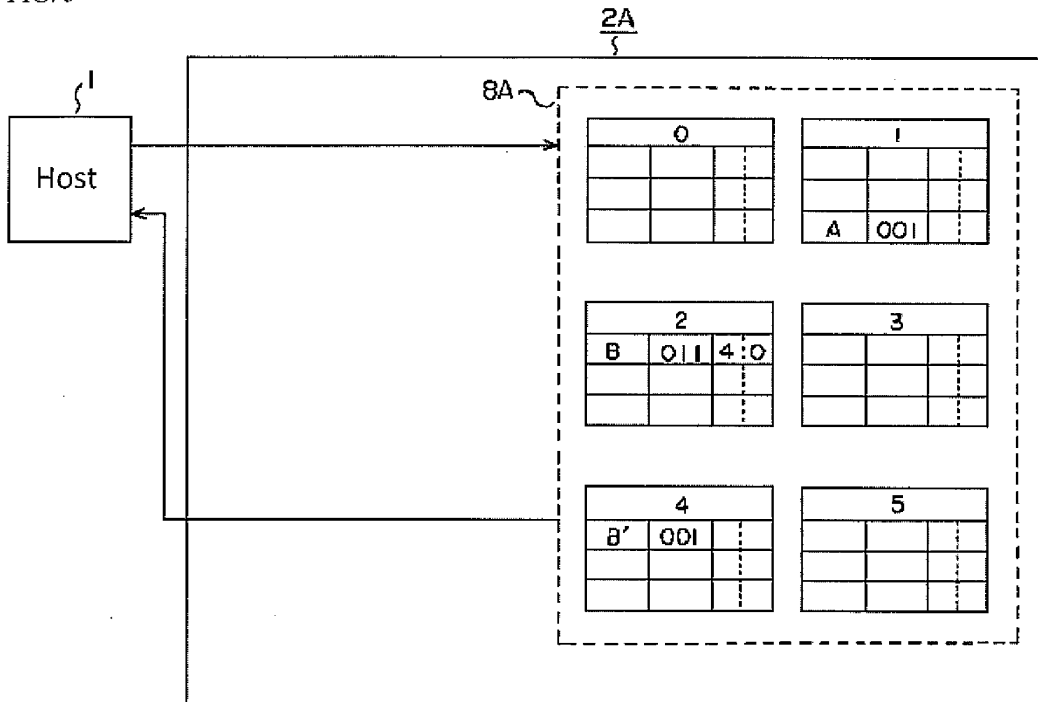


FIG. 6

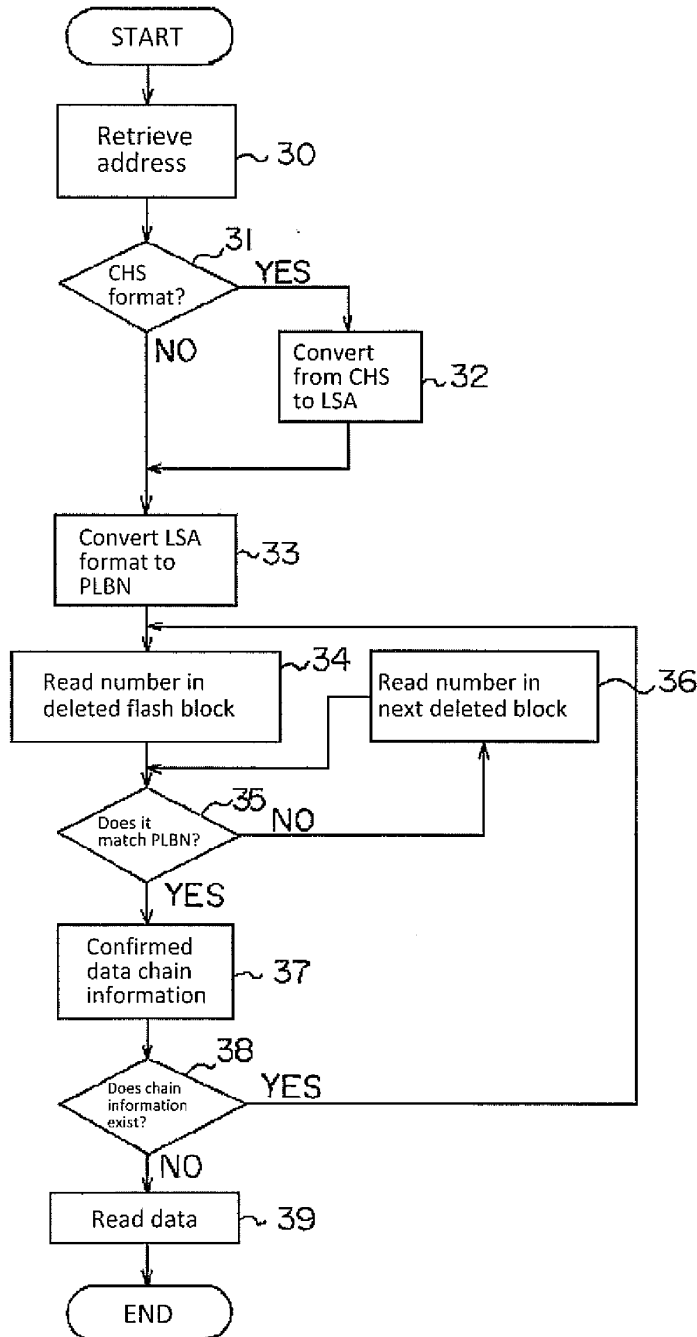


FIG. 7

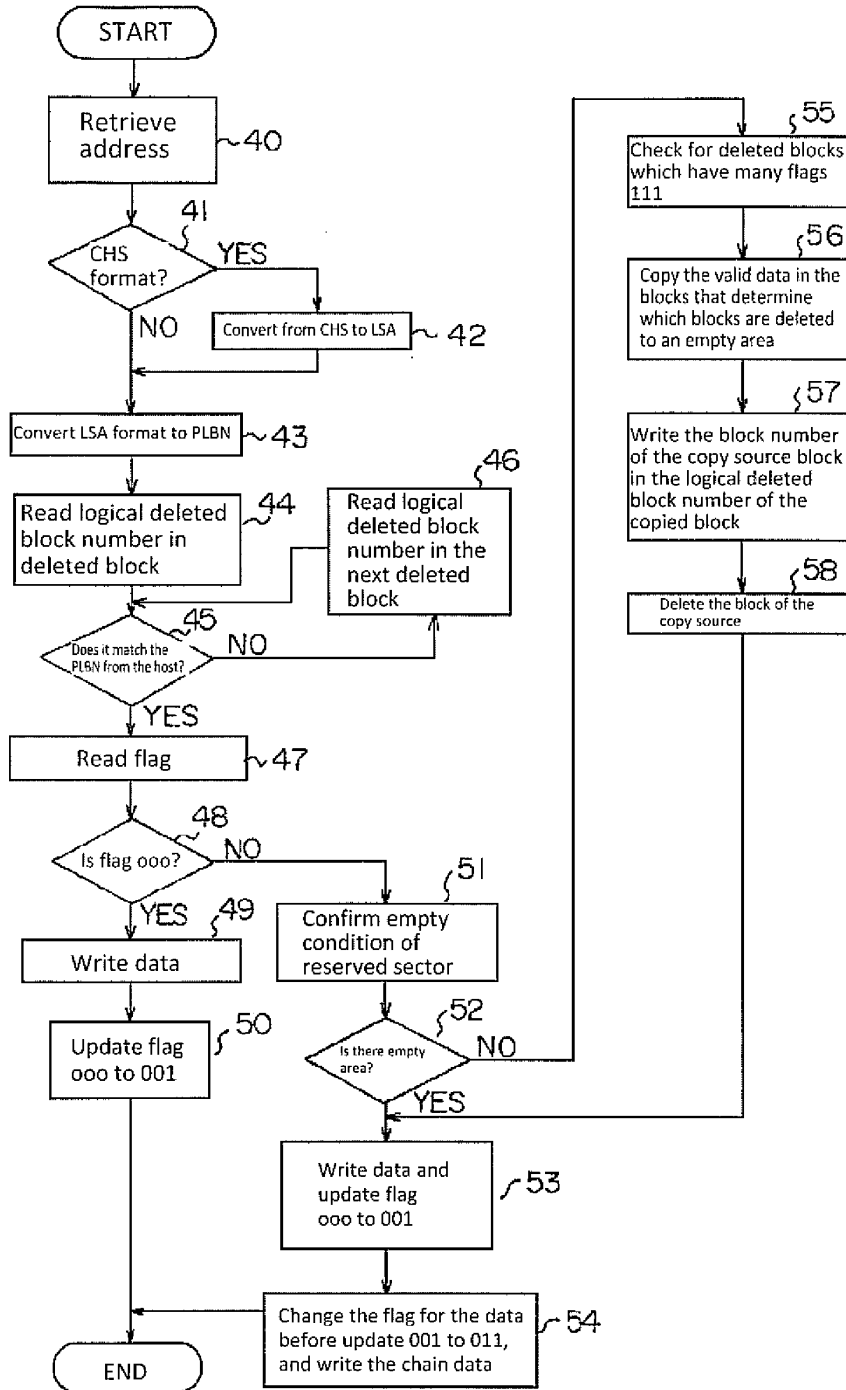


FIG. 8

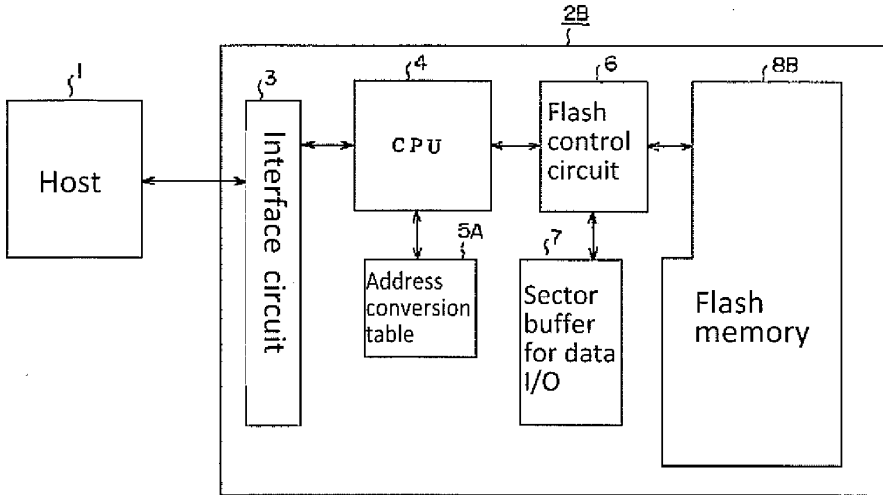


FIG. 9

5A

PLBN	PBN
1	3
2	8
3	n
⋮	⋮
⋮	⋮
⋮	⋮
⋮	⋮
⋮	⋮
n-1	
n	

FIG. 10

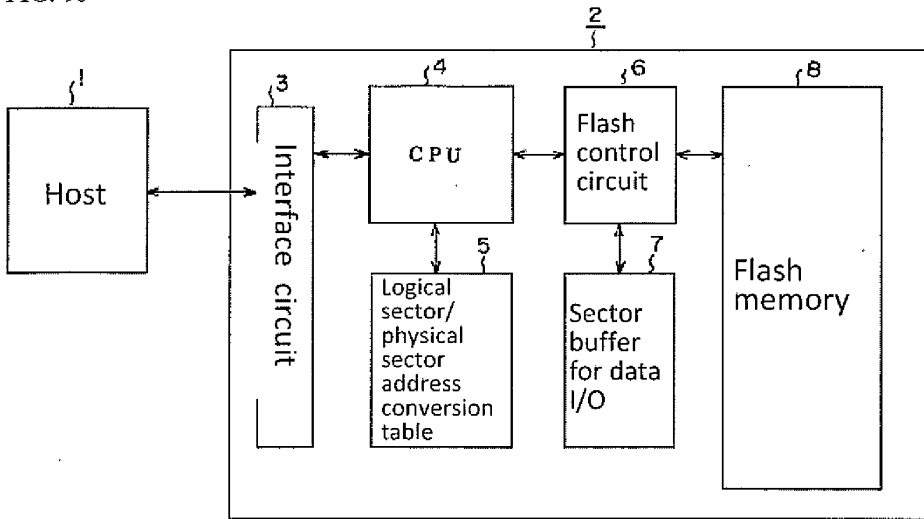


FIG. 11

5
}

LSA	PSA
1	6
2	1
3	3
4	n
5	5
⋮	⋮
⋮	⋮
⋮	⋮
n-1	
n	

FIG. 12

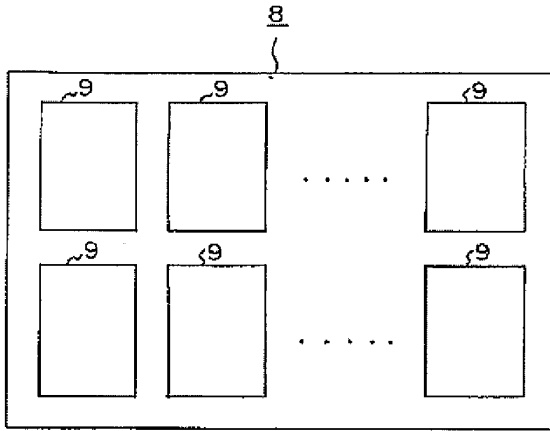


FIG. 13

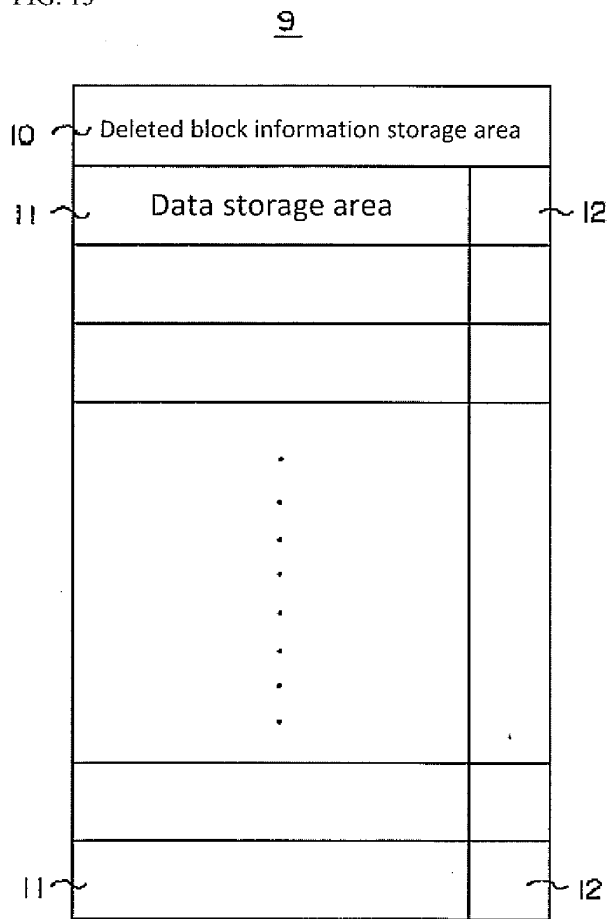


FIG. 14

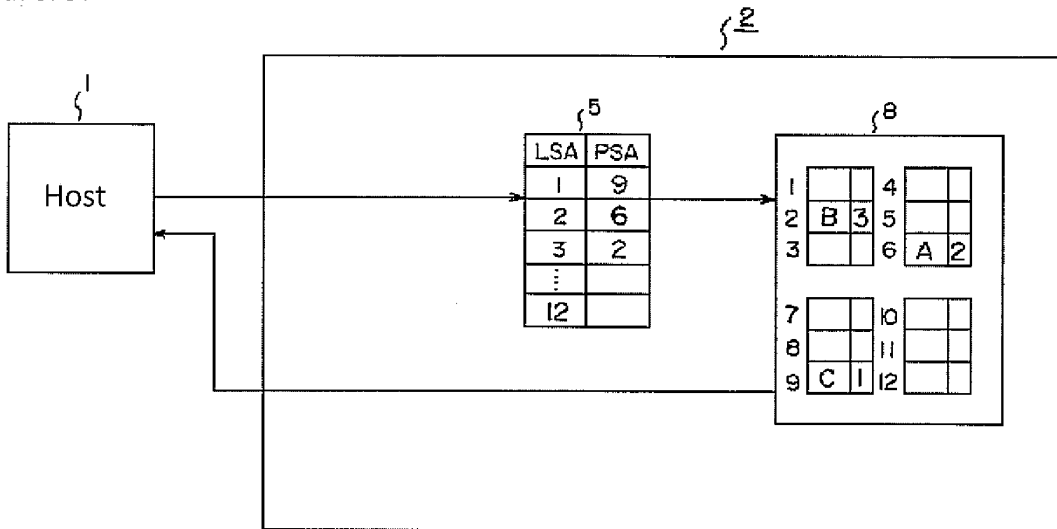


FIG. 15

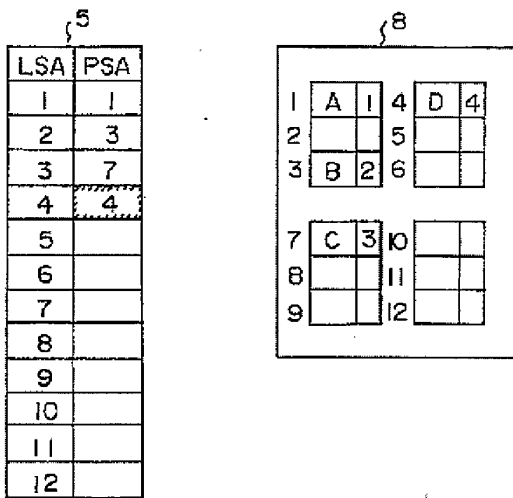
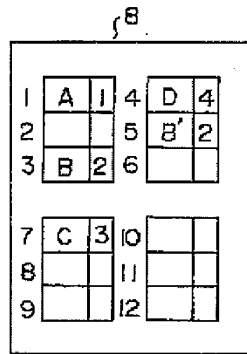


FIG. 16

5

LSA	PSA
1	1
2	5
3	7
4	4
5	
6	
7	
8	
9	
10	
11	
12	



(19) Japan Patent Office (JP)
(12) Japanese Patent Publication (B2)
(11) Patent No. #3,070,539 (P3070539)
(45) Issue Date July 31, 2000
(24) Registration Date May 26, 2000

(51) Int. Cl.7	ID No.	FI			
G06F 12/16	340		G06F 12/16	340P	
	310			310J	
	3/06	305		3/06	305A
	3/08			3/8	C
					H

Number of Claims 3 (Total 25 pages) Continued on last page

(21) Patent Number Application H9-267178
(22) Application Date September 30, 1997
(65) Disclosure No. Unexamined Application H11-110300
(43) Disclosure Date April 23, 1999
Date of Examination Request September 30, 1997

(73) Patent holders 000002185
Sony Corporation
7-35 Kitashinagawa 6-chome, Shinagawa-ku, Tokyo
(72) Inventor FUSE, Hiroaki
c/o Sony Corporation, 7-35 Kitashinagawa 6-chome, Shinagawa-ku, Tokyo
(72) Inventor SASA, Tetsu
c/o Sony Corporation, 7-35 Kitashinagawa 6-chome, Shinagawa-ku, Tokyo
(72) Inventor OGAMI, Jun
c/o Sony Corporation, 7-35 Kitashinagawa 6-chome, Shinagawa-ku, Tokyo
(74) Agent 100067736
Patent Attorney KOIKE, Noboru (and two others)
Chief Examiner UMEMURA, Keiju

Continued on last page

(54) [Title of the Invention] External Storage Device, Data Processing Unit, and Data Processing Method

(57) [Scope of Patent Claims]

[Claim 1] An external storage apparatus characterized by the fact that it is connected to a data processing apparatus and is attachable thereto and removable therefrom, and which stores data from the above-mentioned data processing apparatus, is equipped with a serial interface for the purpose of exchanging serial data with the above-mentioned data processing apparatus, and a nonvolatile memory that stores the above-mentioned data which is separated into multiple blocks that can be erased in a batch, blocks to store the logical address and identification number that indicates the new version of the data stored in the block so that should multiple blocks with the same logical address exist, the new version of the data stored in these blocks is identified based on the identification number, and the new data is deemed to be the invalid data and the old data is deemed to be the valid data.

[Claim 2] In addition, the data processing apparatus is separated into the serial interface for the purpose of exchanging serial data and several blocks that can be erased in a batch. The external storage apparatus is equipped with the nonvolatile memory that stores data in the block and can be attached to or detached from the data processing equipment. When data is stored in the external storage apparatus, the logical address and the identification number that indicates the new version of the data stored in the block are stored in the nonvolatile memory block. When data is read from the external storage apparatus, in the event that multiple blocks with the same logical address exist, the new version of the data that is stored in these blocks is identified based on the identification number, and the new data is deemed to be the invalid data and the old data is deemed to be the valid data. These are the features of the data processing apparatus.

[Claim 3] When data is stored in the external storage apparatus, which contains the nonvolatile memory that is separated into multiple blocks that can be erased in a batch, the block of the nonvolatile memory stores the logical address and the identification number that indicates the new version of the data stored in the block. When data is read from the external storage apparatus, in the event that multiple blocks with the same logical address exist, the new version of the data that is stored in these blocks is identified based on the identification number, and the new data is deemed to be the invalid data and the old data is deemed to be the valid data. These are the features of the data processing method.

[Detailed Explanation of the Invention]

[0001]

[Technical Field of the Invention] The present invention concerns the external storage apparatus in which the storage domain is separated into multiple blocks. In addition, the present invention concerns the data processing apparatus that stores data in the external storage apparatus in which the storage domain is separated into multiple blocks. The present invention also concerns the data processing method at the time when data is stored in the external storage apparatus in which the storage domain is separated into multiple blocks.

[0002]

[Prior Art] There are external storage apparatus equipped with flash memory used with data processing equipment, such as personal computers (PCs) and digital still cameras.

[0003] The external storage apparatus equipped with flash memory is separated into multiple blocks of storage domains, and controls the data domains in units of blocks. Accordingly, each block constitutes a unit for the erasing of data. That is to say, at the time when the data is erased, the initialization process is implemented for the entire block that contains such data. Consequently, such blocks that store data are erased in a batch.

[0004] When data is stored in blocks in this type of external storage apparatus, unique logical addresses are configured for these blocks. Accordingly, each block is controlled by using this logical address.

[0005] In addition, the data stored in the external storage apparatus is usually stored in the external storage apparatus in file units. However, in the event that one (1) file encompasses multiple blocks, consolidated information for these blocks becomes necessary. Accordingly, in the event that one (1) file encompasses multiple blocks, the logical address of the next block is stored in the block that stores such file (hereinafter abbreviated as a "link address").

[0006]

[Problems to be Resolved by the Invention] Previously, in the event that there was a process to detect whether or not there was an error within the storage domain and an error was detected in this type of external storage apparatus, the process that attempted to correct such error was performed each time the external storage apparatus was booted up. Furthermore, this type of process is referred to as an "error detection and correction process" in the following explanation. Usually, this error detection and correction process is one that requires time because the load is comparatively large. Therefore, there was a problem because the existing external storage apparatus could not be booted up quickly for the error detection and correction process.

[0007] In addition, with respect to the external storage apparatus that controls the data domains in units of blocks, in the event that the electrical power is suddenly cut off or the external storage apparatus is forcibly removed from the data processing apparatus when the data is newly input in the block or when the data that is stored in the block is updated, there is a possibility that a situation may arise in which the same logical address exists simultaneously for multiple blocks (hereinafter referred to as a "logical address error") or a situation in which a block that is indicated in the link address does not exist (hereinafter referred to as a "link address error"). Of course, in either situation, the file is unexpectedly linked to the block and the external storage apparatus cannot be used in the normal manner.

[0008] Nevertheless, the existing external storage apparatus was not equipped with the function to detect and properly repair logical address or link address errors. Consequently, in the past, in the event that the electrical power was suddenly cut off or the external storage apparatus was forcibly removed from the data processing apparatus, the external storage apparatus could not be used in the normal manner.

[0009] The present invention has been proposed in light of these above-mentioned past situations and for the purpose of being able to detect and properly repair such errors as logical address and link address errors.

[0010]

[Means for Resolving Problems] The external storage apparatus related to the Invention is connected to the data processing apparatus and can be attached thereto and removed therefrom, is the external storage apparatus that stores the data from the data processing apparatus, and is equipped with a serial interface for the purpose of exchanging serial data with the data processing apparatus and with a nonvolatile memory that stores the data. The nonvolatile memory is separated into multiple blocks that can be erased in a batch, and the block stores the logical address and the identification number that indicates the new version of the data stored in the block. In the event that multiple blocks with the same logical address exist, the new version of the data that is stored in these blocks is identified based on the identification number, and the new data is deemed to be the invalid data and the old data is deemed to be the valid data.

[0011]

[0012] In addition, the data processing apparatus related to the present invention is separated into the serial interface for the purpose of exchanging serial data and the multiple blocks that can be

erased in a batch. The external storage apparatus is equipped with the nonvolatile memory that stores data in the block and can be attached to or removed from the data processing apparatus. When data is stored in the external storage apparatus, the logical address and the identification number that indicates the new version of the data stored in the block are stored in the nonvolatile memory block. When data is read from the external storage apparatus, in the event that multiple blocks with the same logical address exist, the new version of the data that is stored in these blocks is identified based on the identification number, and the new data is deemed to be the invalid data and the old data is deemed to be the valid data.

[0013]

[0014] Furthermore, with respect to the data processing method for the present invention, at the time when the data is stored in the external storage apparatus, which contains the nonvolatile memory separated into multiple blocks that can be erased in a batch, the logical address and the identification number that indicates the new version of the data stored in the block are stored in the nonvolatile memory block. When data is stored in the external storage apparatus, in the event that multiple blocks with the same logical address exist, the new version of the data that is stored in these blocks is identified based on the identification number, and the new data is deemed to be the invalid data and the old data is deemed to be the valid data.

[0015]

[0016]

[Operation of the Invention] The operation of the present invention is explained in detail below by referring to the figures.

[0017] 1. Overall System Configuration

An example of the overall system configuration to which the invention is applied is indicated in FIG.1. This system is configured from data processing apparatus 1, which constitutes the host system, and memory card 2, which is the external storage apparatus connected to the data processing apparatus 1 through the serial interface.

[0018]

[0019] Data processing apparatus 1 is equipped with a central processing unit (CPU), internal memory 4, auxiliary storage apparatus 5 and serial interface circuit 6, and these are connected to each other via the bus 7. This data processing apparatus 1 reads, for example, the programs stored on auxiliary storage apparatus 5, uses such programs in internal memory 4 as the work area, and operates via CPU3. At this time, data is exchanged between serial interface circuit 6 and memory card 2 whenever necessary.

[0020] Furthermore, the exchange of information is not restricted only between data processing apparatus 1, which is used in the system to which the invention is applied, and the external storage apparatus; the present invention can be applied to various data processing equipment, such as personal computers (PCs), digital still cameras, digital video cameras, etc

[0021] The data processing apparatus 1 and the memory card 2 connect through the serial interface; to be specific, through at least three (3) data lines, i.e., SCLK, State and DIO. That is to say, data processing apparatus 1 and memory card 2 are connected through at least the first data line SCLK, which transmits the clock signal when data is transmitted, the second data line State, which transmits the status signal necessary when data is transmitted, and the third data line DIO, which transmits the data written in memory card 2 or the data read from memory card 2 to the serial. Data is exchanged between data processing apparatus 1 and memory card 2 through these lines.

[0022] The exchange of data between data processing apparatus 1 and memory card 2 is usually conducted in file units configured with headers and actual data. The file header also stores, for example, information used to access the file and information, etc. that is necessary for the programs operated in data processing apparatus 1.

[0023] 2. Configuration of Memory Card

As indicated in FIG. 2, memory card 2 is equipped with controller 11 originating from what is called a control IC, and flash memory 12, which is controlled by controller 11.

[0024] Controller 11 is equipped with serial/parallel and parallel/serial interface sequencer 13 for converting serial/parallel and parallel/serial (hereinafter referred to as "S/P & P/S interface sequencer 13"), flash memory interface sequencer 14 that controls the interface of flash memory 12, page buffer 15 that simultaneously stores the data exchanged between S/P & P/S interface sequencer 13, flash memory interface sequencer 14, error correction circuit 16 that conducts the processing of error correction, command generator 17 that conducts the generation, etc. of control commands that control access to flash memory 12, configuration ROM 18 that stores the version information and each type of attribute information, etc. for memory card 2, and oscillator 19 that provides clock signals necessary for these operations for each circuit.

[0025] S/P & P/S interface sequencer 13 is connected to serial interface circuit 6 of the data processing apparatus 1 through at least the above-mentioned three (3) lines, i.e., SCLK, State and DIO, and data is exchanged with data processing apparatus 1 through these data lines SCLK, State and DIO. That is to say, S/P & P/S interface sequencer 13 converts the parallel data that is sent from page buffer 15 into serial data and sends it to the serial interface circuit 6 of data processing apparatus 1. In addition, S/P & P/S interface sequencer 13 converts serial data sent from serial interface circuit 6 of the data processing apparatus 1 to parallel data, and sends it to page buffer 15.

[0026] This transmission between S/P & P/S interface sequencer 13 and the serial interface circuit 6 of data processing apparatus 1 is synchronized with the clock signal sent from data processing apparatus 1 via the first data line SCLK and is conducted via the third data line DIO. Then, the data classification of the serial data exchanged via the third data line DIO is identified by the status signal transmitted via the second data line State. Accordingly, the classification of the serial data consists, for example, of the data to be stored in flash memory 12, the data read from flash memory 12, and the control data used for the purpose of controlling the operation of the memory card 2, etc. Furthermore, the status signal is used to indicate the status of memory card 2. The status of memory card 2 from the status signal may,

for example, indicate that the memory card has started certain processes that do not allow the card to receive input data from data processing apparatus 1, or indicate that memory card 2 may be waiting for input data from the data processing apparatus after completing a process.

[0027] In addition, the S/P & P/S interface sequencer 13 sends the control data to the command generator 17 if the aforementioned control data sent from data processing apparatus 1 controls the operation of memory card 2.

[0028] Command generator 17 generates control commands to control the access to flash memory 12 based on the control data sent from data processing apparatus 1 via S/P & P/S interface sequencer 13 and sends the aforementioned control command to flash memory interface sequencer 14. Flash memory interface sequencer 14 writes data to flash memory 12 and reads data from flash memory 12 based on this control command, as described below.

[0029] Furthermore, this command generator 17 is connected to error deletion prevention switch 20. When error-deletion prevention switch 20 is on, even if control data with instructions to delete data written on flash memory 12 is sent from data processing apparatus 1, command generator 17 does not generate a control command to delete the data written to flash memory 12. In other words, memory card 2 can switch between modes that protect and allow data stored to flash memory 12 to be deleted using the error deletion prevention switch 20.

[0030] Page buffer 15, situated between S/P & P/S interface sequencer 13 and flash memory interface sequencer 14, is a so-called buffer memory that temporarily records data processes between the S/P & P/S interface sequencer 13 and flash memory interface sequencer 14.

[0031] In other words, data sent to flash memory interface sequencer 14 from the S/P & P/S interface sequencer 13 is first sent to page buffer 15 from S/P & P/S interface sequencer 13 and is temporarily recorded by this page buffer 15. At this time, error correction circuit 16 attaches an error correction code to the data recorded to the page buffer 15. Data attached with the error correction code is then sent to flash memory interface sequencer 14 for each page unit (i.e. 1 page = 512 bytes) designated by the page buffer 15.

[0032] Or, data sent from flash memory interface sequencer 14 to S/P & P/S interface sequencer 13 is first sent to page buffer 15 from the flash memory interface sequencer and is temporarily recorded by page buffer 15. At this time, error correction circuit 16 executes an error correction process in the data recorded to page buffer 15. Data executed with the error correction process is then sent to the S/P & P/S interface sequencer 13 for each page unit designated by the page buffer 15.

[0033] Flash memory interface sequencer 14 writes data to flash memory 12 or reads data from flash memory 12 according to the command from command generator 17. In other words, flash memory interface sequencer 14 reads data from flash memory 2 according to the control commands from command generator 17 and sends the aforementioned data to S/P & P/S interface sequencer 13 via page buffer 15 as described above or, flash memory interface sequencer 14 receives data from S/P & P/S interface sequencer 13 according to the control commands from command generator 17 via page buffer 15 as described above, and writes the aforementioned data to flash memory 12.

[0034] Configuration ROM 18 stores this memory card's version and various attribute information. The information stored to configuration ROM 18 is read and used as needed by command generator 17 via S/P & P/S interface sequencer 13. In other words, command generator 17 reads information stored to configuration ROM 18, as deemed necessary, and configures the various settings related to memory card 2 based on this information.

[0035] When the data written to flash memory 12 is sent data processing apparatus 1 as serial data to the above memory card 2, via the three aforementioned data lines, SCLK, State, and DIO, the S/P & P/S interface sequencer 13 converts the aforementioned serial data to parallel data and sends the aforementioned parallel data to page buffer 15. Page buffer 15 temporarily records data sent from S/P & P/S interface sequencer 13. At this time, error correction circuit 16 attaches an error correction code to the data recorded to page buffer 15. The data attached with the error-correction code is then sent to flash memory interface sequencer 14 for each designated page unit. Flash memory interface sequencer 14 then writes data sent from page buffer 15 to flash memory 12 according to the control commands from command generator 17. The above processes writes data sent from data processing apparatus 1 to flash memory 12.

[0036] In addition, when data is read from memory card 2 as described above, flash memory interface sequencer 14 reads data from flash memory 12 first, according to the control commands from command generator 17. Flash memory interface sequencer 14 then sends data read from flash memory 12 to page buffer 15. Page buffer 15 temporarily records data sent from flash memory interface sequencer 14. At this time, error correction circuit 16 executes an error correction process in the data recorded to page buffer 15. Data executed with the error correction process is then sent to S/P & P/S interface sequencer 13 for each designated page unit. S/P & P/S interface sequencer 13 then sends data sent from page buffer 15, after converting it to serial data, to data processing apparatus 1, via the three aforementioned data lines, SCLK, State, and DIO. The above process sends the data read from flash memory 12 to data processing apparatus 1.

[0037] Furthermore, the control data not only controls the processes of writing to flash memory 12 and reading from flash memory 12 when reading and writing data, the data can also be sent from data processing apparatus 1 to S/P & P/S interface sequencer 13 of memory card 2. This control data is sent to command generator 17 from S/P & P/S interface sequencer 13. Command generator 17 then generates control commands that control access to flash memory 12 according to the control data sent from S/P & P/S interface sequencer 13. The control command is then sent to the flash memory interface sequencer 14 and the flash memory interface sequencer 14 accesses the flash memory 12 according to the control command to read or write the data.

[0038] In addition to the aforementioned 3 data lines, SCLK, State, and DIO, power supply lines or unused reserve lines may be prepared for the memory card 2. For example, FIG. 2 and FIG. 3 in the following, provide an example of the aforementioned 3 data lines of SCLK, State, and DIO; the 4 power lines of VSS1, VSS2, VCC, and INT; and the 3 reserve lines of RSV1, RSV2, and RSV3 connected to the memory card 2.

[0039] 3. Dimensions of the memory card

Refer to FIG. 3 for a detailed description of the dimensions of the aforementioned memory card 2.

[0040] The memory card 2 is embedded with a controller 11 and flash memory 12, as described above, in a flat, rectangular, and thin-walled card case 21 created from synthetic resins. This memory card 2 is then placed into the data processing apparatus 1 with a memory card 2 slot.

[0041] There are diagonal notches 22 formed at the front edge of the memory card 2 case 21 and there are 10 grooves 23 formed by the aforementioned notches 22. There are external connection terminals located within these grooves 23 that connect to the connection terminals of the data processing apparatus 1 when the memory card 2 is placed in the memory slot of the data processing apparatus 1. In other words, this memory card 2 has 10 external connection terminals: 24a, 24b, 24c, 24d, 24e, 24f, 24g, 24h, 24i, and 24j. The breakdown of these terminals: 3 data line terminals 24b, 24d, and 24h; 4 electric terminals 24a, 24f, 24i, and 24j; and 3 reserved terminals 24c, 24e, and 24g.

[0042] The error deletion prevention component 25 is attached to the top of the case 21 of the memory card. Error deletion prevention component 25 is connected to the aforementioned error deletion prevention switch 20 installed within the case 21. Sliding the error deletion prevention component 25 switches the error deletion prevention switch 20 on/off.

[0043] The memory card 2 has a semi-circular lock notch 26 (#1) on one side of the case 20 to ensure that the memory card 2 inserted into the data processing apparatus 1 slot does not disengage. The other side of the case 20 also has a rectangular lock notch 27 (#2). When the memory card 2 is inserted into data processing apparatus 1, the lock notches 26, 27 engage with the slot of the data processing apparatus 1 to ensure the memory card 2 does not disengage.

[0044] Furthermore, memory card 2 displayed in FIG. 3 is just one example of an external recording device compatible with this invention. In other words, this invention can be adapted to any external recording device regardless of its external shape.

[0045] 4. Configuration of the storage area

The following describes the storage area configuration of flash memory 12 onboard the above memory card 2.

[0046] As described in FIG 4(a), the storage area of this flash memory 12 is divided into multiple blocks as data deletion units. Included among these blocks is a boot block to store boot data initially read by data processing apparatus 1 when memory card 2 starts up, and a data block where any other data is written. Each of these blocks has a specific physical address attached to it. These blocks are units used for deleting data, as well as, the smallest file management unit. In other words, a file may be stored to one or more blocks, but multiple files cannot be stored to a single block.

[0047] The status of each block is indicated by the attached bits, "1" or "0". The initial mode for all the bits is "1" and the mode can be changed by switching the bit from "1" to "0". In other words,

when writing data as "1" or "0", the aforementioned bit is not changed in the event of "1" but the aforementioned bit is changed from "1" to "0" in the event of "0".

[0048] To delete written data, execute the batch initialization process on the block units to switch all the bits of the aforementioned blocks to "1". This process deletes the data written to the aforementioned blocks in a batch and the blocks are then available to be rewritten with data.

[0049] Furthermore, when the units are switched from "0" to "1", the batch initialization process must be executed on the block units and all bits of the aforementioned blocks are changed to "1". The batch initialization process for block units is unnecessary when the units are switched from "1" to "0". In the following description, the process of block units switching from "1" to "0" without a batch initialization process is called an overwrite.

[0050] This invention is not only compatible with flash memory that can switch each bit between 2 modes (dual type flash memory), as described above, but is also compatible with flash memory that can switch each bit to 3 or more modes (multi-type flash memory).

[0051] Each block of the aforementioned flash memory 12 is composed of a plurality of pages which are units for writing or reading data, as shown in FIG. 4B. In other words, when data is written on flash memory 12, flash memory interface sequencer 14 writes data from page buffer 15 in page units to flash memory 12 in page units, as stated above. When data is read from flash memory 12, flash memory interface sequencer 14 reads data in page units and sends it to page buffer 15.

[0052] Each page has a data area and a redundant area. The data area is a region used to store arbitrary data. The redundant area is a region in which information required to manage data, which is written to the data area, is stored.

[0053] Specifically, information known as distributed administration information is stored in the redundant area on the leading page of the block as information required to administer the block, as shown in FIG. 4C. Also, distributed administration information which is the same as that stored in the redundant area of the leading end page is stored in the redundant area of subsequent pages of the block as spare distributed administration information. However, the redundant area of the last page stores additional data that cannot be administered with the distributed administration information alone. This data is known as additional administration information, rather than as distributed administration information.

[0054] As described above, distributed administration information is stored in the redundant area of each block of flash memory 12. Distributed administration information is information for administering the block in which distributed administration information has been stored.

Information that can be obtained from this distributed administration information includes whether the corresponding block is the leading end of the file, and, in cases where a file consists of multiple blocks, the link between these blocks. More details on distributed administration information are given later.

[0055] Distributed administration information of the blocks in memory card 2 are collected into management information set, which serves as information for administering the overall structure of the flash memory. This management information set is stored as a file in flash memory 12.

[0056] Usually, management information set is used to obtain information required to access each block. That is, when data is communicated between data processing apparatus 1 and memory card 2, data processing apparatus 1 reads management information set from memory card 2 and creates an administration table from it within internal memory 4, which data processing apparatus 1 uses to access memory card 2. This makes it unnecessary to access the distributed administration information stored in each block on every data access, resulting in faster data access.

[0057] 5. Distributed administration information

The following describes distributed administration information in detail.

[0058] Distributed administration information is information for administering the block in which distributed administration information has been stored, and is written to a 16-byte redundant area. Specifically, the distributed administration information is, as shown in FIG. 5, composed of a 1-byte enabled/disabled flag, a 1-byte block flag, a 4-bit final flag, a 4-bit reference flag, a 1-byte administration flag, a 2-byte logical address, a 2-byte link address, a 3-byte reserved area, a 2-byte distributed administration information error-correction code and a 3-byte data error-correction code.

[0059] The enabled/disabled flag indicates whether the block is in an enabled or disabled state. Specifically, it indicates one of 2 states, "enabled" or "disabled". "Enabled" indicates that the block can be used, and "disabled" indicates that the block cannot be used. For example, when an irrecoverable error occurs in the block, this flag is set to "disabled", and the block cannot be used.

[0060] The block flag shows the state of the block. Specifically, the block flag indicates one of four states: "not in use", "in use at leading end", "in use" and "not erased". "Not in use" indicates a state in which the block is not in use or data has been erased from the block. Thus, the block is in the initial state (all bits are 1) and data can be written right away. "In use at leading end" indicates a state in which the block is being used at the leading end of the file. The block flag for a boot block in which boot data is stored is set to "in use at leading end". "In use" indicates a state in which the block is in use other than at the leading end of the file. When the block flag indicates "in use", the block is linked to another block. "Not erased" indicates a state in which the data in the block is invalid. For example, when data is being erased, the block flag is set to "not erased", and the data in the block is erased later when there is spare processing time, making erase operations more efficient.

[0061] The final flag indicates whether or not the file is done. Specifically, it indicates one of 2 states, "block continued" and "final block". "Block continued" indicates that there is a link to the next block. That is, "block continued" indicates that there is more data in the file stored in this block and that the file is continued to another block. "Final block" indicates that the block is the last block for the file. That is, it indicates that the file whose data is stored in this block is complete with this block.

[0062] The reference flag indicates whether the additional administration information should be referenced. Specifically, it indicates one of 2 states, "no reference information" and "reference information exists". "No reference information" indicates that there is no valid additional administration information in the redundant area of the last page of the block. "Reference information" indicates that there is valid additional administration information in the redundant area of the last page of the block.

[0063] The administration flag indicates properties of the block. For example, it indicates whether the block is read only or write enabled. Or, it may indicate whether the block is a boot block or a data block.

[0064] As its name indicates, the logical flag indicates the block's logical address. The logical address is set as needed, for example, when data is overwritten. During normal processing, the logical address is set so that multiple blocks do not have the same logical address value.

[0065] As stated earlier, in flash memory, a block erase must always be done before data is overwritten in the same block. But there is a limit to the number of guaranteed erase/rewrite cycles, so the number of times this cycle is performed must be minimized. Hence, when overwriting block data, the new data is written to a different block. When this happens, the block flag of the block where the previous data was stored is set to "not erased" to indicate that the data in that block is now invalid. Then, in memory card 2, in order that blocks that contain the same data have the same address when data is updated as described above, a dynamic logical address distinct from the preset physical address is assigned to each block, and the blocks are referenced through this logical address.

[0066] The link address indicates the logical address of the block that is linked to the block. That is, the file is continued to another block. In this case, the link address contains the logical address of the next block, which contains the next portion of the data in the file.

[0067] The distributed administration information error correction code is an error correction code for the distributed administration information's administration flag, logical address, link address and reserved area. The enabled/disabled flag, block flag, final flag and reference flag are not subject to error correction through the distributed administration information error correction code. Therefore, the enabled/disabled flag, block flag, final flag and reference flag can be overwritten without updating the distributed administration information error correction code.

[0068] The data error correction code is an error correction code corresponding to the data in the data area of the page where the code is stored.

[0069] The distributed administration information error correction code and data error correction code are used by the error correction circuit 16 inside memory card 2. Therefore, error correction using these error correction codes does not depend on data processing apparatus 1 and any method that works with memory card 2 may be used.

[0070] 6. Additional administration information

The following describes additional administration information in detail.

[0071] Additional administration information is information stored in the 16-byte redundant area of the last page and contains additional data that cannot be administered with the distributed administration information alone.

[0072] Specifically, additional administration information is, as shown in FIG. 6, composed of a 1-byte enabled/disabled flag, a 1-byte block flag, a 4-bit final flag, a 4-bit reference flag, a 1-byte identification number, a 2-byte effective data size, a 5-byte reserved area, a 2-byte additional administration information error correction code and a 3-byte data error correction code.

[0073] The enabled/disabled flag, block flag, final flag, reference flag, reserved area and data error correction code are the same as for distributed administration information. The additional administration information error correction code corresponds to the distributed administration information error correction code, and is used with the additional administration information's identification number, effective data size and the data in the reserved area.

[0074] The identification number and effective data size are the additional information that cannot be administered with the distributed administration information alone.

[0075] The identification number is used in error processing. Its value is incremented each time the block's data is overwritten. When an error occurs such that multiple blocks have the same logical address, the identification numbers are used to determine whether the data in each block is new or old. The identification number takes up one byte, and can take on values between 0 and 255. Its initial value is 0. When an identification number must be incremented and its value is 255, it is reset to 0. When there are multiple data blocks with the same logical address, the data block with the smallest identification number is determined to be valid. However, if a boot block has a backup boot block, those two blocks have the same identification number under normal operation. If due to an abnormal condition, those blocks' identification numbers become different, the one with the larger identification number is determined to be valid.

[0076] The effective data size indicates the size of the valid data within the block. That is, if the block has unused areas, the effective data size is the size of the data written to the data area. Here, the reference flag in the distributed administration information is set to "reference information exists". If the data area of the block does not have any unused areas, the effective data size is assigned 0xffff.

[0077] The above distributed administration information and additional administration information are updated each time the data in the block is updated.

[0078] 7. management information set

The following describes management information set in detail.

[0079] As stated earlier, management information set is created by collecting distributed administration information of the blocks in memory card 2 and is stored as a file in flash memory 12. That is, as shown in FIG. 7, a file consisting of management information set is created for collectively administering all blocks from the distributed administration information of each block. This management information set is stored in the data area of a predetermined block. Now, the management information set, may be saved in one block or may be saved across multiple blocks. Then, data processing apparatus 1, normally, is configured to attempt to derive the necessary information to access every block, based on this management information set.

[0080] In other words, when the useful management information set in memory card 2 is saved as a file, data processing apparatus 1 reads-out that management information set and expands the content to internal memory 4, and constructs a management table to manage memory card 2. Now the physical address of the block where the header of the management information set file is saved is included in the boot data, and the data processing apparatus 1 access the management information set file based on that physical address.

[0081] This management information set has, as shown in FIG. 8, the header of the management information set, and a bit map table showing showing the state of each block, and a conversion table for converting the specified address to the physical address when the block is accessed, and a connection table showing the next block to a specific block.

[0082] In the bit map table, the information on Yes/No flag, block flag, last flag, reference flag and management flag, extracted from the distribution management information of each block, is saved.

[0083] The conversion table is, as shown in FIG. 9, a table specifying the physical address corresponding to the logical address, and the domain where the physical address is stored has two bytes assigned per one entry. When distribution management information is created from this conversion table, the logical address where the distribution management information of the corresponding block is searched, and the physical address corresponding to that block is recorded in the corresponding location of the table. Now, when the logical address is not being used, the corresponding physical address is set to "0xffff".

[0084] The link table is, as shown in FIG. 10, the table where the link address corresponding to the logical address is specified, and the domain where the link address is saved has two bytes assigned per entry. When this link table is being created from the distribution management information, the link address written in the distribution management information of the corresponding block is searched and this link address of that block is recorded in the corresponding location of the table.

[0085] 8. Procedure when the memory card starts up

Next, the procedures when memory card 2 starts up is described while referencing the flow chart in FIG. 11.

[0086] When this memory card 2 is started-up, as shown in FIG. 11, firstly, in step S1, data processing apparatus 1 reads in the boot data from the boot block of memory card 2 and then the process proceeds to step S2.

[0087] In step S2, data processing apparatus 1 checks that the boot data read-in from the boot block was performed normally. If read normally, it proceeds to step S3, and if it was not read normally, it proceeds to step S8.

[0088] In step S3, data processing apparatus 1 determines whether or not memory card 2 corresponds to said data processing apparatus 1, based on the read boot data. If it does correspond, the process proceeds to step S4 and if it does not correspond, the process proceeds to step S8.

[0089] In step S4, data processing apparatus 1 reads the management information set from memory card 2. Now, the physical address of the management information set is indicated in the boot data. Next, the process proceeds to step S5.

[0090] In step S5, data processing apparatus 1 checks that the effective management information set was read normally. If it was read normally, the process proceeds to step S6, and if it was not read normally, the process proceeds to step S7.

[0091] In step S2, data processing apparatus 1 expands the management information set read to internal memory 4, and creates a management table to manage memory card 2. In the above processes, the initial processes on starting-up memory card 2 are terminated, and memory card 2 becomes available for use.

[0092] Moreover, if it was determined in step S5 that effective management data sets were not read normally, the process proceeds to step S7, as mentioned above. In step S7, data processing apparatus 1 reads-out the distribution management information for each block and reconstructs the management information sets. Then, the management information set is expanded to internal memory 4 and a management table to manage memory card 2 is created. In the above processes, the initial processes on starting-up memory card 2 are terminated, and memory card 2 becomes available for use.

[0093] On the other hand, if a determination was made that an error was generated in step S2 in reading the boot data, and it was determined in step S3 that memory card 2 did not correspond to data processing apparatus 1, the process proceeds to step S8, as described above.

[0094] Proceeding to step S8 occurs when memory card 2 can not be used. In this case, in step 8, data processing apparatus 1 runs a predetermined error process and displays a message which states something to the effect that it cannot be used, and terminates the start-up processes of memory card 2.

[0095] 9. Handling management information sets at the time of data update processing
Each time data processing apparatus 1 performs the processes of writing in data to memory card 2, and processes of deleting data from memory card 2 (hereafter these processes are jointly referred to as "data update processing"), the management table held in internal memory 4 is updated to correctly reflect the actual state of memory card 2 (in other words, amending the distribution management information). On the other hand, the management information sets saved as a file in memory card 2, is not updated each time there is data update processing; changed content is updated all at once at an appropriate time.

[0096] In general, there is a maximum number of times that flash memory 12 can be rewritten, but by arranging to rewrite the management information sets more substantially all at once, the rewrite frequency of the blocks where management information sets is saved can be reduced, enabling a long lifetime for memory card 2.

[0097] However, when the data is being updated, the file where the management information sets are saved is disabled before the update. This is done in order to prevent loss of consistency of the management information sets of the distribution management information. When there is an update, an update is made simultaneously of the distribution management information of the corresponding block being processed, but because the content of the management information sets is not updated simultaneously, the distribution management information and the management information sets are in an un-matched state. Therefore, when this state occurs, the file of management information sets in memory card 2 is disabled.

[0098] Specifically, when data is updated, as shown in FIG. 12, firstly, in step S11, data processing apparatus 1 makes a determination as to whether to enable or disable the management information sets saved in memory card 2. If the management information sets are already disabled, then it

moves on directly to the data update process. On the other hand, if the management information sets are enabled, then it proceeds to step S12.

[0099] In step S2, data processing apparatus 1 disables the management information sets.

Specifically, it changes the block flag of the block where the management information sets are stored to "Don't Delete", or performs deletion process in respect of the said block. Then, after disabling the management information sets, it proceeds to data update processing.

[0100] Now, the management information set file saved in memory card 2 is disabled at the time of data update processing in order that the consistency of management information sets and the distribution information is not lost, but the content of the management table in the internal memory 4 of the data processing apparatus 1 is normally updated to be the latest state. Then, data processing apparatus 1, normally, manages each block based on this management table.

[0101] Specifically, the management information sets disabled at the time of data update processing, are rewritten into memory card 2 at an appropriate time, and are enabled once more. Now, an 'appropriate time' means, for example, when the use of memory card 2 is finished and the power source is about to be dropped, or when memory card 2 has not been accessed for more than a predetermined period, or the data has not been rewritten more than a predetermined number of times, and such like.

[0102] Specifically, for example, if the use of memory card 2 is finished and before the power source is dropped, the termination processes shown in FIG. 13 are performed, to enable the management information sets.

[0103] In this termination processing, firstly, in step S21, data processing apparatus 1 makes a determination as to whether the management information sets in memory card 2 are enabled or disabled. Then, if the management information sets are enabled, the processes are terminated at that point. On the other hand, if the management information sets are disabled, it moves to step S22.

[0104] In step S22, data processing apparatus 1 makes a determination as to whether deletion processes have been effected or not, with respect to the block where the file containing management information sets is saved. If deletion processes were not effected, it moves to step S23, if the deletion process were effected, it moves to step S2.

[0105] In step S23, data processing apparatus 1 runs the deletion processes with respect to the block where the file containing management information sets is saved. Thereafter, it moves to step S24.

[0106] In step S24, data processing apparatus 1 writes-in management information sets to memory card 2. At that time, data processing apparatus 1 creates a new management information set file based on the content of the management table stored in internal memory 4, and writes-in that new management information set file to memory card 2. By this means, the enabled management information set showing the latest state of memory card 2 gets saved in memory card 2.

[0107] When the above termination processes are finished, enabled management information sets are in a stored state in memory card 2.

[0108] 10. Writing new files

Next, the process means to write new files to memory card 2 are described. The process means to write new files to memory card 2 are different depending on whether the file size is already known or not.

[0109] 10 - 1 When the file size is already known

When the file size is already known, every time the data of said files are written to a new block, a determination is made as to whether the data will fit in said block. Should the data not fit in the block, then the logical address of the next sequential block is secured, and as well as writing the data to the data area, and then the logical address of the next sequential block is written to the distribution management information as the linked address. At that time, the final flag is set as a "connected block". On the other hand, should the data fit in the block, then the data terminal

number part, in other words the empty area of the data domain, is set to "0xffff". At that time, the last flag is set as the "final block", and is set as an effective data size to write additional management information.

[0110] Following is a detailed description of the procedures used to write a file whose size is known, to memory card 2 using the flow chart in FIG. 14 as reference. The flow chart shown in FIG 14, as well as the flow charts shown in FIG 15 and 17 which are presented later, briefly summarize the checks on error deletion prevent switch 20 of memory card 2, and the processes when some kind of error is generated.

[0111] When writing a file whose size is already known to memory card 2, firstly, in step S31, data processing apparatus 1 prepares the real data to be written to memory card 2, and the header of said data. In other words, in step S31, data processing apparatus 1 prepares the file for writing to the data area of memory card 2. Now, the file size information is included in the header of said file. Next, it moves on to step S32.

[0112] In step S32, data processing apparatus 1 sets the block flag of the block where the first file is saved to "header in use" and secures an empty logical address. Next, it moves on to step S33.

[0113] In step S33, the data processing apparatus 1 searches for an empty physical address. Next, it moves on to step S34.

[0114] In step S34, data processing apparatus 1 determines whether or not a file will fit in the block being processed. If the file will not fit in the block, and the file continues, it proceeds to step S35. On the other hand, if the file will fit in the block, and the file does not continue, it moves on to step S32.

[0115] In step S35, data processing apparatus 1 secures the logical address of the next block in sequence, and sets that logical address as the link address. Next, it moves on to step S37.

[0116] On the other hand, in step S36, data processing apparatus 1 sets the last flag to "final block" and sets the link address to "0xffff". Next, it moves on to step S37.

[0117] In step S37, data processing apparatus 1 creates distribution management information for the block being processed, based on the information set in the above steps. Next, it moves on to step S38.

[0118] In step S38, data processing apparatus 1 sequentially writes to the block being processed in page units. Here, when the file will not fit in the block being processed, a block worth of data can be written in page units. Moreover, when a file can fit entirety into a block being processed, only data corresponding to the necessary pages is written. In this step S38, what can be written to a block is the data of a new file, and the distribution management information created in step S37. Next, it moves on to step S39.

[0119] In step S39, data processing apparatus 1 determines whether or not all the file data was written to memory card 2. If all the file data was not written and there still is remaining data, then it returns to step S33 and repeats the processes. On the other hand, if writing is finished, it moves on to step S40.

[0120] In step S40, data processing apparatus 1 determines whether or not the written data finished in the middle of the block. If the written data finished in the middle of the block, then it moves on to step S41. On the other hand, if the written data continued to the final block, the process is terminated at that point.

[0121] In step S32, data processing apparatus 1 writes in the effective data size to the additional management information saved in the redundant area of the last page. In other words, data processing apparatus 1 writes in the value shown for the data size, written in the data area of the block where the last part of the file is saved, to the additional management information of said block as the effective data size.

[0122] In the above manner, the processes for writing in a file whose size is already known to memory card 2 are complete.

[0123] 10 – 2 When the file size is not known

When the file size is unknown, the logical address of the next block is already secured, and when the data has ended, the last flag of the last block is set to over write. As for other distribution management information and additional management information, they are set in the same way as when the file size is known.

[0124] Following is a detailed explanation of the procedure for writing a file of a known size to memory card 2. FIG. 15 is provided as a reference flow chart.

[0125] When writing a file of a known size to memory card 2, firstly, in step S51, data processing apparatus 1 creates a temporary header for the file to be written to memory card 2. At this stage, because the file size is unknown, the file size information is not included in this temporary file header. Next, proceed to step S52.

[0126] In step S52, data processing apparatus 1 sets the block flag of the block where the first file is saved to "Header used", and secures an empty logical address. Next, proceed to step S53.

[0127] In step S53, data processing apparatus 1 prepares the data for writing to memory card 2. Next, proceed to step S54.

[0128] In step S54, data processing apparatus 1 determines whether or not there is data remaining for writing to memory card 2. If the data is not finished and there is still data remaining, it proceeds to step S55. On the other hand if the data is finished and there is no remaining data, it proceeds to step S61.

[0129] In step S55, data processing apparatus 1 searches for an empty physical address. Next, proceed to step S56.

[0130] In step S56, data processing apparatus 1 secures the next in sequence logical address and sets this logical address as the link address. Next, proceed to step S57.

[0131] In step S57, data processing apparatus 1 creates distribution management information for the block which is being subject to processing, based on the information set in the above steps. Next, proceed to step S58.

[0132] In step S58, data processing apparatus 1 sequentially writes data in page units to the block which is being processed. Here, when the file will not fit in the block being processed, a block worth of data can be written in page units. Moreover, when a file can fit entirely into a block being processed, only data corresponding to the necessary pages is written. Now, what can be written to the block in this step S58 is the data of newly written files and the distribution management information created in Step 57. Next, proceed to step S59.

[0133] In step S59, data processing apparatus 1 determines whether or not the written data ends in the middle of the block or not. Then, if data is saved to the end of the data area, it returns to step S53 and the process is repeated. On the other hand, if the data ends in the middle of the data area, it proceeds to step S60.

[0134] In step S60, data processing apparatus 1 writes in the effective data size to the additional management information stored in the redundant area of the last page of the block being processed. In other words, data processing apparatus 1 writes in the value shown for the data size, written in the data area of the block where the last part of the file is saved, to the additional management information of said block as the effective data size. Next, proceed to step S61.

[0135] In step S61, data processing apparatus 1 sets the final flag of the block being processed to "final block" by overwriting it. Next, proceed to step S62.

[0136] In step S62, data processing apparatus 1 updates the file header. In other words, at this stage, because the file size is known, a new header containing the file size information is created, and the

above mentioned temporary file header is overwritten with the newly created header containing the file size information in the header.

[0137] By the means described above, the processes to write in a file whose size is not already known to memory card 2 is completed.

[0138] 11. Updating Files

Next, the process procedures for updating files already written to memory card 2 are explained.

[0139] When updating a file, a different block is assigned the same logical address as the file to be rewritten, and the new data is written to said block. At this time, the block where the old data is written, is maintained without being opened until the file update is finished. By this means, even if a breakdown is generated during file update, recovery is possible to the state prior to the update.

[0140] A specific example of the procedure of this type of file update is explained while referring to FIG. 16.

[0141] As shown in FIG. 16 (a), the file header is saved in logical address block "1", the next part of the file is saved in logical address block "2", and the next part of the file is saved in logical address block "3". Moreover, the identification number of logical address block "1" is set as "6", the identification number of logical address block "2" is set as "4", the identification number of logical address block "3" is set as "1".

[0142] Then, in this type of situation, the data of logical address block "2" is rewritten. When this occurs, first, as shown in FIG. 16 (b), the logical address "2" is applied to another empty block, and the new data is written to this block. Here, as the identification number for the block where the new data is written, a value is set which is one increment on the identification number where the old data was written, in other words "5".

[0143] At this stage, there exists two block with the same logical address. Therefore, of these two blocks, the block with the higher identification number is the one where the newest data is saved and the block with the lower identification number is the one where the saved data is the old data.

[0144] Then, after the new data is written, the block where the old data is written is delete as shown in FIG. 16 (c). Now, at this point, the deletion processes are not effected in respect of the block where the old data is written. The block flag of said block is merely set to 'Don't Delete', and later, at a suitable time, the deletion processes can be applied to the block.

[0145] A detailed explanation of the procedures for the processes mentioned above is provided below along with the flowchart shown in FIG. 17.

[0146] When a file is updated, first, data processing apparatus 1 selects the block to be updated as in step S71. Then it proceeds to step S72.

[0147] In step S72, data processing apparatus 1 reads out the identification number of the block to be updated and sets the identification number of the block where the new data is to be written as one increment on that value. Moreover, as the logical address of the block where the new data is to be written, it sets it as the same logical address of the block being subjected to updating. The process then proceeds to step S73.

[0148] In step S73, data processing apparatus 1 prepares data to be written to the new block. The process then proceeds to step S74.

[0149] In step S74, data processing apparatus 1 searches for an empty logical address. The process then proceeds to step S75.

[0150] In step S75, data processing apparatus 1 determines whether or not the data update is finished. If the update is not finished, it proceeds to step S76. If the update is finished then it proceeds to step S72.

[0151] In step S76, data processing apparatus 1 secures the logical address of the next block in sequence, and sets that logical address as the link address. The process then proceeds to step S77.

[0152] In step S72, data processing apparatus 1 uses the data from each of the preceding steps to create distribution management information for the block where the new data is written. The process then proceeds to step S78.

[0153] In step S78, data processing apparatus 1 writes the new data in page units to the physical address sought in step S74. Here, when the file will not fit in the block being processed, a block worth of data can be written in page units. Moreover, when the file fits into the corresponding block, only the data of necessary pages is written as page units. In step S78, what is written to the block is the data of the new file and the distribution management information created in step S77. Then, after step 78, it returns to step 73 and the processes are repeated.

[0154] In step S72, data processing apparatus 1 determines whether or not a link address was set for the block last subjected to an update. If a link address was set, the process moves on to step S80 and if a link address was not set, it moves on to step S81.

[0155] In step S72, data processing apparatus 1 sets the last flag of the block where new data was most recently written to "final block". The process then proceeds to step S82.

[0156] In step S72, data processing apparatus 1 sets the link address value for the block set as the most recently updated block to the link address of the block which most recently had new data written. The process then proceeds to step S82.

[0157] In step S72, data processing apparatus 1 updates the file header. In other words, when the file is updated, a new header is created to include the file size because of the possibility that the file size has changed and the file header is updated. The process then proceeds to step S83.

[0158] In step S83, data processing apparatus 1 deletes the block where the old data was written. At this point, the deletion processes have not been applied on the block where the old data is written; the block flag of the said block is merely set to 'Don't Delete.' Later, at a suitable time, the deletion processes can be applied to the block.

[0159] This concludes the update process.

[0160] Error detection and correction processes.

In the type of system above, when new files are being written to memory card 2, or when files saved in memory card 2 are being updated and the like, there is the possibility that states may occur where the power source is suddenly interrupted, where memory card 2 is forcibly removed from data processing apparatus 1, where multiple blocks exist with the same logical address (i.e. logical address errors), or where an indicated link address does not exist (i.e. a link address error).

[0161] In respect of that, in the system applied in the present invention, when building a management information set, error detection and correction processes can be performed to detect and correct logical address errors and link address errors. The following is a detailed explanation of this error detection and correction processes.

[0162] 12 - 1 Error detection table

In the system applied in the present invention, logical address errors and link address errors are detected when management information sets are being constructed and then error detection tables are used to detect link address errors. Error detection tables are tables that are only used to detect errors in link addresses and when process to detect and correct errors are run, this table is stored temporarily in the internal memory of data processing apparatus 1. The domain for setting up the error detection tables is released after the error detection and correction processes are completed.

[0163] These error detection tables are, as shown in FIG. 18, tables which provide a one bit domain showing the connection state of each block, in respect of one logical address. In other words, error detection tables have one bit per entry, and each entry shows the connection state of logical addresses as either '1' or '0'. When there are N blocks subject to processing, the domain held by this error detection table becomes N/8 bytes.

[0164] In these error detection tables, the meaning of the values for the connection state of each block is different when the management information sets are being constructed, and when the construction of the management information sets is completed.

[0165] When the management information set is being built, and the value showing the connection state is '0', this shows that the state of that entry is normal, or shows that the logical address corresponding to that entry is specified in the link address of another block while the processing of the block currently in progress continues. When in that state, as the subsequent processes progress, there is the possibility that the value will become '1', and it is in a state where a determination cannot be made as to whether there is a link address error or not.

[0166] Moreover, when the management information set is being built, and the value showing the connection state is '1', that entry is, while the logical address corresponding to that said entry is the block subject to processing, indicated the link address of another bloc, but is in a state which does not show its physical address. In this state, there is a possibility that the value will become "0," and it is uncertain as to whether or not there is a link address error according to the progress of future processes.

[0167] On the other hand, when the value indicating the linked state is "1" at the stage when the management information set has been built, although the physical address corresponding to such entry is designated as the link address, the entry indicates that a corresponding physical address does not exist. Therefore, in this state, there is a link address error.

[0168] In addition, when the value showing the linked state is "0" at the stage when the management information set has been built, the entry indicates that the link concerning the logical address corresponding to such entry is in a normal state.

[0169] 12-2 Link Address Error Detection

Next, an error detection table such as the above-mentioned one is used, and we provide an explanation by listing specific examples with respect to the link address error.

[0170] For example, as shown in FIG. 19, with respect to the block with the physical address "10," its logical address is "1" and its link address is "3," and with respect to the block with the physical address "17," its logical address is "3" and its link address is "2." In addition, there is no block with the logical address "2".

[0171] At this time, the process is run to re-build the management information set, and once this process is run, the link address error is detected as shown in FIG. 20.

[0172] First, in the initial state, as shown in FIG. 20(a), the values in the error detection table with respect to the respective physical addresses "1," "2" and "3" are all the initial value "0." In addition, all initial values are "0xffff" in the conversion table for the management information set. At this time, there are no values at all in the links table for the management information set.

[0173] Next, the information in the block with the logical address "1" is read. According to such information, as shown in FIG. 20(b), the value in the conversion table corresponding to logical address "1" is the physical address for the block with the logical address "1," i.e., "10." In addition, the figure in the links table corresponding to the logical address "1" is the value of the link address for the block with the logical address "1," i.e., "3."

[0174] Next, the entry for logical address "3" designated and indicated by the value in the links table corresponding to the logical address "1" is confirmed. At this time, in the entry for logical address "3," since the physical block is not assigned, as shown in FIG. 20(c), the value in the error detection table corresponding to the logical address "3" is "1." In addition, the value in the conversion table corresponding to the logical address "3" is the value of the logical address for the original link i.e., "1."

[0175] Next, the information with respect to the block with the logical address "3," which has the value "1" in the error detection table, is read. At this time, the block with the logical address "3"

exists, and the information on such block can be read in a normal manner. Therefore, as shown in FIG. 20(d), the value in the error detection table corresponding to the logical address "3" is "0." In addition, at this time, the value in the conversion table corresponding to the logical address "3" is the value of the physical address for the block with the logical address "3," i.e., "17." In addition, the value in the links table corresponding to the logical address "3" is the value of the link address for the block with the logical address "3," i.e., "2."

[0176] Next, confirm the entry for the logical address "2" that is indicated by the value in the links table corresponding to the logical address "3." At this time, in the entry for logical address "2," since the physical block is not assigned, as shown in FIG. 20(e), the value in the error detection table corresponding to the logical address "2" is "1." In addition, the value in the conversion table corresponding to the logical address "2" is the value of the logical address for the original link, i.e., "3."

[0177] Next, the information with respect to the block with the logical address "2," which constitutes the error detection table value "1," is attempted to be read. However, since the block with the logical address "2" does not exist, at this stage a link address error is identified.

[0178] 12-3 Error Correction Processing

In the system that applies the present invention, the error correction process is conducted, as follows, in response to a logical address error.

[0179] When a logical address error occurs, the respective blocks with the same logical address are searched. Then, in the event that there is only one (1) complete block, the complete block is used, and the remaining blocks become invalid.

[0180] In addition, in the event that there are multiple complete blocks with the same logical address (excluding boot blocks), the identification numbers are compared and the block with the lowest value is used.

Furthermore, in the event that the identification number of one (1) block is "255" and the identification number of the other block is "0," the block with the identification number "255" is used.

[0181] Furthermore, even if there are multiple blocks with the same logical address, the difference between their identification numbers is usually "1." In the event that these conditions are not fulfilled, the error correction process is not conducted automatically by the system and the manual restoration mode is used.

[0182] In addition, in the event that a link address error occurs in the system that applies the present invention, the appropriate error correction process is conducted according to the application software that uses memory card 2 and the data, etc. stored in memory card 2. Specifically, for example, the error correction processes listed below should be conducted.

[0183] That is to say, the blocks indicated by the end link address are assigned to new blocks. Then, the data in the final block is accurately read until the last readable page, and the data up until such page is copied into a new block. At this time, the end flag of the new block is set as the "final block." This error correction process is particularly ideal for the target data that is significant data, even if it is data in progress, for musical data, etc.

[0184] Alternatively, for example, the entire file containing the block in which the link address error occurred is erased. This type of error correction process is particularly ideal in the event that the target data is the data that is in progress and insignificant, such as program and video data, etc.

[0185] 12-4 Rebuilding management information set and error detection and correction processes
Under the system that applies the present invention, a physical address error or a link address error occurs when some sort of malfunction occurs during the data updating process. Then, under the system that applies the present invention, as stated above, the management information set that is stored in the memory card 2 is invalidated prior to the data updating process. Therefore, when a

physical address error or a link address error occurs, the management information set becomes invalid. Then, when the management information set becomes invalid and the memory card 2 is booted up, the distributed management information for all the blocks is searched again, and the process for reconstructing the management information set must be taken without fail.

[0186] Thereupon, under this system, when the management information set is rebuilt, attention is paid so that all the blocks are searched again, and the error detection and correction process is simultaneously conducted at this time. In other words, since there is no possibility of a physical address error or a link address error when the management information set is valid, the error detection and correction process is not conducted at this time. That is to say, under this method, the error detection and correction process is conducted only at the time when the management information set is rebuilt. This operation makes it unnecessary to do any additional access of memory card 2 for the purpose of the running error detection or correction processes. Consequently, for example, it becomes possible to boot up memory card 2 quickly.

[0187] This error detection and correction process is conducted by taking the following steps.

[0188] (1) The conversion table on the internal memory 4 in the data processing apparatus 1 is completely initialized at "Oxffff." In addition, the domain of the error detection table in the internal memory 4 is secured, and said error detection table is completely initialized at "0."

[0189] (2) The operation moves to the lead of the block.

[0190] (3) The distributed management information is read from the block, and the bitmap table is constructed by using such distributed management information. At this time, in the event that the enabled/disabled flag is in the "use disabled" state, or in the event that the block flag is in a "not in use" or "not erased" state, the operation moves to the next block as soon as the bitmap table is created, and the process is repeated.

[0191] (4) The logical address of the block (hereinafter abbreviated as the "logical address A") and the link address (hereinafter abbreviated as the "logical address B") are searched.

[0192] (5) In the event that the logical address A is "Oxffff," the operation moves to the next block and returns to (3), and the process is repeated.

[0193] (6) The logical address A in the error detection table is searched. In the event that the logical address A column in the error detection table is "1," "0" is written instead, and the physical address of the block with the logical address A is written in the logical address A column in the conversion table. In addition, in the event that the logical address A column in the error detection table is "0," the logical address A column in the conversion table is searched. Then, when the value of the logical address A in the conversion table is "Oxffff," the physical address of the block with the logical address A is written there.

[0194] Furthermore, in the event that a value other than "Oxffff" is already written as the value of logical address A in the conversion table, since a logical address error is occurring, the error correction process is conducted in response to the logical address error.

[0195] (7) The value of the logical address B is entered in the logical address A column in the links table.

[0196] (8) The operation searches whether the end flag is the "final block." If it is the "final block," since the link address is invalid, the operation moves to the next block and returns to (3), and the process is repeated.

[0197] (9) The conversion table is used to confirm whether the physical address corresponds to the logical address B. In the event that the value of the logical address B in the conversion table is a value other than "Oxffff," the physical address corresponds to the logical address B. On the other hand, in the event that the value of the logical address B in the conversion table is "Oxffff," at the stage up until the current block, the logical address B does not correspond to the physical address. At this time, "1" is written in the logical address B column in the conversion table and the value of

the logical address A is written in the logical address B column in the conversion table. Thereafter, the operation moves to the next block and returns to (3), and the process is repeated.

[0198] Furthermore, even if the value of the logical address B in the conversion table is "Oxffff," it is unclear whether the physical address really does not correspond to the logical address B at the stage when the process is only conducted for part of the block. That is to say, in the event that the physical address really does not correspond to the logical address B, two (2) copies are possible in the event that the corresponding block appears according to the progress of the future process.

[0199] (10) After the process is conducted for all the blocks, refer to the error detection table. The physical address does not correspond to the logical address that has the value "1" in the error detection table. That is to say, a link address error has occurred. At this time, since the logical address of the original link block is stored in the conversion table, such logical address is used to specify the original block, and the appropriate error correction process is conducted. Furthermore, after the error correction process is conducted, the value in the error detection table is set to "0," and the applicable conversion table value is set to "Oxffff."

[0200] In the system that applies the present invention, as stated above, when the management information set is built, the error detection and correction process is conducted. We provide a further detailed explanation by referring to the flowcharts shown in FIG. 21 through FIG. 23 below with respect to the specific methods of this construction of the management information set and error detection and correction process.

[0201] Furthermore, variables I, A, B, C, D, T(I) are used here, and N is used as the constant. Variable I is the variable input by the physical address, variable A is the variable input by the logical address, variable B is the variable input by the link address, variables C and D are the variables input by the values of the identification numbers, and variable T(I) is the variable that indicates the value in the error detection table corresponding to the logical address "I." In addition, constant N is the constant that indicates the value of all the blocks.

[0202] At the time when the management information set is built, in Step S101 shown in FIG. 21, the data processing apparatus 1 initializes the conversion table, and all values are set at "Oxffff." Next, the operation proceeds to Step S102.

[0203] In Step S102, the data processing apparatus 1 initializes the error detection table, and all values are set at "0." Next, the operation proceeds to Step S103.

[0204] In Step S103, the data processing apparatus 1 replaces variable I with "0." Next, the operation proceeds to Step S104.

[0205] In Step S104, the data processing apparatus 1 reads the distributed management information of the physical address "I" block. Next, the operation proceeds to Step S105.

[0206] In Step S105, the data processing apparatus 1 determines whether the physical address "I" block can be used after referring to the distributed management information flag read in Step S104. Proceed to Step S106 if the block can be used and proceed to Step S129 if the block cannot be used.

[0207] In Step S106, the data processing apparatus 1 determines whether the physical address "I" block is being used after referring to the distributed management information block flag read in Step S104. Specifically, the block flag is determined as either "Use first" or "Use". If the block flag is "Use first" or "Use" and the aforementioned block is being used, proceed to Step S107. Proceed to Step S129 when the aforementioned block is being used.

[0208] In Step S107, the data processing apparatus 1 adds information regarding physical address "I" block to the bit map table. Proceed to Step S108 next.

[0209] In Step S108, the data processing apparatus 1 substitutes the logical address of the physical address "I" block with Variable A and the link address of the physical address "I" block with Variable B, based on the distributed management information read in Step S104. Proceed to Step S109 next.

[0210] In Step S109, the data processing apparatus 1 determines whether the A value is "0xffff". proceed to Step S110 when the value is not "0xffff" and advance to Step S120 in FIG. 22 when the value is "0xffff".

[0211] In Step S110, the data processing apparatus 1 searches for the value of the logical address "A" in the error detection table. Proceed to Step S111 next.

[0212] In Step S111, the data processing apparatus 1 determines whether the value of the logical address "A" in the error detection table is "1". Proceed to Step S112 if the value is "1" and proceed to Step S130 if the value is not "1".

[0213] In Step S112, the data processing apparatus 1 rewrites the value of the logical address "A" in the error detection table as "0". Proceed to Step S113 next.

[0214] In Step S113, the data processing apparatus 1 writes Variable "I" (physical block "I") into the logical address "A" field in the conversion table. Proceed to Step S114 next.

[0215] In Step S114, the data processing apparatus 1 writes Variable "B" (link address "B") into the logical address "A" field in the link table. Proceed to Step S115 next.

[0216] In Step S115, the data processing apparatus 1 determines whether the last flag is the "final block". Proceed to Step S116 if the flag is not the "final block" and proceed to Step S120 in FIG. 22 if the flag is the "final block".

[0217] In Step S116, the data processing apparatus 1 searches for the value within the conversion table corresponding to the logical address "B". Proceed to Step S117 next.

[0218] In Step S117, the data processing apparatus 1 determines whether the value within the conversion table corresponding to the logical address "B" is "0xffff". Proceed to Step S118 if the value is "0xffff" and proceed to step 120 if the value is not "0xffff", as in FIG. 22.

[0219] In Step S118, the data processing apparatus 1 rewrites the logical address "B" field in the error detection table to "1". Proceed to Step S119 next.

[0220] In Step S119, the data processing apparatus 1 writes the logical address "A" into the logical address "B" field in the error detection table. Proceed to Step S120 in FIG. 22 next.

[0221] In Step S120 in FIG. 22, the data processing apparatus 1 compares the Variable I value and the constant N value, indicating the number of total blocks. Proceed to Step S121 if I is not less than N and proceed to Step 128 if $I < N$.

[0222] In Step S121, the data processing apparatus 1 substitutes the Variable I to "0". Proceed to Step S122 next.

[0223] In Step S122, the data processing apparatus 1 determines whether the Variable T (I), indicating the value of the logical address "I" in the error detection table, is "1". Proceed to Step S123 if the value is not "1" and advance to Step S125 if the value is "1".

[0224] In Step S123, the data processing apparatus compares the Variable I value and the constant N value, indicating the number of total blocks. If $I = N$, this process ends. If I does not equal N, proceed to Step S124.

[0225] In Step S124, the data processing apparatus sets the Variable I value by increments of 1. Repeat the process after returning to Step S122.

[0226] In addition, if the Variable T (I) in Step S122 is "1", advance to Step S125, as described above. proceed to Step S125 when a link address error occurs. In Step S125, the data processing apparatus 1 executes designated error correction processes on link address errors. Ensure that the error correction process is applicable to the application software used by the memory card 2 or data stored to the memory card 2 is executed, as described above. Proceed to Step S126 after the error correction process is complete.

[0227] In Step S126, the data processing apparatus 1 switches the value of the logical address "I" in the error detection table to "0". Proceed to Step S127 next.

[0228] In Step S127, the data processing apparatus 1 rewrites the field of logical address "I" in the conversion table to "0xffff". Proceed to Step S123 and execute the above process.

[0229] If $I < N$ in Step S120, proceed to Step S128 as in the above. Proceed to Step S128 when the distributed management information for all the blocks has not been completely read. In Step S128, the data processing apparatus 1 sets the Variable I value by increments of 1 and repeats the process after returning to Step S104 in FIG. 21.

[0230] In addition, if the physical address "I" block in Step S105 in FIG. 21 cannot be used and the physical address "I" block in Step S106 is not being used, proceed to Step S129, as described above.

[0231] In Step S129, the data processing apparatus 1 adds information regarding the physical address "I" block to the bit map table. Proceed to Step S120 in FIG. 22 and execute the above process.

[0232] In addition, if the value of the logical address "A" in the error detection table is not "1" in Step S111 in FIG. 21, proceed to Step 130, as described above. In Step S130, the data processing apparatus 1 searches for the value of the logical address "A" in the conversion table. Proceed to Step S131 next.

[0233] In Step S131, the data processing apparatus 1 determines whether the value of the logical address "A" in the conversion table is "0xffff". Proceed to Step S113 and execute the above process if the value is "0xffff" and proceed to Step S132 in FIG. 23 if the value is not "0xffff".

[0234] Proceed to Step S132 in FIG. 23 when a logical address error is generated and there are two blocks with logical address "A". In Step S132, the data processing apparatus 1 reads the identification number of the two blocks with logical address "A". If the identification number of one of the blocks can be read, the value is substituted with Variable C. If the identification number of the other block can be read, the value is substituted with Variable D. Proceed to Step S133 next.

[0235] In Step S133, the data processing apparatus 1 determines whether the identification number of Step S132 was properly read. Proceed to Step S134 when the identification number substituted by Variable C is read and proceed to Step S137 in all other cases.

[0236] In Step S134, the data processing apparatus 1 sets the block flag of the block corresponding to variable D to "not erased." Next, the operation proceeds to Step S135.

[0237] In Step S135, the data processing apparatus 1 determines whether the process under Step S134, i.e., the process whereby the block flag of the block corresponding to variable D is set to "not erased," is successful. If such process is successful, the operation returns to Step S113 shown in FIG. 21, and the above-mentioned process is conducted. If such process is not successful, the operation proceeds to Step S136.

[0238] In Step S136, the data processing apparatus 1 sets the enabled/disabled flag for the block corresponding to variable D to "use enabled." Thereafter, the operation returns to Step S113 shown in FIG. 21, and the above-mentioned process is conducted.

[0239] In addition, in Step S137, the data processing apparatus 1 determines whether the reading of the identification numbers in Step S132 was conducted in a normal manner. If only the identification number that is substituted for variable D can be read, the operation proceeds to Step S138, and the operation proceeds to Step S141 at all other times.

[0240] In Step S138, the data processing apparatus sets the block flag for the block corresponding to variable C to "not erased." Next, the operation proceeds to Step S139.

[0241] In Step S139, the data processing apparatus 1 determines whether the process under Step S138, i.e., the process whereby the block flag of the block corresponding to variable C is set to "not erased," is successful. If such process is successful, the operation returns to Step S113 shown in FIG. 21, and the above-mentioned process is conducted. If such process is not successful, the operation proceeds to Step S140.

[0242] In Step S140, the data processing apparatus 1 sets the enabled/disabled flag for the block corresponding to variable C to "use enabled." Thereafter, the operation returns to Step S113 shown in FIG. 21, and the above-mentioned process is conducted.

[0243] In addition, in Step S141, the data processing apparatus 1 determines whether the reading of the identification numbers in Step S132 was conducted in a normal manner. When the identification number that is substituted for variable C and the identification number that is substituted for variable D can both be read, the operation proceeds to Step S142. On the other hand, if neither identification number can be read, the operation moves to the manual restoration mode, and the appropriate process that is necessary for the manual correction of the error is conducted.

[0244] In Step S142, the data processing apparatus 1 compares the value of variable C with the value obtained by adding "1" to the value of variable D. If these values are equal, the operation proceeds to Step S134, and the above-mentioned process is conducted. If these values are not equal, the operation proceeds to Step S143.

[0245] In Step S143, the data processing apparatus 1 compares the value of variable D with the value obtained by adding "1" to the value of variable C. If these values are equal, the operation proceeds to Step S138, and the above-mentioned process is conducted. On the other hand, when these values are not correct, it moves to manual recovery mode, and the necessary appropriate processes to correct the error are performed manually.

[0246] By means of the above processes, when the management information sets are rebuilt, the error detection and correction processes can be performed simultaneously. By this means, the necessity for excessive access to the memory card 2 in order to perform error detection and correction processes is obviated. In other words, in a system applying the present invention, when the data write-in finishes normally and when effective management information sets are rewritten into memory card 2, error detection and correction processes are not effected.

[0247] In this manner, in the system applied in this invention, because unnecessary error detection and correction processes are not performed, the access to memory card 2 can be made highly efficient. In particular, because the above mentioned error detection and correction processes are performed simultaneously with the operation of building management information sets from distributed management information, the access to memory card 2 can be reduced to a minimum.

[0248] Moreover, in the system applied in this invention, every block of the memory card 2 has an identification number assigned, and because this identification number is used in dealing with logical address errors, safe update processing of data is enabled. In other words, if some kind of error is generated at the time of data update processing, even if there are multiple blocks existing with the same logical address, by using the identification number, the recovery of the data state before data update processing began is enabled. In addition, in the system applied in this invention, by using error detection tables, not only is the detection of link address errors enabled, the detection of non-existent blocks indicated by the link address is enabled.

[0249]

[Effects of the Invention] As explained in detail above, by means of the present invention, the identification number of every block in the external memory device is saved and because local address errors can be dealt with using these identification numbers, the safe performance of update processing is enabled. In other words, if some kind of error is generated at the time of data update processing, even if there are multiple blocks existing with the same logical address, based on the identification number, the old and new data saved in the block can be distinguished from each other, and the new data can be disabled, and by enabling the old data, the recovery of the data state before data update processing began is enabled.

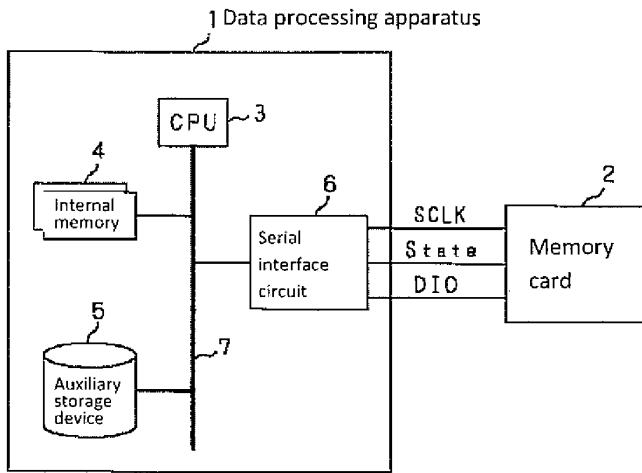
Brief Description of the Drawings

FIG. 1: A drawing showing the overall configuration of a system applying this invention.
FIG. 2: A block drawing showing the configuration of the memory card in a system applying this invention.
FIG. 3: A perspective view showing the outward appearance of the memory card applying this invention.
FIG. 4: A drawing showing the configuration of the memory domains of a memory card applying this invention.
FIG. 5: A drawing showing the configuration of the distributed management information.
FIG. 6: A drawing showing the configuration of the additional management information.
FIG. 7: A drawing showing the appearance of configuration of the management information sets from the distribution management information of every block.
FIG. 8: A drawing showing the configuration of the management information sets.
FIG. 9: A drawing showing the conversion tables.
FIG. 10: A drawing showing the links tables.
FIG. 11: A flow chart showing the procedures at the time of start-up of the memory card.
FIG. 12: A flow chart showing the procedures at the time of data update processing.
FIG. 13: A flow chart showing the procedures of final processing.
FIG. 14: A flow chart showing the procedures when a file of a known size is written into the memory card.
FIG. 15: A flow chart showing the procedures when a file of an unknown size is written into the memory card.
FIG. 16: An outline concept drawing showing a specific example of the file update procedures.
FIG. 17: A flow chart showing the file update procedures.
FIG. 18: A drawing showing the error detection tables.
FIG. 19: A drawing showing a specific example of the linked state between blocks.
FIG. 20: A drawing showing the process flow of the linked state between blocks as shown in FIG. 19 as an example of the file update procedures.
FIG. 21: A flow chart showing the procedures of the error detection and correction processes and building management information sets.
FIG. 22: A flow chart showing the procedures of the error detection and correction processes and building management information sets.
FIG. 23: A flow chart showing the procedures of the error detection and correction processes and building management information sets.

Reference Numerals

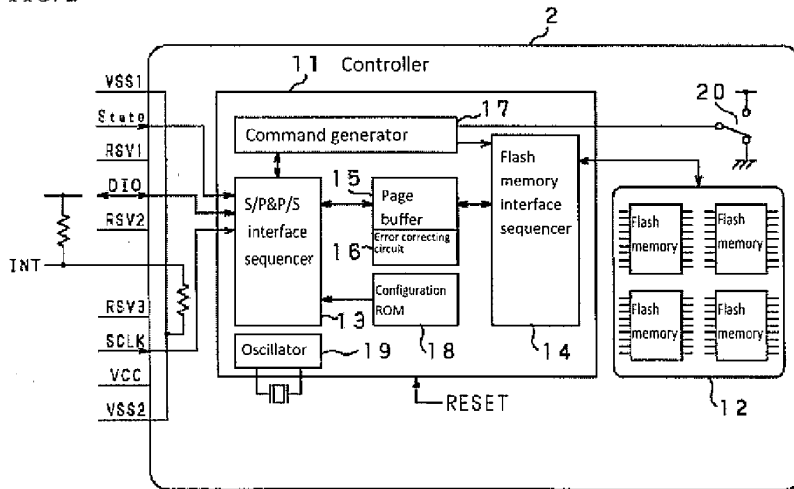
1 Data processing apparatus, 2 Memory card, 3 Computation processing device, 4 Internal memory, 5 Auxiliary memory device
6 Serial interface circuit, 7 Bus, 11 Controller, 12 Flash memory, 13 Serial/parallel, parallel/serial interface sequencer, 14 Flash memory interface sequencer, 15 Page buffer, 16 Error correction circuit, 17 Command generator, 18 Configuration ROM, 19 Oscillator apparatus, 20 Error deletion prevention switch

FIG. 1



Overall structure of system

FIG. 2



Structure of memory card

FIG. 3

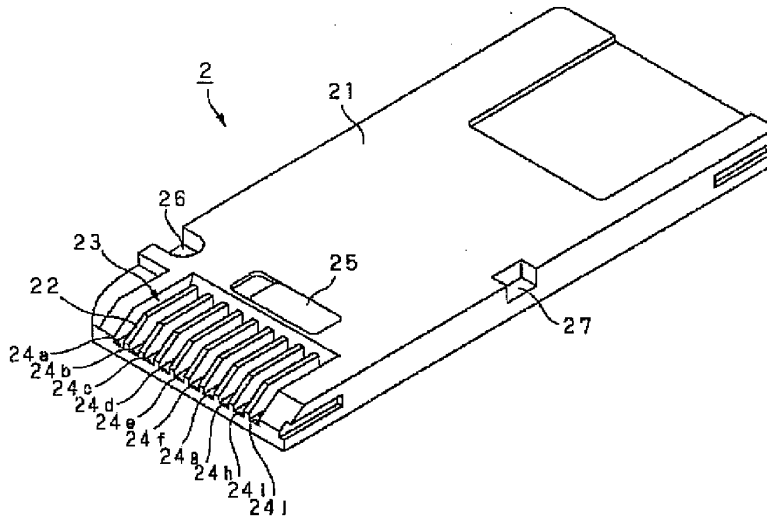
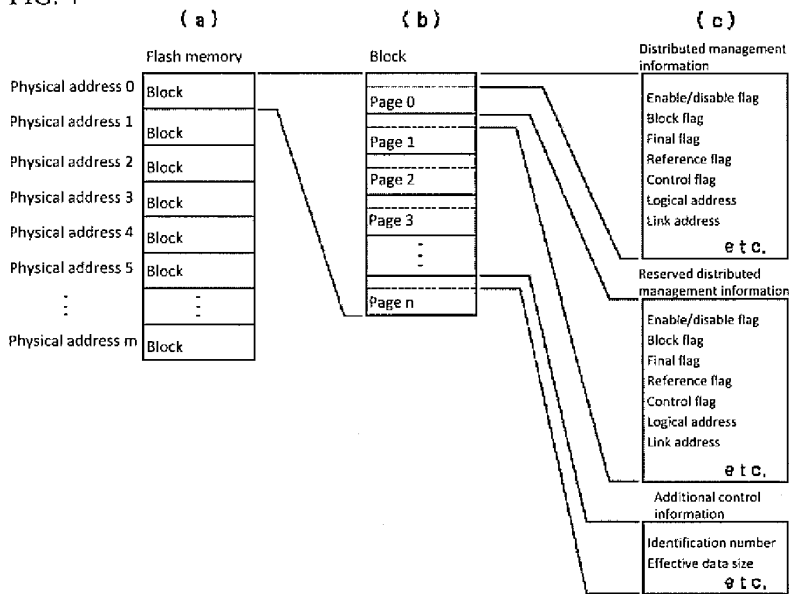
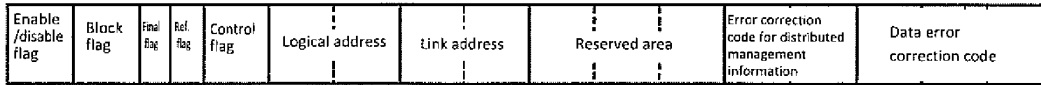


FIG. 4



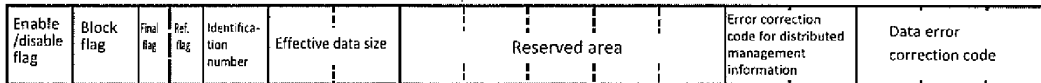
Structure of storage area

FIG. 5



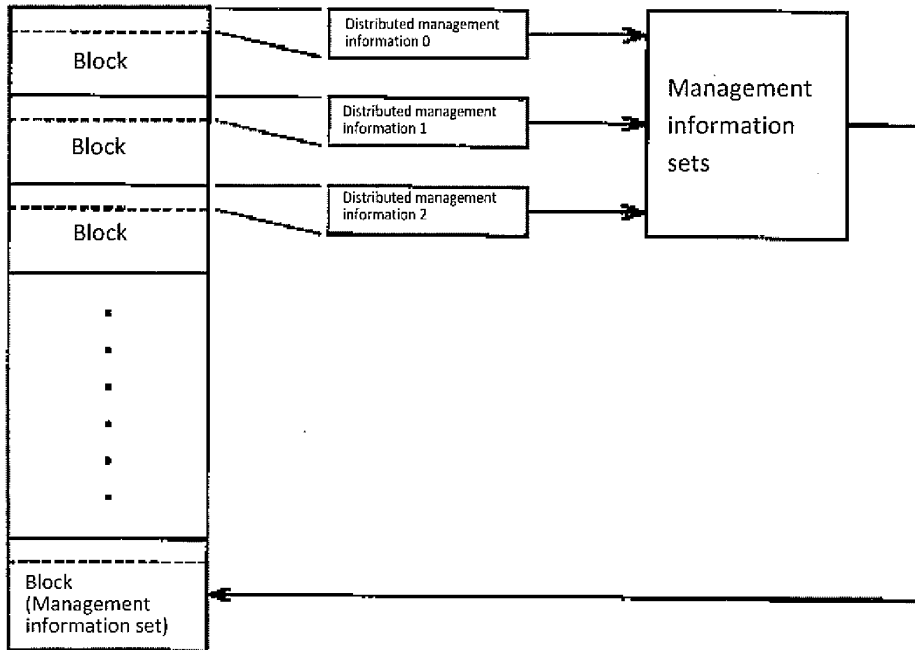
Distributed management information

FIG. 6



Additional control information

FIG. 7



Building management information from distributed management information

FIG. 8

Header
Bitmap table
Conversion table
Link table

Management information sets

FIG. 9

	1 entry 2 byte
Logical address 0	Physical address
Logical address 1	Physical address
Logical address 2	Physical address
⋮	⋮

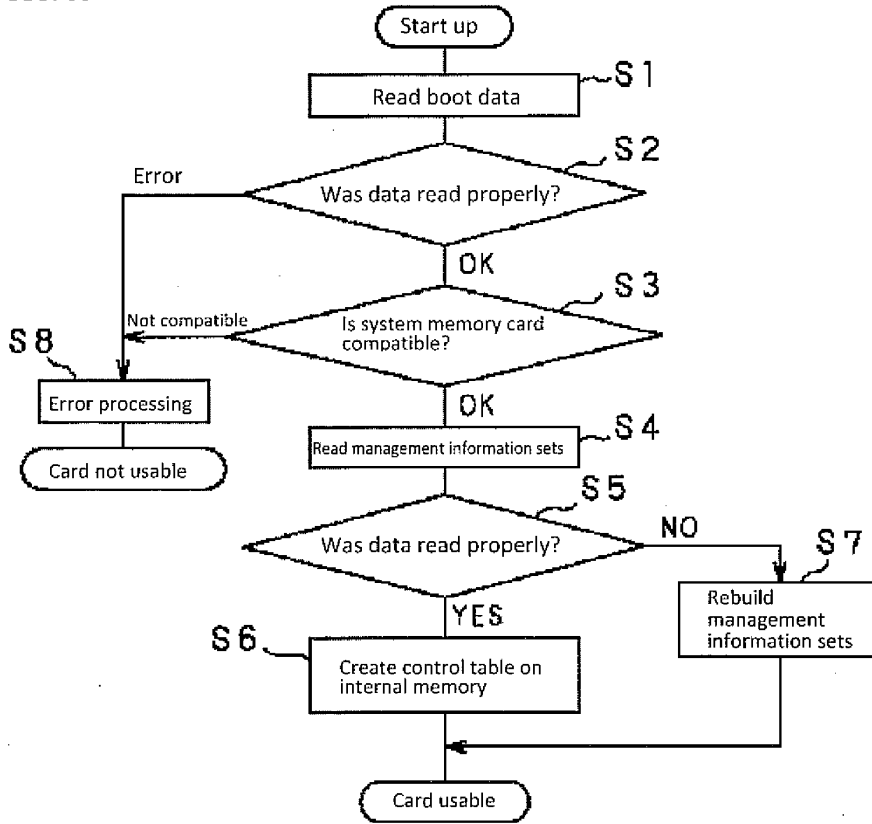
Conversion table

FIG. 10

	1 entry 2 byte
Logical address 0	Link address
Logical address 1	Link address
Logical address 2	Link address
⋮	⋮

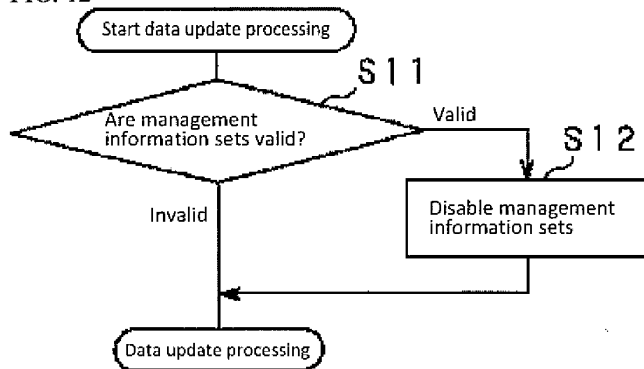
Link table

FIG. 11



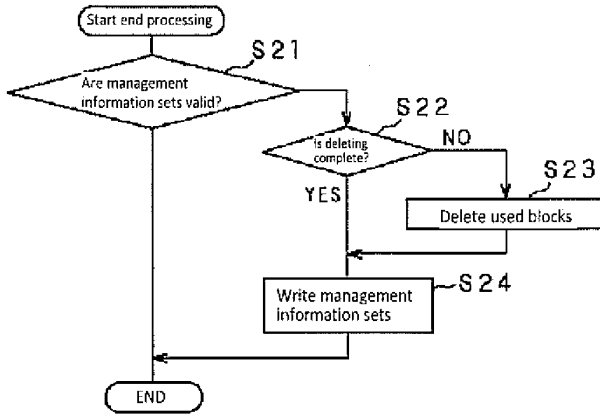
Procedures for memory card startup

FIG. 12



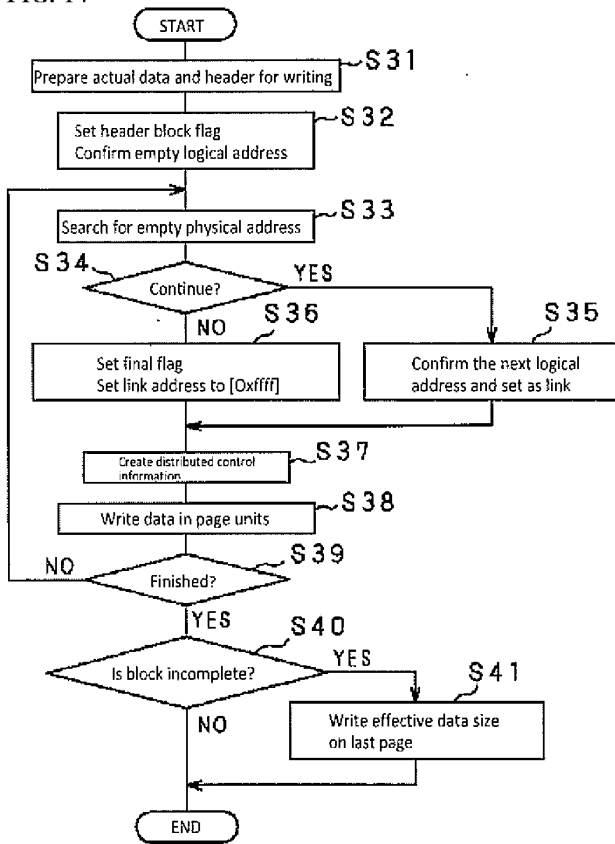
Procedures for starting data update processing

FIG. 13



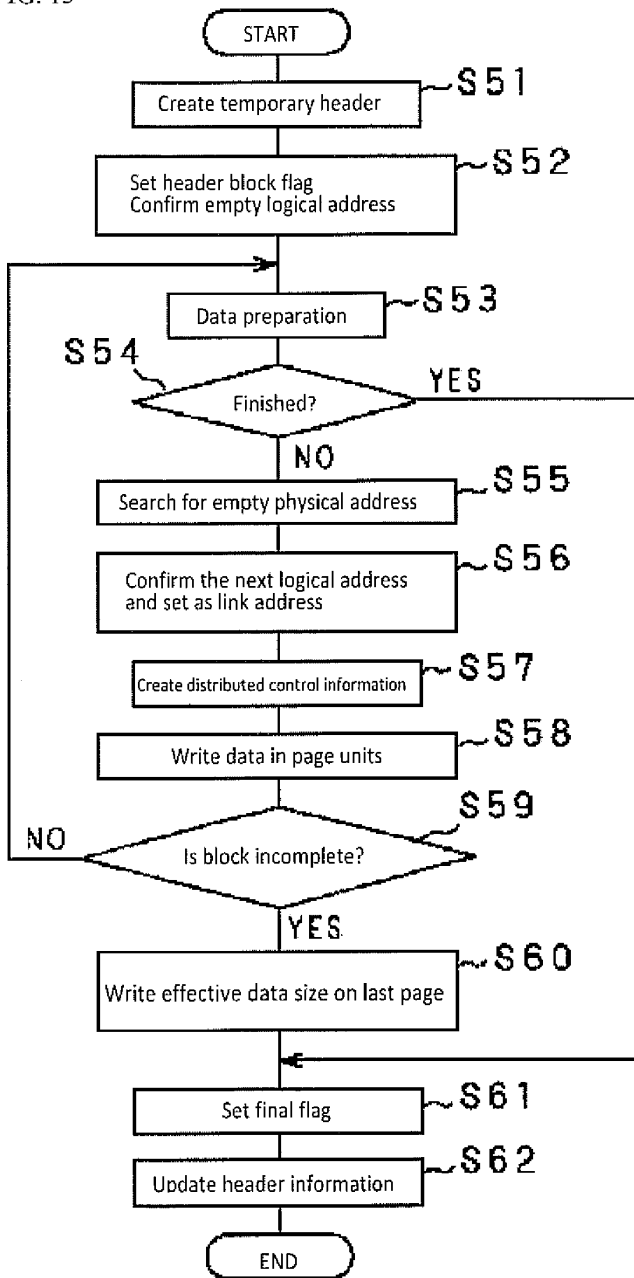
Procedures to end processing

FIG. 14



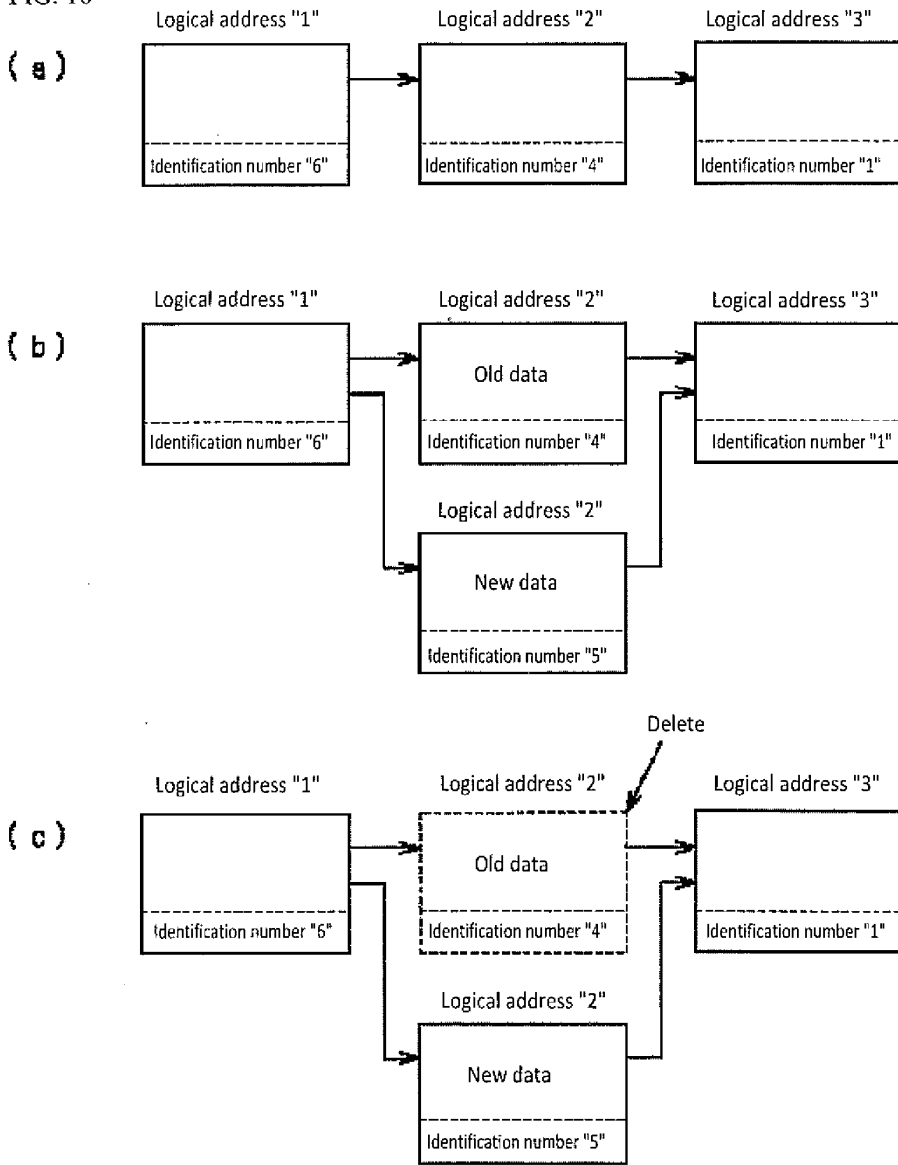
Procedures for writing a file of known size

FIG. 15



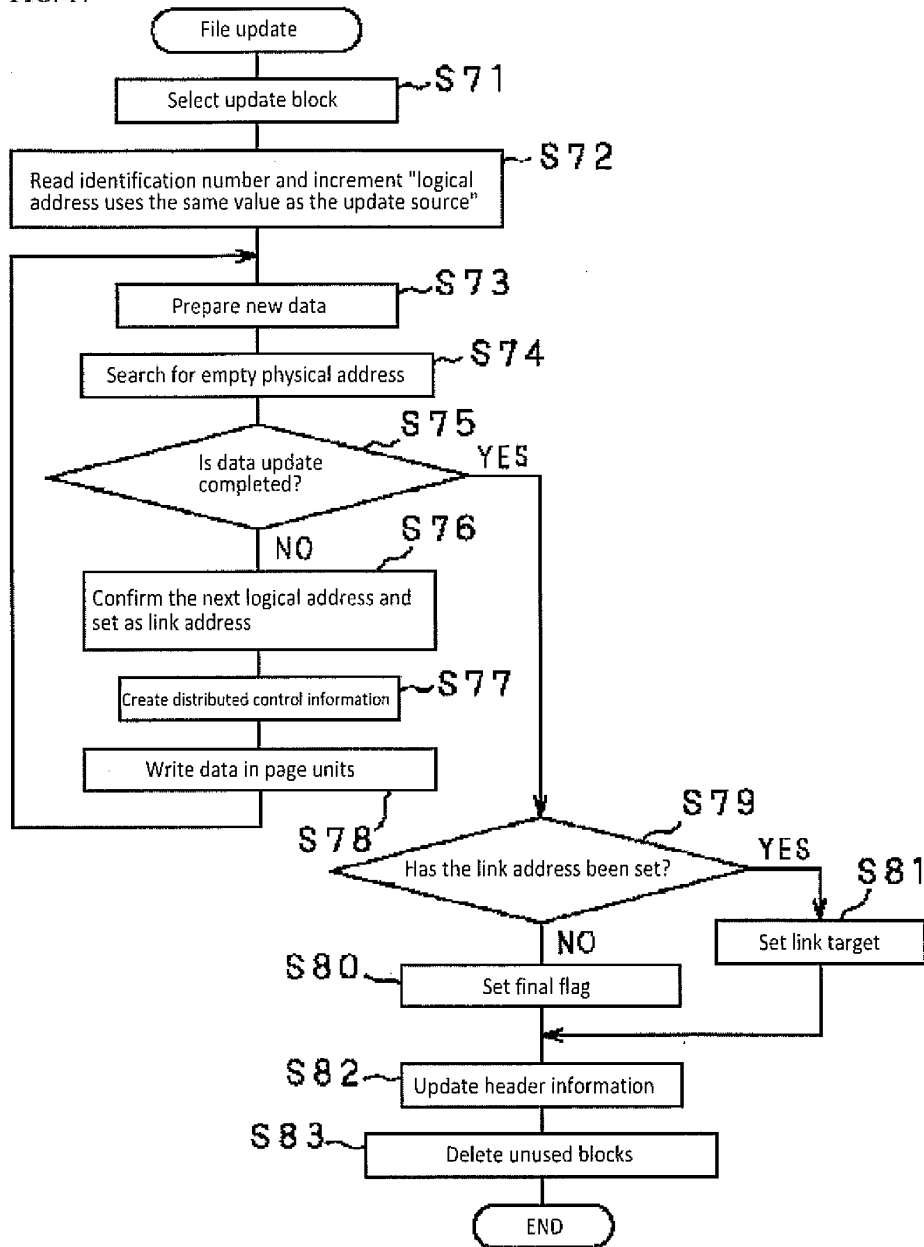
Procedures for writing file of unknown size

FIG. 16



Specific example of file update

FIG. 17



Procedures for updating file

FIG. 18

1 entry 1 bit

Logical address 0	Link status
Logical address 1	Link status
Logical address 2	Link status
Logical address 3	Link status
⋮	⋮

Error detection table

FIG. 19

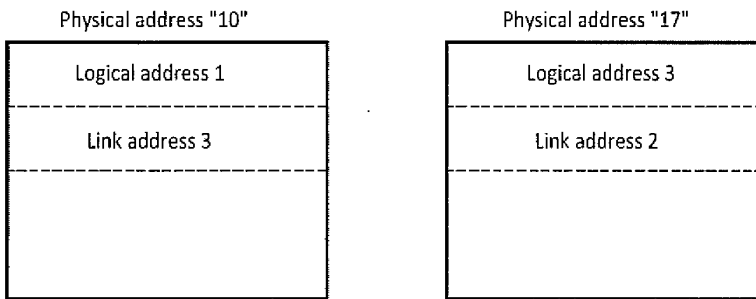
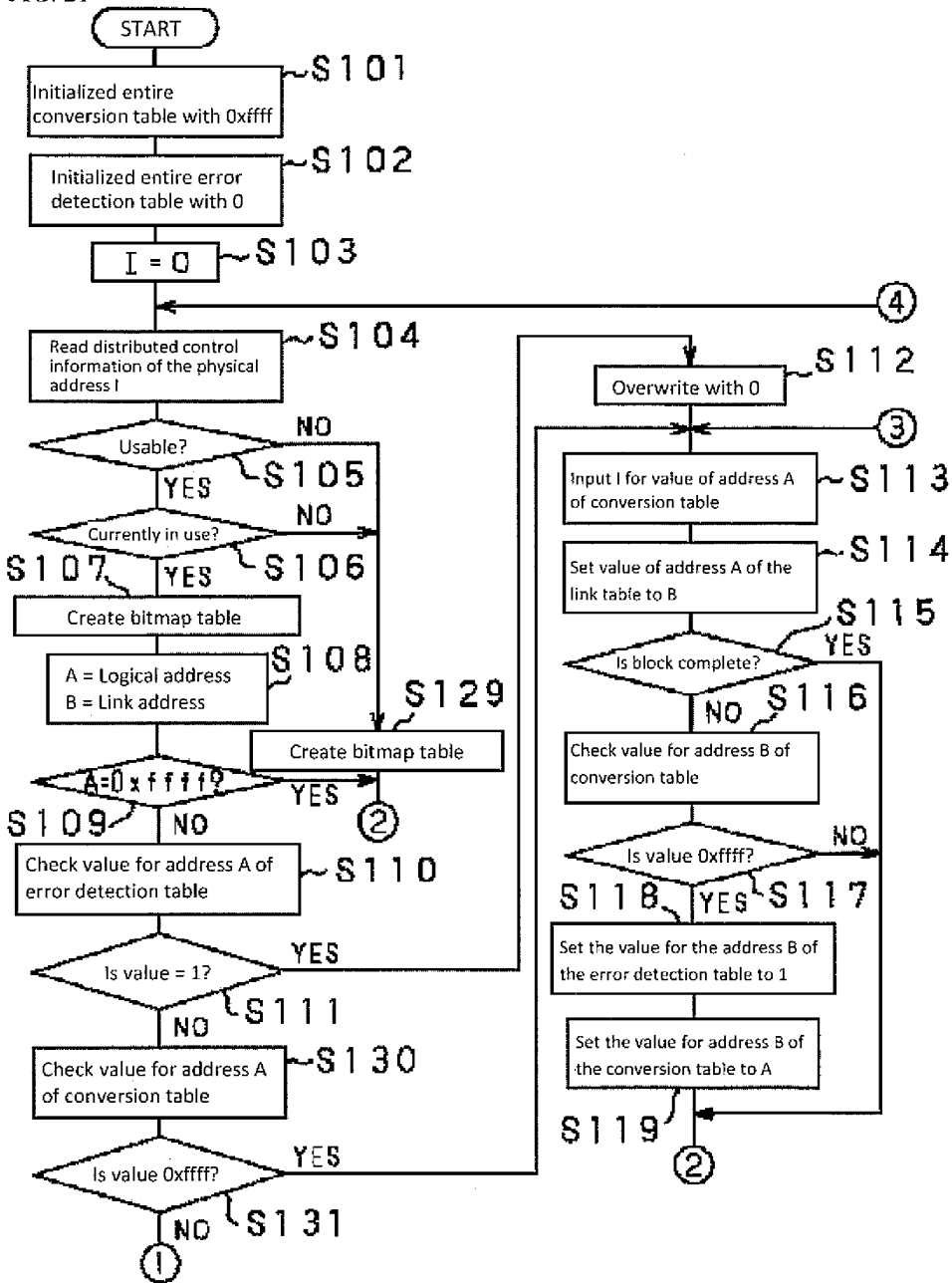


FIG. 20

	Error detection table	Conversion table	Link table
(a)	Logical address 1	0	
	Logical address 2	0	
	Logical address 3	0	
(b)	Logical address 1	10	3
	Logical address 2	0xffff	
	Logical address 3	0xffff	
(c)	Logical address 1	10	3
	Logical address 2	0xffff	
	Logical address 3	1	
(d)	Logical address 1	10	3
	Logical address 2	0xffff	
	Logical address 3	17	2
(e)	Logical address 1	10	3
	Logical address 2	3	
	Logical address 3	17	2

Example of detecting link address

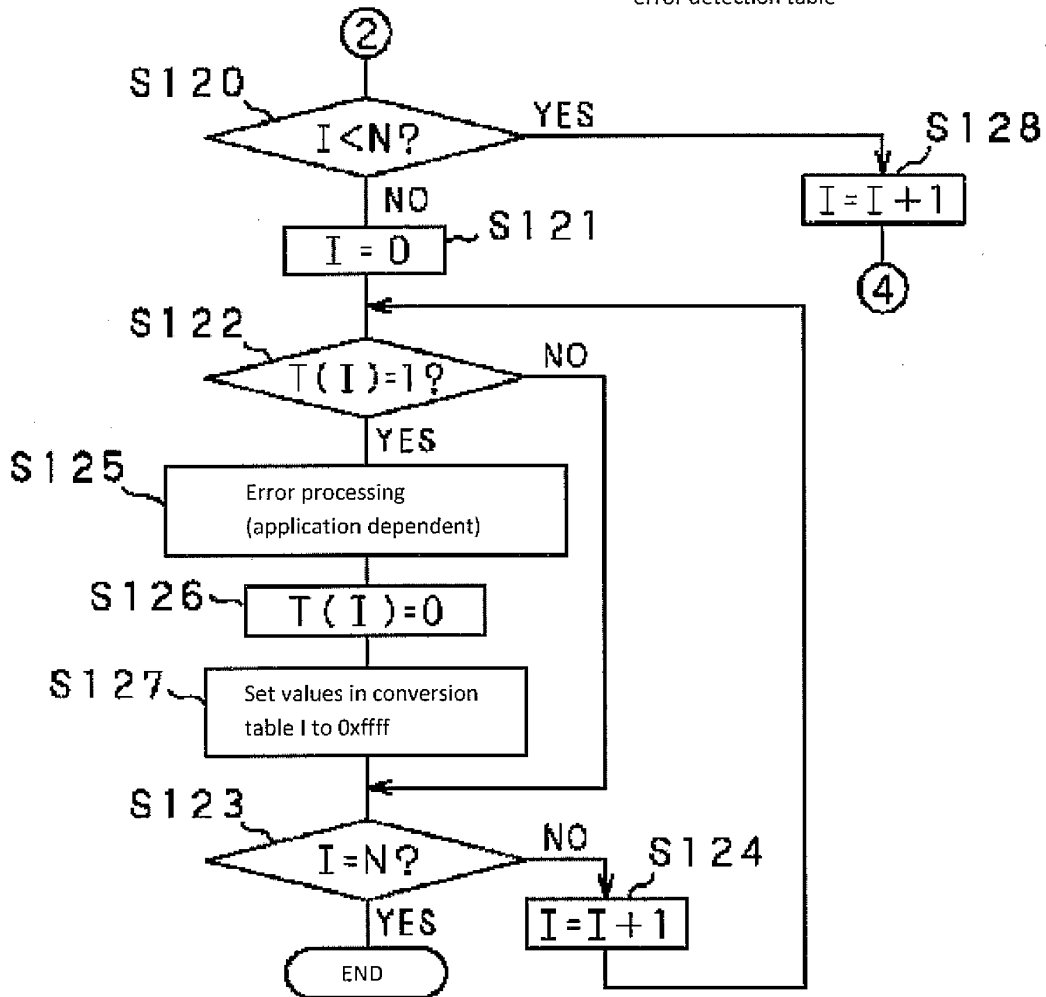
FIG. 21



Procedures for building management information sets and processing error detection and correction

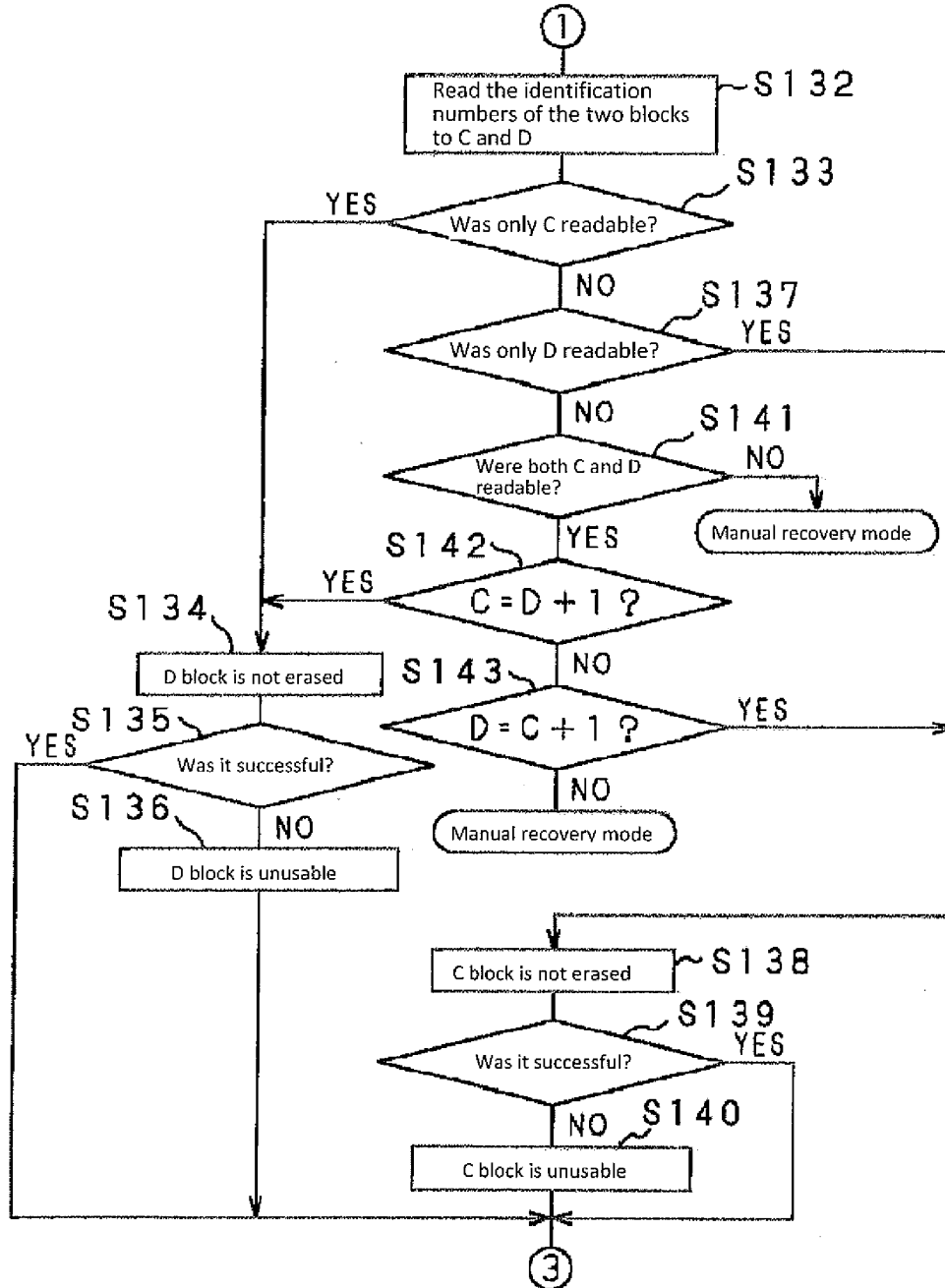
FIG. 22

T(I): Value of logical address of error detection table



Procedures for building management information sets and processing error detection and correction

FIG. 23



Procedures for building management information sets and processing error detection and correction

Continuation of the Front page

(51) Int. Cl.7

Identification Number

FI

G06F

12/00

531

G06F

12/00

531Z

542

542J

(56) Reference Documents

Japanese Unexamined Patent Application Publication No. H08-272698 (JP, A)

Japanese Unexamined Patent Application Publication No. H03-246645 (JP, A)

Japanese Unexamined Patent Application Publication No. H07-248978 (JP, A)

Japanese Unexamined Patent Application Publication No. H04-3218 (JP, A)

Japanese Unexamined Patent Application Publication No. S63-75951 (JP, A)

Japanese Unexamined Patent Application Publication No. H08-221333 (JP, A)

Japanese republished PCT Unexamined Patent Application Publication No. H09-506460 (JP, A)

(58) Fields Searched (Int. Cl.7, Database name)

G06F 12/16

G06F 3/06 - 3/08

G06F 12/00

Electronic Patent Application Fee Transmittal

Application Number:	11250238			
Filing Date:	13-Oct-2005			
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY			
First Named Inventor/Applicant Name:	Kevin M. Conley			
Filer:	Gerald Paul Parsons/Svetlana Stellmach (GPP)			
Attorney Docket Number:	SNDK.156US2			
Filed as Large Entity				
Utility under 35 USC 111(a) Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Miscellaneous:				
Submission- Information Disclosure Stmt	1806	1	180	180
Total in USD (\$)				180

Electronic Acknowledgement Receipt

EFS ID:	4121313
Application Number:	11250238
International Application Number:	
Confirmation Number:	7727
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY
First Named Inventor/Applicant Name:	Kevin M. Conley
Customer Number:	66785
Filer:	Gerald Paul Parsons/Svetlana Stellmach (GPP)
Filer Authorized By:	Gerald Paul Parsons
Attorney Docket Number:	SNDK.156US2
Receipt Date:	15-OCT-2008
Filing Date:	13-OCT-2005
Time Stamp:	17:49:07
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$180
RAM confirmation Number	3160
Deposit Account	040258
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
-----------------	----------------------	-----------	----------------------------------	------------------	------------------

1	Information Disclosure Statement (IDS) Filed (SB/08)	SNDK156US2_Suppl_IDS_10-1 5-08.pdf	101646	no	4
			a44d2a3e883207e7c65c53e1a7b33c438bf 5c80a		
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
2	Foreign Reference	Reference_18_EP1352394B1. pdf	1151699	no	20
			e319e39bd766d7cb13d5e7bc9b5b840109 2775ad		
Warnings:					
Information:					
3	Foreign Reference	Reference_19_JP8-212019A.pdf	1324143	no	16
			f69efc88edcb0a4683f63294d4eb46768567 78a8		
Warnings:					
Information:					
4	Foreign Reference	Reference_21_JP11110300A. pdf	1488434	no	26
			d1e017bc2ea5df70bd52cda9c0e2269a4b 67551		
Warnings:					
Information:					
5	Foreign Reference	Reference_23_JP2000163302A. pdf	1417391	no	24
			af63618f24144beb8b1e4ad042e1397534f3 3a30		
Warnings:					
Information:					
6	NPL Documents	Reference_24_SNDK156JP0_O A_5-13-08.pdf	163685	no	3
			e86bccca3935b8ba00b52b0f54e9c833b794 6a114		
Warnings:					
Information:					
7	NPL Documents	Reference_25_SNDK156KR0_O A_8-25-08.pdf	1556422	no	14
			46266ccc6f8f6e515622ce86bdfbe817b2874 1e70		
Warnings:					
Information:					
8	NPL Documents	Reference_32_Order-No-33.pdf	2537580	no	27
			ec39988cef0a5f6c3c19bc6a8b25a425439d b42d		
Warnings:					
Information:					
9	Foreign Reference	Reference_20_JP8-212019_EN- translation.pdf	813544	no	23
			da59763d4f176c525e94bda929d8ef9532 e58b8		

Warnings:					
Information:					
10	Foreign Reference	Reference_22_JP3070539_EN-translation.pdf	1588861 a9e839d97f4d7f3952c74941f673d13ebc69a78b	no	40
Warnings:					
Information:					
11	NPL Documents	Reference_26_PCMCIA_Vol_1_March-1997.pdf	1431649 1f1d3cb22e46cb36065ac3af0bb129cd8ff1bd5	no	29
Warnings:					
Information:					
12	NPL Documents	Reference_27_PCMCIA_Vol_3_March-1997.pdf	1741762 97ba0031a4c05b57a11dded65a2689f4d562585	no	61
Warnings:					
Information:					
13	NPL Documents	Reference_28_PCMCIA_Vol-7_March-1997.pdf	1883998 d0160fdd8d8331c1ac847c15690fec225392ab9	no	39
Warnings:					
Information:					
14	NPL Documents	Reference_29_Barre_Flash-Memory_Magnetic-Disk-Replace.pdf	170098 4216403ef5f544325d8bbae3a1f961b4320513b9	no	4
Warnings:					
Information:					
15	NPL Documents	Reference_30_Chan_Worlds-Smallest-Solid-State-Storage.pdf	158851 9c6a8889ac1c5c047294f80f2740c09687a73c09	no	4
Warnings:					
Information:					
16	NPL Documents	Reference_31_Wells_Flash-Solid-State-Drive.pdf	197930 bff23645ada4a42417408a06d73ad9120913fab	no	3
Warnings:					
Information:					
17	Fee Worksheet (PTO-06)	fee-info.pdf	30486 f27fabe925914549f4b3fac74ab76bf2e52d92b4	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			17758179		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Kevin M. Conley		
Title:	Block Data Programming and Reading Operations in a Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	October 13, 2005
Examiner:	Ngoc V. Dinh	Group Art Unit:	2189
Docket No.:	SNDK.156US2	Conf. No.:	7727

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

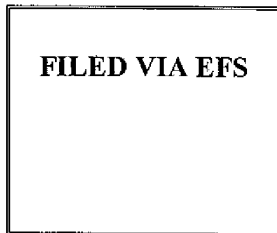
SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Dear Sir:

Pursuant to 37 C.F.R. §§ 1.56, 1.97 and 1.98, the documents listed on the enclosed Form PTO-1449 are being called to the Examiner's attention.

Copies of the listed Other Art are enclosed.

This information disclosure statement is submitted under 37 C.F.R. § 1.97(c). The fee of \$180.00 has been authorized via EFS to Deposit Account 04-0258. The Commissioner is hereby authorized to charge any additional fees, which may be required, or credit any overpayment to Deposit Account 04-0258.



Respectfully submitted,

Gerald P. Parsons November 13, 2008
Gerald P. Parsons Date
Reg. No. 24,486

Davis Wright Tremaine LLP
505 Montgomery Street, Suite 800
San Francisco, CA 94111-6533
(415) 276-6500 (main)
(415) 276-6534 (direct)
(415) 276-6599 (fax)
Email: geraldparsons@dwt.com

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.		Application No.				
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		SNDK.156US2		11/250,238				
		Applicants		Conf. No.				
(Use several sheets if necessary)		Kevin M. Conley		7727				
(Form PTO-1449)		Filing Date		Art Group				
		October 13, 2005		2189				
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
						Translation		
		Document	Date	Country	Class	Subclass	Yes	No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
	1	V. Niles Kynett, "Expert Report on the Invalidity of the '424 Patent," dated August 8, 2008, 99 pages (including Exhibit 4.)						
	2	Exhibits No. 16 through 28 from Kynett's Expert Report, 331 pages.						
	3	Exhibits No. 36 through 41 from Kynett's Expert Report, 69 pages.						
	4	Exhibits No. 46 through 57 from Kynett's Expert Report, 157 pages.						
	5	Exhibits No. 65 through 72 from Kynett's Expert Report, 63 pages.						
Examiner			Date Considered					
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

FILED VIA EFS
Sheet 1 of 1

Electronic Patent Application Fee Transmittal

Application Number:	11250238			
Filing Date:	13-Oct-2005			
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY			
First Named Inventor/Applicant Name:	Kevin M. Conley			
Filer:	Gerald Paul Parsons/Svetlana Stellmach (GPP)			
Attorney Docket Number:	SNDK.156US2			
Filed as Large Entity				
Utility under 35 USC 111(a) Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Miscellaneous:				
Submission- Information Disclosure Stmt	1806	1	180	180
Total in USD (\$)				180

Electronic Acknowledgement Receipt

EFS ID:	4282365
Application Number:	11250238
International Application Number:	
Confirmation Number:	7727
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY
First Named Inventor/Applicant Name:	Kevin M. Conley
Customer Number:	66785
Filer:	Gerald Paul Parsons/Svetlana Stellmach (GPP)
Filer Authorized By:	Gerald Paul Parsons
Attorney Docket Number:	SNDK.156US2
Receipt Date:	13-NOV-2008
Filing Date:	13-OCT-2005
Time Stamp:	12:55:58
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$180
RAM confirmation Number	7780
Deposit Account	040258
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
-----------------	----------------------	-----------	-------------------------------------	------------------	------------------

1	Information Disclosure Statement (IDS) Filed (SB/08)	SNDK156US2_Suppl_IDS_11-1 3-08.pdf	58127 11247d23340d79a4622add95a97c3f10dc3 042ee	no	2
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
2	NPL Documents	Reference-2_Exhibits_16_throu gh_28.pdf	10284106 e06e999837efeeec35175afe9aa7224b18ed6 1609	no	331
Warnings:					
Information:					
3	NPL Documents	Reference-3_Exhibits_36_throu gh_41.pdf	2015924 ac9b2a91c96123784fa91ee8a887903e73d 11fb9	no	69
Warnings:					
Information:					
4	NPL Documents	Reference-4_Exhibits_46_throu gh_57.pdf	4749089 a608c23f15f19192a3ab9c8e59fc5229f14aa 266	no	157
Warnings:					
Information:					
5	NPL Documents	Reference-5_Exhibits_65_throu gh_72.pdf	2108957 407d2e999b358e7f80d91d9d7e3a6e3cad4 212cc	no	63
Warnings:					
Information:					
6	NPL Documents	Reference-1_Kynett_Expert- Report_dtd_8-08.pdf	2883329 91c9877a6fbed2b721cf5241c8e364f89347 4663	no	99
Warnings:					
Information:					
7	Fee Worksheet (PTO-06)	fee-info.pdf	30486 64fde8eaa3784aa28c0eb2776047a3c3954f 2c79	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			22130018		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Kevin M. Conley		
Title:	Partial Block Data Programming And Reading Operations In A Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	October 13, 2005
Examiner:	Dinh, Ngoc V.	Group Art Unit:	2189
Docket No.:	SNDK.156US2	Conf. No.:	7727

Mail Stop Amendment
 Commissioner for Patents
 P.O. Box 1450
 Alexandria, VA 22313-1450

SUPPLEMENTAL AMENDMENT

Sir:

In response to a request made by Examiner Dinh by telephone on November 13, 2008, for corrections to claims 30, 34, 40, 49 and 68, these claims are being amended. In addition, another review of the claims by the undersigned attorney has revealed the desirability for other claim amendments. Therefore, amendments in addition to those requested by Examiner Dinh are being made by this Amendment.

All the claim amendments are reflected in the listing of claims, which begins on page 2 of this paper.

Remarks begin on page 17 of this paper.

Attorney Docket No.: SNDK.156US2
 FILED VIA EFS

Application No.: 11/250,238

CLAIM AMENDMENTS

Please amend the claims by amending claims 30, 34, 40, 46, 49, 52, 54, 55, 57, 59, 63, 65 and 68, and adding new dependent claims 71-74, all without prejudice, as indicated on the following listing of all the claims in the present application after this Amendment:

1 – 3. (Cancelled)

4. (Previously Presented) A method of operating a memory system of reprogrammable non-volatile charge storage elements organized in blocks of a minimum number of storage elements that are erasable together as a unit and in pages of storage elements within the blocks that have specified offset positions within their respective blocks and which are individually programmable as a unit, comprising:

as part of writing data into pages, recording an indication of a time from a clock source that data are written into individual pages,

updating data previously written into one or more pages of one of original data blocks by writing the updated data into one or more update pages of an update data block and logically linking data of the corresponding previously written pages and update data pages, wherein the updated data are caused to be writable into pages of the update data block having different offset positions than the pages of the original data block into which the logically linked data being updated was previously written, and

when reading data of two or more logically linked data pages, read the indications of the times that the data have been stored in the two or more pages and use the data in the two or more pages having more recent time indications without using data in the two or more pages having older time indications.

5 – 6. (Cancelled)

7. (Previously Presented) The method of claim 4, wherein recording the indication of the time that data are written into individual pages includes recording the indication within the pages wherein the data are written.

8. (Previously Presented) The method of claim 4, wherein updating data previously written into one or more pages of one of the original data blocks includes updating data in a number of pages less than all of the pages of the one of the original data blocks while not updating data in a remaining one or more pages of the same original data block.

9. (Previously Presented) The method of claim 8, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

10. (Previously Presented) The method of claim 4, wherein as part of updating data, logically linking the corresponding previously written pages and update data pages comprises writing logical addresses of the updated data in the individual update pages in which the updated data are written.

11. (Previously Presented) The method of claim 4, wherein as part of writing data into pages, data are written in individual storage elements of the pages with more than two storage states, thereby storing more than one bit of data in the individual storage elements.

12. (Previously Presented) The method of claim 4, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

13. (Previously Presented) A method of operating a memory system of reprogrammable non-volatile charge storage elements organized in blocks of a minimum number of storage elements that are erasable together as a unit and in pages of storage elements within

the blocks that have specified offset positions within their respective blocks and which are individually programmable as a unit, comprising:

as part of writing data into pages, data are written into the pages of individual blocks in sequence,

updating data previously written into one or more initial pages of one of original data blocks by writing the updated data into one or more update pages of an update data block and logically linking data of previously written pages with corresponding update data pages, wherein the updated data are caused to be writable into pages of the update data block having different offset positions as the pages of the original data block into which the logically linked data being updated was previously written, and

reading data from the pages of the original data block and update data block in an order that is a reverse of the sequence in which they were written and ignore data in any page that is logically linked with a page from which data have already been read.

14. (Previously Presented) The method of claim 13, wherein updating data previously written into one or more pages of one of the original data blocks includes updating data in a number of pages less than all of the pages of the one of the original data blocks while not updating data in a remaining one or more pages of the same original data block.

15. (Previously Presented) The method of claim 14, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

16. (Previously Presented) The method of claim 13, wherein as part of updating data into pages, logically linking the corresponding previously written pages and update data pages comprises writing logical addresses of updated data in the individual update pages in which the updated data are written.

17. (Previously Presented) The method of claim 13, wherein as part of writing data into pages, data are written in individual storage elements of the pages with more than two storage states, thereby storing more than one bit of data in the individual storage elements.

18. (Previously Presented) The method of claim 13, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

19 – 25. (Cancelled)

26. (Previously Presented) The method of claim 4, wherein logically linking the corresponding previously written pages and update data pages comprises maintaining common logical addresses thereof.

27. (Previously Presented) The method of claim 26, wherein maintaining common logical addresses comprises storing the common logical addresses in the pages of the original data block and in the pages of the update data block along with the respective previously written and updated data therein.

28. (Previously Presented) The method of claim 26, wherein reading the data further comprises organizing the read data by the logical page addresses associated with the read data.

29. (Previously Presented) The method of claim 4, wherein data are written into individual pages of the original and update data blocks in a specified sequence.

30. (Currently Amended) The method of claim 4, wherein the original data previously written into one or more pages of one of original data blocks are written into pages of first and second blocks positioned additionally comprising positioning the one original data block and the update data block in different ones of a plurality of units of the memory system, wherein the units are physically separate groupings of blocks of charge storage elements in

which programming operations may be performed independently, and linking the first and second blocks to cause their charge storage elements to be erasable together and their pages to be programmable together in parallel.

31. (Previously Presented) The method of claim 4, additionally comprising:
updating data previously written into one or more pages of a second of the original data blocks by writing the updated data into one or more update pages of the update data block, and
logically linking data of the corresponding previously written pages of the second of the original data blocks and the update data pages.

32. (Previously Presented) The method of claim 31, wherein updating data previously written into one or more pages of the one and of the second of the original data blocks includes updating data in a number of pages less than all of the pages of the individual original data block while not updating data in a remaining one or more pages of the same individual original data block.

33. (Previously Presented) The method of claim 4, additionally comprising:
subsequently updating for a second time at least some of the data previously written into one or more pages of one of the original data blocks that were previously updated and written into the update data block,
writing the data updated for a second time into the update data block as second updated data pages, and
logically linking data of the corresponding previously written pages and the second updated data pages.

34. (Currently Amended) The method of claim 33, wherein updating data previously written into one or more pages of one of the original data blocks and subsequently updating the data for the second time each includes updating data in a number of pages less than all of the pages of the individual original data block while not updating data in a remaining one or more pages of ~~the same~~ said individual original data block.

35. (Previously Presented) The method of claim 4, wherein the memory system in which the method is practiced is contained within an enclosed card having an electrical connector along one edge thereof for operably connecting with a host system.

36. (Previously Presented) The method of claim 13, wherein logically linking the corresponding previously written pages and update data pages comprises maintaining common logical addresses thereof.

37. (Previously Presented) The method of claim 36, wherein maintaining common logical addresses comprises storing the common logical addresses in the pages of the original data block and in the pages of the update data block along with the respective previously written and updated data therein.

38. (Previously Presented) The method of claim 36, wherein reading the data further comprise organizing the read data by the logical page addresses associated with the read data.

39. (Previously Presented) The method of claim 13, wherein data are written into individual pages of the blocks in a specified sequence.

40. (Currently Amended) The method of claim 13, wherein the data written into one or more initial pages of one of the original data are written into pages of first and second blocks positioned additionally comprising positioning the one original data block and the update data block in different ones of a plurality of units of the memory system, wherein the units are physically separate groupings of blocks of charge storage elements in which programming operations may be performed independently, and linking the first and second blocks to cause their charge storage elements to be erasable together and their pages to be programmable together in parallel.

41. (Previously Presented) The method of claim 13, additionally comprising:

updating data previously written into one or more pages of a second of the original data blocks by writing the updated data into one or more update pages of the update data block, and logically linking data of the corresponding previously written pages of the second of the original data blocks and the update data pages.

42. (Previously Presented) The method of claim 41, wherein updating data previously written into one or more pages of the one and of the second of the original data blocks includes updating data in a number of pages less than all of the pages of the individual original data block while not updating data in a remaining one or more pages of the same individual original data block.

43. (Previously Presented) The method of claim 13, additionally comprising: subsequently updating for a second time at least some of the data previously written into one or more pages of one of the original data blocks that were previously updated and written into the update data block, writing the data updated for a second time into the update data block as second updated data pages, and logically linking data of the corresponding previously written pages and the second updated data pages.

44. (Previously Presented) The method of claim 43, wherein updating data previously written into one or more pages of one of the original data blocks and subsequently updating the data for the second time each includes updating data in a number of pages less than all of the pages of the individual original data block while not updating data in a remaining one or more pages of the same individual original data block.

45. (Previously Presented) The method of claim 13, wherein the memory system in which the method is practiced is contained within an enclosed card having an electrical connector along one edge thereof for operably connecting with a host system.

46. (Currently Amended) In a re-programmable non-volatile semiconductor memory system having a plurality of blocks of a minimum number of memory charge storage elements that are erasable together as a unit, the plurality of blocks individually being divided into a plurality of a given number of pages of memory storage elements that are individually programmable as a unit and which have specified offset positions within their respective blocks, a method of operating the memory system, comprising:

programming original data into individual ones of a first plurality of pages in at least a first block, the pages of original data having logical addresses associated therewith,

thereafter programming, into individual ones of a second plurality of pages in a second block, an updated version of less than the given number of pages of the original data programmed into the first plurality of pages, the pages of the updated version of the original data having logical addresses associated therewith, wherein the logical addresses associated with the pages of the updated version of the original data are the same as the logical addresses associated with the pages of original data,

wherein programming the second plurality of pages additionally comprises ~~causing programming~~ the updated version of the original data ~~to be programmable~~ in those of the second plurality of pages that have different offset positions within the second block than the offset positions of the first plurality of pages within said at least the first block that contain pages of original data with the same associated logical addresses,

thereafter reading data from the first and second plurality of pages, and organizing pages of the read data by their associated logical addresses.

47. (Previously Presented) The method of claim 46, wherein reading data and organizing pages of the read data by their associated logical addresses comprises, for the pages of read data having the same logical addresses associated therewith, utilizing the pages of the updated version of the original data and omitting use of the pages of original data.

48. (Previously Presented) The method of claim 47, wherein reading data and organizing pages of the read data by their associated logical addresses additionally comprises utilizing the pages of original data that have not been updated.

49. (Currently Amended) The method of claim 48, wherein programming original data into individual ones of the first plurality of pages and programming an updated version of the original data into individual ones of the second plurality of pages additionally comprises programming the individual pages with an indication of a relative time of programming the data therein, and

reading data and organizing pages of the read data additionally ~~comprises~~ comprise, for pages of data having the same logical addresses associated therewith, selecting the updated data from the read pages ~~having the more recent~~ having the most recent time indications and omitting use of the ~~original data from others of~~ the pages ~~having the older time indication~~.

50. (Previously Presented) The method of claim 46, wherein programming data into the first plurality of pages in at least the first block and programming data into the second plurality of pages in the second block each comprise programming data into individual pages of the individual blocks in a specified sequence.

51. (Previously Presented) The method of claim 50, wherein reading data and organizing pages of the read data comprise reading the first and second plurality of pages in an order that is reverse to said specified sequence, and ignoring data read from pages having logical addresses that are the same as logical addresses associated with other pages of data that have already been read.

52. (Currently Amended) The method of claim 46, wherein ~~no indication is programmed into individual ones of the first plurality of pages, after the~~ logical addresses associated with the pages of original data previously programmed therein are updated, that indicates that the previously programmed original data were updated individually includes a logical block number and a logical page offset.

53. (Previously Presented) The method of claim 46, wherein programming original data into the first plurality of pages and programming the updated version of the original data

into the second plurality of pages each comprise programming the logical addresses in individual pages along with the data with which the programmed logical addresses are associated.

54. (Currently Amended) The method of claim 46, wherein programming original data into individual ones of the first plurality of pages in at least a first block includes programming the original data into a first plurality of pages in at least the first block and a third one of the additionally comprising positioning the first and second blocks of charge storage elements located in different ones of a plurality of units of the memory system, wherein the plurality of units are physically separate groupings of blocks of charge storage elements in which programming operations may be performed independently, and linking the first and second third blocks together as a metablock to cause their charge storage elements to be erasable together and their pages to be programmable together in parallel.

55. (Currently Amended) The method of claim [[46]] 54, additionally comprising: subsequently programming, into individual ones of the second plurality of pages, a further updated version of at least some of the original data programmed into the first plurality of pages that have previously been updated and the updated version programmed into the second plurality of pages,

wherein the logical addresses associated with the pages of the further updated version of the original data are the same as the logical addresses associated with the pages of original data and the pages of the previously updated version of the original data.

56. (Previously Presented) The method of claim 46, wherein the charge storage elements of the memory system in which the method is carried out comprise electrically conductive floating gates.

57. (Currently Amended) The method of claim [[46]] 54, additionally comprising operating the individual memory system charge storage elements with more than two storage states, thereby storing more than one bit of data in each storage element, wherein programming the pages includes programming the individual memory storage elements into more than two

storage states and reading data includes reading the more than two storage states from the individual memory storage elements.

58. (Previously Presented) The method of claim 46, wherein the memory system in which the method is practiced is contained within an enclosed card having an electrical connector along one edge thereof for operably connecting the memory system with a host system.

59. (Currently Amended) In a re-programmable non-volatile semiconductor memory system having a plurality of blocks of memory charge storage elements that are erasable together as a unit, the plurality of blocks individually being divided into a plurality of a given number of pages of memory storage elements that are programmable together, a method of operating the memory system, comprising:

programming individual ones of a first plurality of said given number of pages in at least a first block with original data, ~~and a~~ the pages of original data having logical addresses associated ~~therewith the page of original data,~~

thereafter programming individual ones of a second plurality of a total number of pages less than said given number in a second block with updated data and a logical address associated with the page of updated data, wherein the logical addresses associated with the pages of updated data programmed into the second plurality of pages are the same as those associated with the pages of original data programmed into the first plurality of pages,

wherein programming the second plurality of pages additionally comprises ~~causing programming~~ the updated version of the original data ~~to be programmable~~ in those of the second plurality of pages that have different offset positions within the second block than the offset positions of the first plurality of pages within said at least the first block that contain pages of original data with the same logical addresses associated therewith,

thereafter reading data from the first and second plurality of pages, and

organizing pages of the read data by the logical addresses associated therewith.

60. (Previously Presented) The method of claim 59, wherein reading data and organizing pages of the read data by the logical addresses associated therewith comprises, for a plurality of pages of read data having the same logical addresses associated therewith, utilizing the pages of the updated version of the original data and omitting use of the pages of original data.

61. (Previously Presented) The method of claim 60, wherein reading data and organizing pages of the read data by the logical addresses associated therewith additionally comprises utilizing the pages of original data that have not been updated.

62. (Previously Presented) The method of claim 59, wherein programming data into the first plurality of pages in at least the first block and programming data into the second plurality of pages in the second block each comprise programming data into individual pages of the individual blocks in a specified sequence.

63. (Currently Amended) The method of claim 59, wherein ~~no indication is programmed into individual ones of the first plurality of pages, after the logical address associated with the page of original data previously programmed therein are updated, that indicates that the previously programmed original data were updated~~ includes a logical block number and a logical page offset.

64. (Previously Presented) The method of claim 59, wherein programming original data into the first plurality of pages and programming the updated version of the original data into the second plurality of pages each comprise additionally programming in the pages the logical addresses for associated with the data programmed therein.

65. (Currently Amended) The method of claim 59, wherein programming original data into individual ones of the first plurality of pages in at least a first block includes programming the original data into a first plurality of pages in at least the first and a third one of the additionally comprising positioning the first and second blocks of charge storage elements

located in different ones of a plurality of units of the memory system, wherein the units are physically separate groupings of blocks of charge storage elements in which programming operations may be performed independently, and linking the first and ~~second~~ third blocks together as a metablock to cause their charge storage elements to be erasable together and their pages thereof to be programmable together in parallel.

66. (Previously Presented) The method of claim 59, additionally comprising: subsequently programming, into individual ones of the second plurality of pages, a further updated version of at least some of the original data programmed into the first plurality of pages that have previously been updated and the updated version programmed into the second plurality of pages,

wherein the logical addresses associated with the further updated version of the original data are the same as the logical addresses associated with the original data and the previously updated version of the original data.

67. (Previously Presented) The method of claim 59, wherein the addresses associated with pages of original and updated data are individually expressed as a logical block number and a logical page offset.

68. (Currently Amended) A method of operating a memory system having an array of reprogrammable non-volatile charge storage elements organized in blocks of storage elements that are erasable together and in pages of storage elements within the blocks that are individually programmable as a unit, comprising:

maintaining an indication of a time separately for individual blocks,
as part of writing data into any one or more of the pages of one of the blocks, updating the indication of a time maintained for the one block to a current time,

when updating data previously written into one or more initial pages, writing the updated data into one or more update pages and identifying the data written into the initial and update pages by common logical addresses, and

as part of reading data having ~~the same~~ common logical addresses from two or more pages of two or more blocks, reading the indications of the times from the two or more blocks and using the data in the two or more blocks having more recent indications of time without using data in the two or more blocks having older indications of time, and reading data from pages within each of the two or more blocks according to a reverse order in physical address to that by which the pages were written, and ignoring data from any page having the same logical address as data of a page from which data have already been read.

69. (Previously Presented) A method of operating a non-volatile memory system having a plurality of blocks of memory storage elements that are individually erasable together, wherein the individual blocks are divided into a plurality of pages of storage elements that are programmable together, the blocks being organized in at least two separate units in which programming may be performed independently, comprising:

linking at least one block from individual ones of said at least two units to form a metablock wherein the storage elements of its component blocks are erased together, and updating pages of original data within any of the metablock component blocks less than all the pages within the block by programming replacement data into pages within another at least one block in only a designated one of the units regardless of which unit the data being updated are stored.

70. (Previously Presented) The method of claim 69, wherein storing the original and replacement data comprises:

identifying the original and replacement data by the same logical address to the memory system, and

distinguishing the replacement data from the original data by determining the relative order in time in which the original and replacement data have been programmed in their respective pages of the memory.

71. (New) The method of claim 46, which additionally comprises maintaining a data structure of the logical addresses associated with the pages of original data and the logical

addresses associated with pages of the updated version of the original data that links (a) physical addresses of a plurality of blocks of memory storage elements that are storing data having common logical addresses with (b) those common logical addresses.

72. (New) The method of claim 59, which additionally comprises maintaining a data structure of the logical addresses associated with the pages of original data and the logical addresses associated with pages of the updated version of the original data that links (a) physical addresses of a plurality of blocks of memory storage elements that are storing data having common logical addresses with (b) those common logical addresses.

73. (New) The method of claim 46, wherein organizing pages of the read data comprises writing the pages of read data into a volatile memory within the memory system.

74. (New) The method of claim 59, wherein organizing pages of the read data comprises writing the pages of read data into a volatile memory within the memory system.

REMARKS

Rather than amending "first and second blocks" of dependent claims 30 and 40 to conform to the antecedent basis, as initially discussed with Examiner Dinh, these claims are being amended to add antecedent for the first and second blocks. In the course of doing this, the focus of claims 30 and 40 has changed. This same type of change is being made to dependent claims 54 and 65. The use of metablocks to store original data is described in at least paragraphs 0062 and 0063 with respect to Figure 15 of the present application.

The amendments to claims 34 and 68 discussed with Examiner Dinh are being made. The amendment to dependent claim 49 is believed to remedy the Examiner's concern.

In addition, amendments are being made to the third from the last paragraph of each of the independent claims 46 and 59 to make that paragraph read better. Also, a clause of the first paragraph after the preamble of claim 59 is being amended to read the same as in claim 46.

The dependencies of claims 55 and 57 are being changed

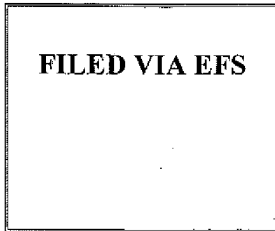
Dependent claims 52 and 63 are being amended to refocus them to define one form of logical address within the scope of their respective independent claims 46 and 59. Use of a logical address having a logical block number and a logical page offset is described in the present application, paragraph 0045 being an example.

New dependent claims 71 and 72 are directed to maintaining a data structure linking physical memory block addresses with the common logical addresses of original and update data stored in different blocks. Examples of this are illustrated by the tables of Figures 9 and 12 of the present application.

New dependent claims 73 and 74 further define the organizing of the read pages of claims 46 and 59, respectively, by use of a volatile memory. An example of this is described in the present application with respect to Figure 13.

Conclusion

The undersigned attorney has tried unsuccessfully to reach Examiner Dinh by telephone over the past week to discuss this Amendment before filing it. It is believed that the Amendment is appropriate to place the claims in better form for allowance. But if the Examiner has any further matters that need to be resolved or any questions about this Amendment, a telephone call to the undersigned attorney at 415-276-6534 (direct line) would be appreciated.



Respectfully submitted,

Gerald P. Parsons November 21, 2008
 Gerald P. Parsons Date
 Reg. No. 24,486

Davis Wright Tremaine LLP
 505 Montgomery Street, Suite 800
 San Francisco, CA 94111-6533
 (415) 276-6500 (main)
 (415) 276-6534 (direct)
 (415) 276-6599 (fax)
 Email: geraldparsons@dwt.com

Electronic Patent Application Fee Transmittal

Application Number:	11250238			
Filing Date:	13-Oct-2005			
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY			
First Named Inventor/Applicant Name:	Kevin M. Conley			
Filer:	Gerald Paul Parsons/Svetlana Stellmach (GPP)			
Attorney Docket Number:	SNDK.156US2			
Filed as Large Entity				
Utility under 35 USC 111(a) Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Claims in excess of 20	1202	4	52	208
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Miscellaneous:				
Total in USD (\$)				208

Electronic Acknowledgement Receipt

EFS ID:	4331783
Application Number:	11250238
International Application Number:	
Confirmation Number:	7727
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY
First Named Inventor/Applicant Name:	Kevin M. Conley
Customer Number:	66785
Filer:	Gerald Paul Parsons/Svetlana Stellmach (GPP)
Filer Authorized By:	Gerald Paul Parsons
Attorney Docket Number:	SNDK.156US2
Receipt Date:	21-NOV-2008
Filing Date:	13-OCT-2005
Time Stamp:	12:26:06
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$208
RAM confirmation Number	9215
Deposit Account	040258
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
-----------------	----------------------	-----------	-------------------------------------	------------------	------------------

1		SNDK156US2_Suppl_Amendm _11-21-08.pdf	595967 97b79d82bcc8de7d5957499fba112cd41b 9404	yes	19
Multipart Description/PDF files in .zip description					
		Document Description	Start	End	
		Miscellaneous Incoming Letter	1	1	
		Supplemental Response or Supplemental Amendment	2	2	
		Claims	3	17	
		Applicant Arguments/Remarks Made in an Amendment	18	19	
Warnings:					
Information:					
2	Fee Worksheet (PTO-06)	fee-info.pdf	30599 fe77b388cf38ccbfe5e429cf3176adc2c3a09 352	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			626566		
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					

LAWYERS

Davis Wright Tremaine LLP



ANCHORAGE BELLEVUE LOS ANGELES NEW YORK PORTLAND SAN FRANCISCO SEATTLE SHANGHAI WASHINGTON, D.C.

SUITE 800
505 MONTGOMERY STREET
SAN FRANCISCO, CA 94111-6333

TEL (415) 276-6500
FAX (415) 276-6599
www.dwt.com

November 21, 2008

Customer No. 66785

Commissioner For Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Re: Applicant: Kevin M. Conley
Title: Partial Block Data Programming And Reading Operations In A Non-Volatile Memory
Application No.: 11/250,238 Filing Date: October 13, 2005
Examiner: Dinh, Ngoc V. Group Art Unit: 2189
Docket No.: SNDK.156US2 Conf. No.: 7727

Dear Sir:

Transmitted herewith are the following documents in the above-identified application:

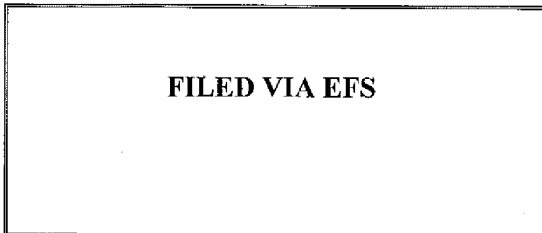
- (1) This Transmittal Letter;
- (2) Supplemental Amendment (18 pages).

The fee has been calculated as shown below:

CLAIMS AS AMENDED

	Claims Remaining <i>After</i> <u>Amendment</u>		Highest No. Previously Paid For		Present Extra	Rate		Additional Fee
Total Claims	62	Minus	58	=	4	x \$52.00	\$	208.00
Independent Claims	6	Minus	6	=	0	x \$220.00	\$	0.00
<input type="checkbox"/>	Fee of _____ for the first filing of one or more multiple dependent claims per application						\$	0.00
<input type="checkbox"/>	Fee for						\$	
Total additional fee for this Amendment:							\$	<u>208.00</u>

- Conditional Petition for Extension of Time: If an extension of time is required for timely filing of the enclosed document(s) after all papers filed with this transmittal have been considered, an extension of time is hereby requested.
- The fee of \$208.00 has been authorized via EFS to Deposit Account 040258. The Commissioner is hereby authorized to charge any additional fees, which may be required, or credit any overpayment to Deposit Account 040258.



Respectfully submitted,

Gerald P. Parsons
Reg. No. 24,486

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875					Application or Docket Number 11/250,238		Filing Date 10/13/2005		<input type="checkbox"/> To be Mailed												
APPLICATION AS FILED – PART I																					
(Column 1)			(Column 2)			SMALL ENTITY <input type="checkbox"/>		OR			OTHER THAN SMALL ENTITY										
FOR		NUMBER FILED		NUMBER EXTRA		RATE (\$)		FEE (\$)		OR		RATE (\$)		FEE (\$)							
<input type="checkbox"/> BASIC FEE (37 CFR 1.16(a), (b), or (c))		N/A		N/A		N/A				OR		N/A									
<input type="checkbox"/> SEARCH FEE (37 CFR 1.16(k), (l), or (m))		N/A		N/A		N/A				OR		N/A									
<input type="checkbox"/> EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))		N/A		N/A		N/A				OR		N/A									
TOTAL CLAIMS (37 CFR 1.16(i))		minus 20 =		*		X \$ =				OR		X \$ =									
INDEPENDENT CLAIMS (37 CFR 1.16(h))		minus 3 =		*		X \$ =				OR		X \$ =									
<input type="checkbox"/> APPLICATION SIZE FEE (37 CFR 1.16(s))		If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$250 (\$125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).																			
<input type="checkbox"/> MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))																					
* If the difference in column 1 is less than zero, enter "0" in column 2.												TOTAL				TOTAL					
APPLICATION AS AMENDED – PART II																					
(Column 1)			(Column 2)			(Column 3)			SMALL ENTITY		OR		OTHER THAN SMALL ENTITY								
AMENDMENT	11/21/2008		CLAIMS REMAINING AFTER AMENDMENT				HIGHEST NUMBER PREVIOUSLY PAID FOR		PRESENT EXTRA		RATE (\$)		ADDITIONAL FEE (\$)		OR		RATE (\$)		ADDITIONAL FEE (\$)		
	Total (37 CFR 1.16(o))		* 62		Minus		** 58		= 4		X \$ =				OR		X \$2=		208		
	Independent (37 CFR 1.16(h))		* 6		Minus		***6		= 0		X \$ =				OR		X \$220=		0		
	<input type="checkbox"/> Application Size Fee (37 CFR 1.16(s))																				
	<input type="checkbox"/> FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))																				
												TOTAL ADD'L FEE		OR		TOTAL ADD'L FEE		208			
AMENDMENT			CLAIMS REMAINING AFTER AMENDMENT				HIGHEST NUMBER PREVIOUSLY PAID FOR		PRESENT EXTRA		RATE (\$)		ADDITIONAL FEE (\$)		OR		RATE (\$)		ADDITIONAL FEE (\$)		
	Total (37 CFR 1.16(o))		*		Minus		**		=		X \$ =				OR		X \$ =				
	Independent (37 CFR 1.16(h))		*		Minus		***		=		X \$ =				OR		X \$ =				
	<input type="checkbox"/> Application Size Fee (37 CFR 1.16(s))																				
	<input type="checkbox"/> FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))																				
												TOTAL ADD'L FEE		OR		TOTAL ADD'L FEE					
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.												Legal Instrument Examiner: /BRENDA HARRISON/									
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".																					
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".																					
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.																					

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Kevin M. Conley		
Title:	Block Data Programming and Reading Operations in a Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	October 13, 2005
Examiner:	Ngoc V. Dinh	Group Art Unit:	2189
Docket No.:	SNDK.156US2	Conf. No.:	7727

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Dear Sir:

Pursuant to 37 C.F.R. §§ 1.56, 1.97 and 1.98, the documents listed on the enclosed Form PTO-1449 are being called to the Examiner's attention.

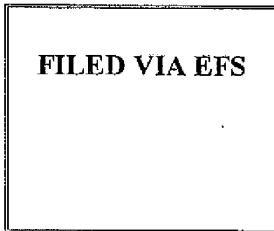
Copies of the listed foreign patent documents and/or Other Art are enclosed .

The newly cited documents are taken from Investigation No. 337-TA-619 of the United States International Trade Commission (ITC), in which claims 17, 18, 24 and 30 of patent no. 6,763,424, grandparent to the present application, are being asserted. A trial in that case was completed in early November, 2008, and a written opinion of the Administrative Law Judge is awaited.

Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

This information disclosure statement is submitted under 37 C.F.R. § 1.97(c). The fee of \$180.00 has been authorized via EFS to Deposit Account 04-0258. The Commissioner is hereby authorized to charge any additional fees, which may be required, or credit any overpayment to Deposit Account 04-0258.



Respectfully submitted,



Gerald P. Parsons
Reg. No. 24,486

December 24, 2008
Date

Davis Wright Tremaine LLP
505 Montgomery Street, Suite 800
San Francisco, CA 94111-6533
(415) 276-6500 (main)
(415) 276-6534 (direct)
(415) 276-6599 (fax)
Email: geraldparsons@dwt.com

U.S. Department of Commerce, Patent and Trademark				Atty. Docket No.			Application No.	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT				SNDK.156US2			11/250,238	
				Applicants			Kevin M. Conley	
(Use several sheets if necessary)				Filing Date			Art Group	
(Form PTO-1449)				October 13, 2005			2189	
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
	1	6,684,289 B1	1/27/2004	Gonzalez et al.				
	2	6,947,332 B2	9/20/2005	Wallace et al.				
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
	3	WO 97/43764 A1	11/20/1997	WIPO				
	4	WO 00/49488 A1	8/24/2000	WIPO				
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
	5	Deposition of Kevin M. Conley dated July 24, 2008, Volume I, 49 pages.						
	6	Deposition of Kevin M. Conley dated July 25, 2008, Volume II, 35 pages.						
	7	Deposition of Kevin M. Conley dated September 11, 2008, Volume III, 35 pages.						
	8	Exhibits No. 1 through No. 12, No. 15 through No. 22, No. 24 through No. 26, No. 28 through No. 32, to Deposition of Kevin M. Conley, total 573 pages.						
	9	Exhibits No. 37 through No. 56, No. 58, No. 59, No. 61 through No. 72, to Deposition of Kevin M. Conley, total 729 pages.						
	10	Deposition of John Mangan dated July 10, 2008, Volume 1, 61 pages.						
	11	Deposition of John Mangan dated July 11, 2008, Volume 2, 37 pages.						

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.	Application No.
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		SNDK.156US2	11/250,238
		Applicants	Conf. No.
(Use several sheets if necessary)		Kevin M. Conley	7727
(Form PTO-1449)		Filing Date	Art Group
		October 13, 2005	2189
12	Exhibits No. 1 through No. 10 to the Deposition of John Mangan, total 540 pages.		
13	Exhibits No. 11 through No. 15, No. 17 through No. 25, No. 29, No. 31, No. 32 to the Deposition of John Mangan, total 504 pages.		
14	SanDisk Corporation, SDP-32 Mbit "Mizer" Low Cost HD, Logical Format 20-10-00050, Rev. 8, December 5, 1997, 32 pages.		
15	Sandisk Israel, MediaClip Memory Card MMC Controller Top Level Design, Version 0.1, November 13, 1996, 74 pages.		
16	SanDisk Corporation, Preliminary MultiMediaCard Product Manual, April 28, 1998, 96 pages.		
17	Photo International, Digital: SanDisk Introduces 64, 80, 96 and 160 MB CompactFlash Memory, February 1999, 1 page.		
18	Hsia et al., "A Reconfigurable Interconnect For In-Silicon Electronic Assembly", <i>IEEE</i> , (1982) pp. 7-16.		
19	Hsia et al., "Adaptive Wafer Scale Integration", Proceeding of the 11 th Conference (1979 International), <i>Japanese Journal of Applied Physics, Volume 19</i> (1980) Supplement 19-1, pp. 193-202.		
20	Hsia, "Memory Applications Of The MNOS", <i>1972 Wescon Technical Papers Volume 16</i> , (1972) pp. 1-5.		
21	Nijima, "Design of a solid-state file using flash EEPROM," IBM J. Res. Develop, Vol. 39, No. 5, September 1995, 15 pages (as marked up by Dr. Harari).		
22	Heller Ehrman, Letter from A. Sharma to J. Yoon identifying Heller's Respondents' Prior Art, dated March 10, 2008, 7 pages.		
23	United States International Trade Commission, Investigation No. 337-TA-619, Notice of Prior Art of Respondents Apacer Technology, Inc. and Apacer Memory America, Inc., Silicon Motion Technology Corporation, Silicon Motion, Inc. (Taiwan), Silicon Motion, Inc. (California) and Silicon Motion International, Inc., Transcend Information, Inc. (Taiwan), Transcend Information, Inc. (California) and Transcend Information Maryland, Inc., dated July 11, 2008, 17 pages.		
24	United States International Trade Commission, Investigation No. 337-TA-619, The Heller Respondents' Notice of Prior Art, dated July 11, 2008, 23 pages.		
25	United States International Trade Commission, Investigation No. 337-TA-619, Respondents Kingston Technology Co., Inc., Kingston Technology Corporation, Memosun, Inc., Payton Technology Corporation, and Phison Electronics Corporation's Notice of Prior Art, dated July 11, 2008, 19 pages.		
26	United States International Trade Commission, Investigation No. 337-TA-619, Respondents LG Electronics, Inc. and LG Electronics U.S.A., Inc's Notice of Prior Art, July 11, 2008, 16 pages.		

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.	Application No.
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		SNDK.156US2	11/250,238
		Applicants	Conf. No.
(Use several sheets if necessary)		Kevin M. Conley	7727
(Form PTO-1449)		Filing Date	Art Group
		October 13, 2005	2189
	27	United States International Trade Commission, Investigation No. 337-TA-619, Niles Kynett's testimony dated October 29, 2008, 22 pages.	
	28	United States International Trade Commission, Investigation No. 337-TA-619, Niles Kynett's testimony dated October 30, 2008, 23 pages.	
	29	United States International Trade Commission, Investigation No. 337-TA-619, witness statement of Kevin Conley dated October 9, 2008, 9 pages.	
	30	United States International Trade Commission, Investigation No. 337-TA-619, Professor M. Ray Mercer's testimony dated November 3, 2008, 30 pages.	
	31	United States International Trade Commission, Investigation No. 337-TA-619, Dr. Thomas Rhyne's testimony dated October 27, 2008, 86 pages.	
	32	United States International Trade Commission, Investigation No. 337-TA-619, Dr. Thomas Rhyne's testimony dated October 28, 2008, 84 pages.	
	33	United States International Trade Commission, Investigation No. 337-TA-619, Dr. Thomas Rhyne's testimony dated October 29, 2008, 34 pages.	
	34	United States International Trade Commission, Investigation No. 337-TA-619, Dr. Subramanian's testimony dated October 30, 2008, 60 pages.	
	35	United States International Trade Commission, Investigation No. 337-TA-619, Dr. Subramanian's testimony dated October 31, 2008, 59 pages.	
Examiner		Date Considered	
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.			

Electronic Patent Application Fee Transmittal

Application Number:	11250238			
Filing Date:	13-Oct-2005			
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY			
First Named Inventor/Applicant Name:	Kevin M. Conley			
Filer:	Gerald Paul Parsons/Svetlana Stellmach (GPP)			
Attorney Docket Number:	SNDK.156US2			
Filed as Large Entity				
Utility under 35 USC 111(a) Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Miscellaneous:				
Submission- Information Disclosure Stmt	1806	1	180	180
Total in USD (\$)				180

Electronic Acknowledgement Receipt

EFS ID:	4521107
Application Number:	11250238
International Application Number:	
Confirmation Number:	7727
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY
First Named Inventor/Applicant Name:	Kevin M. Conley
Customer Number:	66785
Filer:	Gerald Paul Parsons/Svetlana Stellmach (GPP)
Filer Authorized By:	Gerald Paul Parsons
Attorney Docket Number:	SNDK.156US2
Receipt Date:	24-DEC-2008
Filing Date:	13-OCT-2005
Time Stamp:	13:59:49
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$180
RAM confirmation Number	603
Deposit Account	040258
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
-----------------	----------------------	-----------	-------------------------------------	------------------	------------------

1	Foreign Reference	Reference_3_WO-97-43764A1.pdf	738797 1cb28c129626554cf5298874f8dd1d79831da27	no	14
Warnings:					
Information:					
2	Foreign Reference	Reference_4_WO-00-49488A1.pdf	4413260 83210d4dbc93edc86423a3eda2a3c9d30811b07f	no	107
Warnings:					
Information:					
3	NPL Documents	Reference_5_Depo_Kevin_Conley-dtd_7-24-08_Vol-I.pdf	2919631 896aa97e9829f899679645a1625b01c67e2d1260	no	49
Warnings:					
Information:					
4	NPL Documents	Reference_6_Depo_Conley_dtd_7-25-08_Vol-II.pdf	2293766 4e278012696d7a929bfe2d82059f9186220f716	no	35
Warnings:					
Information:					
5	NPL Documents	Reference_7_Depo_Conley_dtd_9-11-08_Vol-III.pdf	2334078 391157914abce57b183b5cf12665615c4e26c836	no	35
Warnings:					
Information:					
6	NPL Documents	Reference_8_Exhibits_1-12_15-22_24-26_28-32_Conley.pdf	18475135 870a9ceb2c1f3af549e16ad112f6de07a64bc5e	no	573
Warnings:					
Information:					
7	NPL Documents	Reference_9_Exhibits_37-56_58-59_61-72_Conley.pdf	23575513 a1b776a74c3d9743f8c449a395ba2d6ccf572d49	no	729
Warnings:					
Information:					
8	NPL Documents	Reference_10_Depo_Mangan_dtd_7-10-08_Vol-1.pdf	284068 73b3ddada346bfc9bcbca210760375c36685b33	no	61
Warnings:					
Information:					
9	NPL Documents	Reference_11_Mangan_dtd_07-11-08_Vol-2.pdf	149452 ca935655801c01f2b77d39b4c65f4b1190e1be1	no	37
Warnings:					
Information:					

10	NPL Documents	Reference_12_Exhibits_1_thr_10_Mangan.pdf	18126619 8138982f82d21f20b15fc2a7303db63c2711415	no	540
Warnings:					
Information:					
11	NPL Documents	Reference_13_Exhibits_11_thr_15_17_thr_25_29_31_32_Mangan.pdf	17684093 0cfb4df01f915243c8a5130e85eae60413b7ca6e	no	504
Warnings:					
Information:					
12	NPL Documents	Reference_14_SanDisk_Mizer_Product_Logical_Format.pdf	1647187 f556807d700254e175917a9ca81f647f93c21f71	no	32
Warnings:					
Information:					
13	NPL Documents	Reference_15_SanDisk_MediaClip_Mem-Card_1996.pdf	3044808 7ab5663089234781689f9ae5ed2a6fb192994706	no	74
Warnings:					
Information:					
14	NPL Documents	Reference_16_SanDisk_MultimediaCard_April-1998.pdf	4083139 cb42a164009237f7be557ea8cb2d45b05e36c4c0	no	96
Warnings:					
Information:					
15	NPL Documents	Reference_17_Article_Photo_Interpreter-dtd-02-99.pdf	100427 c8e53a75dc8544f5dc3dfcd56d0c538c1115210c	no	1
Warnings:					
Information:					
16	NPL Documents	Reference_18_Hsia_A_Reconfigurable_Interconnect.pdf	508765 be31885f8d1d61bc7eeff6b5834e00d33b4591984	no	11
Warnings:					
Information:					
17	NPL Documents	Reference_19_Hsia_Adaptive_Wafer_Scale.pdf	501829 0b20920ff28a3b23968efcd370a3bca7203476db1	no	10
Warnings:					
Information:					
18	NPL Documents	Reference_20_Hsia_Memory_Applications.pdf	235664 e85ec5cd54fe67b95b942ee6324412f0677a355	no	6
Warnings:					
Information:					

19	NPL Documents	Reference_21_Nijima_article_ markedup.pdf	7535548 4c7df0140896881d2c2cd2b3cf81ca04c70b 4eea	no	15
Warnings:					
Information:					
20	NPL Documents	Reference_22_Ltr-from- Sharma-to-Yoon-identifying- Heller-Respondents-prior-art. pdf	128929 5bd52c01799683c57d2aac0c0d59c8f1fb31 db4d	no	7
Warnings:					
Information:					
21	NPL Documents	Reference_23_Apacer_SMI_Tra nscend-Notice-of-Prior-Art.pdf	441692 c5ca1c5e61ed51ff61ec07a1782d6965c646 c9ef	no	17
Warnings:					
Information:					
22	NPL Documents	Reference_24_Heller-Notice-of- Prior-Art.pdf	1920985 6e4e64b45e5465ae65a2bc7f992f53af9954 f832	no	23
Warnings:					
Information:					
23	NPL Documents	Reference_25_Kingston_Phiso n_Notice-of-Prior-Art.pdf	735269 23381dfb432525771db91514ac1fe37f33e1 5e98	no	19
Warnings:					
Information:					
24	NPL Documents	Reference_26_LGE-Notice-of- Prior-Art.pdf	196692 0dbfdb15e97d51d49733433ae85e48534e5 c271d	no	16
Warnings:					
Information:					
25	NPL Documents	Reference_27_Kynett_testimon y_10-29-08.pdf	11295639 c36ca7399eb71adbc480456815abb247194 62d18	no	22
Warnings:					
Information:					
26	NPL Documents	Reference_28_Kynett_testimon y_10-30-08.pdf	11145581 6f549a3e03697cad8800b757c041af49b5ad d421	no	23
Warnings:					
Information:					
27	NPL Documents	Reference_29_Conley_witness _statement_2008-10-10.pdf	359329 53d75befd2d72017eccf8bf570206d175a79 d2fab	no	9
Warnings:					
Information:					

28	NPL Documents	Reference_30_Mercer_Open_T estimon_2008-11-03.pdf	44489 6e17d31693abc4521423b6c5d74311e9a8c 7831a	no	30
Warnings:					
Information:					
29	NPL Documents	Reference_31_Rhyne_Open_Te stimony_2008-10-27.pdf	109340 8d83345974ca7a2d670b54a046026ef30eb f743b	no	86
Warnings:					
Information:					
30	NPL Documents	Reference_32_Rhyne_Open_Te stimony_2008-10-28.pdf	107240 fa4a035fb2c79f100c4fd0b98ef8b1a8588c8 246	no	84
Warnings:					
Information:					
31	NPL Documents	Reference_33_Rhyne_Open_Te stimony_2008-10-29.pdf	50785 cb38bed4429bcb74b29c72fd7c1e3fedd48 6324f	no	34
Warnings:					
Information:					
32	NPL Documents	Reference_34_Subramanian_O pen_Testimony_2008-10-30. pdf	77809 a9f5ae734f63fae033bfcb727f7e6c8539dc7 100	no	60
Warnings:					
Information:					
33	NPL Documents	Reference_35_Subramanian_O pen_Testimony_2008-10-31. pdf	78466 ca38c67f736fb5ccd47ba62375ef9d20dddf 839	no	59
Warnings:					
Information:					
34	Information Disclosure Statement (IDS) Filed (SB/08)	SNDK156US2_Suppl_IDS_12-2 4-08.pdf	264656 a6d6b18946000ae7977af1814488c50ba25e 7948	no	5
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
35	Fee Worksheet (PTO-06)	fee-info.pdf	30486 2710d652264efc4f906ece29100a60eb21ad c749	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			135639166		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

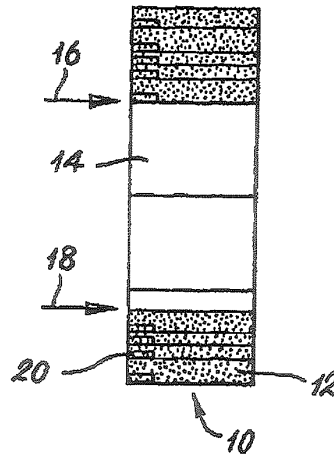
(51) International Patent Classification ⁶ : G11C 7/00, G06F 5/06	A1	(11) International Publication Number: WO 97/43764
		(43) International Publication Date: 20 November 1997 (20.11.97)

(21) International Application Number: PCT/GB97/01241	(81) Designated States: CN, JP, KR, SG, US, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).
(22) International Filing Date: 8 May 1997 (08.05.97)	Published With international search report.
(30) Priority Data: 9609833.0 10 May 1996 (10.05.96) GB	
(71) Applicant (for all designated States except US): MEMORY CORPORATION PLC [GB/GB]; The Computer House, Dalkeith Palace, Dalkeith, Edinburgh EH22 2NA (GB).	
(72) Inventor; and	
(75) Inventor/Applicant (for US only): SINCLAIR, Alan, Welsh [GB/GB]; 102 Lamb's Lane, Cottenham, Cambridge CB4 4TA (GB).	
(74) Agent: CULLIS, Roger; British Technology Group Ltd., Patents Dept., 101 Newington Causeway, London SE1 6BU (GB).	

(54) Title: MEMORY DEVICE

(57) Abstract

An addressable memory device for storing blocks of varying length, utilises a write pointer (18) to indicate the address of the next location to which data are to be written and an erase pointer (16) to indicate the address of the next location from which data are to be erased. It has a sector header (20) appended to each group of data containing information (38) indicating the length of the corresponding sector of data, and the location stored by the write pointer (14), which is selected to ensure that there is always at least one erased block adjacent to the current write block.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

Memory Device

This invention relates to electronic memory devices and, in particular, it relates to memory devices which are capable of storing data in variable length blocks.

Memory systems are used to store data. As data storage requirements increase
5 memory device capacity must also increase. To avoid having to use memories that are physically large or contain a large number of cells, many memory systems incorporate data compression. Data compression is appealing because it has the advantage of increasing the amount of data that can be stored in a memory device without increasing the physical size of the memory device. A number of different data compression techniques exist. Some of
10 these techniques are implemented by hardware others are implemented by software.

In European patent application No. EP 0436104A2 (National Semiconductor) a data communications system with multiple ports using shared data has a transceiver with a FIFO data store. During data reception, a comparator compares a subset of the incoming data with a predetermined reference to decide whether the data should be stored or aborted. This
15 operation sets a memory address value for a commit pointer. The first subset of data behind the commit pointer is selectively stored and the subset after the point is selectively aborted. The reason for this operation is to select data appropriate for the particular port.

In European patent application No. EP 0509722 A2 (NEC) a multiple processor system has an I/O buffer with a predetermined number of buffer areas. An external pointer
20 stores an address corresponding to the initial position of one of the buffer areas. The address is calculated with reference to the ratio of the length of a data block to be transferred and the total length of the buffer areas.

In US patent No. 4297567 (Philips) a content addressable FIFO memory in which the stored signal is addressed in a mode determined by the content of the stored signal.
25 Instructions for processing the signal are stored in a the same memory as a contiguous signal.

In US patent No. 4507760 (AT&T) a FIFO memory system has an addressable cyclic store, a write pointer, a read pointer and a last-word pointer for identifying the end of a multiword message. Before a write operation a comparator checks that the write and
30 read pointers are not at the same setting.. Before reading a comparator checks that the read pointer and the last-word pointer are not at the same setting. The contents of the write

pointer are copied to the last-word pointer register on receipt of an end-of-message signal. On identification of an error in a new memory word, the last word pointer is copied to the write pointer register.

5 In US patent No. US 5410308 (Deutsche Thomson-Brandt) a video signal storage system encodes pixels into variable length data words dependent on discrete cosine transformation of DC and AC components of a pixel. The encoder has an address flag to indicate whether or not a transport block contains data from segmented data blocks having a length greater than a predetermined average length and makes provision for storing such data elsewhere.

10 In US patent No. 5495552 (Mitsubishi) an audio signal is encoded into several hierarchical levels of data corresponding to increasing levels of fidelity. After the available memory is full, recording continues by over-writing successively lower hierarchical level levels. At the conclusion a code is recorded indicating the number of hierarchical levels to be reproduced.

15 US patent No. 4942553 (Zilog) discloses that, in a FIFO memory device controlling data transfer between a microprocessor and peripheral devices, memory over-runs, with consequent loss of data, and under-runs, with consequent transmission of garbage, are obviated by having two user programmable levels for generating a notification to the DMA or co-processor when action is required.

20 In US patent No. 4271480 (Honeywell-Bull) a FIFO store has an input and output interface with stores for input and output data words. Flags indicated when the input and FIFO stores are empty and ready to receive data. A controller sets the width of the output data words in response to an external signal.

25 In Japanese patent application No. 57033469 (Hitachi) increase of throughput and restriction of use of buffer memory is obtained by storing a pointer to the end of stored data. The system employs read, write and data end pointers. Writing is inhibited after the write pointer reaches the value stored in the end pointer. Erasure takes place asynchronously with reading and is controlled by the end pointer.

30 Implementing efficient data compression in memory systems is not simple because the files are split into a number of fixed data block sizes called sectors. If files are compressed before they are split into sectors then there is no problem because each sector

except the final sector is full of compressed data. If, however, files are split into sectors before data compression is applied then the compressed sectors will not be of a standard length, they will be of variable length. That is, one sector may compress to half its original size, whereas another sector may compress to a quarter of its original size. Sectors do not
5 all compress to a common size.

Consider the following case. A sector is compressed and stored at a memory location, a second sector is then compressed and stored in the memory location immediately adjacent to it. If the first sector is then altered it might not be compressed to the same size as before. If it is larger than it was before then it will not fit into the memory
10 space it previously occupied; if it is smaller than it was before then there will be wasted memory space causing disk fragmentation. Disk fragmentation reduces the storage efficiency of the memory device which may offset the benefits gained by data compression.

The invention is concerned with a memory device which stores variable length data blocks such as the blocks produced by data compression of fixed length sectors.

15 Accordingly, the present invention provides an addressable memory device for storing data arranged in groups, said groups of data not being of a fixed length, comprising a memory, a write pointer adapted to indicate an address of the next set of locations to which data are to be written and an erase pointer adapted to indicate the address of the next location from which data are to be erased wherein a sector header is appended to each
20 group of stored data, the location stored by the write pointer being selected to ensure that there is always at least one set of erased memory location adjacent to the set of locations indicated by said write pointer.

An embodiment of the present invention will now be particularly described, by way of example, with reference to the accompanying drawing in which:

25 Figure 1 shows a diagram of a block-erasable memory; and

Figure 2 shows a diagram of a sector header.

Figure 1 shows an addressable block-erasable memory 10 for storing sectors of data
12, where a sector is not of fixed length. The minimum area of memory that can be erased in an erase operation is called an erase block. The memory 10 has an erase pointer 16
30 which indicated the next block of memory from which data are to be erased and a write pointer 18 which indicates the next location to which data are to be written. These two

pointers (16,18) are separated by at least one erase block 14 in the erased state. The erase pointer 16 points to the first sector in an erase block. The write pointer 18 points to the area of memory space to be written to, that is, the memory location immediately adjacent to the sector 12 that was most recently written. After a sector 12 is written, the write pointer 18 is incremented past the end of that sector 12. Any sectors 12 written to the memory 10 directly from a host are compressed according to the compression technique being implemented and stored at the location defined by the write pointer 18.

The pointers (16,18) cycle through the memory space ensuring that the two pointers are separated by at least one erase block 14 in the erased state. Obsolete sectors in the erase block 14 that is currently pointed to by the erase pointer 16 are not relocated; whereas any sectors in that erase block 14 containing valid data are relocated to the location currently indicated by the write pointer 18. Once all of the valid sectors in an erase block 14 have been relocated the erase block 14 is erased and the erase pointer 16 is incremented to point to the first sector 12 in the next erase block 14.

Each sector of data 12 has a sector header 20 associated with it. The sector header is used to indicate how long the sector is and to store control information relating to the sector. The sector header 20 contains a number of fields, as shown in Figure 2.

A write flag 30 consists of one bit which is used to show whether or not data have been written to the sector 12. The write flag 30 is set when a sector of data is written and is cleared whenever an erase block 14 is erased. A compressed flag 32 is a single data bit which is used to indicate whether or not the data in the sector 12 are compressed. The compressed flag 32 is set when the sector data are compressed. A host error correction code (ECC) flag 34 is used when an ECC for a sector is transmitted with the sector data from the host.

A logical sector address field 36 is composed of a plurality of data bits. The logical sector address field 36 is used to store the logical sector address as defined by the host. The logical sector address is the address generated by the host. The number of bits required for the logical sector address 36 will depend on the physical size of the memory.

A sector length field 38 is used to indicate the length of the sector data stored in the block-erasable memory 10. There may also be a header ECC field 40 for providing a means of checking the integrity of the data stored in the sector header 20.

By reading the sector header field 20 the write pointer 18 can calculate where the end of the sector will be. Every time a sector is altered it is relocated so that the sector starts at the memory location adjacent to the sector that was written most recently. Since there is always at least one erase block 14 in the erased condition adjacent to the write pointer 18, a data sector 12 can always be written at the location defined by the write pointer 18. Thus the present invention ensures that data compression techniques can be used efficiently with block erasable memories.

In the above embodiment, the sector length field, which is included in the header defines the logical length of the sector and that logical length may change if data compression is used. In an alternative embodiment the sector length field stores the physical length of the sector. The physical length may vary for two reasons, firstly, if the logical length is variable, perhaps because of data compression and secondly, to accommodate defective memory locations. If a sector spans both good and defective memory regions, the physical length must be altered to take account of the extra addresses which must be allocated because faulty memory cannot be accessed.

It will be appreciated that various modifications may be made to the above described embodiment within the scope of the present invention. For example the present invention may be used with a memory that is byte erasable rather than block-erasable. Some embodiments of the present invention may not use any form of error checking or correction codes.

Claims

1. An addressable memory device for storing data arranged in groups 12, said groups of data not being of a fixed length, comprising a memory 10, a write pointer 18 adapted to indicate an address of the next set of locations to which data are to be written and an erase
5 pointer 16 adapted to indicate the address of the next location from which data are to be erased **characterised in that** a sector header 20 is appended to each group of stored data, the location stored by the write pointer 14 being selected to ensure that there is always at least one set of erased memory location adjacent to the set of locations indicated by said write pointer.
- 10 2. An addressable memory device for storing blocks of data according to claim 1, **characterised in that** each sector header contains information 38 to indicate the length of the corresponding sector of data.
3. An addressable memory device for storing blocks of data, where the blocks of data are not of a fixed length, according to any preceding claim **characterised in that** the sector
15 header contains means 36 for indicating the address generated by the host.
4. An addressable memory device for storing blocks of data, where the blocks of data are not of a fixed length, according to any preceding claim **characterised in that** the sector header contains means 40 for correcting or checking the data bits in the said sector header.
5. An addressable memory device for storing blocks of data, where the blocks of data are
20 not of a fixed length, according to any preceding claim **characterised in that** the sector header contains means 20 for indicating whether the sector data has been compressed or not.
6. An addressable memory device for storing blocks of data, where the blocks of data are not of a fixed length, according to any preceding claim **characterised in that** the physical
25 memory locations 14 between the write pointer and the erase pointer are held in the erased state.
7. A computer system including an addressable memory device according to any one of the preceding claims.

Abstract

An addressable memory device for storing blocks of varying length, utilises a write pointer 18 to indicate the address of the next location to which data are to be written and an erase pointer 16 to indicate the address of the next location from which data are to be erased. It has a sector header 20 appended to each group of data containing information 38 indicating the length of the corresponding sector of data, and the location stored by the write pointer 14, which is selected to ensure that there is always at least one erased block adjacent to the current write block.

Figure 1

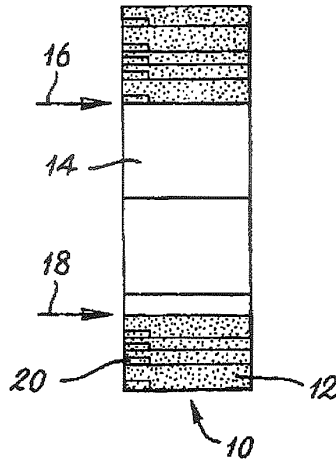


FIG: 1

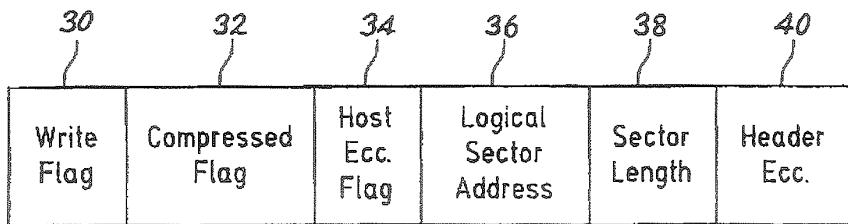


FIG: 2

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 97/01241

A. CLASSIFICATION OF SUBJECT MATTER IPC 6 G11C7/00 G06F5/06		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC 6 G11C G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 436 104 A (NATIONAL SEMICONDUCTOR) 10 July 1991 cited in the application see column 6, line 22 - column 11, line 41; figures 1-3 ---	1
A	EP 0 509 722 A (SUZUKI) 21 October 1992 cited in the application see column 2, line 48 - column 3, line 18; figures 1-3 ---	1
A	US 4 297 567 A (HERZNER) 27 October 1981 see the whole document ---	1
A	US 4 507 760 A (FRASER) 26 March 1985 see column 3, line 18 - column 4, line 2 see claim 1 ---	1
--- /---		
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C.		
<input checked="" type="checkbox"/> Patent family members are listed in annex.		
* Special categories of cited documents:		
A document defining the general state of the art which is not considered to be of particular relevance *E* earlier document but published on or after the international filing date *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) *O* document referring to an oral disclosure, use, exhibition or other means *P* document published prior to the international filing date but later than the priority date claimed *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. *Z* document member of the same patent family		
3	Date of the actual completion of the international search 18 July 1997	Date of mailing of the international search report 29. 07. 97
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl, Fac (+ 31-70) 340-3016		Authorized officer Degraeve, L

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 97/01241

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 410 308 A (KEESEN ET AL) 25 April 1995 see column 3, line 10 - column 5, line 23 see claims 1,2 ---	1
A	US 5 495 552 A (SUGIYAMA ET AL) 27 February 1996 see column 3, line 1 - column 8, line 13 ---	1
A	US 4 942 553 A (DALRYMPLE ET AL) 17 July 1990 see column 2, line 5 - column 6, line 43; figures 1-4 ---	1
A	US 4 271 480 A (VINOT) 2 June 1981 see the whole document ---	1
A	PATENT ABSTRACTS OF JAPAN vol. 6, no. 100 (P-121), 9 June 1982 & JP 57 033469 A (HITACHI) see abstract -----	1

3

Form PCT/ISA/210 (continuation of annex sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 97/01241

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 436104 A	10-07-91	US 5016221 A JP 3241585 A	14-05-91 28-10-91
EP 509722 A	21-10-92	JP 4314163 A AU 661301 B AU 1487592 A CA 2065894 A US 5555380 A	05-11-92 20-07-95 15-10-92 12-10-92 10-09-96
US 4297567 A	27-10-81	DE 2834509 A FR 2433267 A GB 2027959 A,B JP 1373273 C JP 55032199 A JP 61035580 B	14-02-80 07-03-80 27-02-80 07-04-87 06-03-80 13-08-86
US 4507760 A	26-03-85	CA 1191277 A EP 0118446 A JP 3007300 B JP 59501435 T WO 8400835 A	30-07-85 19-09-84 01-02-91 09-08-84 01-03-84
US 5410308 A	25-04-95	AU 8738191 A DE 59107937 D WO 9208289 A EP 0554300 A ES 2089237 T JP 6502051 T	26-05-92 18-07-96 14-05-92 11-08-93 01-10-96 03-03-94
US 5495552 A	27-02-96	DE 4313175 A JP 6259099 A US 5630010 A	21-10-93 16-09-94 13-05-97
US 4942553 A	17-07-90	NONE	
US 4271480 A	02-06-81	FR 2337376 A DE 2659621 A JP 1266091 C JP 52085426 A JP 59044655 B	29-07-77 14-07-77 27-05-85 15-07-77 31-10-84

Form PCT/ISA/210 (patent family annex) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No

PCT/GB 97/01241

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 4271480 A		NL 7614636 A	04-07-77

Form PCT/ISA/218 (patent family annex) (July 1992)

page 2 of 2

SANDISK CONFIDENTIAL BUSINESS INFORMATION SUBJECT TO PROTECTIVE ORDER
HIGHLY CONFIDENTIAL ATTORNEYS' EYES ONLY

SD-I-1883164

APPLE INC.
EXHIBIT 1102 - PAGE 1214

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification: G06F 3/06	A1 (11) International Publication Number: (43) International Publication Date:	WO 00/49488 24 August 2000 (24.08.2000)
(21) International Application Number: PCT/GB00/00550 (22) International Filing Date: 17 February 2000 (17.02.2000) (30) Priority Data: 9903490.2 17 February 1999 (17.02.1999) GB (60) Parent Application or Grant MEMORY CORPORATION PLC [?]; O. SINCLAIR, Alan, Welsh [?]; O. OUSPENSKAJA, Natalia Victorovna [?]; O. TAYLOR, Richard, Michael [?]; O. GOROBETS, Sergey, Anatolievich [?]; O. SINCLAIR, Alan, Welsh [?]; O. OUSPENSKAJA, Natalia Victorovna [?]; O. TAYLOR, Richard, Michael [?]; O. GOROBETS, Sergey, Anatolievich [?]; O. MCCALLUM, William, Potter ; O.	Published	
(54) Title: MEMORY SYSTEM (54) Titre: SYSTEME DE MEMOIRE		
(57) Abstract		
<p>A memory system (10) having a solid state memory (6) comprising non-volatile individually addressable memory sectors (1) arranged in erasable blocks, and a controller (8) for writing to reading from the sectors, and for sorting the blocks into "erased" and "not erased" blocks. The controller performs logical to physical address translation, and includes a Write Pointer (WP) for pointing to the physical sector address to which data is to be written from a host processor. A Sector Allocation Table (SAT) of logical addresses with respective physical addresses is stored in the memory, and the controller updates the SAT less frequently than sectors are written to with data from the host processor. The memory may be in a single chip, or in a plurality of chips. A novel system for arranging data in the individual sectors (1) is also claimed.</p>		
(57) Abrégé		
<p>La présente invention concerne un système de mémoire (10) comportant une mémoire électronique (6) contenant des secteurs de mémoires (1) non volatiles adressables un à un placés dans des blocs effaçables, et un contrôleur (8) assurant les opérations d'écriture et de lecture dans les secteurs, et assurant le classement des blocs en blocs "effacés" et "non effacés". En outre, ce contrôleur assure la conversion d'adresse logique en adresse physique, et applique un pointeur d'écriture (WP) désignant l'adresse secteur physique où les données provenant du processeur hôte doivent être écrites. La mémoire conserve une table d'affectation de secteur (SAT) donnant pour chaque adresse logique une adresse physique correspondante, le contrôleur mettant à jour les SAT moins fréquemment qu'il ne fait d'écriture secteur avec les données provenant du processeur hôte. La mémoire peut être réalisée sous la forme d'un seul microcircuit ou de plusieurs microcircuits. Par ailleurs, cette invention concerne un nouveau système d'organisation des données dans les secteurs (1).</p>		

4-6889
 PENAGAD 800-831-6889
 JH EXHIBIT
 2/1/00
 36
 Conley

SANDISK CONFIDENTIAL BUSINESS INFORMATION SUBJECT TO PROTECTIVE ORDER
HIGHLY CONFIDENTIAL ATTORNEYS' EYES ONLY

SD-I-1883165

PCT

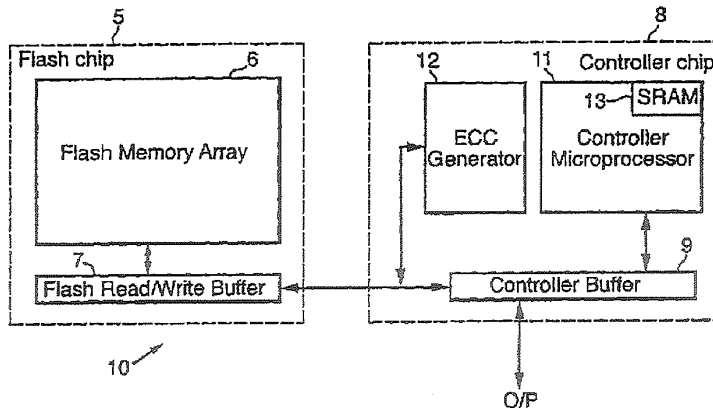
WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 3/06	A1	(11) International Publication Number: WO 00/49488
(21) International Application Number: PCT/GB00/00550	(22) International Filing Date: 17 February 2000 (17.02.00)	(43) International Publication Date: 24 August 2000 (24.08.00)
(30) Priority Data: 9903490.2 17 February 1999 (17.02.99) GB	(71) Applicant (for all designated States except US): MEMORY CORPORATION PLC [GB/GB]; The Computer House, Dalkeith Palace, Dalkeith, Edinburgh EH22 2NA (GB).	(81) Designated States: GB, JP, KR, SG, US, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).
(72) Inventors; and (75) Inventors/Applicants (for US only): SINCLAIR, Alan, Welsh [GB/US]; 14059 Mango Drive #D, Del Mar, CA 92014 (US). OUSPENSKAIA, Natalia Victorovna [RU/RU]; 1-Murinski Prospect 29/20 apt. 63, St. Petersburg, 194100 (RU). TAYLOR, Richard, Michael [GB/GB]; Old Sawmill House, 41 Newmill Road, Dalkeith, Midlothian EH22 2AQ (GB). GOROBETS, Sergey, Anatolievich [RU/GB]; 1PI, 16 East Mayfield, Edinburgh EH9 1SE (GB).	Published <i>With international search report.</i>	
(74) Agents: MCCALLUM, William, Potter et al.; Cruikshank & Fairweather, 19 Royal Exchange Square, Glasgow G1 3AE (GB).		

(54) Title: MEMORY SYSTEM



(57) Abstract

A memory system (10) having a solid state memory (6) comprising non-volatile individually addressable memory sectors (1) arranged in erasable blocks, and a controller (8) for writing to reading from the sectors, and for sorting the blocks into "erased" and "not erased" blocks. The controller performs logical to physical address translation, and includes a Write Pointer (WP) for pointing to the physical sector address to which data is to be written from a host processor. A Sector Allocation Table (SAT) of logical addresses with respective physical addresses is stored in the memory, and the controller updates the SAT less frequently than sectors are written to with data from the host processor. The memory may be in a single chip, or in a plurality of chips. A novel system for arranging data in the individual sectors (1) is also claimed.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	MT	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon	KR	Republic of Korea	PL	Poland		
CN	China	KZ	Kazakhstan	PT	Portugal		
CU	Cuba	LC	Saint Lucia	RO	Romania		
CZ	Czech Republic	LI	Liechtenstein	RU	Russian Federation		
DE	Germany	LK	Sri Lanka	SD	Sudan		
DK	Denmark	LR	Liberia	SE	Sweden		
KE	Kenya			SG	Singapore		

Description

5

10

15

20

25

30

35

40

45

50

55

MEMORY SYSTEM

5
10
15
20
25
30
35
40
45
50
55

The present invention relates to a solid state memory system for data storage and retrieval, and to a memory controller for controlling access to a non-volatile memory of a solid state memory system. In particular, the invention relates to FLASH memory systems and controllers for FLASH memories.

FLASH EEPROM (electrically erasable programmable read only memory) devices are commonly used in the electronics industry for non-volatile data storage. Various types of FLASH memory devices exist, including devices based on NAND type memory cells, AND type memory cells, or NOR type memory cells. Such devices may have different types of interfaces to the host processor system(s) for which they are designed to interface, for example they may use a serial access type interface (as commonly used in many NAND and AND type devices) or a random access type interface (as used in some NOR type devices). The present invention is intended to be applicable, in appropriate forms, to at least some and preferably all of these different types of memory devices.

It is known to use solid state memory systems to try to emulate magnetic disc storage devices in computer systems. It is an aim of the industry to try to increase the speed of operation of solid state memory systems so as to better emulate magnetic disc storage.

According to a first aspect of the present invention we provide a memory system for connection to a host processor, the system comprising:
a solid state memory having non-volatile memory sectors which are individually addressable and which are arranged in erasable blocks of sectors, each said sector having a physical address defining its physical position in the memory;

5 and a controller for writing data structures to and reading
data structures from the memory, and for sorting the blocks of
sectors into blocks which are treated as erased and blocks
10 which are treated as not erased; wherein the controller
5 includes:
means for translating logical addresses received from the host
processor to physical addresses of said memory sectors in the
15 memory;
a write pointer (hereinafter referred to as the Write Pointer
10 (WP)) for pointing to the physical address of a sector to
which data is to be written to from the host processor, said
20 Write Pointer (WP) being controlled by the controller to move
in a predetermined order through the physical addresses of the
memory sectors of any block which is treated as erased and,
25 when the block has been filled, to move to another of the
erased blocks;
wherein the controller is configured so that, when a sector
write command is received from the host processor, the
30 controller translates a logical address received from the host
20 processor to a physical address to which data is written by
allocating for said logical address that physical address to
which said Write Pointer (WP) is currently pointing; and
35 wherein the controller is configured to compile a table of
logical addresses with respective physical addresses which
25 have been allocated therefor by the controller (this table
being hereinafter referred to as the Sector Allocation Table
40 or SAT), and wherein the controller updates the SAT less
frequently than memory sectors are written to with data from
the host processor.
45 30
By not updating the SAT every time data from the host
processor is written to a sector in the memory, but instead
50 updating the SAT on a less frequent basis, the present
invention thus provides very high speed operation of solid

55

5 state memory, for example FLASH memory, thereby enabling good
emulation of magnetic disk memory.

10 The physical sector addresses in the SAT are preferably
5 ordered by logical sector address, whereby the Nth SAT entry
contains the physical address of a sector to which data having
logical address N has been written. When a sector read command
15 is received from the host processor, the controller may look
up a logical sector address received from the host processor
20 in the SAT in order to obtain the physical sector address
which the controller previously allocated to said logical
sector address. The SAT is preferably stored in one or more of
said blocks of memory sectors in the solid state memory, each
25 block which contains any portion of the SAT hereinafter being
referred to as a SAT block. Preferably the SAT is updated by
rewriting one or more blocks of the SAT. By updating a whole
block of SAT sectors at a time this significantly speeds up
operation of the memory system.

30 There may be provided at least one block of sectors
(hereinafter referred to as the Additional SAT Block (ASB)),
containing modified versions of individual sectors of a said
35 SAT block. Each sector in a said ASB block preferably contains
the physical address of the sector of the SAT block which it
25 updates, and the modified version of the said SAT sector. The
purpose of an ASB is to cache individually in solid state
40 memory modified sectors of the SAT so as to reduce the number
of SAT block rewrites. When all the sectors in a said ASB
block are written to with modified versions of SAT sector(s),
45 the respective SAT block is rewritten so as to include all the
modified versions in the ASB block and the ASB block is
erased.

50 It will be appreciated that in the memory system of the
35 present invention the physical address which is allocated to

55

5 any given logical address received from the host processor is
not dependent on the logical address itself. The controller
merely allocates the physical sector address to which the
Write Pointer is currently pointing.

10 5
As described above, the controller fills one said block which
is treated as erased before moving the Write Pointer (WP) on
to another block. The controller may conveniently be
15 configured to move the Write Pointer (WP) in a predetermined
10 order through the blocks which are treated as erased.

20 The controller may conveniently control the Write Pointer (WP)
so as to move sequentially, in ascending numerical order of
physical address, through the erased blocks, as each block is
25 filled with data written thereto. The control of the Write
Pointer (WP) may be cyclic in the sense that once the sectors
in the highest block, according to physical address order,
have been filled with data the WP is controlled by the
30 controller to wrap around to the block of sectors having the
20 numerically lowest physical addresses out of all the blocks
currently being treated by the controller as erased.

35 The controller may, alternatively, use another predetermined
order for writing data to the memory sectors. For example, the
25 controller may control the Write Pointer (WP) to move
sequentially in descending numerical order, according to
40 physical address, through the blocks which are treated as
erased. Another possibility would be to move in non-sequential
order through the physical sector addresses. For example, the
45 30 WP may move in descending numerical address order through the
physical sector addresses in each block which is treated as
erased, and move from block to block in some predetermined
order such as, for example, in ascending numerical order
50 according to the physical address of the first sector in each
35 said block.

55

5

It will be appreciated that many other predetermined orders are possible for writing data to the sectors in the blocks which are treated as erased. Furthermore, the controller could use the erased blocks in any other order which need not be predetermined, or which may be only partially predetermined. Although generally not preferred, the erased blocks could even be used in a random order.

10

15

20

25

30

35

40

45

50

55

The memory sectors in each said block of sectors are preferably erasable together as a unit. The sectors may also be individually erasable (for example where the solid state memory is AND type memory). The controller is preferably configured to control erase operations on the memory so as to only erase whole blocks of memory sectors. A block of sectors will be treated by the controller as an erased block if all the memory sectors therein are erased sectors. If a block contains one or more bad (i.e. defective) sectors, the controller may define the whole block as being bad and treat that block as a not erased block, whereby no data will be written thereto. Alternatively, if a block contains one or more bad sectors the controller may treat that block as an erased block whereby the controller may still use good sectors in the block to store data. In the latter case, though, the memory system preferably includes a table identifying bad sectors and the controller is configured to check whether the next sector address to which the Write Pointer (WP) is to be moved is the address of a bad sector and, if it is the address of a bad sector, to control the Write Pointer to skip this bad sector and move to the next sector address according to the predetermined order in which the sectors are to be written to.

For the avoidance of doubt, any block which contains any good (i.e. not defective) sectors which have already been written to will be treated by the controller as a not erased block.

5 Furthermore, it is intended that the term "erased" sector
covers not only a sector which has been erased, but also
covers a sector which has never yet been written to, and so
10 has not yet ever been erased. Thus, a block of sectors which
5 have never yet been written to is treated by the controller as
an erased block.

15 Each block of sectors preferably has a physical block address
defining its physical position in the memory. The physical
10 address of each said memory sector will preferably include the
physical block address of the block in which it is located.
20 The controller may advantageously be configured to compile a
list of the physical block addresses of at least some of the
blocks of sectors being treated as erased, which may be used
15 by the controller in order to quickly identify the next block
of sectors to be written to. This list of addresses of erased
25 blocks is preferably stored by the controller in a temporary
memory which may be provided in the memory system, which
30 temporary memory may conveniently be an SRAM in a
20 microprocessor of the controller, and may be created from
information already stored in the solid state memory by the
controller identifying the erased state of each block of
35 sectors. (This information will preferably be held in the form
of a bitmap in the solid state memory, in which each block is
25 recorded as an erased block or a not erased block.)

40 The controller is conveniently configured so that, when a
sector write command is received by the controller from the
host processor which command renders obsolete data previously
45 written to another sector, the controller stores in a
temporary memory the address of the sector containing the now
obsolete data. This temporary memory may conveniently be SRAM
or DRAM provided in a microprocessor of the controller. If a
50 sector delete command, generated by a user, is received from
35 the host processor by the controller, the controller

55

5 preferably marks as obsolete the sector to be deleted (without
physically erasing the sector). The controller may allow only
one block at any time, hereinafter referred to as the Current
10 Obsolete Block (COB), to contain one or more sectors
5 containing obsolete data which was written by the Write
Pointer (WP), and when all the sectors in the COB contain
obsolete data, the COB is immediately erased. This is a
15 particularly suitable scheme for the case where the Write
Pointer (WP) moves sequentially through the memory sector
10 addresses in each block which is treated as erased before
moving on to the next block. In such a scheme, a series of
20 obsolete sectors to be deleted (which may, for example,
contain part of a user data file which has been rewritten)
will in most cases all be in the same block. When a series of
25 15 sectors are rewritten in a different order to that in which
they were previously written, this may create obsolete sectors
in more than one block. Where a sector in a block other than
the COB is to contain obsolete data, the controller preferably
30 relocates any data in valid (not obsolete) sectors in the COB
20 to another block, which may be the block to which the Write
Pointer (WP) is currently pointing, and then erases the COB.
Said sector in the block other than the COB is then marked as
35 obsolete and this other block is now the COB. Rather than
writing the relocated data to the current location of the
25 Write Pointer, the memory system may include a second write
pointer, hereinafter referred to as the Relocation Pointer
40 (RP), for pointing to the physical address of the sector to
which such relocated data is to be written, the Relocation
Pointer (RP) always being in a different block of sectors to
45 30 the Write Pointer (WP). This has the advantage of preventing
relocated data from being intermingled with data structures
directly ordered to be written by the host processor i.e.
50 written by the Write Pointer (WP).

55

5 Generally, only two types of data are written to the solid
state memory from the host processor. These are file data and
system data. To further reduce the number of reallocations and
10 erasures, the memory system may further include a third write
5 pointer, hereinafter referred to as the System Write Pointer
(SWP), which points to the physical address of the sector to
which system data is to be written from the host, the SWP
15 always being in a different block to the Write Pointer (WP)
(and in a different block to the Relocation Pointer, if there
10 is one). System data will preferably be identified during
initialisation of the system and will be updated as necessary
20 during operation.

Where both a write pointer (WP) and a system write pointer
25 (SWP) are provided, file data will in this case always be
written to the addresses pointed to by the Write Pointer (WP).
Both the Relocation Pointer (RP) and System Write Pointer
(SWP) are preferably controlled to move through the physical
30 addresses of the memory sectors in said blocks which are
20 treated as erased in a similar manner to the Write Pointer
(WP). Thus, when all the (good) sectors in a said block have
been filled with relocated data or system data, the respective
35 one of the Relocation Pointer (RP) and the System Write
Pointer (SWP) moves on to the next address defined by the
25 controller to be used from the physical addresses of all the
40 sectors in the blocks treated as erased.

Where a System Write Pointer (SWP) is provided, the controller
will preferably allow at least two blocks which contain one or
45 30 more obsolete sectors to exist at any time, one being said COB
and the other being a Current Obsolete System Block (COSB)
containing one or more obsolete system data sectors. If any
50 system data sectors need to be relocated in order to allow the
COSB to be erased, the relocated system data is preferably

55

5 sent to the address to which the System Write Pointer (SWP) is currently pointing.

10 In fact, there may temporarily exist more than two blocks (the
5 COB and COSB) containing obsolete data at any one time. It is possible that when the COB, for example, needs to be erased (obsolete data has just been created in another block) one of the write pointers may be pointing thereto i.e. the WP is still writing to the block which is currently the COB. Where
15 this is the case the controller preferably proceeds with creating the new COB but postpones the erasure of the old COB (which is hereinafter treated as the Pending Obsolete Block (POB)) until all erased sectors in the POB have been filled and the write pointer moves on to the next erased block to be
20 used, as defined by the controller. At this time any valid (not obsolete) data in the POB is relocated and the POB is erased.

30 In addition to writing data structures to the memory from the host processor, the controller may also generate and write to the memory data designated as control information. The controller preferably writes such control information in
35 separate ones of the blocks of memory sectors to those in which data structures received from the host processor are written. Blocks for storing such control information,
40 hereinafter referred to as Control Blocks (CBs), will be updated periodically by the controller and will be accessed during initialisation, and occasionally during operation, of the memory system.

45 30 The controller preferably stores in a temporary memory (which may be a RAM provided in the memory system or which may conveniently be an embedded SRAM or DRAM in a microprocessor
50 of the controller) a list of logical sector addresses for data structures which have been written by the Write Pointer (WP)

55

5 since the SAT was last updated. This list stored in the SRAM
is hereinafter referred to as the ~~Write Sector List (WSL)~~. The
logical addresses in the WSL are advantageously stored in the
10 order in which they were written to the non-volatile sectors
5 in the memory. Conveniently, for a group of consecutively
written sectors, the WSL entry may therefore be written as the
first sector logical address and the sector group length i.e.
15 the number of sectors written. Each said sector group is
defined so as not to span more than one block of sectors.

10
20 The controller advantageously also stores in said temporary
memory the order in which blocks have been used by the Write
Pointer (WP) for writing data since the last update of the
SAT. This is stored in the form of a list of block addresses
25 of the blocks in which the updated sectors whose addresses are
held in the WSL are located. This list of block addresses is
hereinafter referred to as the Write Block List (WBL). It will
be appreciated that since the memory system, by virtue of the
30 WSL and WBL, contains knowledge of the location in physical
20 memory which was allocated for the first logical address in
said group of consecutively written sectors, the controller
can thus always access the correct physical sector for each
35 logical sector address in a said group of consecutively
written sectors written since the last SAT update, using the
25 WSL and WBL. The WSL will preferably have a predetermined size
and once the WSL is full one or more SAT blocks (and/or ASBs)
40 may be updated and the WSL and WBL are emptied.

45 Preferably, the starting physical sector address, and the
30 links between blocks containing sectors to which data has been
written by the controller since the last SAT update, are also
stored in a Control Block of the solid state memory. By
50 storing the logical sector address for the user data stored in
each sector in the sector itself, for example in a header
35 field provided in the sector, the WSL and WBL can therefore

55

5 easily be recreated following any removal and restoration of
power to the system by scanning through the solid state
memory, reading the logical addresses in the sectors written
to since the last update of the SAT, until reaching a block
10 5 which is not full. This is the block which contained the Write
Pointer (WP) before removal or loss of power. This provides
high data security in the event of unexpected power removal
15 from the memory system.

10 Where a Relocation Pointer and a System Write Pointer are
included in the memory system, the controller preferably also
20 stores in said temporary memory (e.g. SRAM or DRAM in the
controller microprocessor) similar lists of logical sector
addresses corresponding to sectors in the memory to which
25 15 relocated data or system data has been written to
respectively, which lists are hereinafter referred to as the
Relocation Sector List (RSL) and Write System Sector List
(WSSL) respectively. The controller may also store in said
30 temporary memory corresponding lists of the order of blocks
20 which have been used by the RP and the SWP, similar to the
Write Block List, and these two lists will hereinafter be
referred to as the Relocation Block List (RBL) and the Write
35 System Block List (WSBL). Moreover, the starting physical
sector address, and the links between blocks containing
25 sectors to which relocated data or system data has been
written since the last SAT update may also be stored in at
40 least one said Control Block (CBs) of the solid state memory
whereby the RSL and WSSL can be recreated following any
removal and restoration of power to the host processor by
45 30 simply scanning the memory and reading the logical addresses
in the sectors written to by the RP and SWP respectively,
since the last update of the SAT.

50 Each said sector in any of the above-described embodiments may
35 consist of a single "page" of memory i.e. one row of memory

55

5 cells in a said block of memory sectors. However the invention
is not limited exclusively to such a sector format and in some
cases (for example when using random access NOR type memory)
each said sector may be less than, or greater than, one page.
10 Moreover, in the latter case not all said sectors need
necessarily be of the same size. For example, a data
organisation scheme such as that described in our earlier
International Patent Application No. PCT/GB99/00188 could be
15 used by the controller to form sectors of appropriate sizes so
20 as to avoid individual defects (of sub-sector size) which may
be present in the solid state memory.

Each sector is, as aforesaid, individually addressable. Each
sector may comprise a plurality of sector portions which are
15 also each individually addressable and the controller may
write to, and read from, each sector portion individually. It
will be appreciated that the smallest possible sector portion
size is the minimum addressable unit of the memory. In NOR
20 type memory, for example, the minimum addressable unit of
memory is commonly 1 byte.

The controller preferably writes data to, and reads data from,
35 the memory sectors in uniformly sized data segments. Where all
the memory sectors are the same size, each said data segment
25 is preferably equal in size to the size of a said memory
sector. Each data segment may comprise data structures from
40 the host processor (e.g. file or system data) and/or data
generated by the controller.

45 Where the solid state memory is based on NAND type devices,
the controller preferably stores in said one or more Control
Blocks a list of the block addresses of blocks in the non-
volatile memory containing bad sectors (hereinafter referred
50 to as the Bad Block List (BBL)), and the controller treats
35 each such block as a "not erased" block, so that it will not

55

5 appear in the list of erased blocks which may be stored in
temporary memory, and the controller will not write any data
to that block.

10 5 Where the memory is based on AND type devices, the controller
preferably stores in said one or more Control Blocks (CBs) a
list of addresses of any bad sectors, and the controller
15 controls the said write pointer(s) to use the good sectors in
any block containing at least one bad sector, and to skip any
20 bad sectors. It will be appreciated that in the latter case
where a block containing one or more bad sectors is to be
erased the good (i.e. non-defective) sectors in the block are
erased individually during a block erase operation.

25 15 The controller advantageously also stores in said one or more
Control Blocks a list of the block addresses of all SAT
blocks. This list is preferably in the form of a plurality of
list portions, each said portion being hereinafter referred to
30 as a Table Block List (TBL), and each said portion containing
the block addresses of a group of logically contiguous SAT
blocks and any corresponding ASBs.

35 The controller preferably stores the block addresses of said
one or more Control Blocks in a dedicated block of the memory
25 hereinafter referred to as the Boot Block (BB). Other
important information required for data security may also be
40 stored in the Boot Block, for example the list of bad blocks
(or bad sectors). Preferably, the first block of sectors in
the memory which does not contain any bad sectors is
45 30 designated as the Boot Block (BB).

50 Preferably, the controller will only use blocks containing all
good sectors as SAT blocks, Control Blocks, ASBs or BBs.

55

5 A cache may be provided in temporary memory (for example RAM
in the memory system, such as SRAM or DRAM in the controller
microprocessor), in which the controller stores a group of
contiguous SAT entries including the SAT entry most recently
10 5 accessed from the SAT (by the controller). This further
improves address translation speed. Further increase in speed
of address translation may be achieved by creating in said
temporary memory a list of physical addresses of all ASBs and
15 the SAT blocks with which they are associated (hereinafter
20 referred to as the ASB List or ASBL) which is updated each
time a SAT sector write operation is performed. Similarly, the
positions of the TBLs in the Control Block(s) may also be
stored in said temporary memory so as to allow even faster
logical-to-physical sector address translation.

25 15
The solid state memory may comprise a single memory array in
the form of a single memory chip, or may comprise a plurality
of memory arrays in the form of a plurality of memory chips.
Where the memory comprises a plurality of chips, the
30 20 controller advantageously forms the memory sectors in the
plurality of memory chips into a multiplicity of virtual
blocks, each said virtual block comprising one erasable block
of memory sectors from each said memory chip, and the
35 controller preferably sorts said virtual blocks into ones
25 which are treated as erased and ones which are treated as not
erased. The controller preferably compiles a list of the
virtual blocks treated as erased and stores this in temporary
memory in the memory system, which may be SRAM in a
40 microprocessor of the controller. The controller preferably
45 30 controls the Write Pointer (WP) (and the RP and SWP, where
provided) to move from one chip to another for each
consecutive sector write operation, starting at one sector in
one erasable block of the virtual block and moving
50 consecutively to one sector in each of the other erasable
35 blocks in the virtual block until one sector has been written

55

5 in each erasable block of the virtual block, and then moving
back to the chip in which the first sector was written and
proceeding in a similar manner to fill another one sector in
10 each erasable block of the virtual block, and so on until the
5 virtual block is full of data. The Write Pointer (WP) then
moves on to the next virtual block in said list of virtual
blocks being treated as erased, and fills this next virtual
15 block in a similar manner. The controller is preferably
configured so that for every n contiguous sector write
10 operations the controller executes, where n is less than or
equal to the number of solid state memory chips in the memory
20 system, the controller writes substantially concurrently to
one sector in each of n of the chips. The controller
preferably carries out erasure of any said virtual block by
25 concurrently erasing all the erasable blocks in the virtual
block.

30 It will be appreciated that the controller of the memory
system may be substantially implemented in circuitry as a
20 controller device, but will preferably be implemented, at
least in part, as firmware held in the memory of a controller
device. The controller may be integrally formed on the same
35 chip (or one of the same chips) as the solid state memory.

25 According to a second aspect of the invention we provide a
40 memory system for connection to a host processor, the memory
system comprising:
a solid state memory comprising a plurality of solid state
memory chips each having non-volatile memory sectors which are
45 individually addressable and which are arranged in erasable
blocks of sectors, each said sector having a physical address
defining its physical position in the memory;
and a controller for writing data structures to and reading
50 data structures from the memory, wherein:

55

5 the controller forms the erasable blocks into virtual blocks,
each said virtual block comprising an erasable block from each
of the memory chips, and the controller sorts the virtual
10 blocks into ones which are treated as erased and ones which
5 are treated as not erased, and the controller fills one
virtual block with data prior to moving on to the next virtual
block to be filled, and each virtual block is filled by
15 writing to the memory sectors thereof in a repeating sequence
in which the controller writes to one memory sector in each of
10 the erasable blocks of the virtual block one after another
whereby consecutively written sectors are in different chips.
20

Preferably, the controller is configured so that for every n
contiguous sector write operations the controller executes for
25 a multiple sector write command received from the host
processor, where n is less than or equal to the number of
solid state memory chips in the memory system, the controller
writes substantially concurrently to one sector in each of the
30 n of the chips.

20
According to a third aspect of the invention we provide a
controller for writing data structures to and reading data
35 structures from a solid state memory having non-volatile
memory sectors which are individually addressable and which
25 are arranged in erasable blocks of sectors, each said sector
40 having a physical address defining its physical position in
the memory, wherein the controller includes:
means for translating logical addresses received from a host
processor of a memory system in which the controller is used
45 30 to physical addresses of said memory sectors in the memory,
and for sorting the blocks of sectors into blocks which are
treated as erased and blocks which are treated as not erased,
50 and a Write Pointer (WP) for pointing to the physical address
of a sector to which is to be written to from the host
35 processor, said Write Pointer (WP) being controlled by the

55

5 controller to move in a predetermined order through the
physical addresses of the memory sectors in any block which is
treated as erased and, when the block has been filled, to move
10 to another of the erased blocks;
5 and wherein, when a sector write command is received by the
controller from the host processor, the controller translates
a logical sector address received from the host processor to a
15 physical address to which data is written by allocating for
said logical address that physical address to which said Write
10 Pointer (WP) is currently pointing;
and wherein the controller is configured to compile a table
20 (the SAT) of logical addresses with respective physical
addresses which have been allocated therefor by the
controller, and to update the SAT less frequently than memory
25 15 sectors are written to with data from the host processor.

According to a fourth aspect of the invention we provide a
method of controlling reading and writing of data structures
30 to and from a solid state memory having non-volatile memory
20 sectors which are individually addressable and which are
arranged in erasable blocks of sectors, each said sector
having a physical address defining its physical position in
35 the memory, the method comprising the steps of:
sorting the blocks of sectors into blocks which are treated as
25 erased and blocks which are treated as not erased;
40 providing a Write Pointer (WP) for pointing to the physical
address of a sector which is to be written to, and controlling
said at least one Write Pointer (WP) so as to move in a
predetermined order through the physical addresses of the
45 30 memory sectors of any block which is treated as erased and,
when the block has been filled, to move to another of the
erased blocks;
50 and, when a sector write command is received from the host
processor, translating a logical address received from the
35 host processor to a physical address to which data is written

55

5 by allocating for said logical address that physical address
to which said Write Pointer (WP) is currently pointing;
storing in non-volatile solid state memory a table (the SAT)
of logical addresses with respective physical addresses which
10 have been allocated therefor by the controller;
and updating the SAT less frequently than memory sectors are
written to with data from the host processor.

15

Preferred embodiments of the invention will now be described
10 by way of example only and with reference to the accompanying
drawings in which:

20

Fig.1 is a schematic illustration of one block of sectors in a
NAND type FLASH memory, showing three sectors therein;

25

Fig.2 is a block diagram of a memory system comprising a FLASH
chip and a controller chip;

30

Fig.3 is a schematic illustration of one page of data in a
NAND or AND type FLASH memory;

35

Fig.4 shows the structure of a Header field of the page of
Fig.3;

20 Fig.5 illustrates the format of a physical address (PA) of a
page;

40

Fig.6 illustrates a Control Block (CB) entry;

Fig.7 illustrates one entry in a Table Block List (TBL);

Fig.8 shows the format of a MAP entry;

25 Fig.9 shows the format of an entry in the ASB List (ASBL);

45

Fig.10 illustrates the format of a Current Obsolete Block
(COB) Structure;

Fig.11 is a table illustrating the order in which sectors are
written to in a virtual block of a multiple FLASH chip memory
30 system according to one embodiment of the invention;

50

Fig.12 shows the format of a Virtual Address (VA);

Fig.13 shows how the PA is obtained from the VA;

55

Fig.14 illustrates the timing of operations during a multiple
sector write to a multiple FLASH chip memory system according
35 to the invention;

60

5 Fig.15 is a block diagram of a controller chip;
Fig.16 is a table showing allocated memory capacity for a
memory system of the invention;
10 Fig.17 is a flow diagram showing an Address Translation
5 process;
Fig.18 is a flow diagram of the steps carried out at box 58 of
Fig.17;
15 Fig.19 is a block diagram of a multiple FLASH chip memory
system comprising four FLASH chips and a controller chip;
10 Fig.20 is a flow diagram of the steps carried out at box 56 of
Fig.17;
20 Fig.21 is a flow diagram of the steps carried out at box 44 of
Fig.17;
Fig.22 is a flow diagram of a sector read operation;
15 Fig.23 is a flow diagram of a sector write operation;
25 Fig.24 is a flow diagram of the steps carried out at box 161
of Fig.23;
Fig.25 is a flow diagram of the steps carried out at box 207
30 of Fig.24;
20 Fig.26 is a flow diagram of the steps carried out at box 160
227 of Fig.23;
Fig.27 is a flow diagram of a sector delete operation;
35 Fig.28 illustrates the physical partitioning of a page in NAND
or AND type FLASH memory;;
25 Fig.29 is an illustration of an alternative way of arranging
the data in the FLASH page of Fig.28;
40 Fig.30 is an illustration of a yet further way of arranging
the data in the FLASH page of Fig.28;
Figs.31(a) illustrates data in a buffer memory of the
45 30 controller prior to a sector write operation;

Fig.31(b) illustrates data in a FLASH page after completion of
50 a write operation, where the data is arranged according to the
embodiment of Fig.30;

55

5 Fig.32 is a table of controller commands used to transfer the
data from the controller buffer to the FLASH memory during the
write operation of Figs.31(a) and (b);
10 Fig.33 illustrates data in a buffer memory of the controller
5 after a read operation;
Fig.34 is a table of of controller commands used to transfer
the data from the FLASH memory to the controller buffer during
15 the read operation of Fig.33; and
Fig.35 is a schematic block diagram of an erasable block of
10 sectors in NOR type FLASH memory, showing three sectors
therein.
20

Fig.1 illustrates schematically the physical page structure in
25 one block 4 of a FLASH memory array based on NAND type memory
cells. Fig. 1 shows three pages 1, 2, 3 in the block 4. The
page 1 in physical terms comprises one row of memory cells in
a block of memory, the memory being partitioned into many such
30 blocks each comprising a multiplicity of rows of memory cells
20 (i.e. a multiplicity of pages). Each page 1, 2, 3 is treated
as one sector of physical memory space in the FLASH memory
system which will be described and is 528 Bytes wide. Each
35 page 1 in the memory is individually addressable (for
read/write and delete operations), and the pages are erasable
25 in blocks. We will now describe a memory system incorporating
such a memory array. We will later additionally describe
40 memory systems based on AND or NOR type FLASH memory.

Fig.2 shows a memory system 10 incorporating a FLASH memory
45 chip 5 and a controller chip 8. The FLASH memory chip 5
comprises a FLASH memory array 6 and a read/write buffer 7
interfaced to a controller buffer 9 in the controller chip 8.
The controller chip 8 further includes a controller
50 microprocessor 11 and an Error Correction Code (ECC) generator
35 and checker 12. The controller buffer 9 interfaces to a host
55

5 computer processor (not shown) connected to the memory system
10 via an output O/P of the controller chip 8. The controller
chip 8 (hereinafter referred to as the "controller"), controls
the reading and writing of data structures to and from the
10 memory array 6. The host processor 2 connected to the memory
system 10 sends read and write commands to the controller 8.
Data can be accessed by the host in 512 Byte portions or "host
15 data sectors", each of which has a logical sector address
(LA). The controller 8 receives an LA from the host processor
20 and translates this to a physical address as will be described
hereinbelow. In the present case (NAND type memory), each
physical address (PA) defines the physical position of a page
1 of the FLASH memory in the array 6. Each LA is in the form
of one 24 bit field. Accessing a PA using an LA is referred
25 to as address translation and is commonly the most frequent
operation needed on every read/write access. The controller 8
writes data to the memory array 6 in data segments, each
segment being 528 Bytes wide. For each 512 Bytes of data
30 received from the host (e.g. user file or system data) the
controller generates 16 Bytes of data comprising a 4 Byte
Header generated by the microprocessor 11 and a 12 Byte ECC
produced by the ECC generator and checker 12. The controller
35 organises this into a 528 Byte data segment which is written
to one page of the memory array 6, via the FLASH buffer 7.
25 The logical address (LA) of a host data sector is stored in
the 4 Byte Header in the FLASH Sector 1 in which that host
40 data sector is written. On a read operation the data stored in
the relevant sector of the FLASH memory array is read from
array 6, via the FLASH read/write buffer 7, into the
45 controller buffer 9 (and concurrently to the ECC generator and
checker to check for errors in the data), and the controller
reads the 4-byte header to check that the stored LA matches
50 the LA requested by the host computer, prior to allowing the
host computer to read the data from the controller buffer 9.

35

55

5 The Controller 8 manages the physical location of data written
to the memory 6 on an individual sector basis. As will be
later described in further detail, the controller stores in
the memory 6 a BitMap (MAP) of erased blocks and compiles in
10 5 SRAM in the microprocessor 11 a list (the Next Erased Block
(NEB) list) of at least some erased blocks, ordered in
ascending order of block physical addresses, which erased
15 blocks are to be used for writing to. The physical page
location at which a host data sector is written does not
20 depend on the logical address received from the host. Each
Host Data sector is written at an address defined by a cyclic
write pointer. Special write pointers are used for different
types of write operations; host file data writes are performed
at an address pointed to by the Data Write Pointer (WP), host
25 15 system data writes at an address pointed to by the System
Write Pointer (SWP). The Relocation Pointer (RP) is used for
writing sectors which were not directly ordered by a host.
Each of these write pointers has an identical behaviour: each
30 pointer moves sequentially through the pages of a block, then
20 moves to the first page of the next erased block in the Next
Erased Block (NEB) list. Blocks containing files which are
not erased are treated as "not erased" blocks and are skipped
35 when a pointer moves from one block to another (and are never
included in the NEB).

25

Sector Relocation Algorithm

40 When a block 4 of sectors is to be erased, so as to recover
sector space containing obsolete data, sectors must be
relocated from a block containing a combination of valid and
45 30 obsolete sectors to allow the block to be erased. In
principle, the controller 8 allows only one block
corresponding to a specific write pointer to contain obsolete
data sectors at any time. When a sector to be written by the
50 host would produce an obsolete sector in a second block, the

55

5 existing block must first be erased, after relocation of valid
sectors, if necessary.

10 Therefore, numerous erasures and extensive relocations of
5 sectors are unavoidable when a majority of blocks are to
contain both valid and obsolete sectors. This occurs only
when a sequence of sectors written by a host as a part of a
15 file differs from the sequence in which they had previously
been written. This is not the normal case in most
20 applications. However, even in case of normal file write
operation, relocation of non-related data must be performed
from blocks containing a "head" and "tail" of a file. There
is a high probability that relocation of system data
intermingled with "other file" data will cause additional
25 erasure of another block and this will produce more
relocations from this block.

To reduce the total number of relocations and erasures system
30 data is therefore specifically identified and is always
written or relocated at the address of the System Write
20 Pointer (SWP). Information about system data is obtained
during initialisation process and is stored in the
35 microprocessor SRAM 13. It will be generally understood that a
data file is written to the FLASH memory by a file system in
the host computer processor. File data is partitioned into
25 clusters by the file system, where each cluster is a group of
contiguous host data sectors of (typically) 512Bytes. The file
40 system maintains tables and data structures relating to the
attributes of the files stored in the memory and the locations
of the clusters which form each file. These tables and
45 structures are stored in the FLASH memory (as system data) and
are updated by the file system whenever file data is written.
When file data is written to the memory it is accompanied by
50 system data (e.g. for Directory and File Allocation Tables)
35 which relate to the file. System data written in the memory

55

5 commonly includes BIOS Parameter configuration information,
one or two copies of the File Allocation Table (FAT) in which
each entry relates to a specific cluster, the Root Directory,
10 and Subdirectory information. The controller is configured to
5 recognise an operation to write a host system data sector,
thereby enabling it to treat this host data sector differently
to a host data sector of file data. A number of methods may be
15 used either singly or together to recognise system sector
writes, as follows:

- 10 1. System data is written with single sector write commands,
whilst file data may be written with multiple sector write
20 commands.
2. All sectors with LAs below the last sector address in the
file system Root Directory are system sectors. This address
25 can be determined from information held in a BIOS parameter
block stored in the memory by the host file system.
3. All sectors within Subdirectories are system sectors.
Subdirectory addresses and sizes can be identified by reading
30 all Root Directory and Subdirectory entries.
- 20 4. System sectors are often read by a file system immediately
before they are rewritten.

35 For the same purpose file data sectors to be relocated are
written at the address defined by a Relocation Pointer (RP),
25 and are therefore not intermingled with sectors written by the
40 host.

In a modified embodiment of the invention, an additional write
pointer may be provided to point to the location to which
45 30 relocated system data is to be written. This additional
pointer is referred to as the System Relocation Pointer (SRP),
and is always located in a different block to the WP and SP.

50 Block Erase Algorithm

55

5 No arbitrary selection of a block to be erased or scheduling
of background erasure is carried out in the present invention.
Listed erasure of a block containing obsolete sectors is
normally immediately performed when an obsolete sector in a
10 5 second block will result from a host sector write command
which is pending. Similarly, a block is immediately erased
when it contains totally obsolete control data structures as a
result of the rewriting of a control block. (Control blocks
15 are where the controller 8 writes certain control data, and
20 are described in further detail later).

20 Therefore, normally, since relocations cannot produce obsolete
data, there can exist not more than two blocks which contain
obsolete data; the Current Obsolete Block (COB) corresponding
25 15 to the Data Write Pointer (WP) and containing obsolete file
data and the Current Obsolete System Block (COSB)
corresponding to the System Write Pointer (SWP) and containing
obsolete system data. However, temporarily there may exist
30 one more obsolete block of each type. This will occur when a
20 block to be erased (obsolete data has just been created in
another block) at this moment also contains a write pointer of
any type. In this case erasure of such a block (designated
35 Pending Obsolete Block (POB)) has to be postponed until all
erased pages in this block have been used and the relevant
25 write pointer will be moved to another block. At this moment
40 the Pending Obsolete Block is immediately erased.

As aforementioned, erased block identity is maintained in a
BitMap (MAP) spanning the whole FLASH block address space, in
45 30 which MAP the erased state of each block is recorded. Erased
blocks are consumed for sector or control data writing
sequentially in block address order. No background erasure is
carried out. Any block containing one or more bad sectors is
50 treated as a Bad Block and is treated as a "not erased" block
35 by the controller.

55

5

Wear Levelling

10

15

20

The use of cyclic write pointers and single sector write management produces inherent wear levelling in the FLASH memory. However, the algorithm of erasing blocks as soon as they are populated with obsolete or deleted data produces a wear levelling characteristic which is a function of the sequences of sector write operations. If any further wear levelling is thought to be necessary, then separate additional techniques may be incorporated, such as occasional relocation of sectors in random blocks so as to allow these blocks to be erased.

Address Translation Principles

25

30

35

40

The principal address translation means is the Sector Address Table (SAT), which basically is a list of physical addresses of sectors, ordered by logical address. Thus the Nth SAT entry normally contains the physical address for sector with logical address N. The SAT is organised as a number of independent blocks (SAT blocks), and is updated by rewriting individual pages of the SAT blocks. A SAT block may have a dedicated Additional SAT Block (ASB) linked with it to allow modification of individual pages of the SAT Block. SAT pages are not rewritten after each sector write, but on a much less frequent basis, to minimise impact on sector write performance.

45

50

55

Therefore, the SAT will not contain the correct physical address for sectors written since the SAT was last updated. The logical addresses of such sectors are stored by the processor in its SRAM 13 in lists called the Write Sector List (WSL), Relocation Sector List (RSL) and Write System Sector List (WSSL). These lists exactly match the ordering of the sectors themselves which were written by a host or relocated from blocks before erasure. In the case of consecutively

5 written sectors the WSL and RSL entry defines first sector
logical address and the sector group length. The sector
groups cannot jump from one block to another. The
10 microprocessor 11 has knowledge of the starting point of the
5 sector series in Flash memory, and also the order in which
blocks were used for sector writing (special lists,
complimentary to those described above and created in the
15 processor SRAM, namely the Write Block List (WBL), Write
System Block List (WSBL) and Relocation Block List (RBL), are
10 used to store this information and are described in further
detail later), and so can calculate the physical location of
20 the sector.

The WSL, RSL and WSSL (and lists, complimentary to them: WBL,
25 WSBL and RBL) can be recreated by the microprocessor 11 after
removal and restoration of power to the memory system 10 by
reading the logical addresses in the headers of the sectors in
the series written since the last SAT rewrite. The starting
30 sector address in the series and the links between blocks
20 containing sectors in the series is obtained by the
microprocessor from entries in a special data structure called
the Control Block (CB) in Flash memory. (The Control Block
35 (CB) is described in detail later). This method gives high
security of data in the event of unexpected power removal from
25 the card.

40 It will be appreciated that where a System Relocation Pointer
(SRP) is included, as mentioned above, a System Relocation
Sector List (SRSL) and complementary System Relocation Block
45 List (SRBL) are also created and used in a similar manner as
described above with regard to the WSL, RSL and WSSL, and the
WBL, RBL and WSBL, respectively.

50 The organisation of data in a FLASH Sector (i.e. page) 1 to
35 which data is written by the controller 8, according to the

55

5 present embodiment, is illustrated in Fig. 3. The Sector 1
contains a first, 512 Byte information portion 1a which may,
for example, consist of a host data sector, followed by a 4
10 byte header portion 1b, followed in turn by 12 Bytes of ECC
5 1c. As shown in Fig. 4, the Header portion itself comprises a
Data Structure Type portion 20 and a Header Parameter 22 (e.g.
comprising the logical sector address (LA) of a host data
15 sector written in the information portion 1a. The Data
Structure Type can have values representing any one of the
10 following : Data Sector; Deleted Data Sector; Sector Address
Table (SAT) page; Additional SAT Block (ASB) page; Control
20 Block (CB) page; and Boot Block (BB) page.

Deleted Data Sector

25 A deleted data sector physically exists temporarily in the
FLASH memory only in blocks which are permitted to contain
obsolete or deleted sector data i.e. the COB or COSE. It is
identified by the "all zero" state of the Data Structure Type
30 field in the header, or other means as appropriate.

20

Sector Address Table (SAT)

35 The SAT is a series of entries containing the physical address
of logical sectors. Entry N contains the physical address for
logical sector N. Entries within a page occupy the 512 bytes
25 of the information portion 1a of the data stored in the page.
Because a SAT page is written in a single operation, the EEC
40 field may be used to protect the complete page, and separate
ECC fields for each entry in the SAT page are not required.

45 30 The SAT is actually stored as a series of blocks, each block
containing up to 64 SAT pages. A separate data structure, the
Table Block List (TBL), is held in the Control Block
50 (described later) to define the block addresses of all SAT
blocks. Each SAT entry defines the physical address of a
35 sector and occupies 3 bytes and as shown in Fig.5 comprises:

55

5

Chip Number 5 bits, allows 32 chips to be addressed
Block Number 13 bits, allows there to be 8192 blocks per
chip
5 *Sector Number* 6 bits, allows up to 64 sectors per block.

10

The Header Parameter field on a SAT page data structure
15 contains the SAT block and page number.

15

10 A Flash card with capacity 8MB and 8VB blocks can store
approximately 16K sectors, and its SAT will therefore have
20 approximately 16K entries. A 512-byte page within a SAT will
contain 170 entries, and the SAT will therefore occupy almost
96 pages, which will occupy 6 blocks. A SAT for a large FLASH
25 memory card (2GB, 8VB blocks) occupies 1543 blocks.

25

Additional SAT Block (ASB)

30 An Additional SAT Block (ASB) is a dedicated block which may
be linked to a specific SAT block to allow single pages of the
20 SAT block to be modified (i.e. rewritten). There can be
several ASBs, each of which acts as an extension to the SAT
block to which it is linked. When a SAT block is to be
35 modified, it generally contains modified data in only a small
number of its pages. The ASB allows only these modified pages
25 to be rewritten. This is a much faster operation than the
writing of every page and erasure of the obsolete block which
40 is required for rewriting a SAT block. The Header Parameter
portion of an ASB page contains the SAT block to which it is
linked and the SAT page number within that block which it
45 30 replaces. The format of the information portion of data
stored in an ASB page is identical to that of a SAT page.

35

40

45

Control Blocks (CBs)

50 The Controller 8 stores some control and address information
35 in one or more Control Blocks (CBs). The CBs are updated

50

55

5 periodically by the controller 8 and must be accessed during
initialisation of the memory system 10 or occasionally during
operation. Information is stored in the CBs in entries of a
fixed size which can be independently written. There are 9
10 5 entries per page in each CB. An entry relates to one of the
following list of data types, which are identified by a CB
Header field in the entry itself :

- 15 * Table Block List (TBL)
- * Map of blocks with some fields in the entry corresponding to
10 a file data write operation (WMAP)
- 20 * Map of blocks corresponding to a system data write operation
(SMAP)
- * Map of blocks corresponding to a relocate sector operation
(RMAP)

25 15 When new data must be added to the CB, an additional entry of
the appropriate type is added immediately following the last
valid entry. For large cards the CB may occupy more than one
block. The addresses of all CB blocks are stored in a Boot
30 Block (BB) in the FLASH memory 6.

20
For purposes of data security the first page (designated
35 Header Page) of each block of a CB (and the BB also) does not
contain entries and has a full page format of like that of
Fig. 3. The Header Parameter field of the Header 16 of this
25 page consists of a Signature, which identifies the CB, and its
Block Number (which is the block's serial number within the
40 set of Control Blocks).

45 The Information Field 1a of the Header Page of the first CB
30 block is occupied by a Block Link Info data. The Block Link
Info data provides all necessary information to restore the
WBL, WSBL and RBL if the system has to be initialised
50 following a CB rewrite operation. It comprises Link fields
collected from all MAPs (WMAPs, SMAPs and RMAPs) written since

55

5 the last SAT page write operation was performed and has to be
written to the Header Page of the first block of a new CB
during its rewrite operation. The Block Link Info is a timing
10 ordered list of 4 bytes entries, each of which contains a
5 block address of a block being visited by one of the write
pointers and a flag identifying which pointer it was. A
maximum number of blocks in memory space is allowed for CBs
15 and when this becomes full, active entries are rewritten to
the next available erased block(s) from the NEB (i.e. it is
20 compacted and corresponding Header Pages are added), and the
old CBs are erased. (At the same time a corresponding entry
has to be added to the Boot Block.) The information field of
a CB page contains 9 entries of equal length. The EEC field
of a CB page is not used or may be used for other purposes.

15
25 An entry in a CB is 56 Bytes wide and has the format shown in
Fig.6 and comprises a Header 24 identifying the data type of
the entry, an information field 26, and an EEC field 28. The
30 EEC is of the same form as is used for a full page.

20

Table Block List (TBL)

35 The CBs contain the Table Block List (TBL). The TBL contains
the addresses of a group of contiguous blocks of the SAT and
any corresponding ASBs. Multiple TBLs are normally required
25 to define all required SAT block addresses. The relevant TBL
is written immediately after a SAT block write operation or
40 allocation of a new ASB, to record the modified SAT or ASB
block locations. The TBL also contains fields which are used
to record write pointer positions at the moment when a SAT
45 30 page write operation is to be performed. Therefore, a TBL is
also written whenever a SAT page write operation is carried
out. One entry of the TBL is shown in Fig. 7 and occupies the
50 information field 26 of a CB entry. The first 1 Byte of the
TBL entry is No, a sequential number of the TBL entry. Each
35 TBL entry keeps values of 8 SAT - ASB block pairs, in other

55

5 words each TBL entry keeps values for SAT blocks from N to N+7
where N is No*8.

10 WP is the Write Pointer Page field, RP is the Relocation
5 Pointer field, and SWP is the System Write Pointer field.
These fields determine the position of the WP, RP and SWP
within the blocks after WSL, RSL, SSL release operation. WP,
15 RP and SWP are valid only when WSL, RSL or SSL release
operation is complete (i.e. at the last SAT write operation).
10 This condition is set with the Flag bit in the entry Header.
Flag=1 means the entry is the last one written during WSL or
20 RSL release and so WP field is valid.

3 reserved bytes are left for possible future additions.

15 25 SAT-ASB Pairs follow reserved fields, this is an array of 8
entries, each consists of SAT and ASB block addresses.

30 20 SAT_N is the number of the Nth SAT block and ASB_N is the number
of the ASB linked to the SAT_N . If SAT or ASB doesn't exist
the value of this field should be equal to zero.

35 25 New TBL entry should be added to CB each time SAT block is
relocated or new ASB is linked to SAT block.

40 MAP (WMAP, SMAP and RMAP)

The CBs contain various MAP entries. There are three
different types of MAP entry each of which corresponds to a
different type of write operation; WMAP - to a file data write
45 30 or delete operation, SMAP - to a system data write and RMAP -
to a sector relocation operation.

50 35 The information field of all the MAP entries has the same
format. It contains a BitMap defining the erase state of a
group of consecutive blocks. The erase state of 256 blocks is

55

5 defined by the 256 bits in a 32 byte field in the MAP. The
information field contains a Range field identifying the group
of blocks to which it relates. Another field defines the
10 destination block for a transition of the write pointer
5 between blocks. The MAP also contains fields identifying the
location of blocks in flash memory containing obsolete data;
ObsC is used for the COB (or COSB) and ObsP - for the Pending
15 Obsolete Block. If obsolete block is not present
corresponding field is set to 0. The EB field contains an
10 address of a block in which erasure is caused by current write
or delete sector operation. If there is no such block the EB
20 field is set to 0. This MAP entry format is illustrated in
Fig.8.

25 15 When one of the write pointers is moved from one block to
another, a corresponding MAP entry must be added to show the
use of an erased block (the BitMap field is updated) and to
record the link between the blocks (the Link field is
30 updated). When a write (or delete sector) operation produces
20 obsolete (or deleted) data in a new block, a corresponding MAP
also must be added to record a new obsolete block position
(ObsC or/and ObsP fields are updated), to indicate that a
35 block has to be erased (the EB field is updated) and to show
that a new erased block is going to appear (the BitMap field
25 is updated). Therefore, normally, at least two fields of a
MAP are written at the same time and this may be achieved in a
40 single page write operation.

Boot Block (BB)

45 30 The function of the Boot Block is to provide high security of
data in the event of unexpected power removal from the card
and at the same time avoid extensive scanning during
initialisation procedure. For large cards the BB contains
50 more than one block. The BB always occupies the first
35 nondefective block(s) in the card. For purpose of data

55

5 security there is a copy of the BB occupying the next
nondefective block(s) in the card. Both the BB and its copy
must be put to the same place after being rewritten.

10 5 The BB has the same structure as the Control Block and
contains the following types of entries :

- 15 ◦ Signature
- Interleaving Enable.
- Bad Block List
- 10 ◦ Control Block Pointers Table (CBPT)

20 When new data must be added to the BB, an additional entry of
the appropriate type is added immediately following the last
valid entry. The Signature and the BBL entries have just the
same format as described before (of course, the signature
15 field in the signature entry is different and unique). The
Control Block Pointers Table entry contains pointers to all
blocks of the CB and has to be updated immediately after the
CB is rewritten.

30

20 Previous Link (PL)

The CB(s) also contain the Previous Link (PL). The purpose of
the Previous Link is to provide all necessary information to
35 restore the WBL (described later) if the system has to be
initialised following a CB rewrite operation. It comprises
25 Link fields collected from all MAPs written since the last SAT
page write operation has been performed. The PL has to be
40 written only to a new CB during its rewrite operation.

45 Data Structures Stored in SRAM of Controller

30 Various data structures are stored in the SRAM 13 of
microprocessor including the following :

50 Write Sector List (WSL) (or "whistle")

55

5 The Write Sector List records the logical addresses of sectors
written after the last SAT write. Its purpose is to allow
correct logical-to-physical address translation for such
sectors. The WSL has capacity for 128 entries of length 4
10 5 bytes, where each entry stores the logical address of the
first sector in a group of consecutively written sectors and
the group length. The sector groups cannot jump from one
15 block to another. The WSL is empty immediately after a SAT
write.

10 The ordering of logical sector addresses in the WSL exactly
20 matches the order in which they were written. A non-volatile
copy of the WSL in Flash memory is therefore automatically
provided by the headers of the actual sectors written at
15 consecutive locations starting at that defined by the Write
Pointer (WP) at the time the SAT was last written. There is
therefore no need to explicitly make copies of the WSL to
Flash memory 6. If necessary, these sectors can be scanned
30 from a starting address defined in a Control Block field which
20 contains the position of the Write Pointer (WP) at the time
the SAT was last written, along with the Link fields from
subsequent MAP entries.

35 Search of the WSL is performed in reverse order, since only
25 the last entry for any logical sector is valid. Duplicate
earlier entries may not have any corresponding obsolete sector
40 located in Flash memory because sector relocations and block
erasures may have been performed. If preferred, the controller
is configured to simply remove duplicate entries from the WSL.

45 30 Two similar lists, the Relocation Sector List (RSL) and System
Sector List (SSL), are also compiled in the microprocessor
SRAM 13 recording the logical addresses of relocated sectors
50 and System data sectors (written to addresses pointed to by
35 the RP and SWP respectively) written since the last SAT write.

55

5

An ASB and/or a SAT block is supplemented with WSL, RSL or SSL entries every time the WSL, RLS or SSL respectively is full. This procedure is called WSL, RSL or SSL release. This release can cause ASB release if necessary. ASB release occurs when an ASB is full. When an ASB is full the respective SAT block is rewritten and the ASB is erased. Information about written pages in all ASBs should be stored in RAM to avoid frequent ASB scanning. An ASB List (ASBL) for this purpose is stored in the SRAM 13.

10

15

20

The ASBL is a list of all the ASB blocks currently linked with SAT blocks, there being one entry in the ASBL for each ASB. Fig.9 illustrates the format of one entry in the ASBL, where :

15 LWP = number of last written page in this ASB block
NVP = number of the valid pages in ASB block minus one.
ASB Page 0 ... ASB Page n = an array, index is ASB page number, value is corresponding SAT page number.
30 N = pages per block.

25

30

20

Write Block List (WBL)

The Write Block List is complementary to the Write Sector List and is created in microprocessor SRAM 13 to define the blocks within which the sectors in the WSL are located. The WBL is 25 empty immediately after WSL release.

35

40

The WSL and WBL are recreated by a scanning process during initialisation of the memory system. The lists in SRAM which are created in this way will exactly match the lists which 45 30 existed before power was last removed.

45

50

Two similar lists to the WBL named the Relocation Block List (RBL) and System Block List (SBL), are also compiled and stored in the SRAM 13, these lists being complimentary to the 35 RSL and SSL respectively. The RBL and SBL define the blocks

55

5 within which the sectors in the RSL and SSL respectively are
physically located, and are of similar format to the WBL. The
RSL, RBL, SSL and SBL can also be recreated by a scanning
10 process during initialisation of the memory system.

5

Current Obsolete Block (COB)

15 Only one block is allowed to exist which contains obsolete or
deleted sector data written by the Write Pointer (WP). This
is named the Current Obsolete Block (COB). When obsolete or
20 deleted sector data is created in another block, an erase
operation must be performed immediately on the block defined
as the COB. The current block address of the COB and also a
list of sectors which became obsolete or were deleted in this
25 structure hereinafter referred to as the COB Structure. The
COB Structure is established during initialisation by copying
a field containing obsolete block address from the latest MAP
entry, and then adding obsolete sector addresses after the
30 reconstructed WSL and WBL have been analysed and reading
deleted sector addresses from this block. The COB Structure
is updated every time a new delete sector operation is
performed or a new obsolete sector is created, and the block
35 address is copied to a current MAP entry in the CB every time
obsolete data is created in a new block.

25

40 There is also allowed to exist one block which contains
obsolete or deleted sector data written by the SWP, named the
Current Obsolete System Block (COSB). The COSB Structure is
also stored in the SRAM 13 in a similar manner to the COB
45 Structure. The COB and COSB each have the format shown in
Fig. 10, namely a 4 byte Block Number field 28 (this is the
block address of the COB or COSB) and a 32 byte Obsolete or
Deleted Sector Mask 30 which is a BitMap containing ones in
50 positions corresponding to obsolete or deleted sectors inside

55

5 this block. For a block comprising 256 pages this mask 30
takes 32 bytes.

Next Erased Block list (NEB)

10 5 The Next Erased Block list is created in the microprocessor
SRAM 13 in order to provide fast identification of the next
available erased block when incrementing the WP, SWP and RP
15 between blocks. Available erased blocks are used in ascending
order of their physical addresses. The NEB contains M erased
20 block addresses, (e.g. M=8). The NEB list is a list of next
available erased blocks, starting with the erased block with
block address closest to and higher than the address of the
last erased block to be allocated for use. Thus, although the
25 number of entries in the NEB is limited (e.g. to 8), the NEB
itself may contain information about more than the next eight
erased blocks.

The Next Erased Block list is derived from the MAP entry
30 (stored in CB) appropriate to the region of Flash memory
address space being accessed by the write pointers. This will
20 remain in SRAM as the active NEB until all erased blocks
defined by it are used, at which point it must be recreated
35 from the appropriate MAP entries. The CB contains sufficient
MAP entries to define the erase state of every block in Flash
25 memory. Both the NEB and the corresponding MAP entry are
updated by addition and removal of blocks during operation of
40 the memory system. An entry is removed from the NEB when an
erased block is allocated for sector or control data storage.
An entry is added to the NEB (unless the NEB is already full)
45 30 when an erased block is created at a block address which lies
within the range spanned by the other blocks in the NEB.

50 A single NEB entry defines a group erased blocks at contiguous
addresses, and defines the block start address (the first

55

5 block address in the contiguous group of blocks) and the group
length (number of contiguous blocks in the group).

10 TBL Pointers (TBLP)

5 The TBL Pointers identify positions of TBL entries in the CB.
They are also stored in the microprocessor SRAM and are used
to provide fast sector address translation. The TBLP is
15 created during initial scanning of the Control Block, and then
is updated every time a new TBL entry is created in the CB.

10

ASB List (ASBL)

20 As aforementioned, the ASBL is created in SRAM 13 and supports
fast address translation. It identifies the physical
addresses of all ASB blocks and the SAT blocks with which they
25 are associated. It is created during initialisation and has
to be updated each time a SAT page write operation is
performed. The ASBL entries are listed in order of previous
accesses, that is, when an entry is accessed it is promoted to
30 the top of the list. When a SAT block which does not
currently have an associated ASB is accessed, an ASB is
allocated and an entry for it is added at the top of the ASBL.
The bottom entry of the ASBL, representing the least recently
35 accessed ASB, is eliminated.

25 SAT Cache

40 A cache is maintained in the SRAM 13 for 32 contiguous SAT
entries incorporating the entry most recently accessed from a
SAT in Flash memory.

45

Capacity Map

Total Flash memory capacity in the memory system 10 is
allocated to data and control structures as follows :

50

1. Logical Sectors

Capacity is allocated to storage of one valid data sector
35 for each logical address within the declared logical

55

- 5 capacity of the card. This declared capacity is the
available physical capacity minus items 2 to 8 below, and
is defined by the formatter during card manufacture.
- 10 2. Boot Block
5 At least one block is allocated to the boot block.
Preferably, a second block is allocated for storing
another copy of the boot block.
- 15 3. Control Block
A number of blocks are allocated to storage of Control
10 Block entries (Signature, BBL, TBL and MAP). The fully
compacted Control Block occupies less than one block in
20 most cases. Additional blocks are allocated to the
Control Block for entries written between
compaction/rewrite operations.
- 25 4. Sector Address Table
This is the capacity allocated to the storage of the
blocks of the SAT. It is proportional to the logical
capacity of the card.
- 30 5. Additional SAT Blocks
20 A fixed number of blocks are allocated for ASBs to be
associated with defined SAT blocks.
- 35 6. Obsolete Sectors
One block is allocated for the COB. Another block is
allocated for the COSB and one further block is allocated
25 for the POB (for embodiments which allow the existence of
a COSB and POB). The maximum number of permitted obsolete
40 data sectors is therefore set by the number of pages in a
block.
- 45 7. Erased Buffer
30 This is a buffer of erased blocks which must be allocated
for correct operation of the system. At least one erased
block must be allocated for data sector relocation and
one for control structure relocation which may take place
50 concurrently.
- 35 8. Spare Blocks

55

5 Spare blocks may be allocated for use to maintain
declared logical capacity in the event of a failure
during operating life. The number of spare blocks is
determined by the formatter during card manufacture.

10

5

A Capacity Allocation Table illustrating, for example
purposes, capacity allocated to the above items 1-8, for an
8MB card, 64MB Card and 512MB Card (FLASH Cards) is shown in
15 Fig. 16 (this shows only one block allocated for obsolete
sectors, namely for the COB, but it will be appreciated that
more blocks will be allocated for obsolete sectors where there
is provision for a COSB and/or POB in the memory system).

20

SAT WRITE OPERATIONS

25

30

35

40

45

15 The SAT is rewritten when the WSL, WBL, SSL, SBL, RSL or RBL,
is full and, in the majority of cases, it is done with a
granularity of one page, that is, only SAT pages containing
modified sector addresses are rewritten. However, in a small
minority of cases, full SAT block rewriting is required
20 (designated "SAT block write"). When a SAT rewrite is
required, an Additional SAT Block (ASB) is created, into which
only the required pages are written (designated "SAT page
write"). There can only be one ASB dedicated to a specific
SAT block, and totally, there may exist a limited number N of
25 ASBs. The value of N will be chosen as an appropriate
compromise between write performance and capacity overhead
(for example, N=8).

50

55

Each ASB exists until a "SAT block write" is performed on its
30 associated SAT block. A SAT block write is performed when a
SAT write page is required on a SAT block whose ASB is full or
an ASB is to be deallocated. When a SAT page write is
required and the corresponding SAT block does not have an
associated ASB and all N ASBs are already allocated, one of
35 the existing ASBs has to be deallocated before allocation of a

55

5 new ASB. The ASB to be deallocated is selected as the one
which has not been written for the longest time. Note that
deallocation of an ASB is rather time consuming procedure as
it requires writing of a SAT block and also erasure of both
10 the obsolete SAT block and ASB.

Interleaved Chip Access

15 The above described operations and data structures inherently
allow interleaved write operations to be performed on several
10 Flash chips 5 and this can significantly increase performance.
The controller chip 8 may thus control a plurality of FLASH
20 chips 5, for example an array of four FLASH chips,
incorporated in the memory system. The memory space of the
four FLASH chips is organised by the controller as a set of
25 virtual blocks, each of which consists of four blocks 4, one
block from each of the four chips 5 (which chips are
permanently linked together). The four blocks in each virtual
block are the blocks with the same block address within a
30 chip. This organisation allows linked blocks forming a virtual
20 block to be treated as one large block and, therefore, to use
all the same described algorithms and data structures as
previously described for a single FLASH memory system 10,
35 using virtual blocks instead of the individual erasable blocks
of sectors. Page write operations are performed concurrently
25 across all interleaved chips. The Write Pointer (WP) and the
RP and SWP each move sequentially through page addresses, and
the ordering of address bits supplied to a hardware chip
enable decoder provided in the controller 8 ensures that pages
40 of linked blocks are sequentially accessed in the order
45 30 illustrated in Fig. 11 i.e. each pointer WP, SWP, RP moves
from one chip to another when moving from one PA to another
PA. This is achieved by the use of Virtual Addresses. A
50 unique Virtual Address (VA) is allocated to each physical
sector (e.g. in the above-described NAND based memory system a
35 VA is allocated for each page) in all of the chips. The

55

5 Virtual Addresses are allocated such that incrementing the VA
by one moves the write pointer from one chip to the next chip,
the Virtual addresses incrementing from chip to chip, though
10 linked blocks of each virtual block, in a repeating pattern as
shown in Fig. 11.

The controller in effect treats the array of memory chips as a
15 single column of virtual blocks. The Virtual Address of a
sector takes the format shown in Fig. 12. This consists of a
Virtual Block portion comprising a ChipHigh portion and the 13
20 bit Block address of the sector, and a Virtual Page portion
comprising the 6 bit page address and a ChipLow portion. The
ChipHigh portion is C_{high} bits of the 5 bit chip number (of the
physical address - see Fig. 5) and the ChipLow portion is C_{low}
25 bits of the 5 bit chip number, where:

C_{high} = column number of chip in array of chips; and
 C_{low} = row number of chip in array of chips.

To obtain the Physical Address (PA) from the Virtual Address
30 (VA), the controller simply re-organises the VA so as to move
the ChipLow portion back between the ChipHigh and Block
address portion, as shown in Fig. 13. Thus, it will be
appreciated that in a single chip memory system for any sector
35 the VA is equal to the PA.

25 For simplicity, only a number of chips which is a binary
multiple may be interleaved, for example, 2 or 4. Erase
operations on virtual blocks are performed as concurrent
erasures of all linked blocks of interleaved chips. If a block
40 in a chip is a bad block the controller treats all the blocks
having the equivalent block address in the other chips as bad
45 blocks.

Interleaving is enabled or disabled according to the status of
50 a control byte in the Boot Block which is set in enabled or

55

5 disabled status by the manufacturer of the memory system when
the FLASH memory is formatted.

10 Where block addresses or PAs were previously used in the
5 above-described single chip NAND type FLASH embodiments, we
now use Virtual Block addresses and VAs, respectively. On
receipt of a host data sector write command the controller
15 translates an incoming LA to a PA by allocating the PA to
which the relevant write pointer is pointing. The controller
10 controls the write pointers to each move through the PAs so as
to, in effect, move sequentially through the Virtual Addresses
20 (VAs) of the sectors of those of the virtual blocks which are
erased (which erased virtual blocks are identified in the
NEB).

25 Fig. 14 illustrates the timing of the various operations
involved in a multiple sector write to interleaved chips.
Fig. 14 will now be described with reference to Fig. 15 and
30 Fig. 19. Fig 15 is a block diagram illustrating in detail the
controller chip 8 of the memory system. (The controller chip
8 of Fig. 2 may be of the form shown in Fig. 15 and like parts
in both Fig. 2 and Fig. 15 are numbered with the same
35 reference numerals). Fig. 19 is a schematic diagram of a
memory system 10 comprising the controller chip 81 and four
25 FLASH chips 1¹, 2¹, 3¹, 4¹ each with its own read/write buffer
71¹, 72¹, 73¹, 74¹.

40 Fig. 15 shows the controller chip 8¹, with : an input/output
port O/P e.g. a PC Card ATA Port, Compact Flash or IDE Port,
45 for connection to a host computer; a Host Interface &
Registers 80 connected to O/P and a Dual-Port SRAM Sector
Buffer 9 connected thereto; A Datapath controller 82, the ECC
50 Generator & Checker 12, and a Flash Memory Interface (FMI) 84,
all connected to the Sector Buffer 9, the FMI also being
35 connected to a Flash Memory Port 86. The controller 8 also

55

5 includes microprocessor 11 (in the form of a RISC Processor);
processor SRAM 13,; a processor Mask ROM 88; and a port for an
external Program RAM or ROM 92. An optional Debug Port 94
10 may also be provided for the RISC processor 11. Data and
5 commands are communicated between the various components of
the controller 8¹ (except for the Sector Buffer 9) via a
microprocessor Bus 91.

15 As shown in Fig. 14, when a multiple sector write command (in
10 this case a 4-sector write comprising Host Data sectors 1, 2,
3 & 4) is received at the ATA Port O/P, Sector 1 is written
20 into a Buffer 1 of the Dual-Port Sector Buffer 9. This leaves
a Buffer 2 of the Sector Buffer 9 available for controller
data management operations. When Sector 2 is received it is
25 written directly into Buffer 2 and at the same time Sector 1
is moved from Buffer 1 to the Flash Memory Port 86 from where
it is written into the read/write buffer 71¹ of one of the
four FLASH chips (chip 1¹). Sector 2 is then sent from buffer
30 2 to the Flash port 86 and on to the read/write buffer 72¹ of
one of the other four FLASH chips (chip 2¹). While this is
happening Sector 3 is received directly into Buffer 1 of the
Sector Buffer 9. Sector 3 is written to the buffer 73¹ of
35 Chip 3 and Sector 4 is received into Buffer 2 of the Sector
Buffer 9, and is then written to the buffer 74¹ of chip 4.
25 Sectors 1,2,3 and 4 are then written into the relevant
allocated physical sectors in the memory arrays 61¹,62¹,63¹,64¹
40 of chips 1¹, 2¹, 3¹ & 4¹ respectively. Although Fig. 14 shows
each such sector Write operation starting shortly after the
previous one, in practice, to all intents and purposes, it
45 will be appreciated that the four Sectors 1,2,3,4 are written
substantially concurrently to the Flash Chips 1¹ to 4¹.
Moreover it will be appreciated that the physical addresses of
the sectors to which the Host Data Sectors 1 to 4 are written
50 are determined by the algorithms previously described which

55

5 determine the position of the relevant write pointer (i.e. sequential use of Virtual Addresses).

10 It will be appreciated that where a multiple-sector write command of more than four sector writes is sent from the host processor to the controller, the controller partitions the multiple-sector write into groups of (for the present embodiment using four memory chips) four sectors, each such group to be written to FLASH memory in an interleaved write
15 sequence as described above with reference to Fig.14.

20 Address Translation

The process of address translation for an LA to a VA will now be described in further detail, with reference to read and
15 write operations.

Address translation is an operation performed on the logical address of a sector which returns one of the following three
30 values:

- 20 * Valid sector physical address
- * Information that logical sector has been deleted (or has never been written)
- 35 * Information that an error condition has occurred

25 Fig. 17 is a flow chart illustrating the address translation process. This process is carried out for every read operation. When a logical sector address (LA) to be translated is received by the controller 8¹ from the host processor an algorithm is implemented (box 40) to identify the
40 possibility that a sector whose logical address is to be translated has previously been written or deleted since its SAT entry was last written. A conclusion that a sector has not been previously written or deleted must be 100% certain; a
45 conclusion that a sector may have been previously written or deleted should have a high probability of being correct. This
50
55

5 is carried out by the processor by keeping a limited number of
pairs of values representing the start and end addresses of
contiguous sequences of sectors, and identifying whether an
address to be translated lies within any of these ranges.
10 5 These ranges may eventually become non-contiguous, leading to
some uncertainty in a conclusion that a sector may have been
previously written or deleted. If the LA lies within any of
the ranges then we answer "IS Repeat Possible?" (box 42 of
15 Fig.17) with YES. If the LA does not lie within the ranges we
answer NO and go to the SAT or SAT Cache to find VA (box 44).
From here we determine whether the physical sector is Bad (46)
or Deleted (50). If the LA corresponds to an unwritten sector
20 this results in VA=Deleted (58) at box 50. If we answer Yes
at box 42, then we search 52 the WSL or SSL (depending on
25 whether the LA corresponds to file or system data). If at box
54 the LA is found (YES) the VA is calculated 56 and the
logical address stored in the header 1b is of the physical
sector is read by the controller microprocessor (at 58). If
30 LA=LAI (box 60) then the calculated VA is correct. If the LA
is not found at 54 then we search 62 for it in the RSL and if
it is not found in the RSL we go to the SAT or SAT Cache and
get the VA from there 44. If the VA found in the SAT or Sat
35 Cache is not Bad or Deleted 46, 50, then we get LA1 from the
VA 58 and check if LA=LAI, as before.

25

40 The process steps carried out at box 56 (Calculate VA) of
Fig.17 are illustrated in detail in the flow diagram of Fig.
20. Fig.20 illustrates the steps carried out in order to
obtain the VA for an LA found in the WSL or RSL, for a memory
45 system using only two write pointers, WP and RP. It will be
appreciated that this flow diagram would be extended to also
allow the VA for an LA found in the SSL, where the memory
system also incorporates a System Write Pointer (SWP). The
50 process starts at box 100 where we set NumFromEnd (NFE), where
35 NumFromEnd = number of sectors written beginning from the end

55

5 of the WSL (or RSL) up to the given sector (found in the WSL
 or RSL. If the LA was found in the WSL we set P=WP and if LA
 was found in the RSL we set P=RP; then we set PG= P.Page where
 P.Page is the page of the write pointer indicated by the value
 10 5 of P (see box 102). If $104 PG > NFE$ (i.e. LA is in the last
 written block) then $VA = P - NFE - 1$, namely NFE-1 sectors away
 from the position of the relevant write pointer. If $PG < NFE$
 15 then we determine 106 if $P = 0$, namely if a block corresponding
 the last WBL/RBL entry is fully written. If it is (i.e. $P = 0$)
 10 we set $NotLast = 0$, and if it is not we set $NotLast = 1$. We then
 calculate 108 number of blocks, Nblock, between the last one
 20 and the block where the given sector lies, using the following
 algorithm:

Nsect is a number of sectors between last written block page 0
 25 and the given sector;
 $Nsect = NumFromEnd - PG$;
 $Nblock = Nsect / BlockSize + NotLast$

30 We then calculate 110 page number in a block, PageNum, where
 $20 PageNum = BlockSize - Nsect \% BlockSize$. If LA is in the WSL we
 then 112 get Block Address (BLAddr) from the WBL, or if LA is
 in the RSL we get 114 Block Address (BLAddr) from the RBL,
 35 where Block Address is a Virtual address of a block containing
 the given sector, using the following:
 25 If LA is I WSL, then $BlAddr = RBL[LEBL - Nblock]$, where LEBL is
 an index of the last entry in the WBL;
 40 If LA is in RSL, then $BlAddr = RBL[LRBL + Nblock]$, where LRBL
 is an index of the last entry in the RBL.

45 30 Then we calculate 116 the VA using: $VA = Page0 + PageNum$, where
 $Page0 =$ Virtual address of page 0 in the block containing the
 given sector.

50 Fig.21 is a flow diagram of the process used to Get VA from
 35 SAT or SAT Cache (box 44 in Fig.17). Fig.21 is generally self-

55

5 explanatory but further comments regarding specifically
labeled boxes is given as follows:

10 Box 120 (Is LA in SAT Cache): LA is in the SAT Cache if $LA \geq$
5 FirstCacheEntry.LA $\leq LA < \text{FirstCacheEntry.LA} + \text{CacheSize}$,
where FirstCacheEntry.LA = the LA corresponding to the first
SAT Entry in the Cache, and (global) CacheSize = number of
15 entries in the SAT Cache;

20 10 Box 122 (Calculate Block and Page in SAT): We calculate SBNum
which is a SAT block number for the given LA, and Spage which
is a SAT page number for the given LA;

25 Box 124 (Calculate number of TBL entry): TBLNum is a number of
the required TBL Entry, where $TBLNum = SBNum/8$; and

30 Box 126 (Store part of SAT page in cache): If it is possible,
32 entries starting with the last entry accessed are cached.
If there are not enough entries then a group of 32 entries
20 ending with the last entry in a page and including the last
sector accesses is cached.

35 Fig.18 is a flow diagram illustrating the process steps for
Box 58 (Get LA1 from VA) of Fig.17. It should be noted that
25 the Header Parameter (HP) stored in the Page Header will be
the value of the logical address (LA) incremented by one. This
40 is because deleted sectors are marked by setting all bits in
their headers to zero. This LA=0 cannot be stored in a header.
We therefore set $LA1 = HP - 1$.

45 30 Read Operations

50 Fig.22 is a flow diagram illustrating the sequence of steps
carried out to read a host data sector from a physical sector.
The controller starts by translating the LA (received from
35 the host) to a VA (box 130). This is done by carrying out the

55

5 process illustrated in the Fig.17 flow chart, already
described. Once the LA has been translated to a VA, the
content of the physical sector with address VA is read into
the buffer 9 of the controller. The controller then checks
10 (box 132) if the sector (i.e. the content of the sector) is
deleted or bad. If it is deleted or bad, the controller sets
all bytes of the host processor's data buffer to 0xFF (box
134). If the sector is not a deleted sector (box 136), the
15 controller returns error status to the host (box 138). If the
20 sector is a deleted sector, the controller returns valid
status to the host (box 137). If at box 132 the controller
determines that the sector is not deleted or bad, the
controller goes straight to box 137 i.e. returns valid status
to the host.

15

Write Operations

25 Fig.23 is a flow diagram illustrating the sequence of steps
carried out to write a host data sector to a physical sector.
30 Fig.23 deals only with write operations for host file data,
and thus written by the Write Pointer (WP), but it will be
appreciated that the operations of Fig.23 would be
appropriately extended to deal with separate system data
35 writes where the memory system uses a separate write pointer
for system data (i.e. the SWP). The controller starts by
25 translating the LA (received from the host) to a VA (box
150). This is done by carrying out the process illustrated in
the Fig.17 flow chart, already described. If 152 the sector is
Bad, the controller returns error status 154 to the host. If
the sector is not Bad, then check if the sector is deleted 156
45 and if it is deleted then check if WP is valid or invalid 158.
WP is invalid (WP==0) when a full block has just been written
and WP has to be moved to an erased block. If WP is not valid,
we set the WP to a new (valid) physical sector address 160.
50 When WP is valid, we add the LA to the WSL 162 and perform any
35 WSL or RSL release, and/or CB and CBPT compaction, which is

55

5 necessary. We then update 164 the ranges from the Evaluate
Repeat Possibility algorithm (box 40 of Fig.17) and write the
sector 166 from the controller buffer to the address of the
WP, and return a valid status value 168 to the host. If, at
10 5 box 156, we find the sector is not deleted we then check 157
if the VA is in the COB. (The VA is in the COB if VA coincides
with VB, where VB is a Virtual Block Number field (this is the
Virtual Block address - see Fig.12) in the COB structure
15 stored in SRAM 13. If the VA is in the COB, we record VA as
20 obsolete 159 in the COB structure stored in the controller
SRAM 13 (this is done by setting a corresponding bit to 1 in
the BitMask field of the COB Structure in SRAM 13.) and then
go on to box 158 (is WP valid). If the VA is not in the COB we
change the COB 161 and then move on to box 159 (Mark VA as
15 Obsolete).

25 Fig.24 is a flow diagram of the steps implemented at box 161
of Fig.23 (Change the COB). Fig.24 is generally self-
explanatory but should be read in conjunction with the
30 following notes:

35 Box 200 (VA.B1, VB): VA.B1 is a Virtual Block field of VA, and
VB is as above-described;

40 Box 202 (Is COB invalid): COB is invalid if VB=0. If VB is
25 equal to zero this indicates that there are no obsolete data
at this moment;

45 Boxes 203,204 (Calculate MaxRel): MaxRel is a maximum number
of sectors to be relocated from COB. MaxRel=P.Page-1, where
P.Page is a Page field (address) of the WP or RP;

50 30 Boxes 205,206 (Add Dummy Entries to WSL): If a block to be
relocated is not fully written yet, corresponding "dummy"
sector LAs must be added to the last WSL (RSL) Entry;

55 Box 207 (Relocate Sectors): See Fig.25;

5 Box 208 (Write WMAP): Write WMAP to the CB where EB+VB and
 corresponding bit in the BitMap is set to 1. Perform CB
 rewrite if necessary;

10 Box 209 (Update Lists): Find WRBArray entry equal to VB and
 mark it and any other entries for the same VB in WRBArray as
 invalid. The WRBArray is in fact the WBL and WSL lists which
 are actually stored in the same area of memory with the WBL
15 entries at the start counting up and the RBL entries at the
 end counting down. The WRBArray is full when the two lists
10 meet in the middle.

20 Box 210 (Setup COB): Update COB Structure in SRAM, VB field is
 set to VA.Bl, Obs and Del Mask bit corresponding to VA.Page is
 set to 1, all other bits are set to 0.

25 Fig.25 illustrates the steps implemented at box 207 (Relocate
 Sectors) of Fig.24. This should be read in conjunction with
 the following notes:

30 Boxes 220, 222 and 230: Perform a loop going through Obs and
 Del Mask field of the COB Structure in SRAM;

20 Box 223 (Is Sector Valid?): Sector is valid if ODMask[i]=0;
 zero value in COB Obs and Del Mask indicates that this page
 contains a valid sector;

35 Boxes 224 and 225: RP=0 if a block pointed to by the RP is
 already fully written;

25 Box 226 (Store LA from Page Header): LA got from Page Header
 is temporarily stored to be used in Add Entry to RSL;

40 Box 225 (RP=0): Add Entry to the RSL. Perform WSL/RSL release,
 if necessary.

45 Fig.26 illustrates the steps implemented in order to set the
 Write Pointer (WP), at box 160 in Fig.23. A similar process is
 used to set the RP at box 227 in Fig.25. Fig.26 should be read
 in conjunction with the following comments:

50

55

5 Box 240 (Is WRArray not full): WRArray is not full if
Last<LastRE-1, where Last (global) is an index of the last WSL
entry, and LastRE is an index of the last RSL Entry;
10 Box 242 (Is WRBArray not full): WRBArray is not full if
5 LBL<LRBL-1, where LBL and LRBL are as previously defined with
regard to Fig.20;
 Box 244 (Lists Release): Perform SAT Page Write operation, SAT
Block Write and CB rewrite if necessary;
15 Box 246 (Fill NEB): Select next N (N=NEBSize) erased blocks
10 from MAPs stored in the CB;
 Box 248 (Write WMAP): Write WMAP with Link field set to
ErBlock and corresponding bit in BitMap field set to 0.
20 Perform CB rewrite if necessary.

25 Fig.27 is a flow diagram of the process steps carried out in a
delete sector operation. This is in the initial steps similar
to a write operation (see Fig.23): the LA is translated 250 to
a VA. If 252 the sector is deleted we return error status to
the host 254. If the sector is not deleted we check 256 if the
30 20 VA is in the COB and if it is not 261 in the COB we Change the
COB 261 (same as Fig.24). If VA is in the COB we Mark VA in
the COB as deleted 269, then we fill one of the buffers of the
35 Dual port SRAM 9 (in the controller) with zeros 271, and then
write this "all zeros" page from the buffer to the sector to
25 be deleted (thereby deleting the sector). We then return valid
status 275 (confirming sector has been deleted) to the host.

40

Initialisation

 Initialisation procedures and power-loss recovery procedures
45 30 will now be described. For simplicity, these are described
with reference to a single write pointer system (i.e. only WP)
but it will be appreciated that the procedures will be readily
extended as appropriate for the multiple write pointer system
50 (WP, SWP, RP).

35

55

5 All data and control structures are specially constructed to
avoid generalised scanning during initialisation. Almost all
control structures (except WSL and WBL) normally are derived
10 from corresponding information stored in the CB. During
5 initialisation of the card, it is necessary to perform
following operations.

1. To read the last Control Block Pointers Table entry from the
15 Boot Block and so identify the CB block(s) locations.
2. To reconstruct the TBLP by scanning the CB.
- 10 3. To scan header/ECC fields of pages sequentially following
the write pointer position defined in the last TEL entry in
20 the Control Block to identify sectors written since the last
SAT rewrite and to construct the WSL and WBL.
4. To construct the NEB from corresponding MAP entries in the
25 CB.
5. To construct the COB and ASBL.
6. To check if a block referenced in the ErB field of the last
MAP is really erased. If not, to complete erase operation.

30

20 Construction of WSL and WBL

During initialisation of the card, the last value of the Write
35 Pointer (WP) to be stored is read from the latest TBL entry in
the CB and a scan of page headers at successive pages from
that location is performed to rebuild the WSL and WBL in
25 processor SRAM. When an erased location is encountered, the
end of the sequence of sectors written since the last SAT
40 rewrite has been reached.

This sector scan must take account of the fact that the Write
45 30 Pointer (WP) may jump from the end of a block to the beginning
of a non-adjacent block. All block transitions made by the
(WP) are recorded in the Link fields of MAP entries in the CB.

50

Construction of COB and ASBL

55

5 These structures can be reconstructed by copying corresponding
 entries from the CB. In addition, to construct the COB
 structure (to identify deleted sectors in it) it is necessary
 to scan a current block containing these sectors, whose
10 address is defined in the Obs field of the last MAP entry in
 the CB. To identify obsolete sectors in this block, it is
 also necessary to scan WSL and WBL. In order to record ASBL
15 pages we have to identify ASB addresses from the TBL and then
 to scan their header/ECC fields.

10

Power-Loss Recovery

20 It is a requirement for the memory system that it should be
 able to operate normally, and that no stored data should be
 lost, when power is restored, whatever the circumstances under
25 which power has been removed. However, it is not necessary to
 restore the full normal state of the memory system immediately
 after power-on, only to allow it to operate normally... A
 normal state can be restored later as an exception, whenever
30 any abnormal state is detected.

20

 The normal state of the Memory System may be degraded if the
 supply voltage is removed whilst any of the following
35 operations is being performed.

1. Writing of a data sector from a host
- 25 2. Writing of a data sector which is being relocated
3. Writing of an entry to a control data block (CB or BB)
- 40 4. Writing of a page to a control data block (SAT or CB)
5. Erasure of any block with obsolete sector or control data

45

30 Power loss during writing of a data sector from a host

 In this case, the data being written may be lost, but the host
 had not been informed that the write command had completed and
 may write the sector again. An incompletely written sector
50 may exist in Flash memory as a result of the partial write
35 operation. This is detected during initialisation when the

55

5 value of the Write Pointer is established by reading the page
headers in the block defined by the last Link parameter in the
CB. The last detected sector should read fully to check its
10 ECC, and the next page should be read to check that it is
5 completely erased. If a partially written sector is detected,
all other sectors should be relocated to a new Write Pointer
position at the beginning of the next erased block, then the
15 block should be erased.

10 Power loss during writing of a data sector which is being
relocated

20 This is detected during the process of establishing the Write
Pointer during initialisation, as above. The same action of
relocating sectors and erasing the block should be taken. In
25 addition, an incomplete relocation operation should be
detected by comparing logical sector addresses immediately
preceding the Write Pointer with those of obsolete sectors in
the block defined by the Obs parameter in the CB. Any pending
30 sector relocations should be completed during initialisation.

20 Power loss during writing of an entry to a control data block
(CB or BB)

35 This condition may be detected during normal initialisation
when entries in the CB and BB are read and their ECCs are
25 checked. The entry should be ignored if its ECC shows an
error. The earlier operation which initiated the CB or BB
40 entry write operation had not completed correctly and the
entry will later be written correctly when this operation is
repeated during normal operation.

30 Power loss during writing of a page to a control data block
(ASB)

50 This condition may be detected during normal initialisation
when pages in the ASB are read and their ECCs are checked.
35 The page should be ignored if its ECC shows an error. The

55

5 earlier operation which initiated the ASB page write operation
has not completed correctly and the page will later be written
correctly when this operation is repeated during normal
10 operation.

5

Power loss during writing of a full control data block
(SAT or CB)

15 This will result in an incomplete control data block existing
in Flash memory, with no references to it by other data
10 structures. This condition need not be detected during
initialisation, and the block may be allowed to exist as a
20 "lost block". The earlier operation which initiated the block
write operation had not completed correctly and the block will
later be written correctly when this operation is repeated
15 during normal operation. At a later stage of normal
operation, the lost block will be detected by a discrepancy
with its MAP state, or by the discovery of a discrepancy in
the number of erased blocks in the system (see Capacity Map in
30 Fig. 16). Exception routines may then identify and erase the
20 block, by full FLASH memory scanning if necessary.

35 Power loss during erasure of a block with obsolete sector or
control data

This will result in an incompletely erased block existing in
25 Flash memory. This condition is detected during
initialisation when the state of the block referenced by the
40 ErB field in the last MAP entry in the CB is checked. Re-
erasure of this block can be performed, if necessary.

45 Further Alternative Embodiments

Various modifications to the above-described embodiments are
possible without departing from the scope of the invention.
50 For example, one alternative way of handling erasure
operations is to always allow two COBs (and two COSBs) to
35 exist: the advantage of this would be to make the best use of

55

5 memory capacity. In the above-described embodiment, we only
allow one COB, but also allow a POB to exist temporarily when
there is a write pointer in a block which we wish to make the
COB. This means that there must at all times be enough erased
10 memory capacity to allocate for a POB, should it be necessary
to have a POB. It therefore is attractive to make the best use
of this memory capacity and one way of ensuring this is to
15 always allow two COBs to exist, therefore eliminating the need
for a POB (the second COB can act as a POB, when required). In
20 such a two COB system, when it becomes necessary to create a
new COB we erase the older one of the two COBs (unless it has
a write pointer in it in which case we erase the younger one).

With reference to Fig.3, and the description of the
25 arrangement of data in each of the FLASH pages in the memory
system, we also propose some alternative ways of storing the
data within a page. Fig.28 shows the physical partitioning of
a typical 528 Byte NAND or AND type FLASH memory page 1. The
30 page comprises a 512 Byte "data area" 300 and a 16-Byte "Spare
area" 302. In the embodiment described above with reference to
Fig.3, the controller 8 stores 512Bytes of information 1a
(e.g. one host data sector) in the Data Area 300, and stores
35 the Header 1b and ECC 1c (together referred to hereinafter as
Overhead Data (OD)) in the Spare Area 302. However, other
25 arrangements of data within the page 1 are possible. For
example, as shown in Fig.29, the Header 1b and ECC 1c could
40 equally be stored in a first portion 303 of the Data Area 300,
and the Information 1a stored in the portion 304 consisting of
the Spare Area 302 and the remainder of the Data Area 300.

45 30 Another possibility, shown in Fig.30, is to write the Header
1b and ECC 1c at a position offset from the start of the FLASH
page, and to write the Host Data Sector (which may be referred
50 to as the "user data") in the remaining space either side of
35 the Header and ECC. By how much (Offset S) the OD is offset

55

5 may, for example, be determined by a function of either: (a)
the physical address (PA) of the page 1; or (b) one or more
bits within the first byte of user data (i.e. the host data
sector) written to the page 1. Fig.31(a) illustrates the
10 arrangement of data in the controller buffer 320 before the
start of a sector write operation, the data being arranged as
a first portion 322 of user data and a second portion 324 of
Header data. Fig.31(b) shows the arrangement of the data in a
15 FLASH memory page, following completion of the write operation
in which the offset S is determined by one or more bits within
the first byte of user data (option (b) above). The data is
stored as a first portion 326 of the user data, followed by a
20 second portion 328 of the user data, followed by the Header 1b
and ECC 1c, followed by the third and final portion 330 of the
15 user data. The length of portion 326 + portion 328 is
dependent on data within portion 326. The length of portion
326 is defined to be less than or equal to the minimum offset,
and the length of 328 is calculated on the basis of data
30 within portion 326 to provide the correct Offset S . The first
and second portions 326, 328 of user data are separately
identified so that the first portion 326 may be read from the
FLASH memory by the controller in one operation, and evaluated
35 by the controller in order to determine the length of the
second portion 328 which should be read by the controller in a
25 subsequent operation Fig.32 is a table detailing the sequence
of controller commands used to transfer the data from the
controller buffer to the FLASH memory during the write
40 operation.

45 One advantage of choosing the offset S to be a function of one
or more bits of the user data is that the overhead data is
therefore not inserted at the same position in the 528Byte
data segment in every sector. This protects the valuable
50 overhead data from simultaneous loss on a large number of

55

5 sectors in the event of a systematic failure in FLASH memory,
such as a memory array column failure.

Fig.33 shows the resulting arrangement of data in the
10 controller buffer after completion of a read operation on the
FLASH memory page of Fig.31(b). From Fig.33 it will be seen
that the data in the buffer is arranged back to a first
15 portion 322 of user data and a second portion 324 of Header
data, now followed by a third and final portion 325 of ECC.
20 Fig.34 is a table detailing the sequence of controller
commands used to transfer the data from the FLASH memory to
the controller buffer during the read operation.

25 Additionally, with reference to interleaved write operations
to multiple FLASH chips, as described already with reference
to multiple FLASH chip memory systems, we also propose that
this technique for writing substantially concurrently to a
30 plurality of chips may also be used for writing data to a
single memory chip in which the physical page size is a
multiple of the size of a sector write by the controller e.g.
each page of the memory is four times the size of a segment of
35 (user + overhead) data written by the controller, where the
controller writes data in uniformly sized segments.

40 It will further be appreciated that the invention is
applicable not only to NAND-type memories but also to other
types of memory, such as AND and NOR type memories. In the
case of AND type FLASH memory, each page 1 of a block has the
same format as the NAND page format of Fig.28 and we can use
45 any of the possible arrangements of data within the pages as
afore-described. We design the controller to still erase
memory in blocks of sectors, although in blocks containing bad
sectors the individual good sectors in the block to be erased
50 will be erased individually. Thus, the controller does not
35 treat any blocks containing bad sectors as bad blocks, but

55

5 instead treats them as good (erasable) blocks and makes use of
the good sectors within those blocks. In AND type embodiments
though we ensure that the controller only uses blocks
10 containing all good sectors for SAT blocks or ASBs.

5
Where AND type FLASH memory is being used and the memory
system is a multiple FLASH chip system utilizing interleaved
15 chip write operations as described above, where any block of
sectors (pages) in one of the virtual blocks contains a bad
20 sector, the controller causes the write pointers to skip this
sector and go to the next good sector in the block e.g. where
c=chip and s=sector, if a burst of four sector writes is c3s5,
c4s5, c1s6, c2s6 then if c1s6 is a bad sector the sequence
becomes c3s5, c4s5, c2s6, c3s6. This is in contrast to
25 15 embodiments based on NAND type memory, where if one block in a
virtual block contains one or more bad sectors the controller
treats that block as a bad block and treats the whole virtual
block as a bad virtual block.

30
20 Where we use NOR type FLASH memory, our preferred embodiment
is one in which we design the controller of the memory system
to still read and write data structures to and from the FLASH
35 memory in uniformly sized sectors, each sector being 528 Bytes
wide. Fig.35 illustrates schematically three such sectors
25 1,2,3 in a block 4' of NOR memory. Due to the fact that one
row of memory in a NOR block is only 512 Bytes wide it will be
40 appreciated that each of our sectors in NOR therefore fills
one row and wraps round to fill a part of the next row.
Nevertheless, it would be possible to define our sectors in
45 30 NOR memory in a different manner, and we may choose to use
sectors of smaller or larger size than 528 Bytes, and a block
could even contain sectors of more than one size. The
controller may arrange the data within each sector in any of
50 the various different ways already described with reference to

55

5 NAND and AND type memory pages, each sector including user and
overhead data.

10 It will be appreciated from the foregoing that the physical
sectors of the memory system, whether the memory system is
based on NAND, AND or NOR type memory arrays, need not have
any particular relationship to the physical architecture of
15 the memory array itself, for example a sector need not
correspond to a row (page) of the array, and no physical
10 feature need be present to identify the boundary between one
sector and a neighbouring sector. A sector can be interpreted
20 as a group of memory cells which are always treated by the
controller as a unit. Different sectors need not be of the
same size. Moreover, the physical structure of a sector has no
25 dependence on data which might be stored in the sector. Also,
embodiments are possible in which defective regions within a
row (page) of memory cells are tolerated and are simply
skipped over by the controller when writing to physical
30 sectors.

20 With reference to the SAT, while as above-described the SAT is
preferably stored in one or more blocks of the memory array 6,
35 it would alternatively be possible to provide in the memory
system 10 a separate non-volatile memory which is accessible
25 to the controller, in which separate memory the controller
40 stores the SAT.

Finally, in a modified version of the above-described
45 embodiment, instead of always using available erased blocks in
30 ascending order of their physical addresses as above-
described, the controller uses the erased blocks in another
order. In this modified embodiment, the NEB list contains a
50 chosen subset of all the currently available erased blocks,
the first block address in the NEB list is the next erased
35 block to be used, and this first block address is removed from

55

5 the NEB list when it has been allocated for data storage use.
Any new erased block which is created (e.g. due to creation of
10 obsolete data, following a delete command from the host) is
added to the bottom of the NEB list. This continues for a
5 period determined by the controller (which could be a
predetermined number of sector write commands from the host,
for example), at the end of which period the controller re-
15 compiles the NEB list by replacing the entries in the NEB with
a new subset of the currently available erased blocks.
10 Conveniently, subsets of the whole set of all erased blocks
may be used sequentially in order of ascending physical block
20 addresses. This modified embodiment may have some advantage in
reducing memory space requirements in connection with
monitoring and storing the erased state of all blocks.

25

30

35

40

45

50

55

Claims

5

10

15

20

25

30

35

40

45

50

55

5

CLAIMS

10

1. A memory system for connection to a host processor, the
5 system comprising:

15

a solid state memory having non-volatile memory sectors which
are individually addressable and which are arranged in
erasable blocks of sectors, each said sector having a physical
address defining its physical position in the memory;

20

10 and a controller for writing data structures to and reading
data structures from the memory, and for sorting the blocks of
sectors into blocks which are treated as erased and blocks
which are treated as not erased; wherein the controller
includes:

25

15 means for translating logical addresses received from the host
processor to physical addresses of said memory sectors in the
memory;

30

a Write Pointer (WP) for pointing to the physical address of a
sector to which data is to be written to from the host

35

20 processor, said Write Pointer (WP) being controlled by the
controller to move in a predetermined order through the
physical addresses of the memory sectors of any block which is
treated as erased and, when the block has been filled, to move
to another of the erased blocks;

40

25 wherein the controller is configured so that, when a sector
write command is received from the host processor, the
controller translates a logical address received from the host
processor to a physical address to which data is written by
allocating for said logical address that physical address, to

45

30 which said Write Pointer (WP) is currently pointing, and
wherein the controller is configured to compile a Sector
Allocation Table (SAT) of logical addresses with respective
physical addresses which have been allocated therefor by the
controller, and to update the SAT less frequently than memory

50

35 sectors are written to with data from the host processor.

55

- 5 2. A memory system according to claim 1, wherein said Write
Pointer (WP) is controlled by the controller to move in a
predetermined order through the blocks which are treated as
10 erased.
- 15 3. A memory system according to claim 1 or claim 2, wherein
the physical sector addresses in the SAT are ordered by
logical sector address (LSA), whereby the Nth SAT entry
10 contains the physical address of a sector to which data having
logical address N has been written.
- 20 4. A memory system according to claim 3, wherein the
controller is configured so that when a sector read command is
received from the host processor the controller looks up a
25 logical sector address (LSA) received from the host processor
in the SAT in order to obtain the physical sector address
which the controller previously allocated to said logical
sector address.
- 30 5. A memory system according to any preceding claim, wherein
the SAT is stored in at least one of said blocks of memory
sectors in the solid state memory.
- 35 6. A memory system according to claim 5, wherein the
controller is configured to update the SAT by rewriting the
SAT in whole blocks.
- 40 7. A memory system according to claim 5 or claim 6, wherein
there is provided at least one block (ASB) of sectors
45 containing modified versions of individual sectors of a SAT
block.
- 50 8. A memory system according to claim 7, wherein each sector
in a said ASB block contains the physical address of the

5 sector of the SAT block which it updates, and the modified
version of the said sector of the SAT block.

10 9. A memory system according to claim 7 or claim 8, wherein
5 when all the sectors in a said ASB block are written to with
modified versions of SAT sector(s), the respective SAT block
is rewritten so as to include all the modified versions in the
15 ASB block and the ASB block is erased.

10 10. A memory system according to any preceding claim, wherein
the controller is configured to control the Write Pointer (WP)
20 so as to move sequentially, in ascending numerical order of
physical address, through the erased blocks, as each block is
filled with data written thereto.

15 11. A memory system according to claim 10, wherein the control
of the Write Pointer (WP) is cyclic in the sense that once the
sectors in the highest block, according to physical address
30 order, have been filled with data the WP is controlled by the
20 controller to wrap around to the block of sectors having the
numerically lowest physical block address out of all the
blocks currently being treated by the controller as erased.

35 12. A memory system according to any of claims 1 to 9, wherein
25 the controller is configured to control the Write Pointer (WP)
to move non-sequentially, according to physical address order,
40 through the erased blocks.

45 13. A memory system according to any of claims 1 to 12,
30 wherein each said memory sector (1) is physically partitioned
into a data area (300) and a spare area (302) and the
controller is configured so as to write overhead data (OD)
50 comprising header data and error correction code data (ECC) at
a position in the sector which is offset from the start of the
35 data area (300) of the sector and to write user data, received

55

5 from the host processor, in the space remaining in the sector,
on either side of the overhead data (OD).

10 14. A memory system according to claim 13, wherein said
5 overhead data (OD) is offset by an amount which is determined
by at least one bit of the user data to be written to the
sector.

15 15. A memory system according to any preceding claim, wherein
10 the memory sectors in each said block of sectors are erasable
together as a unit.

20 16. A memory system according to claim 16, wherein the memory
sectors in each said block of sectors are also individually
25 erasable.

30 17. A memory system according to any preceding claim, wherein
the controller is configured to control erase operations on
the memory so as to only erase whole blocks of memory sectors,
20 and wherein a block of sectors is treated by the controller as
an erased block if all the memory sectors therein are erased
sectors.

35 18. A memory system according to claim 17, wherein if a block
25 contains one or more bad sectors, the controller defines the
whole block as being bad and treats that block as a not erased
block, whereby no data will be written thereto.

40 19. A memory system according to claim 16, wherein if a block
45 contains one or more bad sectors the controller treats that
block as an erased block whereby the controller may still use
good sectors in the block to store data, and wherein the
50 memory system includes a table identifying bad sectors and the
controller is configured to check whether the next sector
35 address to which the Write Pointer (WP) is to be moved is the

55

5 address of a bad sector and, if it is the address of a bad
sector, to control the Write Pointer to skip this bad sector
and move to the next sector address according to the
predetermined order in which the sectors are to be written to.

10

5

20. A memory system according to any preceding claim, wherein
each block of sectors has a physical block address defining
its physical position in the memory and the physical address
of each said memory sector includes the physical block address
15 of the block in which it is located, and wherein the
controller is configured to compile a list of the physical
block addresses of at least some of the blocks of sectors
being treated as erased, listed in an order in which the WP is
to move through the blocks, which list is used by the
20 controller in order to quickly identify the next block of
sectors to be written to, and the memory system further
includes temporary memory means in which said list is stored
by the controller.

30

20 21. A memory system according to any preceding claim, wherein
the controller is configured so that, when a sector write
command is received by the controller from the host processor
which command renders obsolete data previously written to
another sector, the controller stores in a temporary memory of
25 the memory system the address of the sector containing the now
obsolete data.

40

22. A memory system according to claim 21, wherein the
controller is further configured so that if a sector delete
45 command, generated by a user, is received from the host
processor by the controller, the controller marks as obsolete
the sector to be deleted and stores the address of the sector
in said temporary memory.

50

55

5 23. A memory system according to claim 21 or claim 22, wherein
the controller is configured so as to allow only a fixed
predetermined number of blocks at any time, herein referred to
as the Current Obsolete Blocks (COBs), to contain one or more
10 5 sectors containing obsolete data which was written by the
Write Pointer (WP), and so that when all the sectors in a said
COB contain obsolete data, the said COB is immediately erased.

15 24. A memory system according to claim 23, wherein the
10 controller is configured so that where a sector in a block
other than a said COB is to contain obsolete data, the
controller: relocates any data in valid (not obsolete) sectors
20 in a said COB to another block and then erases the said COB;
marks said sector in the said block other than a COB as
15 obsolete; and designates said other block as a new COB.

25 25. A memory system according to claim 23 or claim 24, wherein
said fixed predetermined number of COBs is one.

30 26. A memory system according to claim 24, wherein said block
to which the controller relocates said valid data is the block
in which the WP is currently located.

35 27. A memory system according to claim 24, wherein the memory
25 system includes a further write pointer, herein referred to as
the Relocation Pointer (RP), for pointing to the physical
address of the sector to which said valid data is to be
relocated, the RP always being in a different block of sectors
to the Write Pointer (WP).

40 28. A memory system according to claim 27, wherein the memory
30 system includes a further write pointer, referred to as the
System Write Pointer (SWP), which points to the physical
50 address of the sector to which system data is to be written

55

5 from the host, the SWP always being in a different block to
the Write Pointer (WP).

10 29. A memory system according to claim 28, wherein the
5 controller is configured so as to allow at least two blocks
which contain one or more obsolete sectors to exist at any
time, one being said COB and the other being a Current
15 Obsolete System Block (COSB) containing one or more obsolete
system data sectors and, if any system data sectors need to be
10 relocated in order to allow the COSB to be erased, the
relocated system data is sent to the address to which the
20 System Write Pointer (SWP) is currently pointing.

25 30. A memory system according to claim 28, wherein the memory
15 system includes another write pointer, herein referred to as
the System Relocation Pointer (SRP), for pointing to the
physical address of the sector to which valid system data is
to be relocated, the SRP always being in a different block of
30 sectors to the Write Pointer (WP) and the System Write Pointer
20 (SWP).

35 31. A memory system according to any of claims 28 to 30,
wherein the controller is configured so that if the COB
contains one of said write pointers (WP, RP, SWP, SRP) at the
25 time when the controller needs to erase a said COB because
obsolete data has just been created in another block, the
40 controller proceeds with creating a new COB but postpones the
erasure of the old COB, herein referred to as the Pending
Obsolete Block (POB), until all erased sectors in the POB have
45 been filled and said Pointer moves on to the next erased block
to be used, as defined by the controller, at which time any
valid (not obsolete) data in the POB is relocated by the
50 controller and the POB is erased.

55

5 32. A memory system according to claim 29, wherein the
controller is configured to store in a temporary memory of the
memory system respective lists of logical sector addresses
corresponding to sectors in the memory to which relocated data
10 has been written to by the RP (herein referred to as the
Relocation Sector List or RSL), the SWP (herein referred to as
the Write System Sector List or WSSL), and the SRP (herein
referred to as the System Relocation Sector List or SRSL)
15 since the SAT was last updated, and the controller is
20 configured to store in said temporary memory corresponding
lists of the order of blocks which have been used by the RP,
SWP and SRP (herein referred to as the Relocation Block List
(RBL), the Write System Block List (WSBL) and the System
Relocation Block List(SRBL)).

15 25 33. A memory system according to any preceding claim, wherein
in addition to writing data structures to the memory from the
host processor, the controller also generates and writes to
30 the memory data designated as control information, and the
controller is configured so as to write such control
20 information in one or more different ones (Control Blocks or
CBs) of the blocks of memory sectors to those in which data
structures received from the host processor are written.
35

25 34. A memory system according to claim 33, wherein the
controller is configured to store in at least one said Control
40 Block a list of the block addresses of all the SAT blocks.

35 45 35. A memory system according to claim 33 or claim 34, wherein
the controller is configured to store the block addresses of
said one or more Control Blocks in a dedicated block (the Boot
Block or BB) of the memory, this dedicated block being the
50 first block of sectors in the memory which does not contain
any bad sectors.

35

55

5 36. A memory system according to claim 34, as dependent from
claim 7, 8 or 9, wherein said list of all the SAT block
addresses is in the form of a plurality of list portions
10 (Table Block Lists or TBLs), and each said portion contains
5 the block addresses of a group of logically contiguous SAT
blocks and any ASBs corresponding thereto.

15 37. A memory system according to any of claims 7 to 9, wherein
the controller is configured to store in a temporary memory of
10 the memory system a list (the Write Sector List or WSL) of
logical sector addresses for data structures which have been
20 written by the Write Pointer (WP) since the SAT was last
updated.

25 38. A memory system according to claim 37, wherein the
controller is configured to also store in said temporary
memory the order in which blocks have been used by the Write
Pointer (WP) for writing data since the last update of the
30 SAT, this order being stored in the form of a list (the Write
20 Block List or WBL) of block addresses of the blocks in which
the updated sectors whose addresses are held in the WSL are
located.

35 39. A memory system according to claim 38, wherein the WSL has
25 a predetermined size and once the WSL is full at least one SAT
block or ASB block is updated and the WSL and WBL are emptied.
40

45 40. A memory system according to claim 38, wherein the
controller stores a starting physical sector address, and the
30 links between blocks containing sectors to which data has been
written by the controller since the last update of a SAT or
ASB block, in a said Control Block of the solid state memory.
50

55

- 5 41. A memory system according to any preceding claim, wherein
each said sector consists of a single "page" of memory, namely
one row of memory cells in a said block of memory sectors.
- 10 42. A memory system according to any preceding claim, wherein
the controller is configured to write data to, and read data
from, the memory sectors in uniformly sized data segments.
- 15 43. A memory system according to claim 42, wherein all the
10 memory sectors are the same size and each said data segment is
equal in size to the size of a said memory sector.
- 20 44. A memory system according to any preceding claim, further
including a temporary cache memory in which the controller is
15 configured to store a group of contiguous SAT entries
including the SAT entry most recently accessed from the SAT by
the controller.
- 30 45. A memory system according to claim 43, when dependent from
20 claim 8, wherein the controller is configured to create in
said temporary cache memory a list (ASBL) of physical
addresses of all ASBs and the SAT blocks with which they are
35 associated which is updated each time a SAT sector write
operation is performed.
- 25 46. A memory system according to any preceding claim, wherein
the solid state memory comprises a single memory array in the
form of a single memory chip.
- 40 47. A memory system according to any of claims 1 to 45,
wherein the solid state memory comprises a memory array formed
by a plurality of memory chips.
- 45 48. A memory system according to any of claims 1 to 45,
35 wherein the solid state memory comprises a plurality of memory

55

5 arrays in the form of a plurality of memory chips, and wherein
the controller is configured to form the memory sectors in the
plurality of memory chips into a multiplicity of virtual
10 blocks, each said virtual block comprising one erasable block
of memory sectors from each said memory chip, and to sort said
virtual blocks into ones which are treated as erased and ones
which are treated as not erased.

15 49. A memory system according to claim 48, wherein the
20 controller is configured to compile a list of the virtual
blocks treated as erased and store this in temporary memory in
the memory system, and to control the Write Pointer (WP) to
move from one chip to another for each consecutive sector
write operation, starting at one sector in one erasable block
25 of the virtual block and moving consecutively to one sector in
each of the other erasable blocks in the virtual block until
one sector has been written in each erasable block of the
virtual block, and then moving back to the chip in which the
30 first sector was written and proceeding in a similar manner to
fill another one sector in each erasable block of the virtual
block, and so on until the virtual block is full of data, and
then to move the Write Pointer (WP) on to the next virtual
35 block in said list of virtual blocks being treated as erased,
and fill this next virtual block in a similar manner.

40 50. A memory system according to claim 49, wherein the
controller is configured so that for every n contiguous sector
write operations the controller executes for a multiple sector
write command received from the host processor, where n is
45 30 less than or equal to the number of solid state memory chips
in the memory system, the controller writes substantially
concurrently to one sector in each of n of the chips.

50 51. A memory system according to claim 49 or claim 50, wherein
35 the controller is configured to carry out erasure of any said

55

5 virtual block by concurrently erasing all the erasable blocks
in the virtual block.

10 52. A memory system for connection to a host processor, the
5 memory system comprising:

15 a solid state memory comprising a plurality of solid state
memory chips each having non-volatile memory sectors which are
individually addressable and which are arranged in erasable
16 blocks of sectors, each said sector having a physical address
10 defining its physical position in the memory;

20 and a controller for writing data structures to and reading
data structures from the memory, wherein:

25 the controller forms the erasable blocks into virtual blocks,
each said virtual block comprising an erasable block from each
15 of the memory chips, and the controller sorts the virtual
blocks into ones which are treated as erased and ones which
are treated as not erased, and the controller fills one
virtual block with data prior to moving on to the next virtual
30 block to be filled, and each virtual block is filled by

20 writing to the memory sectors thereof in a repeating sequence
in which the controller writes to one memory sector in each of
the erasable blocks of the virtual block one after another
35 whereby consecutively written sectors are in different chips.

25 53. A memory system according to claim 52, wherein the
40 controller is configured so that for every n contiguous sector
write operations the controller executes for a multiple sector
write command from the host processor, where n is less than or
equal to the number of solid state memory chips in the memory
45 30 system, the controller writes substantially concurrently to
one sector in each of n of the chips.

50 54. A controller for writing data structures to and reading
data structures from a solid state memory having non-volatile
35 memory sectors which are individually addressable and which

55

5 are arranged in erasable blocks of sectors, each said sector
having a physical address defining its physical position in
the memory, wherein the controller includes:
10 means for translating logical addresses received from a host
processor of a memory system in which the controller is used
to physical addresses of said memory sectors in the memory,
and for sorting the blocks of sectors into blocks which are
15 treated as erased and blocks which are treated as not erased;
and a Write Pointer (WP) for pointing to the physical address
of a sector which is to be written to from the host processor,
said Write Pointer (WP) being controlled by the controller to
20 move in a predetermined order through the physical addresses
of the memory sectors in any block which is treated as erased
and, when the block has been filled, to move to another of the
15 erased blocks, and wherein the controller is configured so
that, when a sector write command is received by the
25 controller from the host processor, the controller translates
a logical sector address received from the host processor to a
physical address to which data is written by allocating for
30 said logical address that physical address to which said Write
Pointer (WP) is currently pointing;
and wherein the controller is configured to compile a table
35 (the SAT) of logical addresses with respective physical
addresses which have been allocated therefor by the
controller, and to update the SAT less frequently than memory
25 sectors are written to with data from the host processor.

40
55. A method of controlling reading and writing of data
structures to and from a solid state memory having non-
45 volatile memory sectors which are individually addressable and
which are arranged in erasable blocks of sectors, each said
sector having a physical address defining its physical
50 position in the memory, the method comprising the steps of:
sorting the blocks of sectors into blocks which are treated as
35 erased and blocks which are treated as not erased;

5 providing a Write Pointer (WP) for pointing to the physical
address of a sector which is to be written to, and controlling
said at least one Write Pointer (WP) so as to move in a
predetermined order through the physical addresses of the
10 memory sectors of any block which is treated as erased, and
when the block has been filled to move to another of the
erased blocks and, when a sector write command is received
from the host processor, translating a logical address
15 received from the host processor to a physical address to
20 which data is written by allocating for said logical address
that physical address to which said Write Pointer (WP) is
currently pointing;
storing in non-volatile solid state memory a table (the SAT)
of logical addresses with respective physical addresses which
25 have been allocated therefor by the controller;
and updating the SAT less frequently than memory sectors are
written to with data from the host processor.

30 56. A memory system for connection to a host processor, the
20 system comprising:
a solid state memory having non-volatile memory sectors which
are individually addressable and which are arranged in
35 erasable blocks of sectors, each said sector having a physical
address defining its physical position in the memory;
25 and a controller for writing data structures to and reading
40 data structures from the memory, wherein the controller
includes means for translating logical addresses received from
the host processor to physical addresses of said memory
sectors in the memory; and wherein
45 30 each said memory sector (1) is physically partitioned into a
data area (300) and a spare area (302) and the controller is
configured so as to write overhead data (OD) comprising header
data and error correction code data (ECC) at a position in the
50 sector which is offset from the start of the data area (300)
35 of the sector and to write user data, received from the host

55

5

processor, in the space remaining in the sector, on either side of the overhead data (OD).

10

57. A memory system according to claim 56, wherein said overhead data (OD) is offset by an amount which is determined by at least one bit of the user data to be written to the sector.

15

20

25

30

35

40

45

50

55

Fig. 1.

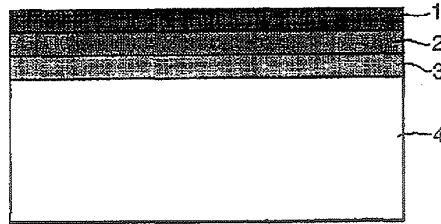


Fig. 2.

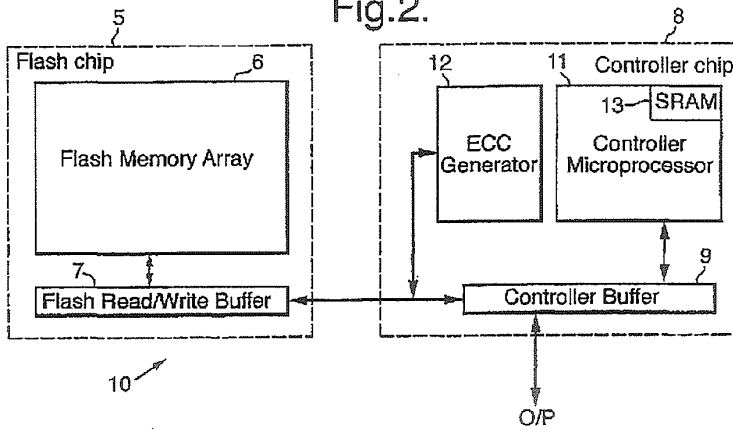


Fig. 3.

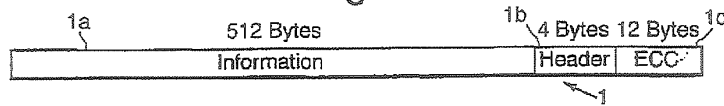
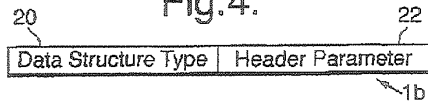


Fig. 4.



SUBSTITUTE SHEET (RULE 26)

Fig.5.

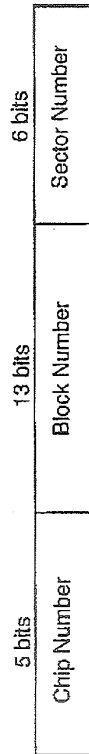


Fig.6.

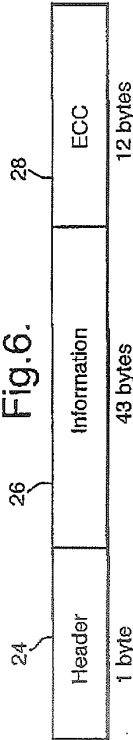


Fig.7.

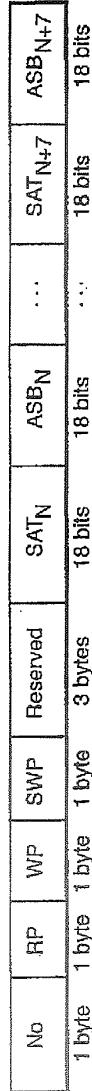
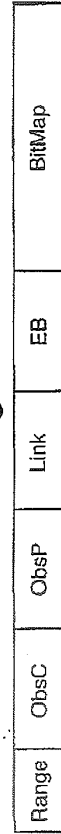


Fig.8.



SUBSTITUTE SHEET (RULE 26)

Fig.9.

SAT Block Index	SAT Block Address	ASB Address	LWP	NVP	ASB Page 0	...	ASB Page N
2 bytes	3 bytes	3 bytes	1 byte	1 byte	1 byte	...	1 byte

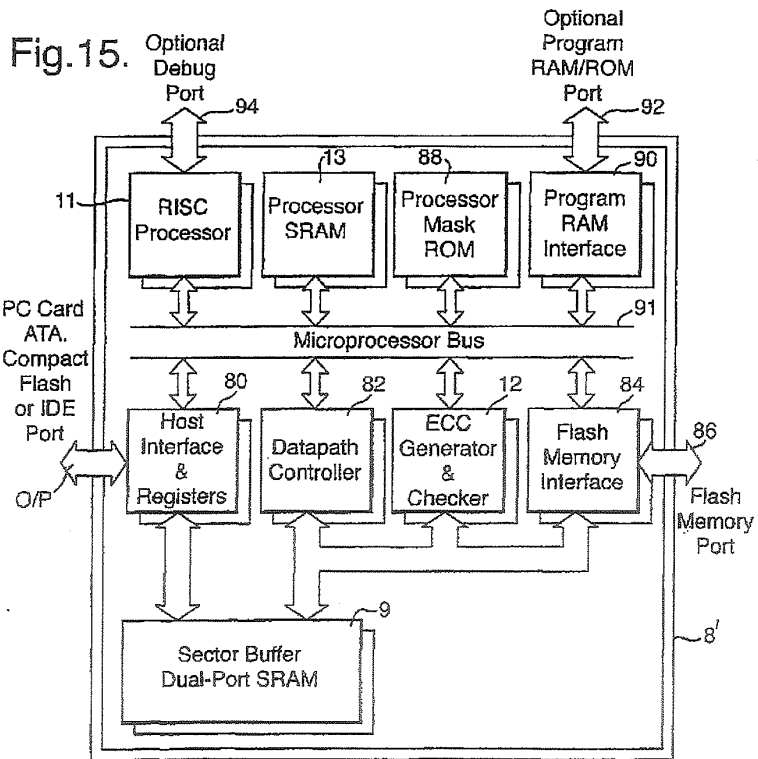
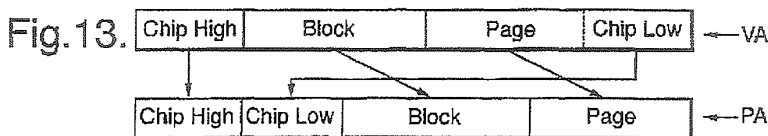
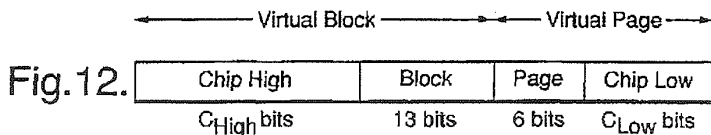
Fig.10.

Block Number	Obsolete or Deleted Sector Mask
4 bytes	32 bytes

Fig.11.

Chip 0 Block 0	Chip 1 Block 0	Chip 2 Block 0	Chip 3 Block 0
0	1	2	3
4	5	6	Etc.

SUBSTITUTE SHEET (RULE 26)



SUBSTITUTE SHEET (RULE 26)

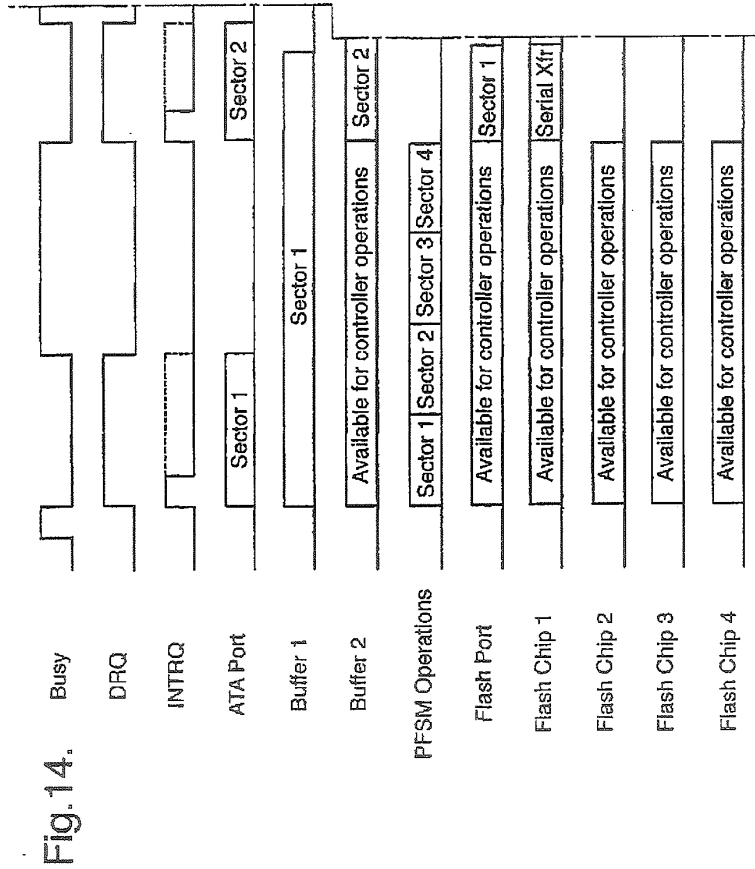
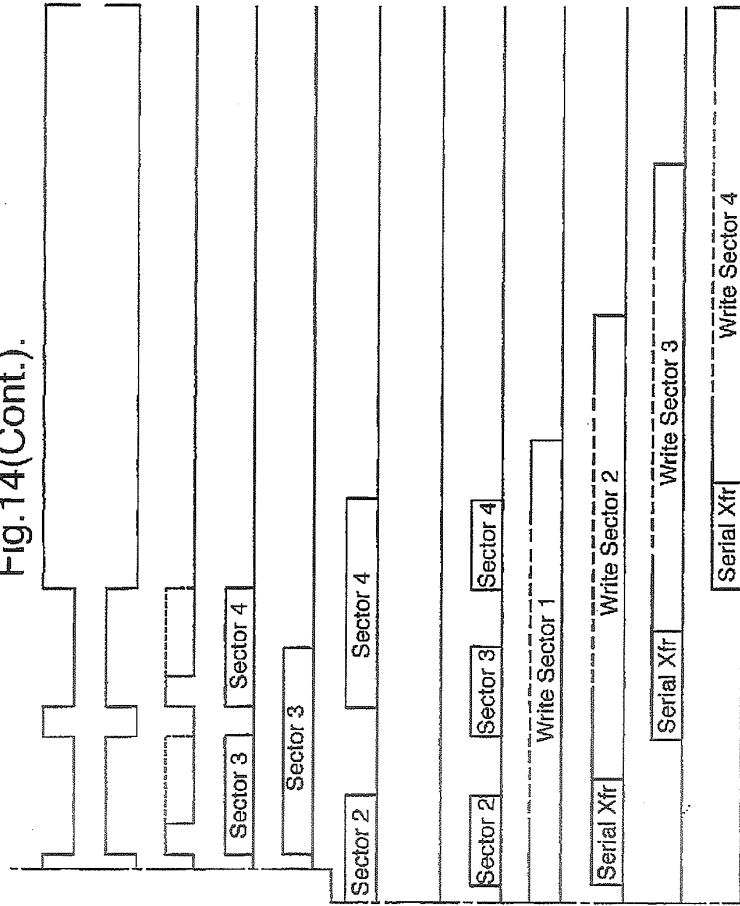


Fig. 14.

SUBSTITUTE SHEET (RULE 26)

Fig. 14(Cont.).



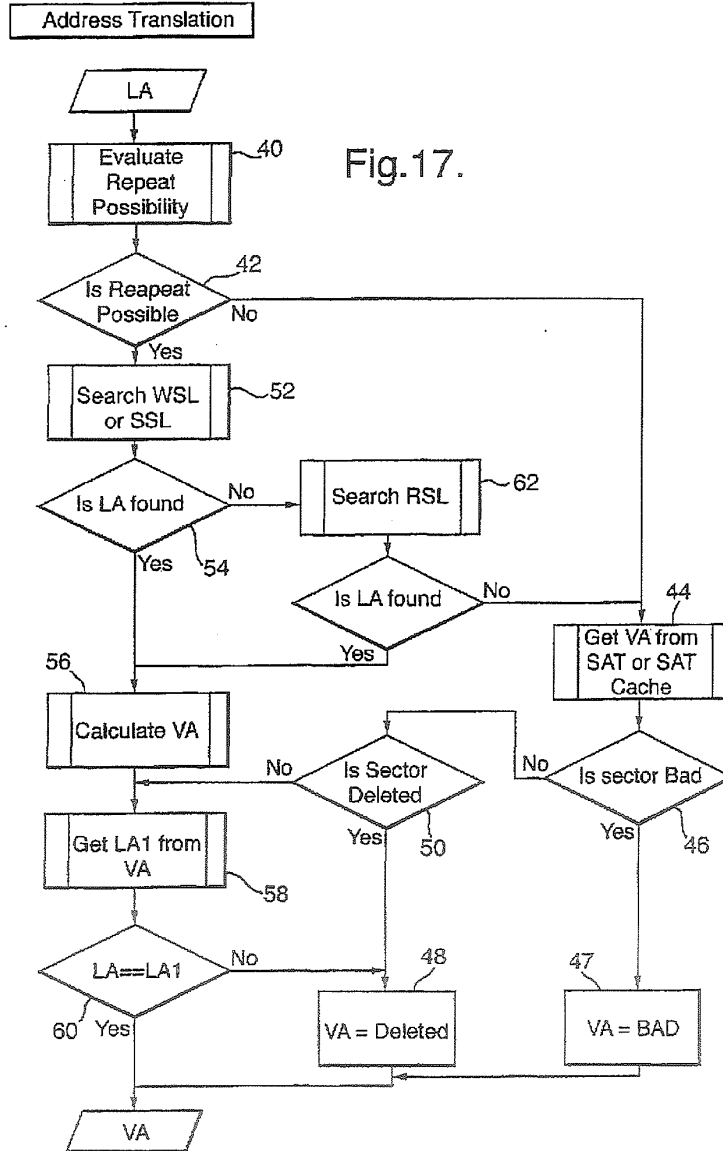
SUBSTITUTE SHEET (RULE 26)

Capacity Allocation Table

	8MB Card		64MB Card		512MB Card	
	Number of Blocks	% of Capacity	Number of Blocks	% of Capacity	Number of Blocks	% of Capacity
Total Capacity	1024	100%	8192	100%	65536	100%
Sectors	1007	98.1%	8132	99.2%	65131	99.4%
Boot Block	1	0.1%	1	0.01%	1	0.00%
Control Block	3	0.3%	4	0.05%	4	0.00%
Sector Address Table	6	0.6%	48	0.6%	384	0.6%
Additional SAT Blocks	6	0.6%	8	0.1%	8	0.01%
Obsolete Sectors	1	0.1%	1	0.01%	1	0.00%
Erased Buffer	2	0.2%	2	0.02%	2	0.00%
Spare Blocks						

Fig. 16.

SUBSTITUTE SHEET (RULE 26)



SUBSTITUTE SHEET (RULE 26)

9/21

Fig.18.

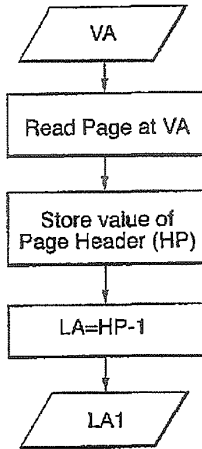
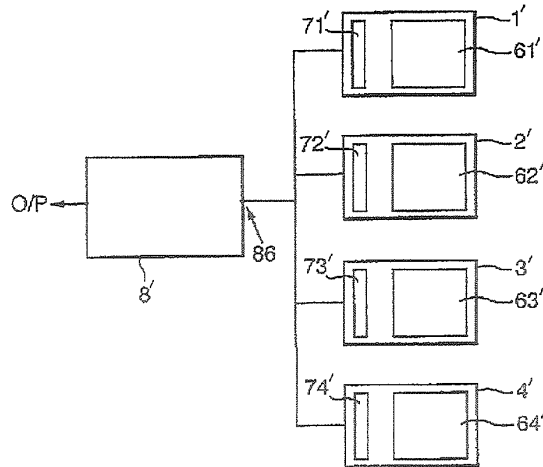


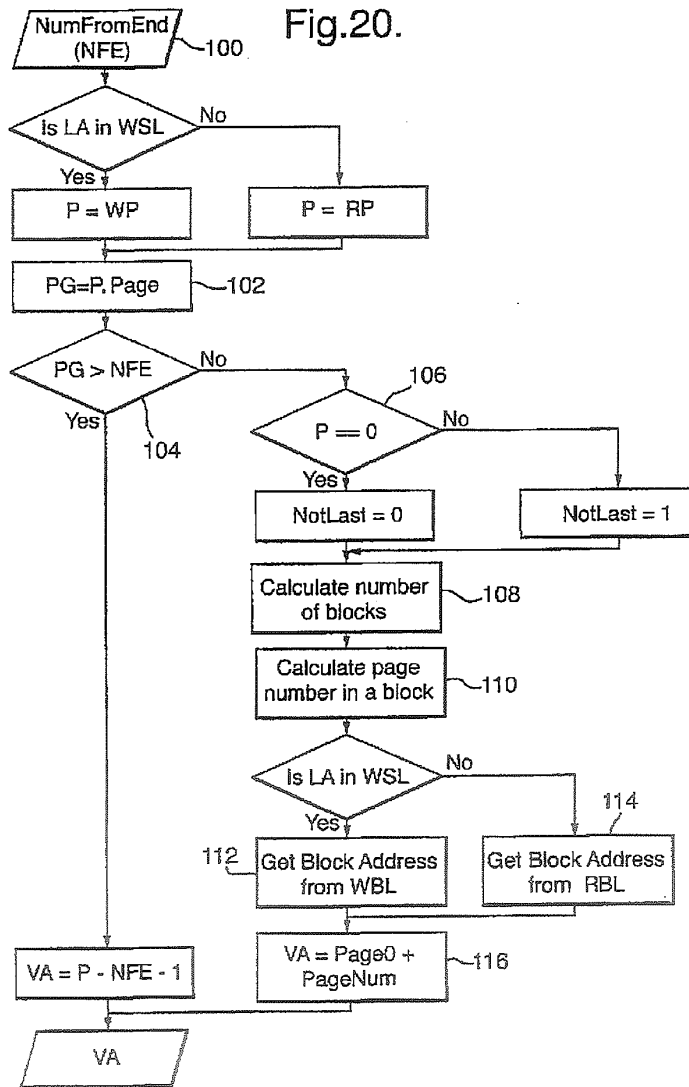
Fig.19.



SUBSTITUTE SHEET (RULE 26)

10/21

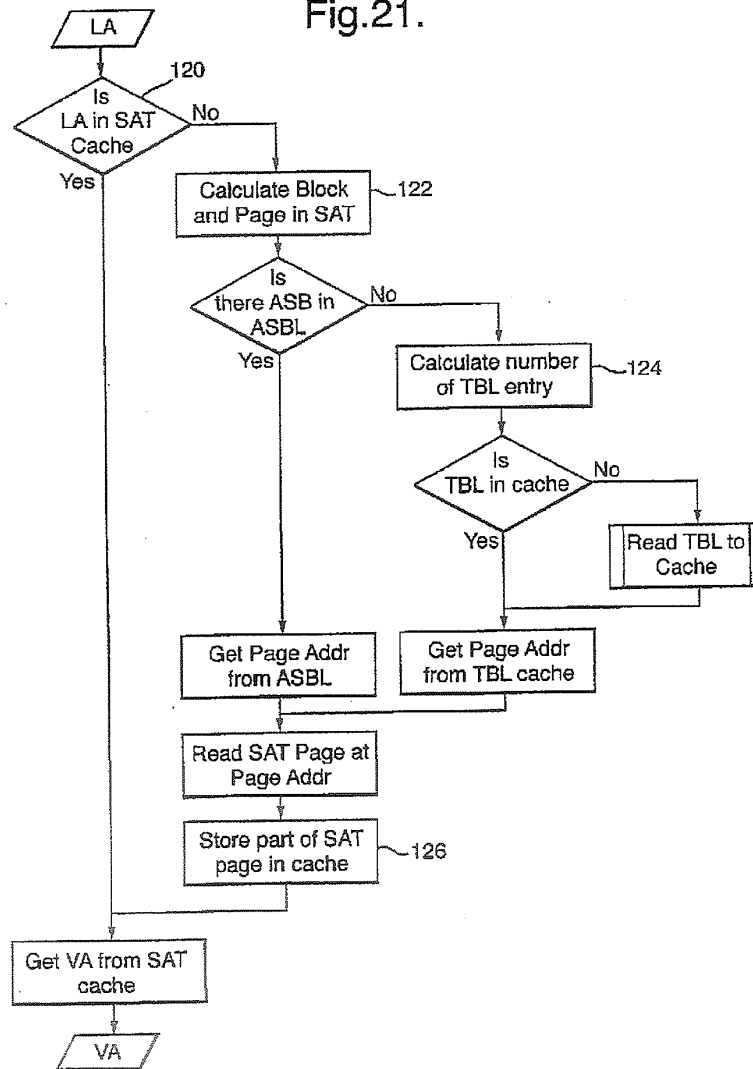
Fig.20.



SUBSTITUTE SHEET (RULE 26)

11/21

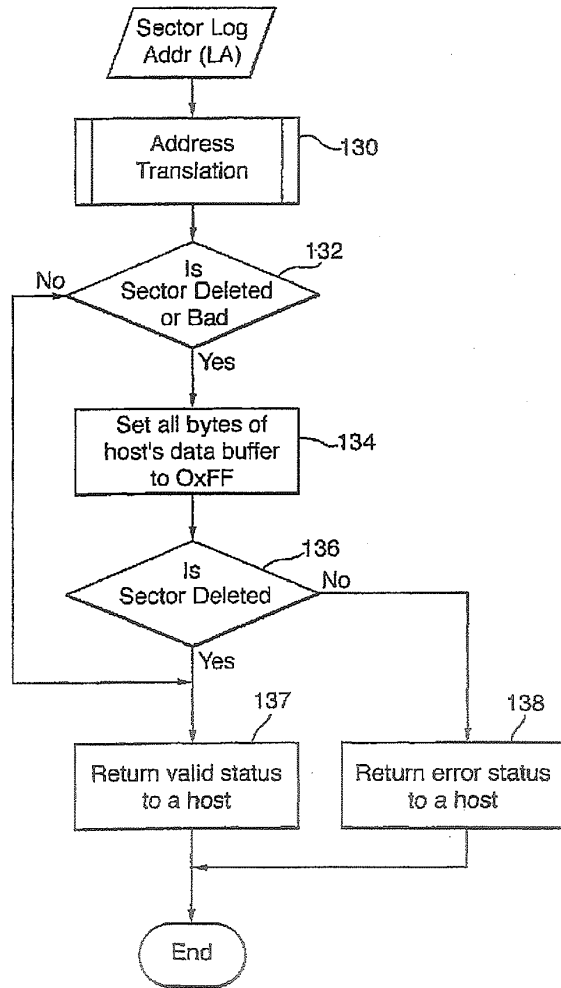
Fig.21.



SUBSTITUTE SHEET (RULE 26)

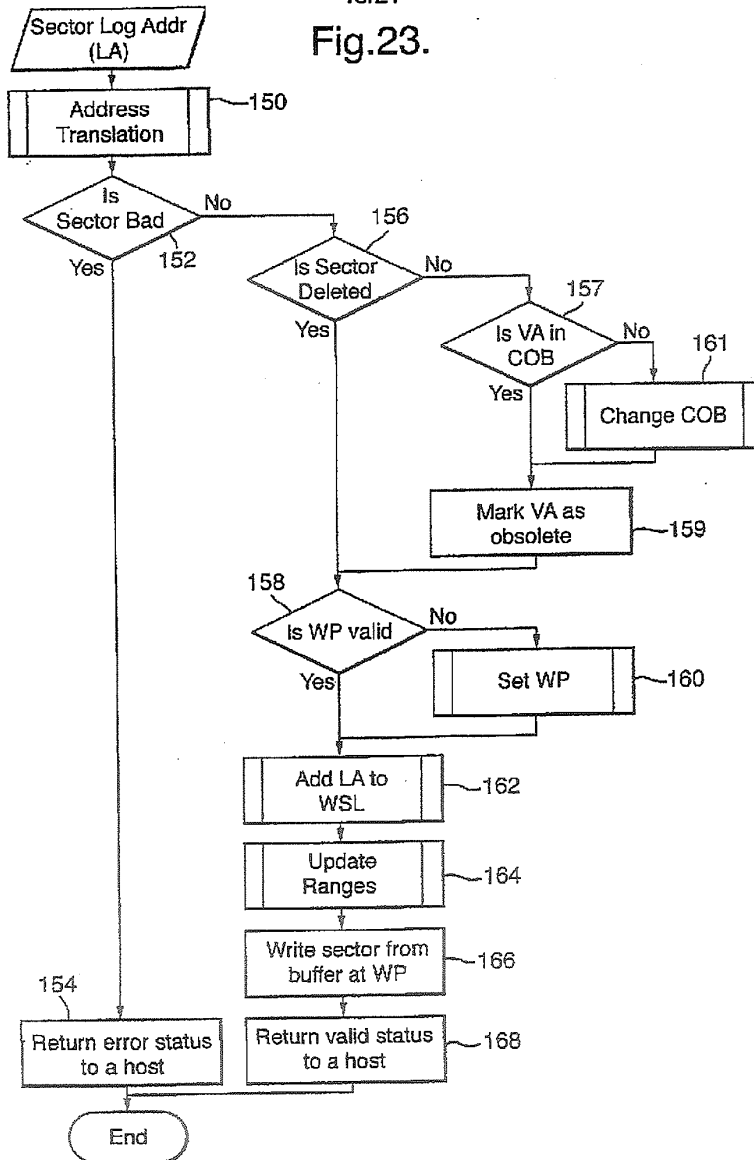
12/21

Fig.22.



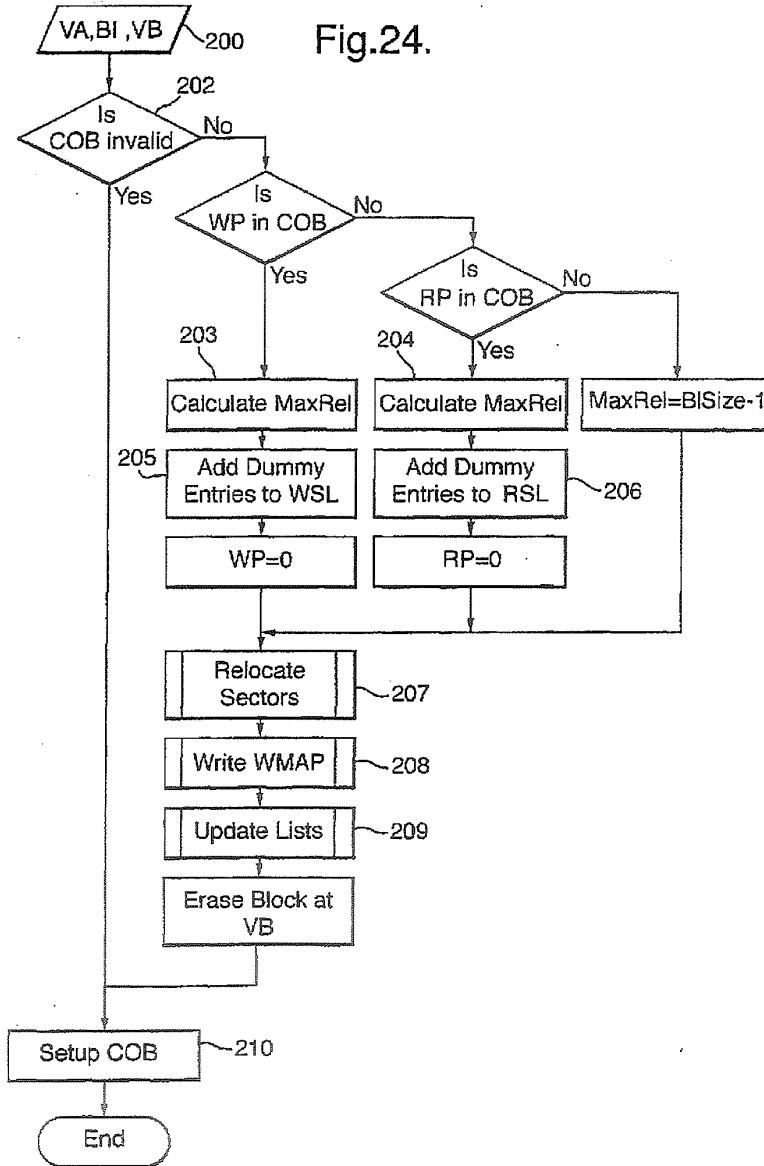
SUBSTITUTE SHEET (RULE 26)

13/21
Fig.23.



SUBSTITUTE SHEET (RULE 26)

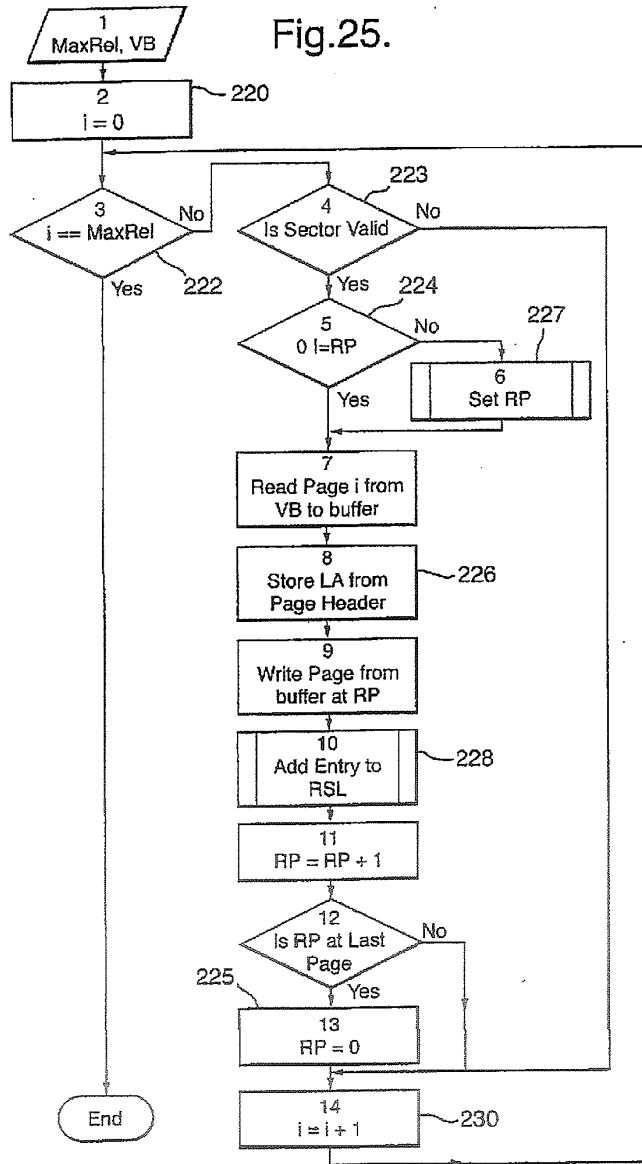
Fig.24.



SUBSTITUTE SHEET (RULE 26)

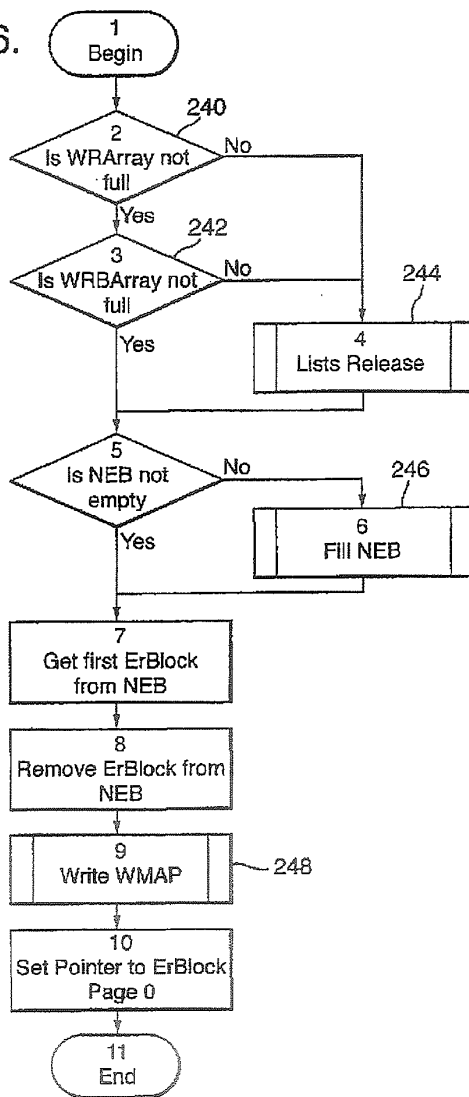
15/21

Fig.25.



SUBSTITUTE SHEET (RULE 26)

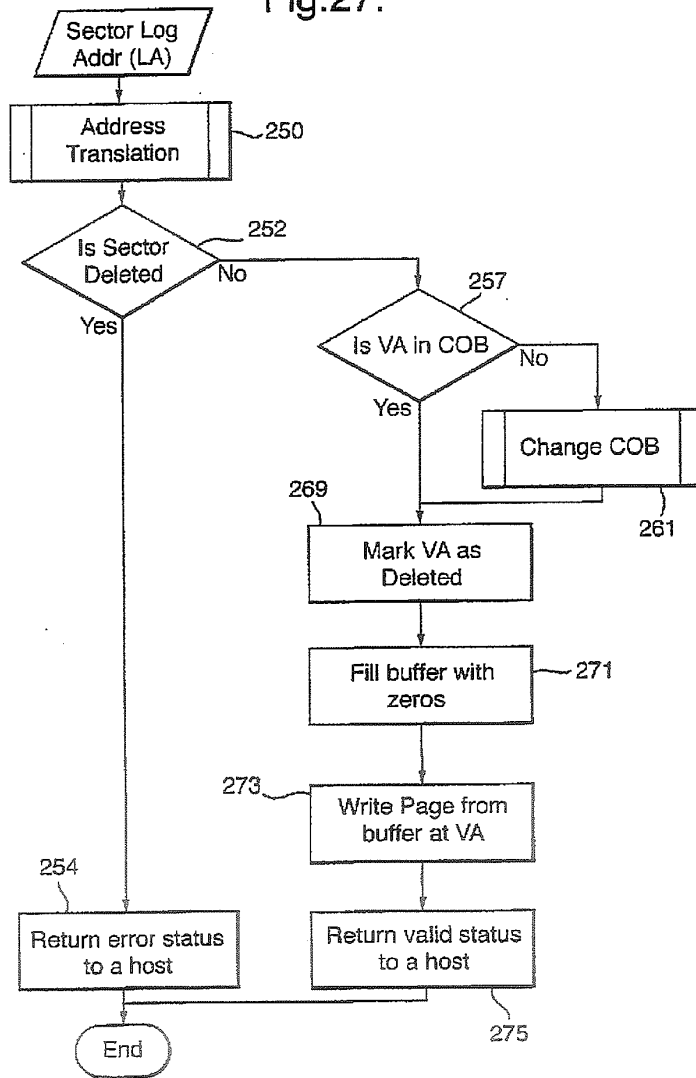
Fig.26.



SUBSTITUTE SHEET (RULE 26)

17/21

Fig.27.



SUBSTITUTE SHEET (RULE 16)

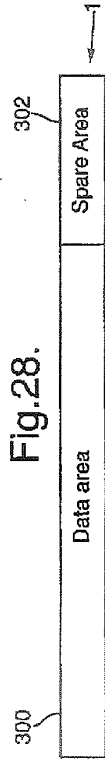


Fig. 28.

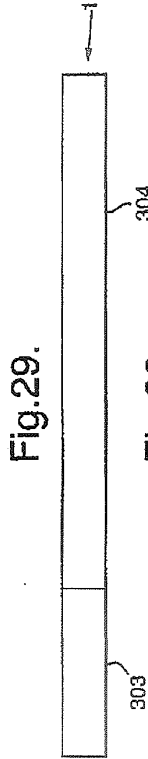


Fig. 29.

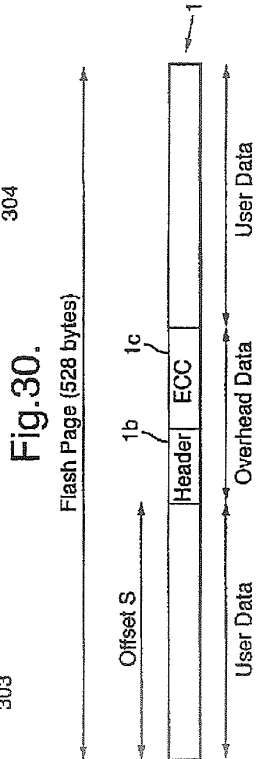


Fig. 30.

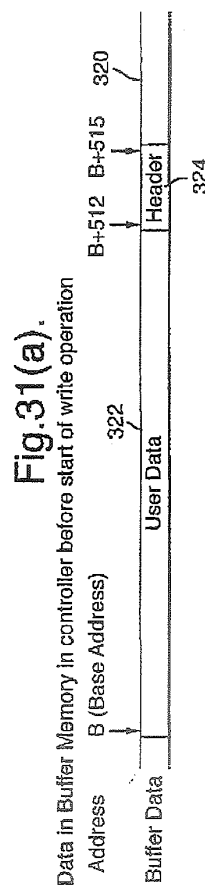


Fig. 31(a).

Data in Buffer Memory in controller before start of write operation

Address B (Base Address)

Buffer Data

User Data

Header

324

B+512 B+515 320

SUBSTITUTE SHEET (RULE 26)

Fig.31(b).

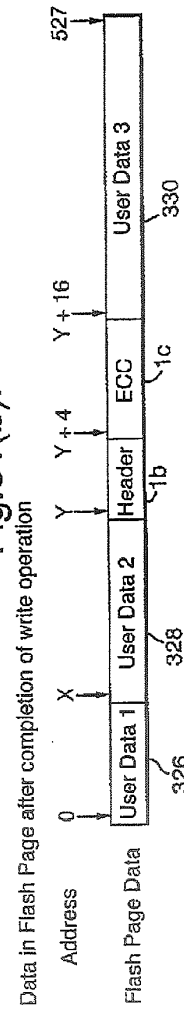


Fig.33.

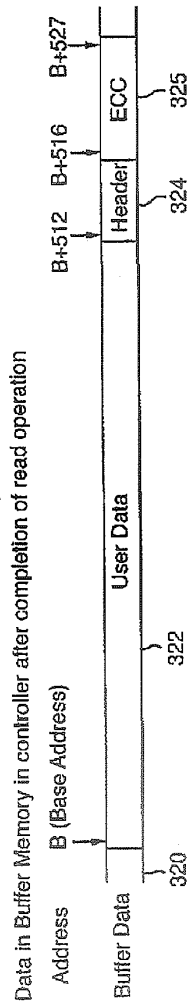
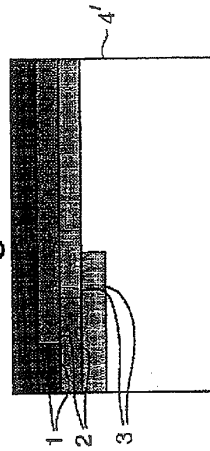


Fig.35.



SUBSTITUTE SHEET (RULE 26)

Fig.32.
Sequence of commands to controller data transfer hardware

Command Sequence	Command Parameters			Command Description		
	Flash Start Address	Buffer Start Address	No. of Data Xfr Cycles	ECC Generator Mode	Flash Clock	Summary
Crmd 1	n/a	B	516	Generator on Output register disabled	off	Generate ECC for 512 bytes of User Data + Header
Crmd 2	0	B	Y	Generator off Output register disabled	on	Transfer User Data 1 + User Data 2 to Flash buffer
Crmd 3	continue	B + 512	4	Generator off Output register disabled	on	Transfer Header to Flash buffer
Crmd 4	continue	n/a	12	Generator off Output register enabled	on	Transfer ECC to Flash buffer
Crmd 5	continue	B + Y	512 - Y	Generator off Output register disabled	on	Transfer User Data 3 to Flash buffer Write Flash buffer to Flash array

SUBSTITUTE SHEET (RULE 26)

Fig.34.

Sequence of commands to controller data transfer hardware

Command Sequence	Command Parameters			Command Description		
	Flash Start Address	Buffer Start Address	No. of Data Xfr Cycles	ECC Generator Mode	Flash Clock	Summary
Cmdnd 1	0	B	X	Generator on Output register disabled	on	Read Flash page to Flash buffer Transfer User Data 1 from Flash buffer
Cmdnd 2	continue	B + X	Y - X	Generator on Output register disabled	on	Transfer User Data 2 from Flash buffer
Cmdnd 3	continue	B + 512	16	Generator off Output register disabled	on	Transfer Header + ECC from Flash buffer
Cmdnd 4	continue	B + Y	512 - Y	Generator on Output register disabled	on	Transfer User Data 3 to Flash buffer
Cmdnd 5	n/a	B + 512	16	Generator on Output register disabled	off	Transfer Header + ECC from Buffer to ECC Generator

SUBSTITUTE SHEET (RULE 26)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 00/00550

A. CLASSIFICATION OF SUBJECT MATTER IPC 7 06F3/06 According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC 7 06F Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practical, search terms used)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	GB 2 291 991 A (MEMORY CORP PLC) 7 February 1996 (1996-02-07) page 5, line 15 -page 8, line 5; figures	1,2,4, 10,11, 15,17, 20,46, 47,52, 54-56
A	WO 97 37296 A (SINCLAIR ALAN WELSH ;MEMORY CORP PLC (GB)) 9 October 1997 (1997-10-09) page 3, line 21 -page 7, line 6; figures -/-	1,52, 54-56
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C. <input checked="" type="checkbox"/> Patent family members are listed in annex.		
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but published on or after the international filing date "L" document which may throw doubt on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document relating to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. "Z" document member of the same patent family		
Date of the actual completion of the international search		Date of mailing of the international search report
25 May 2000		02/06/2000
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patanilaan 2 NL - 2280 HV Rijswijk Tel: (+31-70) 940-2040, Tx. 31 651 00010, Fax: (+31-70) 940-3016		Authorized officer Moens, R

Form PCT/ISA210 (second sheet) (July 1999)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 00/00550

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 712 067 A (MITSUBISHI ELECTRIC CORP) 15 May 1996 (1996-05-15) column 4, line 37 -column 6, line 32; figures	1-4, 10, 15, 17, 24, 46, 47, 52, 54-56
A	EP 0 522 780 A (IBM) 13 January 1993 (1993-01-13) claims 1,9,13	1, 52, 54-56
A	WO 94 20906 A (SYSTEMS LTD M ;SYSTEMS INC M (US)) 15 September 1994 (1994-09-15) page 2, line 21 -page 5, line 2	1, 48, 49, 52
A	US 5 602 987 A (HARARI ELIYAHOU ET AL) 11 February 1997 (1997-02-11) column 8, line 39 - line 67; figures IA, 2	1, 13, 14, 16, 56, 57

1

Form PCT/GB/AU/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/GB 00/00550

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
GB 2291991 A	07-02-1996	CN 1198226 A	04-11-1998
		EP 0852766 A	15-07-1998
		WO 9712325 A	03-04-1997
		JP 11511879 T	12-10-1999
WO 9737296 A	09-10-1997	EP 0896699 A	17-02-1999
EP 0712067 A	15-05-1996	JP 8137634 A	31-05-1996
		US 5905993 A	18-05-1999
EP 0522780 A	13-01-1993	JP 2582487 B	19-02-1997
		JP 5027924 A	05-02-1993
		DE 69223287 D	08-01-1998
		DE 69223287 T	28-05-1998
		US 5524230 A	04-06-1996
WO 9420905 A	15-09-1994	US 5404485 A	04-04-1995
		AU 6269994 A	26-09-1994
		CN 1098526 A	08-02-1995
		DE 69414556 D	17-12-1998
		DE 69414556 T	06-05-1999
		EP 0688450 A	27-12-1995
		FI 954235 A	08-11-1995
		IL 108766 A	05-12-1996
		JP 8510072 T	22-10-1996
		ZA 9401446 A	26-09-1994
		US 5602987 A	11-02-1997
US 5535328 A	09-07-1996		
US 5991517 A	23-11-1999		
DE 69024086 D	25-01-1996		
DE 69024086 T	20-06-1996		
DE 69033262 D	30-09-1999		
DE 69033262 T	24-02-2000		
DE 69033438 D	02-03-2000		
EP 0392895 A	17-10-1990		
EP 0617363 A	28-09-1994		
EP 0618535 A	05-10-1994		
EP 0675502 A	04-10-1995		
EP 0935255 A	11-08-1999		
JP 2292798 A	04-12-1990		
US 5671229 A	23-09-1997		
US 5719808 A	17-02-1998		
US 5936971 A	10-08-1999		
US 5418752 A	23-05-1995		
US 5877986 A	02-03-1999		
US 5862080 A	19-01-1999		
US 5999446 A	07-12-1999		

Form PCT/ISA/210 (patent family annex) (July 1992)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Kevin M. Conley
Title: Partial Block Data Programming And Reading Operations In A Non-Volatile Memory
Application No.: 11/250,238 Filing Date: October 13, 2005
Examiner: Dinh, Ngoc V. Group Art Unit: 2189
Docket No.: SNDK.156US2 Conf. No.: 7727

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUPPLEMENTAL AMENDMENT

Sir:

In response to a request made by Examiner Dinh by telephone on November 13, 2008, for corrections to claims 30, 34, 40, 49 and 68, these claims are being amended. In addition, another review of the claims by the undersigned attorney has revealed the desirability for other claim amendments. Therefore, amendments in addition to those requested by Examiner Dinh are being made by this Amendment.

All the claim amendments are reflected in the listing of claims, which begins on page 2 of this paper.

Remarks begin on page 17 of this paper.

Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

U.S. Department of Commerce, Patent and Trademark				Atty. Docket No.			Application No.	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT				SNDK.156US2			11/250,238	
				Applicants			Kevin M. Conley	
(Use several sheets if necessary)				Filing Date			Art Group	
(Form PTO-1449)				October 13, 2005			2189	
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
/N.D./	1	6,684,289 B1	1/27/2004	Gonzalez et al.				
/N.D./	2	6,947,332 B2	9/20/2005	Wallace et al.				
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
/N.D./	3	WO 97/43764 A1	11/20/1997	WIPO				
/N.D./	4	WO 00/49488 A1	8/24/2000	WIPO				
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
/N.D./	5	Deposition of Kevin M. Conley dated July 24, 2008, Volume I, 49 pages.						
	6	Deposition of Kevin M. Conley dated July 25, 2008, Volume II, 35 pages.						
	7	Deposition of Kevin M. Conley dated September 11, 2008, Volume III, 35 pages.						
	8	Exhibits No. 1 through No. 12, No. 15 through No. 22, No. 24 through No. 26, No. 28 through No. 32, to Deposition of Kevin M. Conley, total 573 pages. February 02, 2000.						
	9	Exhibits No. 37 through No. 56, No. 58, No. 59, No. 61 through No. 72, to Deposition of Kevin M. Conley, total 729 pages. January 13, 2003.						
	10	Deposition of John Mangan dated July 10, 2008, Volume 1, 61 pages.						
/N.D./	11	Deposition of John Mangan dated July 11, 2008, Volume 2, 37 pages.						

FILED VIA EFS
Sheet 1 of 3

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.	Application No.
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		SNDK.156US2	11/250,238
		Applicants	Conf. No.
(Use several sheets if necessary)		Kevin M. Conley	7727
(Form PTO-1449)		Filing Date	Art Group
		October 13, 2005	2189
/N.D./	12	Exhibits No. 1 through No. 10 to the Deposition of John Mangan, total 540 pages. September 12, 1989	
	13	Exhibits No. 11 through No. 15, No. 17 through No. 25, No. 29, No. 31, No. 32 to the Deposition of John Mangan, total 504 pages. April 29, 2008 .	
	14	SanDisk Corporation, SDP-32 Mbit "Mizer" Low Cost HD, Logical Format 20-10-00050, Rev. 8, December 5, 1997, 32 pages.	
	15	Sandisk Israel, MediaClip Memory Card MMC Controller Top Level Design, Version 0.1, November 13, 1996, 74 pages.	
	16	SanDisk Corporation, Preliminary MultiMediaCard Product Manual, April 28, 1998, 96 pages.	
	17	Photo International, Digital: SanDisk Introduces 64, 80, 96 and 160 MB CompactFlash Memory, February 1999, 1 page.	
	18	Hsia et al., "A Reconfigurable Interconnect For In-Silicon Electronic Assembly", <i>IEEE</i> , (1982) pp. 7-16.	
	19	Hsia et al., "Adaptive Wafer Scale Integration", Proceeding of the 11 th Conference (1979 International), <i>Japanese Journal of Applied Physics, Volume 19</i> (1980) Supplement 19-1, pp. 193-202.	
	20	Hsia, "Memory Applications Of The MNOS", <i>1972 Wescon Technical Papers Volume 16</i> , (1972) pp. 1-5.	
	21	Nijima, "Design of a solid-state file using flash EEPROM," IBM J. Res. Develop, Vol. 39, No. 5, September 1995, 15 pages (as marked up by Dr. Harari).	
	22	Heller Ehrman, Letter from A. Sharma to J. Yoon identifying Heller's Respondents' Prior Art, dated March 10, 2008, 7 pages.	
	23	United States International Trade Commission, Investigation No. 337-TA-619, Notice of Prior Art of Respondents Apacer Technology, Inc. and Apacer Memory America, Inc., Silicon Motion Technology Corporation, Silicon Motion, Inc. (Taiwan), Silicon Motion, Inc. (California) and Silicon Motion International, Inc., Transcend Information, Inc. (Taiwan), Transcend Information, Inc. (California) and Transcend Information Maryland, Inc., dated July 11, 2008, 17 pages.	
	24	United States International Trade Commission, Investigation No. 337-TA-619, The Heller Respondents' Notice of Prior Art, dated July 11, 2008, 23 pages.	
	25	United States International Trade Commission, Investigation No. 337-TA-619, Respondents Kingston Technology Co., Inc., Kingston Technology Corporation, Memosun, Inc., Payton Technology Corporation, and Phison Electronics Corporation's Notice of Prior Art, dated July 11, 2008, 19 pages.	
/N.D./	26	United States International Trade Commission, Investigation No. 337-TA-619, Respondents LG Electronics, Inc. and LG Electronics U.S.A., Inc's Notice of Prior Art, July 11, 2008, 16 pages.	

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.	Application No.
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		SNDK.156US2	11/250,238
		Applicants	Conf. No.
(Use several sheets if necessary)		Kevin M. Conley	7727
(Form PTO-1449)		Filing Date	Art Group
		October 13, 2005	2189
/N.D./	27	United States International Trade Commission, Investigation No. 337-TA-619, Niles Kynett's testimony dated October 29, 2008, 22 pages.	
	28	United States International Trade Commission, Investigation No. 337-TA-619, Niles Kynett's testimony dated October 30, 2008, 23 pages.	
	29	United States International Trade Commission, Investigation No. 337-TA-619, witness statement of Kevin Conley dated October 9, 2008, 9 pages.	
	30	United States International Trade Commission, Investigation No. 337-TA-619, Professor M. Ray Mercer's testimony dated November 3, 2008, 30 pages.	
	31	United States International Trade Commission, Investigation No. 337-TA-619, Dr. Thomas Rhyne's testimony dated October 27, 2008, 86 pages.	
	32	United States International Trade Commission, Investigation No. 337-TA-619, Dr. Thomas Rhyne's testimony dated October 28, 2008, 84 pages.	
	33	United States International Trade Commission, Investigation No. 337-TA-619, Dr. Thomas Rhyne's testimony dated October 29, 2008, 34 pages.	
	34	United States International Trade Commission, Investigation No. 337-TA-619, Dr. Subramanian's testimony dated October 30, 2008, 60 pages.	
	35	United States International Trade Commission, Investigation No. 337-TA-619, Dr. Subramanian's testimony dated October 31, 2008, 59 pages.	
/N.D./			
Examiner /Ngoc Dinh/		Date Considered 01/05/2009	
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.			

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.		Application No.				
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		SNDK.156US2		11/250,238				
		Applicants		Conf. No.				
(Use several sheets if necessary)		Conley		7727				
(Form PTO-1449)		Filing Date		Art Group				
		October 13, 2005		2189				
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
/N.D./	1	Japanese Patent Office, "Decision of Refusal," corresponding Japanese Patent Application No. 2002-558275 on November 27, 2007, 3 pages (including translation).						
/N.D./	2	The Patent Office of the People's Republic of China, "Notification of the First Office Action," corresponding Chinese Patent Application No. 200610142358.3 on December 14, 2007, 3 pages.						
Examiner	/Ngoc Dinh/		Date Considered	12/23/2008				
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

FILED VIA EFS
Sheet 1 of 1

U.S. Department of Commerce, Patent and Trademark INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Use several sheets if necessary) (Form PTO-1449)	Atty. Docket No. SNDK.156US2 Applicants Kevin M. Conley Filing Date October 13, 2005	Application No. 11/250,238 Conf. No. 7727 Art Group 2189
---	---	---

U.S. Patent Documents

*Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate

U.S. Published Patent Application Documents

*Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate

Foreign Patent Documents

*Examiner Initial	Document	Date	Country	Class	Subclass	Translation	
						Yes	No

OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)

/N.D./	1	V. Niles Kynett, "Expert Report on the Invalidity of the '424 Patent," dated August 8, 2008, 99 pages (including Exhibit 4.)
	2	Exhibits No. 16 through 38 from Kynett's Expert Report, 331 pages.
	3	Exhibits No. 36 through 41 from Kynett's Expert Report, 60 pages.
	4	Exhibits No. 46 through 57 from Kynett's Expert Report, 157 pages.
	5	Exhibits No. 65 through 72 from Kynett's Expert Report, 63 pages.

Examiner /Ngoc Dinh/	Date Considered 12/23/2008
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.	

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S246	17	S244 same ((identical "same" common) near4 (logical virtual lba lsa lbn))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 14:35
S247	323	S244 and ((identical "same" common) near4 (logical virtual lba lsa lbn))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 14:39
S248	2604	read\$3 with ((most more) with (updated recent) with (version data block sector page))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 14:41
S249	24	S248 same ((identical "same" common) near4 (logical virtual lba lsa lbn))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 14:41
S250	158	S248 same S243	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 14:48
S251	125484	(read\$3 same (program\$5)) same S243	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 15:23
S252	89059	(read\$3 with (program\$5)) same S243	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 15:23
S253	11295	(eprom eprom flash non \$volatile) with (partition\$3 divid\$3)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 15:24

S254	1209	S252 same S253	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 15:24
S255	1209	(read\$3 with (program\$5)) same S253	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 15:25
S256	4489	(eprom eeprom flash non \$volatile) near3 (partition\$3 divid\$3)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 15:26
S257	492	(read\$3 with (program\$5)) same S256	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 15:26
S258	117	metablock same S243	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 15:45
S259	592442	prom eeprom flash non \$volatile	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/26 08:52
S260	25414	read\$3 with ((recent up\$to \$date new\$3) near5 (version data))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/26 08:54
S261	80	S260 same (("same" identical common) with (logical lbn lsn lu))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/26 08:55
S262	45	S259 and S261	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/26 08:55

10/28/08 10:00:06 AM

C:\Documents and Settings\ndinh\My Documents\EAST\workspaces\11250238.wsp

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	87775	(flash eeprom eeprom non \$volatile) same (block page)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 14:40
L2	4890	(read\$3 write writing updat\$3) same ("same" identical common) near4 (logical lba lbn lsa lsn))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 14:40
L3	4143	read\$3 with ((most more) near3 recent)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 14:40
L4	21	L2 same L3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 14:40
L5	18	L1 and L4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 14:40
L6	2999	(read\$3) same ("same" identical common) near4 (logical lba lbn lsa lsn))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 14:52
L7	10738	(read\$3) same ((more most) near3 recent)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 14:53
L8	32	7 same ("same" identical common) near4 (logical lba lbn lsa lsn))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 14:53
L9	1589	read\$3 with ((most more) near3 recent) with (data version)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 15:18

L10	81	(flash eeprom non \$volatile) same 9	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 15:18
L11	87775	(flash eeprom non \$volatile) same (block page)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 15:43
L12	20592	partial with (program\$5 writing write)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 15:44
L13	340	11 same 12	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 15:44
L14	44	13 and (metablock)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 15:45

10/16/08 5:20:01 PM

C:\Documents and Settings\ndinh\My Documents\EAST\workspaces\11250238.wsp

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1	"20020024756"	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 12:10
L2	1	"6288862".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 12:14
L3	1	"7275714".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 12:21
L4	1	"7257714".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 12:22
L5	1	"6547130".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 12:24
L6	8565	((most new\$3 more) near2 recent) with (sector page block cell)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 12:54
L7	15893	("same" identical common) near3 (logical)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 12:55
L8	404	6 with (read reading)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 12:56
L9	16273	7 ssame 8	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 12:56
L10	8	7 same 8	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 12:56
L11	851	6 same (read reading)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 12:58
L12	8	7 same 11	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 13:03
L13	116	7 and 11	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 13:03

L14	17709	((most new\$3 more) near2 recent) with (time count time\$stamp)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 13:11
L15	1141	6 same 14	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 13:12
L16	412	((most new\$3 more) near2 recent) with (sector page block cell) same (time \$stamp count)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 13:12
L17	1	"6948026".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 13:22
L18	1	"6684289".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/03/13 13:33

3/13/08 2:11:15 PM

C:\Documents and Settings\ndinh\My Documents\EAST\workspaces\11250238.wsp

U.S. Department of Commerce, Patent and Trademark	Atty. Docket No.	Application No.
INFORMATION DISCLOSURE STATEMENT BY APPLICANT	SNDK.156US2	11/250,238
	Applicants	Conf. No.
(Use several sheets if necessary)	Conley	7727
(Form PTO-1449)	Filing Date	Art Group
	October 13, 2005	2189

U.S. Patent Documents


*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate
/N.D./	1	5,388,248	2/7/1995	Robinson et al.			
	2	5,544,356	8/6/1996	Robinson et al.			
	3	5,627,783	5/6/1997	Miyauchi			
	4	5,682,499	10/28/1997	Bakke et al.			
	5	5,740,396	4/14/1998	Mason			
	6	5,822,781	10/13/1998	Wells et al.			
	7	5,867,417	2/2/1999	Wallace et al.			
	8	6,202,138 B1	3/13/2001	Estakhri et al.			
	9	6,219,752 B1	4/17/2001	Sekido			
	10	6,262,918 B1	6/17/2001	Estakhri et al.			
	11	6,288,862 B1	9/11/2001	Baron et al.			
	12	6,330,633 B1	12/11/2001	Kusakabe et al.			
	13	6,584,579 B1	6/24/2003	Komatsu et al.			
	14	6,725,321 B1	4/20/2004	Sinclair et al.			
	15	6,845,438 B1	1/18/2005	Tanaka et al.			
	16	6,925,012 B2	8/2/2005	Yamagami et al.			
/N.D./	17	7,167,944 B1	1/23/2007	Estakhri			

Foreign Patent Documents

							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
/N.D./	18	EP 1 352 394 B1	5/24/2006	Europe				
	19	JP 8-212019 A	8/20/1996	Japan			Abstract	
	20	JP 8-212019 A	8/20/1996	Japan			X	
	21	JP 11-110300 A	4/23/1999	Japan			Abstract	
	22	JP P3070539 B2	7/31/2000	Japan			X	
	23	JP 2000-163302 A	6/16/2000	Japan			Abstract	
/N.D./								

FILED VIA EFS
Sheet 1 of 2


U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.	Application No.
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		SNDK.156US2	11/250,238
		Applicants	Conf. No.
(Use several sheets if necessary)		Conley	7727
(Form PTO-1449)		Filing Date	Art Group
		October 13, 2005	2189
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)			
/N.D./	24	Japanese Patent Office, "Notification of Reasons for Refusal," corresponding Japanese Patent Application No. 2002-558275, mailed on May 13, 2008, 3 pages.	
↓	25	Korean Patent Office, "Office Action," corresponding Korean Patent Application No. 2003-7009551, mailed on August 25, 2008, 14 pages (including translation.)	
↓	26	PCMCIA, PC Card Standard 6.0. PCMCIA March 1997, Volume 1, Overview and Glossary, 29 pages.	
↓	27	PCMCIA, PC Card Standard 6.0. PCMCIA March 1997, Volume 3, Physical Specification, 61 pages.	
↓	28	PCMCIA, PC Card Standard 6.0. PCMCIA March 1997, Volume 7, Media Formats Storage, Specification, 39 pages.	
↓	29	Barre, "Flash Memory Magnetic Disk Replacement," IEEE Transactions on Magnetics, Vol. 29, No. 6, November 1993, 4 pages.	
↓	30	Chan, "World's Smallest Solid State Storage Device Sets New Industry Standard," IEEE, November 4-6, 1997, pp. 484-487.	
↓	31	Wells et al. "Flash Solid State Drive with 6MB/s Read/Write Channel and Data Compression," ISSCC 93/Session 3/Non-Volatile, Dynamic, and Experimental Memories/Paper WP 3.6, 4 pages.	
↓	32	United States International Trade Commission, In the Matter of Certain Flash Memory Controllers, Drivers, Memory Cards, and Media Palyers and Products Containing same, Inv. No. 337-TA-619, Order No. 33: Order Construing the Terms of the Asserted Claims of the Patents at Issue, July 15, 2008, pages i-iii, 1-10 and 54-67.	
/N.D./			
Examiner	/Ngoc Dinh/	Date Considered	12/23/2008
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.			

Index of Claims 	Application/Control No. 11250238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.
	Examiner NGOC V DINH	Art Unit 2188

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claims renumbered in the same order as presented by applicant
 CPA
 T.D.
 R.1.47


CLAIM		DATE									
Final	Original	12/02/2008									
	1	-									
	2	-									
	3	-									
1	4	=									
	5	-									
	6	-									
2	7	=									
3	8	=									
4	9	=									
5	10	=									
6	11	=									
7	12	=									
18	13	=									
19	14	=									
20	15	=									
21	16	=									
22	17	=									
23	18	=									
	19	-									
	20	-									
	21	-									
	22	-									
	23	-									
	24	-									
	25	-									
8	26	=									
9	27	=									
10	28	=									
11	29	=									
12	30	=									
13	31	=									
14	32	=									
15	33	=									
16	34	=									
17	35	=									
24	36	=									

Index of Claims 	Application/Control No. 11250238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.
	Examiner NGOC V DINH	Art Unit 2188

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claims renumbered in the same order as presented by applicant
 CPA
 T.D.
 R.1.47


CLAIM		DATE									
Final	Original	12/02/2008									
25	37	=									
26	38	=									
27	39	=									
28	40	=									
29	41	=									
30	42	=									
31	43	=									
32	44	=									
33	45	=									
34	46	=									
35	47	=									
36	48	=									
37	49	=									
38	50	=									
39	51	=									
40	52	=									
41	53	=									
42	54	=									
43	55	=									
45	56	=									
44	57	=									
46	58	=									
49	59	=									
50	60	=									
51	61	=									
52	62	=									
53	63	=									
54	64	=									
55	65	=									
56	66	=									
57	67	=									
60	68	=									
61	69	=									
62	70	=									
47	71	=									
58	72	=									

<i>Index of Claims</i> 	Application/Control No. 11250238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.
	Examiner NGOC V DINH	Art Unit 2188

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claims renumbered in the same order as presented by applicant
 CPA
 T.D.
 R.1.47

CLAIM		DATE								
Final	Original	12/02/2008								
48	73	=								
59	74	=								

Issue Classification 	Application/Control No. 11250238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.
	Examiner NGOC V DINH	Art Unit 2188

ORIGINAL						INTERNATIONAL CLASSIFICATION														
CLASS			SUBCLASS			CLAIMED					NON-CLAIMED									
711			103			G	0	6	F	9 / 24 (2006.01.01)										
CROSS REFERENCE(S)																				
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)																			
711	113		115																	

<input type="checkbox"/> Claims renumbered in the same order as presented by applicant <input type="checkbox"/> CPA <input checked="" type="checkbox"/> T.D. <input type="checkbox"/> R.1.47															
Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
	1	22	17	15	33	37	49	55	65						
	2	23	18	16	34	38	50	56	66						
	3		19	17	35	39	51	57	67						
1	4		20	24	36	40	52	60	68						
	5		21	25	37	41	53	61	69						
	6		22	26	38	42	54	62	70						
2	7		23	27	39	43	55	47	71						
3	8		24	28	40	45	56	58	72						
4	9		25	29	41	44	57	48	73						
5	10	8	26	30	42	46	58	59	74						
6	11	9	27	31	43	49	59								
7	12	10	28	32	44	50	60								
18	13	11	29	33	45	51	61								
19	14	12	30	34	46	52	62								
20	15	13	31	35	47	53	63								
21	16	14	32	36	48	54	64								

/NGOC V DINH/ Examiner.Art Unit 2188 (Assistant Examiner)	12/02/08 (Date)	Total Claims Allowed: 62	
/Hyung S Sough/ Supervisory Patent Examiner.Art Unit 2188 (Primary Examiner)	12/22/2008 (Date)	O.G. Print Claim(s) 1	O.G. Print Figure 14



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

66785 7590 01/08/2009

DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION
505 MONTGOMERY STREET
SUITE 800
SAN FRANCISCO, CA 94111

EXAMINER

DINH, NGOC V

ART UNIT PAPER NUMBER

2188

DATE MAILED: 01/08/2009

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.

11/250,238 10/13/2005 Kevin M. Conley SNDK.156US2 7727

TITLE OF INVENTION: PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY

Table with 7 columns: APPLN. TYPE, SMALL ENTITY, ISSUE FEE DUE, PUBLICATION FEE DUE, PREV. PAID ISSUE FEE, TOTAL FEE(S) DUE, DATE DUE

nonprovisional NO \$1510 \$300 \$0 \$1810 04/08/2009

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

- A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.
B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

- A. Pay TOTAL FEE(S) DUE shown above, or
B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PART B - FEE(S) TRANSMITTAL

**Complete and send this form, together with applicable fee(s), to: Mail Mail Stop ISSUE FEE
 Commissioner for Patents
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 or Fax (571)-273-2885**

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

66785 7590 01/08/2009

DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION
 505 MONTGOMERY STREET
 SUITE 800
 SAN FRANCISCO, CA 94111

Certificate of Mailing or Transmission
 I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

_____ (Depositor's name)
_____ (Signature)
_____ (Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/250,238	10/13/2005	Kevin M. Conley	SNDK.156US2	7727

TITLE OF INVENTION: PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	\$1510	\$300	\$0	\$1810	04/08/2009

EXAMINER	ART UNIT	CLASS-SUBCLASS
DINH, NGOC V	2188	711-103000

<p>1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).</p> <p><input type="checkbox"/> Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.</p> <p><input type="checkbox"/> "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. Use of a Customer Number is required.</p>	<p>2. For printing on the patent front page, list</p> <p>(1) the names of up to 3 registered patent attorneys or agents OR, alternatively, 1 _____</p> <p>(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed. 2 _____</p> <p>3 _____</p>
---	---

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE _____ (B) RESIDENCE: (CITY and STATE OR COUNTRY) _____

Please check the appropriate assignee category or categories (will not be printed on the patent): Individual Corporation or other private group entity Government

<p>4a. The following fee(s) are submitted:</p> <p><input type="checkbox"/> Issue Fee</p> <p><input type="checkbox"/> Publication Fee (No small entity discount permitted)</p> <p><input type="checkbox"/> Advance Order - # of Copies _____</p>	<p>4b. Payment of Fee(s): (Please first reapply any previously paid issue fee shown above)</p> <p><input type="checkbox"/> A check is enclosed.</p> <p><input type="checkbox"/> Payment by credit card. Form PTO-2038 is attached.</p> <p><input type="checkbox"/> The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).</p>
---	--

5. Change in Entity Status (from status indicated above)

a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27. b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.
Row 1: 11/250,238, 10/13/2005, Kevin M. Conley, SNDK.156US2, 7727
Row 2: 66785, 7590, 01/08/2009, [EXAMINER], []
Row 3: [] [] [] [] []
Row 4: [] [] [] [] []
Row 5: [] [] [] [] []
Row 6: [] [] [] [] []
Row 7: [] [] [] [] []
Row 8: [] [] [] [] []
Row 9: [] [] [] [] []
Row 10: [] [] [] [] []
Row 11: [] [] [] [] []
Row 12: [] [] [] [] []
Row 13: [] [] [] [] []
Row 14: [] [] [] [] []
Row 15: [] [] [] [] []
Row 16: [] [] [] [] []
Row 17: [] [] [] [] []
Row 18: [] [] [] [] []
Row 19: [] [] [] [] []
Row 20: [] [] [] [] []
Row 21: [] [] [] [] []
Row 22: [] [] [] [] []
Row 23: [] [] [] [] []
Row 24: [] [] [] [] []
Row 25: [] [] [] [] []
Row 26: [] [] [] [] []
Row 27: [] [] [] [] []
Row 28: [] [] [] [] []
Row 29: [] [] [] [] []
Row 30: [] [] [] [] []
Row 31: [] [] [] [] []
Row 32: [] [] [] [] []
Row 33: [] [] [] [] []
Row 34: [] [] [] [] []
Row 35: [] [] [] [] []
Row 36: [] [] [] [] []
Row 37: [] [] [] [] []
Row 38: [] [] [] [] []
Row 39: [] [] [] [] []
Row 40: [] [] [] [] []
Row 41: [] [] [] [] []
Row 42: [] [] [] [] []
Row 43: [] [] [] [] []
Row 44: [] [] [] [] []
Row 45: [] [] [] [] []
Row 46: [] [] [] [] []
Row 47: [] [] [] [] []
Row 48: [] [] [] [] []
Row 49: [] [] [] [] []
Row 50: [] [] [] [] []
Row 51: [] [] [] [] []
Row 52: [] [] [] [] []
Row 53: [] [] [] [] []
Row 54: [] [] [] [] []
Row 55: [] [] [] [] []
Row 56: [] [] [] [] []
Row 57: [] [] [] [] []
Row 58: [] [] [] [] []
Row 59: [] [] [] [] []
Row 60: [] [] [] [] []
Row 61: [] [] [] [] []
Row 62: [] [] [] [] []
Row 63: [] [] [] [] []
Row 64: [] [] [] [] []
Row 65: [] [] [] [] []
Row 66: [] [] [] [] []
Row 67: [] [] [] [] []
Row 68: [] [] [] [] []
Row 69: [] [] [] [] []
Row 70: [] [] [] [] []
Row 71: [] [] [] [] []
Row 72: [] [] [] [] []
Row 73: [] [] [] [] []
Row 74: [] [] [] [] []
Row 75: [] [] [] [] []
Row 76: [] [] [] [] []
Row 77: [] [] [] [] []
Row 78: [] [] [] [] []
Row 79: [] [] [] [] []
Row 80: [] [] [] [] []
Row 81: [] [] [] [] []
Row 82: [] [] [] [] []
Row 83: [] [] [] [] []
Row 84: [] [] [] [] []
Row 85: [] [] [] [] []
Row 86: [] [] [] [] []
Row 87: [] [] [] [] []
Row 88: [] [] [] [] []
Row 89: [] [] [] [] []
Row 90: [] [] [] [] []
Row 91: [] [] [] [] []
Row 92: [] [] [] [] []
Row 93: [] [] [] [] []
Row 94: [] [] [] [] []
Row 95: [] [] [] [] []
Row 96: [] [] [] [] []
Row 97: [] [] [] [] []
Row 98: [] [] [] [] []
Row 99: [] [] [] [] []
Row 100: [] [] [] [] []

DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION
505 MONTGOMERY STREET
SUITE 800
SAN FRANCISCO, CA 94111

EXAMINER
DINH, NGOC V
ART UNIT
PAPER NUMBER
2188
DATE MAILED: 01/08/2009

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 0 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 0 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

Notice of Allowability	Application No.	Applicant(s)	
	11/250,238	CONLEY, KEVIN M.	
	Examiner	Art Unit	
	NGOC V. DINH	2188	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. This communication is responsive to amendment filed 11/21/08.
2. The allowed claim(s) is/are 4, 7-18, 26-74 (renumbered as 1-62).
3. Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some* c) None of the:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
5. CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
 - (a) including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
 - 1) hereto or 2) to Paper No./Mail Date _____.
 - (b) including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

- | | |
|--|---|
| <ol style="list-style-type: none"> 1. <input type="checkbox"/> Notice of References Cited (PTO-892) 2. <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) 3. <input checked="" type="checkbox"/> Information Disclosure Statements (PTO/SB/08),
Paper No./Mail Date <u>10/15/08, 11/13/08, 12/24/08</u> 4. <input type="checkbox"/> Examiner's Comment Regarding Requirement for Deposit of Biological Material | <ol style="list-style-type: none"> 5. <input type="checkbox"/> Notice of Informal Patent Application 6. <input type="checkbox"/> Interview Summary (PTO-413),
Paper No./Mail Date _____. 7. <input checked="" type="checkbox"/> Examiner's Amendment/Comment 8. <input checked="" type="checkbox"/> Examiner's Statement of Reasons for Allowance 9. <input type="checkbox"/> Other _____. |
|--|---|

--	--

DETAILED ACTION

1. This office action is a response to the amendment filed on 11/21/08.

INFORMATION DISCLOSURE STATEMENT

2. The Applicant's submission of the IDS filed 10/15/2008, 11/13/08 and 12/24/08 have been considered, except those references are crossed. The examiner has also corrected some of the references to include the date. As required by M.P.E.P. 609 C(2), copies of the PTOL-1449 is attached to the instant office action.

EXAMINER'S AMENDMENT

3. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it **MUST** be submitted no later than the payment of the issue fee.

Authorization for this Examiner's Amendment was given in a telephone interview with Gerald Parsons (Reg. 24,486) on 11/13/08 and 11/02/08.

In the claim section:

The supplemental Amendment to the claims filed on 11/21/2008 by the applicant has been reviewed and is accepted.

REASON FOR ALLOWANCE

4. The primary reasons for allowance of claims 4, 13, 46, 59, 68, 69 in the instant application is the combination with the inclusion of at least the limitations set forth in lines 10-13 of claims 4, 13; lines 15-19 of claim 46; lines 13-17 of claim 59; lines 11-17 of claim 68; lines 8-11 of claim 69.

Because claims 7-12, 26-35; 14-18, 36-45; 47-58, 71, 73; 60-67, 72, 74 and 70 depend directly or indirectly on claims 4, 13, 46, 59, 69. These claims are considered allowable for at least the same reasons noted above.

Conclusion

5. Any response to this action should be mailed to:

Under Secretary of Commerce for intellectual Property and Director of the
United States Patent and Trademark Office
PO Box 1450
Alexandria, VA 22313-1450

or faxed to:

(571) 273-8300, (for Official communications intended for entry).

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PMR) system. Status information for published Applications may be obtained from either Private PMR or Public PMR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pak-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ngoc Dinh whose telephone number is (571) 272-4191. The examiner can normally be reached on Monday-Friday 8:30 AM-5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung Sough, can be reached on (571) 272-6799.


/N. V. D./

Examiner, Art Unit 2188

/Hyung S. Sough/

Supervisory Patent Examiner, Art Unit 2188

12/22/08

Search Notes 	Application/Control No. 11250238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.
	Examiner NGOC V DINH	Art Unit 2188

SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES		
Search Notes	Date	Examiner
Limited classified search of Class/subclass. East text search w/o classified/search. See printout.	12/2/08	ND
Consulted Kevin Verbrugge (AU 2189).	10/23/08	ND
Consulted Kevin Ellis.	10/29/08	ND

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner
711/103, 711/113, 711/115	Intereference search EAST.	12/2/08	ND

--	--

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1746	("same" identical common) near5 (logical adj2 address)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/12 16:19
L2	581446	(eeprom flash non \$volatile)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/12 16:20
L3	16563	(programming program\$4 programing update updat\$3) same (time \$stamp (time adj stamp))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/12 16:20
L4	402	L2 same L3	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/12 16:20
L5	6	(most near2 recent near3 (page block sector)) same L4	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/12 16:20
L6	2131	(eeprom eprom flash non\$volatile) with separate with (unit portion section module)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/12 16:20
L7	2131	(eeprom eprom flash non\$volatile) with separate with (unit portion section module)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/12 16:20
L8	52	((program\$4 write writting written) with (offset)) and L7	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/12 16:20
L9	146	(erase eras\$3) with metablock	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/12 16:21
L10	0	7 same 9	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/12 16:21
L11	2	7 and 9	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/12 16:21

EAST Search History

L12	52	6 and 8	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/12 16:21
L13	0	4 and offset and L7	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/12 16:22
L14	19	3 and 9	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/12 16:22

11/12/08 4:22:46 PM

C:\Documents and Settings\ndinh\My Documents\EAST\workspaces\11250238.wsp



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
 United States Patent and Trademark Office
 Address: COMMISSIONER FOR PATENTS
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 7727

SERIAL NUMBER 11/250,238	FILING or 371(c) DATE 10/13/2005 RULE	CLASS 711	GROUP ART UNIT 2189	ATTORNEY DOCKET NO. SNDK.156US2		
APPLICANTS Kevin M. Conley, San Jose, CA; ** CONTINUING DATA ***** This application is a CON of 10/841,388 05/07/2004 PAT 6,968,421 which is a CON of 09/766,436 01/19/2001 PAT 6,763,424 ** FOREIGN APPLICATIONS ***** ** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ** 11/03/2005						
Foreign Priority claimed <input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	35 USC 119(a-d) conditions met <input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	<input type="checkbox"/> Met after Allowance Initials _____	STATE OR COUNTRY CA	SHEETS DRAWINGS 9	TOTAL CLAIMS 3	INDEPENDENT CLAIMS 2
ADDRESS DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION 505 MONTGOMERY STREET SUITE 800 SAN FRANCISCO, CA 94111 UNITED STATES						
TITLE PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY						
FILING FEE RECEIVED 1000	FEES: Authority has been given in Paper No. _____ to charge/credit DEPOSIT ACCOUNT No. _____ for following:		<input type="checkbox"/> All Fees <input type="checkbox"/> 1.16 Fees (Filing) <input type="checkbox"/> 1.17 Fees (Processing Ext. of time) <input type="checkbox"/> 1.18 Fees (Issue) <input type="checkbox"/> Other _____ <input type="checkbox"/> Credit			

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	87775	(flash eeprom eeprom non \$volatile) same (block page)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 09:34
L2	2320	(read\$3 write writing updat \$3) with (("same" identical common) near4 (logical lba lbn lsa lsn))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 09:37
L3	4890	(read\$3 write writing updat \$3) same (("same" identical common) near4 (logical lba lbn lsa lsn))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 09:37
L4	4143	read\$3 with ((most more) near3 recent)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 09:39
L5	736	1 and 2 and 3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 09:39
L6	21	3 same 4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 09:39
L7	736	1 and 2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 09:39

EAST Search History

L8	18	1 and 6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 09:40
L9	644	(write writing written program\$4) with (different near3 offset)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 09:53
L10	2	1 same 9	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 09:53
L11	137	1 and 9	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/10/16 09:53

10/16/08 10:28:47 AM

C:\Documents and Settings\ndinh\My Documents\EAST\workspaces\11250238.wsp

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1	"5860124".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 13:08
L2	85629	read\$3 same ((most more updated) near3 (version data block sector))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 14:26
L3	39392	read\$3 with ((most more updated) near3 (version data block sector))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 14:26
L4	116	3 same ((identical "same" common) near4 (logical virtual lba lsa lbn))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 14:28
L5	40808	read\$3 with ((most more updated) near3 (version data block sector page))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 14:33
L6	589277	(eprom eprom flash non \$volatile)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 14:34
L7	2907	5 same 6	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 14:34
L8	1700	5 with 6	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 14:34
L9	17	7 same ((identical "same" common) near4 (logical virtual lba lsa lbn))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 14:35

L10	323	7 and ((identical "same" common) near4 (logical virtual lba lsa lbn))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 14:39
L11	2604	read\$3 with ((most more) with (updated recent) with (version data block sector page))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 14:41
L12	24	11 same ((identical "same" common) near4 (logical virtual lba lsa lbn))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 14:41
L13	158	11 same 6	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/10/20 14:48

10/20/08 3:22:19 PM

C:\Documents and Settings\ndinh\My Documents\EAST\workspaces\11250238.wsp

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	595865	(prom eeprom flash non\$volatile)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:12
L2	143	(erase eras\$3) with metablock	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:13
L3	88	1 same 2	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:13
L4	1740	("same" identical common) near5 (logical adj2 address)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:19
L5	4256	711/103 711/711/102	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:19
L6	1967	((new old superceded updated original modified) with (version data)) same (logical near2 address)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:20
L8	566	1 same 6	US-PGPUB; USPAT; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/11/06 08:21
L9	307	5 and 8	US-PGPUB; USPAT; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/11/06 08:21
L10	7101	711/103 711/200 711/203	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:21
L11	25262	(write writing written programm \$3) same (recent timestamp)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:21

L12	437	L10 and L11	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:21
L13	437	10 and 12	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:21
L14	248455	(original old out-of-date superceded) same (replacement updated new)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:21
L15	1740	("same" identical common) near5 (logical adj2 address)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:21
L16	227	L14 same L15	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:21
L17	126	10 and 16	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:22
L18	288	((most least) near3 (recently) near3 (block page sector)) with (time timing (time \$stamp))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:22
L19	1122	((program\$4 write writting written) with (offset) same 1	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:23
L20	0	15 same 19	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:23
L21	62	15 and 19	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:23
L22	57603	(sanitiz\$4 eras\$4 programm\$4) near5 (part portion partial partially)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:23
L23	7	19 same 22	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:24
L24	406	19 and 22 and 1	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:24

L25	594515	(eprom eprom flash non\$volatile)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:24
L26	121	metablock same L25	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:24
L27	98	10 and 26	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:24
L28	98	27 and (eras\$3)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:25
L29	2128	(eprom eprom flash non\$volatile) with separate with (unit portion section module)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:26
L30	3	((program\$4 write writting written) with (offset)) same 29	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:26
L31	52	((program\$4 write writting written) with (offset)) and 29	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2008/11/06 08:26

11/6/08 8:27:13 AM

C:\Documents and Settings\ndinh\My Documents\EAST\workspaces\11250238.wsp

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Kevin M. Conley		
Title:	Partial Block Data Programming And Reading Operations In A Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	October 13, 2005
Examiner:	Dinh, Ngoc V.	Group Art Unit:	2189
Docket No.:	0084567-156US2	Conf. No.:	7727

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUPPLEMENTAL AMENDMENT

Sir:

This Supplemental Amendment is being filed concurrently with a Request for Continued Examination (RCE) of the above-referenced patent application, which has been allowed. A Supplemental Information Disclosure Statement is also being filed herewith.

Claim amendments are reflected in the listing of claims, which begins on page 2 of this paper.

Remarks begin on page 19 of this paper.

Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

CLAIM AMENDMENTS

Please amend the claims by amending claims 55, 57-59, 64, 69, 71 and 73 and adding new dependent claims 75-84, all without prejudice, as indicated on the following listing of all the claims in the present application after this Amendment:

1 – 3. (Cancelled)

4. (Previously Presented) A method of operating a memory system of reprogrammable non-volatile charge storage elements organized in blocks of a minimum number of storage elements that are erasable together as a unit and in pages of storage elements within the blocks that have specified offset positions within their respective blocks and which are individually programmable as a unit, comprising:

as part of writing data into pages, recording an indication of a time from a clock source that data are written into individual pages,

updating data previously written into one or more pages of one of original data blocks by writing the updated data into one or more update pages of an update data block and logically linking data of the corresponding previously written pages and update data pages, wherein the updated data are caused to be writable into pages of the update data block having different offset positions than the pages of the original data block into which the logically linked data being updated was previously written, and

when reading data of two or more logically linked data pages, read the indications of the times that the data have been stored in the two or more pages and use the data in the two or more pages having more recent time indications without using data in the two or more pages having older time indications.

5 – 6. (Cancelled)

7. (Previously Presented) The method of claim 4, wherein recording the indication of the time that data are written into individual pages includes recording the indication within the pages wherein the data are written.

8. (Previously Presented) The method of claim 4, wherein updating data previously written into one or more pages of one of the original data blocks includes updating data in a number of pages less than all of the pages of the one of the original data blocks while not updating data in a remaining one or more pages of the same original data block.

9. (Previously Presented) The method of claim 8, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

10. (Previously Presented) The method of claim 4, wherein as part of updating data, logically linking the corresponding previously written pages and update data pages comprises writing logical addresses of the updated data in the individual update pages in which the updated data are written.

11. (Previously Presented) The method of claim 4, wherein as part of writing data into pages, data are written in individual storage elements of the pages with more than two storage states, thereby storing more than one bit of data in the individual storage elements.

12. (Previously Presented) The method of claim 4, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

13. (Previously Presented) A method of operating a memory system of reprogrammable non-volatile charge storage elements organized in blocks of a minimum number of storage elements that are erasable together as a unit and in pages of storage elements within

the blocks that have specified offset positions within their respective blocks and which are individually programmable as a unit, comprising:

as part of writing data into pages, data are written into the pages of individual blocks in sequence,

updating data previously written into one or more initial pages of one of original data blocks by writing the updated data into one or more update pages of an update data block and logically linking data of previously written pages with corresponding update data pages, wherein the updated data are caused to be writable into pages of the update data block having different offset positions as the pages of the original data block into which the logically linked data being updated was previously written, and

reading data from the pages of the original data block and update data block in an order that is a reverse of the sequence in which they were written and ignore data in any page that is logically linked with a page from which data have already been read.

14. (Previously Presented) The method of claim 13, wherein updating data previously written into one or more pages of one of the original data blocks includes updating data in a number of pages less than all of the pages of the one of the original data blocks while not updating data in a remaining one or more pages of the same original data block.

15. (Previously Presented) The method of claim 14, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

16. (Previously Presented) The method of claim 13, wherein as part of updating data into pages, logically linking the corresponding previously written pages and update data pages comprises writing logical addresses of updated data in the individual update pages in which the updated data are written.

17. (Previously Presented) The method of claim 13, wherein as part of writing data into pages, data are written in individual storage elements of the pages with more than two storage states, thereby storing more than one bit of data in the individual storage elements.

18. (Previously Presented) The method of claim 13, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

19 – 25. (Cancelled)

26. (Previously Presented) The method of claim 4, wherein logically linking the corresponding previously written pages and update data pages comprises maintaining common logical addresses thereof.

27. (Previously Presented) The method of claim 26, wherein maintaining common logical addresses comprises storing the common logical addresses in the pages of the original data block and in the pages of the update data block along with the respective previously written and updated data therein.

28. (Previously Presented) The method of claim 26, wherein reading the data further comprises organizing the read data by the logical page addresses associated with the read data.

29. (Previously Presented) The method of claim 4, wherein data are written into individual pages of the original and update data blocks in a specified sequence.

30. (Previously Presented) The method of claim 4, wherein the original data previously written into one or more pages of one of original data blocks are written into pages of first and second blocks positioned in different ones of a plurality of units of the memory system, wherein the units are physically separate groupings of blocks of charge storage elements in which programming operations may be performed independently, and linking the first and

second blocks to cause their charge storage elements to be erasable together and their pages to be programmable together in parallel.

31. (Previously Presented) The method of claim 4, additionally comprising:
updating data previously written into one or more pages of a second of the original data blocks by writing the updated data into one or more update pages of the update data block, and
logically linking data of the corresponding previously written pages of the second of the original data blocks and the update data pages.

32. (Previously Presented) The method of claim 31, wherein updating data previously written into one or more pages of the one and of the second of the original data blocks includes updating data in a number of pages less than all of the pages of the individual original data block while not updating data in a remaining one or more pages of the same individual original data block.

33. (Previously Presented) The method of claim 4, additionally comprising:
subsequently updating for a second time at least some of the data previously written into one or more pages of one of the original data blocks that were previously updated and written into the update data block,
writing the data updated for a second time into the update data block as second updated data pages, and
logically linking data of the corresponding previously written pages and the second updated data pages.

34. (Previously Presented) The method of claim 33, wherein updating data previously written into one or more pages of one of the original data blocks and subsequently updating the data for the second time each includes updating data in a number of pages less than all of the pages of the individual original data block while not updating data in a remaining one or more pages of said individual original data block.

35. (Previously Presented) The method of claim 4, wherein the memory system in which the method is practiced is contained within an enclosed card having an electrical connector along one edge thereof for operably connecting with a host system.

36. (Previously Presented) The method of claim 13, wherein logically linking the corresponding previously written pages and update data pages comprises maintaining common logical addresses thereof.

37. (Previously Presented) The method of claim 36, wherein maintaining common logical addresses comprises storing the common logical addresses in the pages of the original data block and in the pages of the update data block along with the respective previously written and updated data therein.

38. (Previously Presented) The method of claim 36, wherein reading the data further comprise organizing the read data by the logical page addresses associated with the read data.

39. (Previously Presented) The method of claim 13, wherein data are written into individual pages of the blocks in a specified sequence.

40. (Previously Presented) The method of claim 13, wherein the data written into one or more initial pages of one of the original data are written into pages of first and second blocks positioned in different ones of a plurality of units of the memory system, wherein the units are physically separate groupings of blocks of charge storage elements in which programming operations may be performed independently, and linking the first and second blocks to cause their charge storage elements to be erasable together and their pages to be programmable together in parallel.

41. (Previously Presented) The method of claim 13, additionally comprising:
updating data previously written into one or more pages of a second of the original data blocks by writing the updated data into one or more update pages of the update data block, and

logically linking data of the corresponding previously written pages of the second of the original data blocks and the update data pages.

42. (Previously Presented) The method of claim 41, wherein updating data previously written into one or more pages of the one and of the second of the original data blocks includes updating data in a number of pages less than all of the pages of the individual original data block while not updating data in a remaining one or more pages of the same individual original data block.

43. (Previously Presented) The method of claim 13, additionally comprising:
subsequently updating for a second time at least some of the data previously written into one or more pages of one of the original data blocks that were previously updated and written into the update data block,

writing the data updated for a second time into the update data block as second updated data pages, and

logically linking data of the corresponding previously written pages and the second updated data pages.

44. (Previously Presented) The method of claim 43, wherein updating data previously written into one or more pages of one of the original data blocks and subsequently updating the data for the second time each includes updating data in a number of pages less than all of the pages of the individual original data block while not updating data in a remaining one or more pages of the same individual original data block.

45. (Previously Presented) The method of claim 13, wherein the memory system in which the method is practiced is contained within an enclosed card having an electrical connector along one edge thereof for operably connecting with a host system.

46. (Previously Presented) In a re-programmable non-volatile semiconductor memory system having a plurality of blocks of a minimum number of memory charge storage

elements that are erasable together as a unit, the plurality of blocks individually being divided into a plurality of a given number of pages of memory storage elements that are individually programmable as a unit and which have specified offset positions within their respective blocks, a method of operating the memory system, comprising:

programming original data into individual ones of a first plurality of pages in at least a first block, the pages of original data having logical addresses associated therewith,

thereafter programming, into individual ones of a second plurality of pages in a second block, an updated version of less than the given number of pages of the original data programmed into the first plurality of pages, the pages of the updated version of the original data having logical addresses associated therewith, wherein the logical addresses associated with the pages of the updated version of the original data are the same as the logical addresses associated with the pages of original data,

wherein programming the second plurality of pages additionally comprises programming the updated version of the original data in those of the second plurality of pages that have different offset positions within the second block than the offset positions of the first plurality of pages within said at least the first block that contain pages of original data with the same associated logical addresses,

thereafter reading data from the first and second plurality of pages, and organizing pages of the read data by their associated logical addresses.

47. (Previously Presented) The method of claim 46, wherein reading data and organizing pages of the read data by their associated logical addresses comprises, for the pages of read data having the same logical addresses associated therewith, utilizing the pages of the updated version of the original data and omitting use of the pages of original data.

48. (Previously Presented) The method of claim 47, wherein reading data and organizing pages of the read data by their associated logical addresses additionally comprises utilizing the pages of original data that have not been updated.

49. (Previously Presented) The method of claim 48, wherein

programming original data into individual ones of the first plurality of pages and programming an updated version of the original data into individual ones of the second plurality of pages additionally comprises programming the individual pages with an indication of a relative time of programming the data therein, and

reading data and organizing pages of the read data additionally comprise, for pages of data having the same logical addresses associated therewith, selecting the updated data from the read pages having the most recent time indications and omitting use of the data from others of the pages.

50. (Previously Presented) The method of claim 46, wherein programming data into the first plurality of pages in at least the first block and programming data into the second plurality of pages in the second block each comprise programming data into individual pages of the individual blocks in a specified sequence.

51. (Previously Presented) The method of claim 50, wherein reading data and organizing pages of the read data comprise reading the first and second plurality of pages in an order that is reverse to said specified sequence, and ignoring data read from pages having logical addresses that are the same as logical addresses associated with other pages of data that have already been read.

52. (Previously Presented) The method of claim 46, wherein the logical addresses associated with the pages of original data individually includes a logical block number and a logical page offset.

53. (Previously Presented) The method of claim 46, wherein programming original data into the first plurality of pages and programming the updated version of the original data into the second plurality of pages each comprise programming the logical addresses in individual pages along with the data with which the programmed logical addresses are associated.

54. (Previously Presented) The method of claim 46, wherein programming original data into individual ones of the first plurality of pages in at least a first block includes programming the original data into a first plurality of pages in at least the first block and a third one of the blocks of charge storage elements located in different ones of a plurality of units of the memory system, wherein the plurality of units are physically separate groupings of blocks of charge storage elements in which programming operations may be performed independently, and linking the first and third blocks together as a metablock to cause their charge storage elements to be erasable together and their pages to be programmable together in parallel.

55. (Currently Amended) The method of any one of claims 46 – 54, additionally comprising:

subsequently programming, into individual ones of the second plurality of pages, a further updated version of at least some of the original data programmed into the first plurality of pages that have previously been updated and the updated version programmed into the second plurality of pages,

wherein the logical addresses associated with the pages of the further updated version of the original data are the same as the logical addresses associated with the pages of original data and the pages of the previously updated version of the original data.

56. (Previously Presented) The method of claim 46, wherein the charge storage elements of the memory system in which the method is carried out comprise electrically conductive floating gates.

57. (Currently Amended) The method of any one of claims 46 – 54, additionally comprising operating the individual memory system charge storage elements with more than two storage states, thereby storing more than one bit of data in each storage element, wherein programming the pages includes programming the individual memory storage elements into more than two storage states and reading data includes reading the more than two storage states from the individual memory storage elements.

58. (Currently Amended) The method of any one of claims 46 – 54, wherein the memory system in which the method is practiced is contained within an enclosed card having an electrical connector along one edge thereof for operably connecting the memory system with a host system.

59. (Currently Amended) In a re-programmable non-volatile semiconductor memory system having a plurality of blocks of memory charge storage elements that are erasable together as a unit, the plurality of blocks individually being divided into a plurality of a given number of pages of memory storage elements that are programmable together, a method of operating the memory system, comprising:

programming individual ones of a first plurality of ~~said given number~~ of pages in at least a first block with original data, the pages of original data having logical addresses associated therewith,

thereafter programming individual ones of a second plurality of a total number of pages less than said given number in a second block with updated data and a logical address associated with the page of updated data, wherein the logical addresses associated with the pages of updated data programmed into the second plurality of pages are the same as those associated with the pages of original data programmed into the first plurality of pages,

wherein programming the second plurality of pages additionally comprises programming the updated version of the original data in those of the second plurality of pages that have different offset positions within the second block than the offset positions of the first plurality of pages within said at least the first block that contain pages of original data with the same logical addresses associated therewith,

thereafter reading data from the first and second plurality of pages, and
organizing pages of the read data by the logical addresses associated therewith.

60. (Previously Presented) The method of claim 59, wherein reading data and organizing pages of the read data by the logical addresses associated therewith comprises, for a plurality of pages of read data having the same logical addresses associated therewith, utilizing

the pages of the updated version of the original data and omitting use of the pages of original data.

61. (Previously Presented) The method of claim 60, wherein reading data and organizing pages of the read data by the logical addresses associated therewith additionally comprises utilizing the pages of original data that have not been updated.

62. (Previously Presented) The method of claim 59, wherein programming data into the first plurality of pages in at least the first block and programming data into the second plurality of pages in the second block each comprise programming data into individual pages of the individual blocks in a specified sequence.

63. (Previously Presented) The method of claim 59, wherein the logical address associated with the page of original data includes a logical block number and a logical page offset.

64. (Currently Amended) The method of claim 59, wherein programming original data into the first plurality of pages and ~~programming the updated version of the original data into the second plurality of pages each~~ comprise additionally comprises programming in the pages the logical addresses ~~for~~ associated with the data programmed therein.

65. (Previously Presented) The method of claim 59, wherein programming original data into individual ones of the first plurality of pages in at least a first block includes programming the original data into a first plurality of pages in at least the first and a third one of the blocks of charge storage elements located in different ones of a plurality of units of the memory system, wherein the units are physically separate groupings of blocks of charge storage elements in which programming operations may be performed independently, and linking the first and third blocks together as a metablock to cause their charge storage elements to be erasable together and pages thereof to be programmable together in parallel.

66. (Previously Presented) The method of claim 59, additionally comprising:
subsequently programming, into individual ones of the second plurality of pages, a further updated version of at least some of the original data programmed into the first plurality of pages that have previously been updated and the updated version programmed into the second plurality of pages,

wherein the logical addresses associated with the further updated version of the original data are the same as the logical addresses associated with the original data and the previously updated version of the original data.

67. (Previously Presented) The method of claim 59, wherein the addresses associated with pages of original and updated data are individually expressed as a logical block number and a logical page offset.

68. (Previously Presented) A method of operating a memory system having an array of reprogrammable non-volatile charge storage elements organized in blocks of storage elements that are erasable together and in pages of storage elements within the blocks that are individually programmable as a unit, comprising:

maintaining an indication of a time separately for individual blocks,
as part of writing data into any one or more of the pages of one of the blocks, updating the indication of a time maintained for the one block to a current time,
when updating data previously written into one or more initial pages, writing the updated data into one or more update pages and identifying the data written into the initial and update pages by common logical addresses, and

as part of reading data having common logical addresses from two or more pages of two or more blocks, reading the indications of the times from the two or more blocks and using the data in the two or more blocks having more recent indications of time without using data in the two or more blocks having older indications of time, and reading data from pages within each of the two or more blocks according to a reverse order in physical address to that by which the pages were written, and ignoring data from any page having the same logical address as data of a page from which data have already been read.

69. (Currently Amended) A method of operating a non-volatile memory system having a plurality of blocks of memory storage elements that are individually erasable together, wherein the individual blocks are divided into a plurality of pages of storage elements that are programmable together, the blocks being organized in at least two separate units in which programming may be performed independently, comprising:

linking at least one block from individual ones of said at least two units to form a metablock wherein the storage elements of its component blocks are erased together, and updating one or more pages of original data within any of the metablock component blocks less than all the pages within the block by programming replacement data into pages ~~within~~ another at least one block in only a designated one of the units regardless of which unit the data being updated are stored.

70. (Previously Presented) The method of claim 69, wherein storing the original and replacement data comprises:

identifying the original and replacement data by the same logical address to the memory system, and

distinguishing the replacement data from the original data by determining the relative order in time in which the original and replacement data have been programmed in their respective pages of the memory.

71. (Currently Amended) The method of any one of claims 46 – 54, which additionally comprises maintaining a data structure of the logical addresses associated with the pages of original data and the logical addresses associated with pages of the updated version of the original data that links (a) physical addresses of a plurality of blocks of memory storage elements that are storing data having common logical addresses with (b) those common logical addresses.

72. (Previously Presented) The method of claim 59, which additionally comprises maintaining a data structure of the logical addresses associated with the pages of original data

and the logical addresses associated with pages of the updated version of the original data that links (a) physical addresses of a plurality of blocks of memory storage elements that are storing data having common logical addresses with (b) those common logical addresses.

73. (Currently Amended) The method of any one of claims 46 – 54, wherein organizing pages of the read data comprises writing the pages of read data into a volatile memory within the memory system.

74. (Previously Presented) The method of claim 59, wherein organizing pages of the read data comprises writing the pages of read data into a volatile memory within the memory system.

75. (New) The method of claim 55, which additionally comprises maintaining a data structure of the logical addresses associated with the pages of original data and the logical addresses associated with pages of the updated version of the original data that links (a) physical addresses of a plurality of blocks of memory storage elements that are storing data having common logical addresses with (b) those common logical addresses.

76. (New) The method of claim 55, wherein organizing pages of the read data comprises writing the pages of read data into a volatile memory within the memory system.

77. (New) The method of claim 55, additionally comprising operating the individual memory system charge storage elements with more than two storage states, thereby storing more than one bit of data in each storage element, wherein programming the pages includes programming the individual memory storage elements into more than two storage states and reading data includes reading the more than two storage states from the individual memory storage elements.

78. (New) The method of claim 65, additionally comprising:

subsequently programming, into individual ones of the second plurality of pages, a further updated version of at least some of the original data programmed into the first plurality of pages that have previously been updated and the updated version programmed into the second plurality of pages,

wherein the logical addresses associated with the further updated version of the original data are the same as the logical addresses associated with the original data and the previously updated version of the original data.

79. (New) The method of claim 78, wherein the addresses associated with pages of original and updated data are individually expressed as a logical block number and a logical page offset.

80. (New) The method of claim 69, wherein updating one or more pages of original data further includes programming a logical address associated with both the original and replacement data.

81. (New) The method of claim 80, wherein programming a logical address associated with both the original and replacement data includes programming both a logical block number and a logical page offset.

82. (New) The method of claim 69, wherein the memory storage elements of the non-volatile memory system in which the method is carried out comprise electrically conductive floating gates.

83. (New) The method of claim 69, additionally comprising programming the original and replacement data into the individual memory storage elements with more than two storage states, thereby storing more than one bit of data the individual memory storage elements.

84. (New) The method of claim 69, wherein the non-volatile memory system in which the method is practiced is contained within an enclosed card having an electrical

connector along one edge thereof for operably connecting the memory system with a host system.

REMARKS

Continued examination of the present application is being requested primarily in order to have considered by the Examiner the above claim amendments and information provided by an accompanying Supplemental Information Disclosure Statement.

By this Amendment, minor changes are being made to independent claims 59 and 69. Claims 55, 57 and 58, originally dependent upon claim 46, are being amended into multiple dependent form.

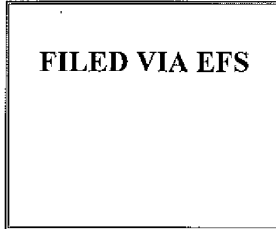
New dependent claims 75-77 are the same as claims 71, 73 and 57, respectively, except that they have been made to depend from multiple dependent claim 55. New dependent claims 78-79 are the same as claims 66-67 except that they have been made dependent on claim 65. New claims 80-84 depend from independent claim 69.

Since the amended and new dependent claims each depend from an independent claim that has been allowed, these amended and new claims are submitted to also be allowable.

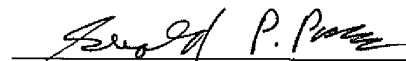
The Supplemental Information Disclosure Statement being filed herewith includes materials from Investigation No. 337-TA-619 of the United States International Trade Commission (ITC) of alleged infringements of grandparent patent no. 6,763,424, assigned to SanDisk Corporation. Claims 17, 18, 24 and 30 of patent no. 6,763,424 have been asserted by Complainant SanDisk Corporation in that ITC action to be infringed by "Certain Flash Memory Controllers, Drives, Memory Cards, and Media Players, and Products Containing Same" being imported by Respondents. A trial in this matter before the Administrative Law Judge of the ITC was completed in early November 2008.

Conclusion

It is believed that the present application remains allowable after this amendment, and an early formal allowance is earnestly solicited. But if the Examiner has any further matters that need to be resolved or any questions about this Supplemental Amendment, a telephone call to the undersigned attorney at 415-276-6534 (direct line) would be appreciated.



Respectfully submitted,


Gerald P. Parsons
Reg. No. 24,486

February 4, 2009
Date

Davis Wright Tremaine LLP
505 Montgomery Street, Suite 800
San Francisco, CA 94111-6533
(415) 276-6500 (main)
(415) 276-6534 (direct)
(415) 276-6599 (fax)
Email: geraldparsons@dwt.com

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Kevin M. Conley		
Title:	Partial Block Data Programming And Reading Operations In A Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	October 13, 2005
Examiner:	Dinh, Ngoc V.	Group Art Unit:	2189
Docket No.:	0084567-156US2	Conf. No.:	7727

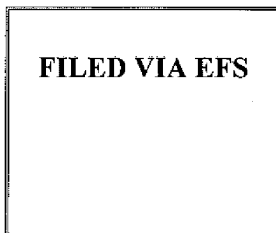
Mail Stop RCE
 Commissioner for Patents
 P.O. Box 1450
 Alexandria, VA 22313-1450

REQUEST FOR CONTINUED EXAMINATION (RCE)

Dear Sir:

This is a Request for Continued Examination (RCE) under 37 C.F.R. § 1.114 of the above-identified application. Please consider the Supplemental Amendment and Supplemental Information Disclosure Statement, which are being filed herewith.

The RCE fee of \$810.00 required under 37 C.F.R. § 1.17(e) has been authorized via EFS to Deposit Account 04-0258. The Commissioner is hereby authorized to charge any additional fees, which may be required, or credit any overpayment to Deposit Account 04-0258. Please contact the undersigned with any questions concerning this request or the above-identified patent application.



Respectfully submitted,

Gerald P. Parsons February 4, 2009
 Gerald P. Parsons Date
 Reg. No. 24,486

DAVIS WRIGHT TREMAINE LLP
 505 Montgomery Street, Suite 800
 San Francisco, California 94111-6533
 (415) 276-6500 (main)
 (415) 276-6534 (direct)
 (415) 276-6599 (fax)
 Email: geraldparsons@dwt.com

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Kevin M. Conley		
Title:	Block Data Programming and Reading Operations in a Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	October 13, 2005
Examiner:	Ngoc V. Dinh	Group Art Unit:	2189
Docket No.:	SNDK.156US2	Conf. No.:	7727

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

INFORMATION DISCLOSURE STATEMENT

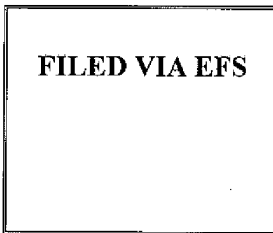
Dear Sir:

The attached Form PTO-1449 is an annotated copy of a page previously filed since the Examiner lined through the last four of the five items on that form because their dates were not apparent from them. The date of August 8, 2008 has now been added to each these four items on the attached Form PTO-1449. This is the same date as originally given in the first item of the attached Form PTO-1449 since all five items are part of one Report, the last four items being attached to the Report as exhibits. It was perceived during the last filing that there were too many pages to electronically file them as a single document, so the Report was broken up into the five pieces listed.

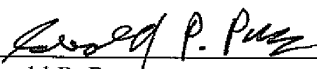
Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

If the Examiner believes there is any reason why the entire Report cannot now be made of record in the file of the present application, it is requested that the undersigned attorney be called at his direct telephone number (415-276-6534), so that any perceived deficiency can be addressed.



Respectfully submitted,



Gerald P. Parsons
Reg. No. 24,486

February 4, 2009

Date

Davis Wright Tremaine LLP
505 Montgomery Street, Suite 800
San Francisco, CA 94111-6533
(415) 276-6500 (main)
(415) 276-6534 (direct)
(415) 276-6599 (fax)
Email: geraldparsons@dwt.com

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.		Application No.				
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		0084567-156US2		11/250,238				
		Applicants		Conf. No.				
(Use several sheets if necessary)		Kevin M. Conley		7727				
(Form PTO-1449)		Filing Date		Art Group				
		October 13, 2005		2189				
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
	1	V. Niles Kynett, "Expert Report on the Invalidity of the '424 Patent," dated August 8, 2008, 99 pages (including Exhibit 4.)						
	2	Exhibits No. 16 through 28 from Kynett's Expert Report, dated August 8, 2008, 331 pages.						
	3	Exhibits No. 36 through 41 from Kynett's Expert Report, dated August 8, 2008, 69 pages.						
	4	Exhibits No. 46 through 57 from Kynett's Expert Report, dated August 8, 2008, 157 pages.						
	5	Exhibits No. 65 through 72 from Kynett's Expert Report, dated August 8, 2008, 63 pages.						
Examiner			Date Considered					
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

Electronic Patent Application Fee Transmittal

Application Number:	11250238			
Filing Date:	13-Oct-2005			
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY			
First Named Inventor/Applicant Name:	Kevin M. Conley			
Filer:	Gerald Paul Parsons/Svetlana Stellmach (GPP)			
Attorney Docket Number:	0084567-156US2			
Filed as Large Entity				
Utility under 35 USC 111(a) Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Claims in excess of 20	1202	35	52	1820
Multiple dependent claims	1203	1	390	390
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Request for continued examination	1801	1	810	810
Total in USD (\$)				3020

Electronic Acknowledgement Receipt

EFS ID:	4737207
Application Number:	11250238
International Application Number:	
Confirmation Number:	7727
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY
First Named Inventor/Applicant Name:	Kevin M. Conley
Customer Number:	66785
Filer:	Gerald Paul Parsons/Svetlana Stellmach (GPP)
Filer Authorized By:	Gerald Paul Parsons
Attorney Docket Number:	0084567-156US2
Receipt Date:	04-FEB-2009
Filing Date:	13-OCT-2005
Time Stamp:	20:10:41
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$3020
RAM confirmation Number	6972
Deposit Account	040258
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
-----------------	----------------------	-----------	-------------------------------------	------------------	------------------

1		SNDK156US2_Trans_Suppl_Amend_RCE_IDS_2-4-09.pdf	806578 28275cce0fb72c3009805d7d3f8e05a4a1de bdd7	yes	25
Multipart Description/PDF files in .zip description					
		Document Description	Start	End	
		Miscellaneous Incoming Letter	1	1	
		Amendment Submitted/Entered with Filing of CPA/RCE	2	2	
		Claims	3	19	
		Applicant Arguments/Remarks Made in an Amendment	20	21	
		Amendment Submitted/Entered with Filing of CPA/RCE	22	22	
		Information Disclosure Statement (IDS) Filed (SB/08)	23	25	
Warnings:					
Information:					
2	NPL Documents	Reference-1_Kynett_Expert-Report_dtd_8-08.pdf	2883329 91c9877a6fbed2b721cf5241c8e364f89347 4663	no	99
Warnings:					
Information:					
3	NPL Documents	Reference-2_Exhibits_16_through_28.pdf	10284106 e06e999837efeecc35175afe9aa7224b18ed6 1609	no	331
Warnings:					
Information:					
4	NPL Documents	Reference-3_Exhibits_36_through_41.pdf	2015924 ac9b2a91c96123784fa91ee8a887903e73d 11fb9	no	69
Warnings:					
Information:					
5	NPL Documents	Reference-4_Exhibits_46_through_57.pdf	4749089 a608c23f15f19192a3ab9c8e59fc5229f14aa 266	no	157
Warnings:					
Information:					
6	NPL Documents	Reference-5_Exhibits_65_through_72.pdf	2108957 407d2e999b358e7f80d91d9d7e3a6e3cad4 212cc	no	63
Warnings:					
Information:					

7	Fee Worksheet (PTO-06)	fee-info.pdf	33899	no	2
			a1b8e16167bbd97702a567aa0b2cadb5fed41c06		

Warnings:

Information:

Total Files Size (in bytes):	22881882
-------------------------------------	----------

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Davis Wright Tremaine LLP



ANCHORAGE BELLEVUE LOS ANGELES NEW YORK PORTLAND SAN FRANCISCO SEATTLE SHANGHAI WASHINGTON, D.C.

SUITE 800
505 MONTGOMERY STREET
SAN FRANCISCO, CA 94111-6533

TEL (415) 276-6500
FAX (415) 276-6599
www.dwt.com

February 4, 2009

Customer No. 66785

Commissioner For Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Re: Applicant: Kevin M. Conley
Title: Partial Block Data Programming and Reading Operations in a Non-Volatile Memory
Application No.: 11/250,238 Filing Date: October 13, 2005
Examiner: Dinh, Ngoc V. Group Art Unit: 2189
Docket No.: 0084567-156US2 Conf. No.: 7727

Dear Sir:

Transmitted herewith are the following documents in the above-identified application:

- (1) This Transmittal Letter;
- (2) Supplemental Amendment (20 pages);
- (3) Request for Continued Examination (1 page);
- (4) Information Disclosure Statement (3 pages);
- (5) 5 References enclosed.

The fee has been calculated as shown below:

CLAIMS AS AMENDED

	Claims Remaining <u>After</u> <u>Amendment</u>		Highest No. Previously <u>Paid For</u>	=	Present <u>Extra</u>	x	<u>Rate</u>	\$	Additional <u>Fee</u>
Total Claims	97	Minus	62	=	35	x	\$52.00	\$	1,820.00
Independent Claims	6	Minus	6	=	0	x	\$220.00	\$	0.00
<input checked="" type="checkbox"/>	Fee of \$390.00 for the first filing of one or more multiple dependent claims per application							\$	390.00
<input checked="" type="checkbox"/>	Fee for filing Request for Continued Examination							\$	810.00
<u>Total additional fee for this Amendment:</u>								\$	<u>3,020.00</u>

- Conditional Petition for Extension of Time: If an extension of time is required for timely filing of the enclosed document(s) after all papers filed with this transmittal have been considered, an extension of time is hereby requested.
- The fee of \$3,020.00 has been authorized via EFS to Deposit Account 040258. The Commissioner is hereby authorized to charge any additional fees, which may be required, or credit any overpayment to Deposit Account 040258.

FILED VIA EFS

Respectfully submitted,

Gerald P. Parsons
Reg. No. 24,486

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875				Application or Docket Number 11/250,238		Filing Date 10/13/2005		<input type="checkbox"/> To be Mailed			
APPLICATION AS FILED – PART I											
(Column 1)			(Column 2)			SMALL ENTITY <input type="checkbox"/>		OR		OTHER THAN SMALL ENTITY	
FOR		NUMBER FILED	NUMBER EXTRA		RATE (\$)	FEE (\$)	OR		RATE (\$)	FEE (\$)	
<input type="checkbox"/> BASIC FEE (37 CFR 1.16(a), (b), or (c))		N/A	N/A		N/A				N/A		
<input type="checkbox"/> SEARCH FEE (37 CFR 1.16(k), (l), or (m))		N/A	N/A		N/A		N/A				
<input type="checkbox"/> EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))		N/A	N/A		N/A		N/A				
TOTAL CLAIMS (37 CFR 1.16(i))		minus 20 =	*		X \$ =		OR		X \$ =		
INDEPENDENT CLAIMS (37 CFR 1.16(h))		minus 3 =	*		X \$ =		OR		X \$ =		
<input type="checkbox"/> APPLICATION SIZE FEE (37 CFR 1.16(s))		If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$250 (\$125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).									
<input type="checkbox"/> MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))											
* If the difference in column 1 is less than zero, enter "0" in column 2.											
APPLICATION AS AMENDED – PART II											
(Column 1)			(Column 2)			SMALL ENTITY		OR		OTHER THAN SMALL ENTITY	
AMENDMENT	02/04/2009	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)	ADDITIONAL FEE (\$)	OR		RATE (\$)	ADDITIONAL FEE (\$)
	Total (37 CFR 1.16(o))	* 97	Minus	** 62	= 35	X \$ =				X \$2=	1820
	Independent (37 CFR 1.16(h))	* 6	Minus	***6	= 0	X \$ =		X \$220=	0		
	<input type="checkbox"/> Application Size Fee (37 CFR 1.16(s))										
	<input type="checkbox"/> FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))										
						TOTAL ADD'L FEE		OR		TOTAL ADD'L FEE	1820
AMENDMENT		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)	ADDITIONAL FEE (\$)	OR		RATE (\$)	ADDITIONAL FEE (\$)
	Total (37 CFR 1.16(o))	*	Minus	**	=	X \$ =				X \$ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X \$ =		X \$ =			
	<input type="checkbox"/> Application Size Fee (37 CFR 1.16(s))										
	<input type="checkbox"/> FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))										
						TOTAL ADD'L FEE		OR		TOTAL ADD'L FEE	
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.											
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".											
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".											
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.											
Legal Instrument Examiner: /ELMIRA HALL/											

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

66785 7590 04/24/2009

DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION
505 MONTGOMERY STREET
SUITE 800
SAN FRANCISCO, CA 94111

EXAMINER: DINH, NGOC V
ART UNIT: 2188
PAPER NUMBER:
DATE MAILED: 04/24/2009

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.
11/250,238 10/13/2005 Kevin M. Conley 0084567-156US2 7727

TITLE OF INVENTION: PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY

Table with 7 columns: APPLN. TYPE, SMALL ENTITY, ISSUE FEE DUE, PUBLICATION FEE DUE, PREV. PAID ISSUE FEE, TOTAL FEE(S) DUE, DATE DUE
nonprovisional NO \$1510 \$300 \$0 \$1810 07/24/2009

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

- A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.
B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

- A. Pay TOTAL FEE(S) DUE shown above, or
B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PART B - FEE(S) TRANSMITTAL

**Complete and send this form, together with applicable fee(s), to: Mail Mail Stop ISSUE FEE
 Commissioner for Patents
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 or Fax (571)-273-2885**

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

66785 7590 04/24/2009

DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION
 505 MONTGOMERY STREET
 SUITE 800
 SAN FRANCISCO, CA 94111

Certificate of Mailing or Transmission
 I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

_____ (Depositor's name)
_____ (Signature)
_____ (Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/250,238	10/13/2005	Kevin M. Conley	0084567-156US2	7727

TITLE OF INVENTION: PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	\$1510	\$300	\$0	\$1810	07/24/2009

EXAMINER	ART UNIT	CLASS-SUBCLASS
DINH, NGOC V	2188	711-103000

<p>1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).</p> <p><input type="checkbox"/> Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.</p> <p><input type="checkbox"/> "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. Use of a Customer Number is required.</p>	<p>2. For printing on the patent front page, list</p> <p>(1) the names of up to 3 registered patent attorneys or agents OR, alternatively, 1 _____</p> <p>(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed. 2 _____</p> <p>3 _____</p>
---	---

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE _____ (B) RESIDENCE: (CITY and STATE OR COUNTRY) _____

Please check the appropriate assignee category or categories (will not be printed on the patent): Individual Corporation or other private group entity Government

<p>4a. The following fee(s) are submitted:</p> <p><input type="checkbox"/> Issue Fee</p> <p><input type="checkbox"/> Publication Fee (No small entity discount permitted)</p> <p><input type="checkbox"/> Advance Order - # of Copies _____</p>	<p>4b. Payment of Fee(s): (Please first reapply any previously paid issue fee shown above)</p> <p><input type="checkbox"/> A check is enclosed.</p> <p><input type="checkbox"/> Payment by credit card. Form PTO-2038 is attached.</p> <p><input type="checkbox"/> The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).</p>
---	--

5. Change in Entity Status (from status indicated above)

a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27. b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.
Row 1: 11/250,238, 10/13/2005, Kevin M. Conley, 0084567-156US2, 7727
Row 2: 66785, 7590, 04/24/2009, [EXAMINER], [DINH, NGOC V]
Row 3: [ART UNIT], [PAPER NUMBER]
Row 4: 2188, DATE MAILED: 04/24/2009

DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION
505 MONTGOMERY STREET
SUITE 800
SAN FRANCISCO, CA 94111

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 0 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 0 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

Notice of Allowability	Application No.	Applicant(s)	
	11/250,238	CONLEY, KEVIN M.	
	Examiner	Art Unit	
	NGOC V. DINH	2188	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. This communication is responsive to RCE filed 02/04/09.
2. The allowed claim(s) is/are 4, 7-18, 26-84 (renumbered as 1-72).
3. Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some* c) None of the:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
5. CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
 - (a) including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
 - 1) hereto or 2) to Paper No./Mail Date _____.
 - (b) including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

- | | |
|--|--|
| <ol style="list-style-type: none"> 1. <input type="checkbox"/> Notice of References Cited (PTO-892) 2. <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) 3. <input checked="" type="checkbox"/> Information Disclosure Statements (PTO/SB/08),
Paper No./Mail Date <u>02/04/09</u> 4. <input type="checkbox"/> Examiner's Comment Regarding Requirement for Deposit of Biological Material | <ol style="list-style-type: none"> 5. <input type="checkbox"/> Notice of Informal Patent Application 6. <input type="checkbox"/> Interview Summary (PTO-413),
Paper No./Mail Date _____. 7. <input type="checkbox"/> Examiner's Amendment/Comment 8. <input checked="" type="checkbox"/> Examiner's Statement of Reasons for Allowance 9. <input type="checkbox"/> Other _____. |
|--|--|

--	--

DETAILED ACTION

This office action is a response to the RCE filed on 02/04/09. Claims 75-84 are new added.

INFORMATION DISCLOSURE STATEMENT

The Applicant's submission of the IDS filed 02/04/09 have been considered. As required by M.P.E.P. 609 C(2), copies of the PTOL-1449 is attached to the instant office action.

REASON FOR ALLOWANCE

The primary reasons for allowance of claims 4, 13, 46, 59, 68, 69 in the instant application is the combination with the inclusion of at least the limitations set forth in lines 10-13 of claims 4, 13; lines 15-19 of claim 46; lines 13-17 of claim 59; lines 11-17 of claim 68; lines 8-11 of claim 69.

Because claims 7-12, 26-35; 14-18, 36-45; 47-58, 71, 73, 75-77; 60-67, 72, 74, 78-79 and 70, 80-84 depend directly or indirectly on claims 4, 13, 46, 59, 69. These claims are considered allowable for at least the same reasons noted above.

Conclusion

Any response to this action should be mailed to:

Under Secretary of Commerce for intellectual Property and Director of the
United States Patent and Trademark Office

PO Box 1450

Alexandria, VA 22313-1450

or faxed to:

(571) 273-8300, (for Official communications intended for entry).

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PMR) system. Status information for published Applications may be obtained from either Private PMR or Public PMR. Status information for unpublished applications is available through Private PAIR only. For

more information about the PAIR system, see <http://pak-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ngoc Dinh whose telephone number is (571) 272-4191. The examiner can normally be reached on Monday-Friday 8:30 AM-5:00 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung Sough, can be reached on (571) 272-6799.

/N. V. D./

Examiner, Art Unit 2188

/Hyung S. Sough/

Supervisory Patent Examiner, Art Unit 2188

04/01/09



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
 United States Patent and Trademark Office
 Address: COMMISSIONER FOR PATENTS
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 7727

SERIAL NUMBER 11/250,238	FILING or 371(c) DATE 10/13/2005 RULE	CLASS 711	GROUP ART UNIT 2188	ATTORNEY DOCKET NO. 0084567-156US2		
APPLICANTS Kevin M. Conley, San Jose, CA; ** CONTINUING DATA ***** This application is a CON of 10/841,388 05/07/2004 PAT 6,968,421 which is a CON of 09/766,436 01/19/2001 PAT 6,763,424 ** FOREIGN APPLICATIONS ***** ** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ** 11/03/2005						
Foreign Priority claimed <input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	35 USC 119(a-d) conditions met <input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	<input type="checkbox"/> Met after Allowance Initials	STATE OR COUNTRY CA	SHEETS DRAWINGS 9	TOTAL CLAIMS 3	INDEPENDENT CLAIMS 2
ADDRESS DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION 505 MONTGOMERY STREET SUITE 800 SAN FRANCISCO, CA 94111 UNITED STATES						
TITLE PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY						
FILING FEE RECEIVED 6054	FEES: Authority has been given in Paper No. _____ to charge/credit DEPOSIT ACCOUNT No. _____ for following:		<input type="checkbox"/> All Fees <input type="checkbox"/> 1.16 Fees (Filing) <input type="checkbox"/> 1.17 Fees (Processing Ext. of time) <input type="checkbox"/> 1.18 Fees (Issue) <input type="checkbox"/> Other _____ <input type="checkbox"/> Credit			

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	7495	711/103 711/200 711/203	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 12:57
L2	1808	("same" identical common) near5 (logical adj2 address)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 12:57
L3	418	L1 and L2	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 12:57
L4	1330	(updat\$3 same offset) same (page pag\$3 mapp\$3)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 12:59
L5	725	(map mapping pag \$3) same (different near4 offset)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 13:00
L6	21	1 and 5	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 13:00
L7	8679	((original old) adj2 data) with ((updated new replacement) adj2 data)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 13:00
L8	1808	("same" identical common) near5 (logical adj2 address)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 13:00
L9	108	L7 same L8	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 13:00
L10	78	1 and 9	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 13:01
L11	367894	(new old superseded updated) with (version data)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 13:01

L12	1020	((("same" common identical) near2 (logical adj address))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 13:01
L13	205	L11 same L12	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 13:01
L14	30	((time adj stamp) time\$stamp) and L13	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 13:01
L15	5122	(read\$3 write writing updat\$3) same (("same" identical common) near4 (logical lba lbn lsa lsn))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/03/02 13:01
L16	4308	read\$3 with ((most more) near3 recent)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/03/02 13:01
L17	23	L15 same L16	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/03/02 13:01
L18	2220	(eprom eprom flash non\$volatile) with separate with (unit portion section module)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 13:02
L19	2220	(eprom eprom flash non\$volatile) with separate with (unit portion section module)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 13:02
L20	56	((program\$4 write writing written) with (offset)) and L19	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 13:02
L21	56	L18 and L20	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 13:02
L22	13	1 and 21	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/03/02 13:03

3/2/09 1:04:48 PM

C:\Documents and Settings\ndinh\My Documents\EAST\workspaces\11250238.wsp

OK to enter
04/01/2009
ND

SDK0156.002US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Kevin M. Conley		
Title:	Partial Block Data Programming And Reading Operations In A Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	October 13, 2005
Examiner:	Dinh, Ngoc V.	Group Art Unit:	2189
Docket No.:	0084567-156US2	Conf. No.:	7727

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUPPLEMENTAL AMENDMENT

Sir:


This Supplemental Amendment is being filed concurrently with a Request for Continued Examination (RCE) of the above-referenced patent application, which has been allowed. A Supplemental Information Disclosure Statement is also being filed herewith.

Claim amendments are reflected in the listing of claims, which begins on page 2 of this paper.

Remarks begin on page 19 of this paper.

Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238


Search Notes 	Application/Control No. 11250238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.
	Examiner NGOC V DINH	Art Unit 2188

SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES		
Search Notes	Date	Examiner
Limited classified search of Class/subclass. East text search w/o classified/search. See printout.	4/3/09	ND
Consulted Kevin Verbrugge (AU 2189).	10/23/08	ND
Consulted Kevin Ellis.	10/29/08	ND

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner
711/103, 711/113, 711/115	Intereference search EAST.	4/3/09	ND


--	--

Index of Claims 	Application/Control No. 11250238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.
	Examiner NGOC V DINH	Art Unit 2188

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claims renumbered in the same order as presented by applicant
 CPA
 T.D.
 R.1.47

CLAIM		DATE									
Final	Original	03/31/2009									
	1	-									
	2	-									
	3	-									
1	4	=									
	5	-									
	6	-									
2	7	=									
3	8	=									
4	9	=									
5	10	=									
6	11	=									
7	12	=									
18	13	=									
19	14	=									
20	15	=									
21	16	=									
22	17	=									
23	18	=									
	19	-									
	20	-									
	21	-									
	22	-									
	23	-									
	24	-									
	25	-									
8	26	=									
9	27	=									
10	28	=									
11	29	=									
12	30	=									
13	31	=									
14	32	=									
15	33	=									
16	34	=									
17	35	=									
24	36	=									

Index of Claims 	Application/Control No. 11250238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.
	Examiner NGOC V DINH	Art Unit 2188

✓	Rejected
=	Allowed


-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

Claims renumbered in the same order as presented by applicant
 CPA
 T.D.
 R.1.47

CLAIM		DATE									
Final	Original	03/31/2009									
25	37	=									
26	38	=									
27	39	=									
28	40	=									
29	41	=									
30	42	=									
31	43	=									
32	44	=									
33	45	=									
34	46	=									
35	47	=									
36	48	=									
37	49	=									
38	50	=									
39	51	=									
40	52	=									
41	53	=									
42	54	=									
43	55	=									
48	56	=									
47	57	=									
49	58	=									
52	59	=									
53	60	=									
54	61	=									
55	62	=									
56	63	=									
57	64	=									
58	65	=									
61	66	=									
62	67	=									
65	68	=									
66	69	=									
67	70	=									
50	71	=									
63	72	=									

<i>Index of Claims</i> 	Application/Control No. 11250238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.
	Examiner NGOC V DINH	Art Unit 2188

✓	Rejected
=	Allowed

-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

Claims renumbered in the same order as presented by applicant
 CPA
 T.D.
 R.1.47

CLAIM		DATE								
Final	Original	03/31/2009								
51	73	=								
64	74	=								
44	75	=								
45	76	=								
46	77	=								
59	78	=								
60	79	=								
68	80	=								
69	81	=								
70	82	=								
71	83	=								
72	84	=								

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.		Application No.				
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		0084567-156US2		11/250,238				
		Applicants		Conf. No.				
(Use several sheets if necessary)		Kevin M. Conley		7727				
(Form PTO-1449)		Filing Date		Art Group				
		October 13, 2005		2189				
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
/N.D./	1	V. Niles Kynett, "Expert Report on the Invalidity of the '424 Patent," dated August 8, 2008, 99 pages (including Exhibit 4.)						
↓	2	Exhibits No. 16 through 28 from Kynett's Expert Report, dated August 8, 2008, 331 pages.						
↓	3	Exhibits No. 36 through 41 from Kynett's Expert Report, dated August 8, 2008, 69 pages.						
↓	4	Exhibits No. 46 through 57 from Kynett's Expert Report, dated August 8, 2008, 157 pages.						
↓	5	Exhibits No. 65 through 72 from Kynett's Expert Report, dated August 8, 2008, 63 pages.						
/N.D./								
Examiner /Ngoc Dinh/		Date Considered		03/31/2009				
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

FILED VIA EFS
Sheet 1 of 1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Kevin M. Conley		
Title:	Block Data Programming and Reading Operations in a Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	October 13, 2005
Examiner:	Ngoc V. Dinh	Group Art Unit:	2188
Docket No.:	0084567-156US2	Conf. No.:	7727

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Dear Sir:

Pursuant to 37 C.F.R. §§ 1.56, 1.97 and 1.98, Applicant(s) call(s) the documents listed on the enclosed Form PTO-1449 to the Examiner's attention in this patent application.

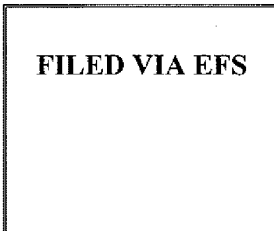
A copy of the listed foreign patent documents and/or Other Art is enclosed.

Citation of these documents shall not be construed as (1) an admission that the documents are prior art with respect to the invention or inventions claimed in this application, (2) a representation that a search has been made (other than as indicated by any cited document), or (3) an admission that the cited information is, or is considered to be, material to patentability as defined in § 1.56(b).

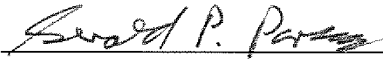
Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

This information disclosure statement is submitted under 37 C.F.R. § 1.97(d). In accordance with § 1.97(e)(1), each item contained in this information disclosure statement was first cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this information disclosure statement. The fee of \$180.00 has been authorized via EFS to Deposit Account 04-0258. The Commissioner is hereby authorized to charge any additional fees, which may be required, or credit any overpayment to Deposit Account 04-0258.



Respectfully submitted,



Gerald P. Parsons
Reg. No. 24,486

April 29, 2009
Date

Davis Wright Tremaine LLP
505 Montgomery Street, Suite 800
San Francisco, CA 94111-6533
(415) 276-6500 (main)
(415) 276-6534 (direct)
(415) 276-6599 (fax)
Email: geraldparsons@dwt.com

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.		Application No.				
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		0084567-156US2		11/250,238				
		Applicants		Conf. No.				
(Use several sheets if necessary)		Kevin M. Conley		7727				
(Form PTO-1449)		Filing Date		Art Group				
		October 13, 2005		2188				
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
	1	Korean Patent Office, "Notice of Preliminary Rejections," corresponding Korean Patent Application No. 2008-7028861, mailed on March 10, 2009, 8 pages (including translation.)						
Examiner			Date Considered					
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

FILED VIA EFS
Sheet 1 of 1

Electronic Patent Application Fee Transmittal

Application Number:	11250238			
Filing Date:	13-Oct-2005			
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY			
First Named Inventor/Applicant Name:	Kevin M. Conley			
Filer:	Gerald Paul Parsons/Svetlana Stellmach (GPP)			
Attorney Docket Number:	0084567-156US2			
Filed as Large Entity				
Utility under 35 USC 111(a) Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Miscellaneous:				
Submission- Information Disclosure Stmt	1806	1	180	180
Total in USD (\$)				180

Electronic Acknowledgement Receipt

EFS ID:	5242112
Application Number:	11250238
International Application Number:	
Confirmation Number:	7727
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY
First Named Inventor/Applicant Name:	Kevin M. Conley
Customer Number:	66785
Filer:	Gerald Paul Parsons/Svetlana Stellmach (GPP)
Filer Authorized By:	Gerald Paul Parsons
Attorney Docket Number:	0084567-156US2
Receipt Date:	29-APR-2009
Filing Date:	13-OCT-2005
Time Stamp:	14:42:29
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$180
RAM confirmation Number	982
Deposit Account	040258
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
-----------------	----------------------	-----------	-------------------------------------	------------------	------------------

1	Information Disclosure Statement (IDS) Filed (SB/08)	SNDK156US2_Suppl_IDS_4-29-09.pdf	76899 1df281a977476ed394a4b6cacd2be04d6157c853	no	3
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
2	NPL Documents	Reference_SNDK156KR1_OA_3-10-09.pdf	582307 ce05439d58cb84ffe6160cf2e0a7d95021e41e	no	8
Warnings:					
Information:					
3	Fee Worksheet (PTO-875)	fee-info.pdf	30586 2b32a62f94ec01a359b283e1b1b3d82b378954fc	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			689792		
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.

11/250,238 10/13/2005 Kevin M. Conley 0084567-156US2 7727

66785 7590 05/06/2009
DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION
505 MONTGOMERY STREET
SUITE 800
SAN FRANCISCO, CA 94111

Table with 1 column: EXAMINER

DINH, NGOC V

Table with 2 columns: ART UNIT, PAPER NUMBER

2188

Table with 2 columns: MAIL DATE, DELIVERY MODE

05/06/2009 PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES DEPARTMENT OF COMMERCE
U.S. Patent and Trademark Office
 Address: COMMISSIONER FOR PATENTS
 P.O. Box 1450
 Alexandria, Virginia 22313-1450

APPLICATION NO./ CONTROL NO.	FILING DATE	FIRST NAMED INVENTOR / PATENT IN REEXAMINATION	ATTORNEY DOCKET NO.
11250238	10/13/05	CONLEY, KEVIN M.	0084567-156US2

DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION
 505 MONTGOMERY STREET
 SUITE 800
 SAN FRANCISCO, CA 94111

EXAMINER

NGOC V. DINH

ART UNIT	PAPER
2188	20090505

2188 20090505

DATE MAILED:

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner for Patents

The IDS filed 04/29/09 has been considered by the examiner. As required by M.P.E.P. 609 C(2), a copy of the PTOL-1449 is attached to the instant office action.

/Hyung S. Sough/
 Supervisory Patent Examiner, Art Unit 2188
 05/05/09

PTO-90C (Rev.04-03)

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.		Application No.				
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		0084567-156US2		11/250,238				
		Applicants		Conf. No.				
(Use several sheets if necessary)		Kevin M. Conley		7727				
(Form PTO-1449)		Filing Date		Art Group				
		October 13, 2005		2188				
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
/N.D./	1	Korean Patent Office, "Notice of Preliminary Rejections," corresponding Korean Patent Application No. 2008-7028861, mailed on March 10, 2009, 8 pages (including translation.)						
Examiner	/Ngoc Dinh/		Date Considered	05/05/2009				
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

FILED VIA EFS
Sheet 1 of 1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Kevin M. Conley		
Title:	Partial Block Data Programming And Reading Operations In A Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	October 13, 2005
Examiner:	Dinh, Ngoc V.	Group Art Unit:	2188
Docket No.:	0084567-156US2	Conf. No.:	7727

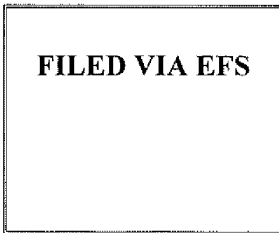
Mail Stop RCE
 Commissioner for Patents
 P.O. Box 1450
 Alexandria, VA 22313-1450

REQUEST FOR CONTINUED EXAMINATION (RCE)

Dear Sir:

This is a Request for Continued Examination (RCE) under 37 C.F.R. § 1.114 of the above-identified application. Please consider the Supplemental Amendment and Information Disclosure Statement, which are being filed herewith.

The RCE fee of \$810.00 required under 37 C.F.R. § 1.17(e) has been authorized via EFS to Deposit Account 04-0258. The Commissioner is hereby authorized to charge any additional fees, which may be required, or credit any overpayment to Deposit Account 04-0258. Please contact the undersigned with any questions concerning this request or the above-identified patent application.



Respectfully submitted,

Gerald P. Parsons

 Gerald P. Parsons
 Reg. No. 24,486

June 8, 2009

 Date

DAVIS WRIGHT TREMAINE LLP
 505 Montgomery Street, Suite 800
 San Francisco, California 94111-6533
 (415) 276-6500 (main)
 (415) 276-6534 (direct)
 (415) 276-6599 (fax)
 Email: geraldparsons@dwt.com

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Kevin M. Conley		
Title:	Block Data Programming and Reading Operations in a Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	October 13, 2005
Examiner:	Ngoc V. Dinh	Group Art Unit:	2188
Docket No.:	0084567-156US2	Conf. No.:	7727

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

INFORMATION DISCLOSURE STATEMENT

Dear Sir:

Pursuant to 37 C.F.R. §§ 1.56, 1.97 and 1.98, Applicant calls the documents listed on the enclosed Form PTO-1449 to the Examiner's attention in this patent application.

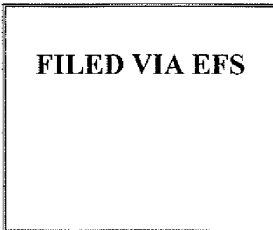
A copy of the listed Other Art is enclosed.

Citation of these documents shall not be construed as (1) an admission that the documents are prior art with respect to the invention or inventions claimed in this application, (2) a representation that a search has been made (other than as indicated by any cited document), or (3) an admission that the cited information is, or is considered to be, material to patentability as defined in § 1.56(b).

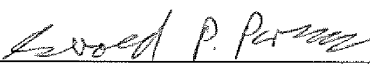
Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

This information disclosure statement is submitted under 37 C.F.R. § 1.97(b) and consequently no fee should be required. The Commissioner is authorized, however, to charge any fee that may be required, or to credit any overpayment, against Deposit Account No. 04-0258.



Respectfully submitted,


Gerald P. Parsons
Reg. No. 24,486

June 8, 2009
Date

Davis Wright Tremaine LLP
505 Montgomery Street, Suite 800
San Francisco, CA 94111-6533
(415) 276-6500 (main)
(415) 276-6534 (direct)
(415) 276-6599 (fax)
Email: geraldparsons@dwt.com

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.		Application No.				
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		0084567-156US2		11/250,238				
		Applicants		Conf. No.				
(Use several sheets if necessary)		Kevin M. Conley		7727				
(Form PTO-1449)		Filing Date		Art Group				
		October 13, 2005		2188				
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
	1	United States International Trade Commission, " <i>In the Matter of Certain Flash Memory Controllers, Drives, Memory Cards and Media Players, and Products Containing Same,</i> " Inv. No. 337-TA-619, Initial Determination on Violation of Section 337 and Recommended Determination on Remedy and Bond, Administrative Law Judge Charles E. Bullock, Public Version, May 5, 2009, 398 pages.						
Examiner			Date Considered					
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

Electronic Patent Application Fee Transmittal

Application Number:	11250238			
Filing Date:	13-Oct-2005			
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY			
First Named Inventor/Applicant Name:	Kevin M. Conley			
Filer:	Gerald Paul Parsons/Svetlana Stellmach (GPP)			
Attorney Docket Number:	0084567-156US2			
Filed as Large Entity				
Utility under 35 USC 111(a) Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Claims in excess of 20	1202	86	52	4472
Independent claims in excess of 3	1201	2	220	440
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Request for continued examination	1801	1	810	810
Total in USD (\$)				5722

Electronic Acknowledgement Receipt

EFS ID:	5476703
Application Number:	11250238
International Application Number:	
Confirmation Number:	7727
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY
First Named Inventor/Applicant Name:	Kevin M. Conley
Customer Number:	66785
Filer:	Gerald Paul Parsons/Svetlana Stellmach (GPP)
Filer Authorized By:	Gerald Paul Parsons
Attorney Docket Number:	0084567-156US2
Receipt Date:	08-JUN-2009
Filing Date:	13-OCT-2005
Time Stamp:	19:05:19
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$5722
RAM confirmation Number	4984
Deposit Account	040258
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
-----------------	----------------------	-----------	----------------------------------	------------------	------------------

1		SNDK156US2_Trans_Suppl-Amend_RCE_IDS_6-8-09.pdf	1066127	yes	32
			3f395e547004eb1eda3716fbd30fcb2cda8da088		
Multipart Description/PDF files in .zip description					
		Document Description	Start	End	
		Miscellaneous Incoming Letter	1	1	
		Amendment Submitted/Entered with Filing of CPA/RCE	2	2	
		Claims	3	26	
		Applicant Arguments/Remarks Made in an Amendment	27	28	
		Request for Continued Examination (RCE)	29	29	
		Transmittal Letter	30	31	
		Information Disclosure Statement (IDS) Filed (SB/08)	32	32	
Warnings:					
Information:					
2	NPL Documents	Reference_337-TA-619_PUBLIC_Final_ID.pdf	9137332	no	398
			9eb0f3dbec8722cd2c591d49ea64b347c5738f3		
Warnings:					
Information:					
3	Fee Worksheet (PTO-875)	fee-info.pdf	33739	no	2
			59b99497f2784702c4f5492500fe7b735a9878c6		
Warnings:					
Information:					
Total Files Size (in bytes):			10237198		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

June 8, 2009

Customer No. 66785

Commissioner For Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Re: Applicant: Kevin M. Conley
Title: Partial Block Data Programming and Reading Operations in a Non-Volatile Memory
Application No.: 11/250,238 Filing Date: October 13, 2005
Examiner: Dinh, Ngoc V. Group Art Unit: 2188
Docket No.: 0084567-156US2 Conf. No.: 7727

Dear Sir:

Transmitted herewith are the following documents in the above-identified application:

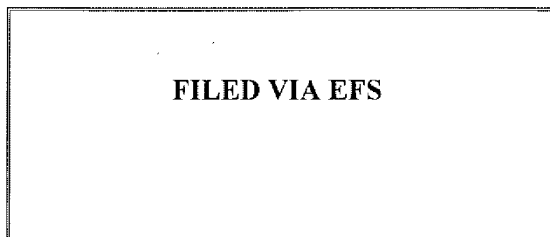
- (1) This Transmittal Letter;
- (2) Supplemental Amendment (27 pages);
- (3) Request for Continued Examination (1 page);
- (4) Information Disclosure Statement (3 pages);
- (5) 1 Reference enclosed.

The fee has been calculated as shown below:

CLAIMS AS AMENDED

	Claims Remaining <i>After</i> <u>Amendment</u>		Highest No. Previously <u>Paid For</u>	=	Present <u>Extra</u>	Rate		Additional <u>Fee</u>	
Total Claims	189	Minus	103	=	86	x \$52.00	\$	4,472.00	
Independent Claims	8	Minus	6	=	2	x \$220.00	\$	440.00	
<input type="checkbox"/>	Fee of \$390.00 for the first filing of one or more multiple dependent claims per application							\$	0.00
<input checked="" type="checkbox"/>	Fee for filing Request for Continued Examination							\$	810.00
<u>Total additional fee for this Amendment:</u>								\$ 5,722.00	

- Conditional Petition for Extension of Time: If an extension of time is required for timely filing of the enclosed document(s) after all papers filed with this transmittal have been considered, an extension of time is hereby requested.
- The fee of \$5,722.00 has been authorized via EFS to Deposit Account 040258. The Commissioner is hereby authorized to charge any additional fees, which may be required, or credit any overpayment to Deposit Account 040258.



Respectfully submitted,

Gerald P. Parsons
Reg. No. 24,486

Anchorage
Bellevue
Los Angeles

New York
Portland
San Francisco

Seattle
Shanghai
Washington, D.C.

www.dwt.com

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Kevin M. Conley
Title: Partial Block Data Programming And Reading Operations In A Non-Volatile Memory
Application No.: 11/250,238 Filing Date: October 13, 2005
Examiner: Dinh, Ngoc V. Group Art Unit: 2188
Docket No.: 0084567-156US2 Conf. No.: 7727

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SECOND SUPPLEMENTAL AMENDMENT

Sir:

This Second Supplemental Amendment is being filed concurrently with a Request for Continued Examination (RCE) of the above-referenced patent application, which has been allowed. A Supplemental Information Disclosure Statement is also being filed herewith.

Claim amendments are reflected in the listing of claims, which begins on page 2 of this paper.

Remarks begin on page 26 of this paper.

Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

CLAIM AMENDMENTS

Please amend the claims by adding new claims 85-118, without prejudice, as indicated on the following listing of all the claims in the present application after this Amendment:

1 – 3. (Cancelled)

4. (Previously Presented) A method of operating a memory system of reprogrammable non-volatile charge storage elements organized in blocks of a minimum number of storage elements that are erasable together as a unit and in pages of storage elements within the blocks that have specified offset positions within their respective blocks and which are individually programmable as a unit, comprising:

as part of writing data into pages, recording an indication of a time from a clock source that data are written into individual pages,

updating data previously written into one or more pages of one of original data blocks by writing the updated data into one or more update pages of an update data block and logically linking data of the corresponding previously written pages and update data pages, wherein the updated data are caused to be writable into pages of the update data block having different offset positions than the pages of the original data block into which the logically linked data being updated was previously written, and

when reading data of two or more logically linked data pages, read the indications of the times that the data have been stored in the two or more pages and use the data in the two or more pages having more recent time indications without using data in the two or more pages having older time indications.

5 – 6. (Cancelled)

7. (Previously Presented) The method of claim 4, wherein recording the indication of the time that data are written into individual pages includes recording the indication within the pages wherein the data are written.

8. (Previously Presented) The method of claim 4, wherein updating data previously written into one or more pages of one of the original data blocks includes updating data in a number of pages less than all of the pages of the one of the original data blocks while not updating data in a remaining one or more pages of the same original data block.

9. (Previously Presented) The method of claim 8, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

10. (Previously Presented) The method of claim 4, wherein as part of updating data, logically linking the corresponding previously written pages and update data pages comprises writing logical addresses of the updated data in the individual update pages in which the updated data are written.

11. (Previously Presented) The method of claim 4, wherein as part of writing data into pages, data are written in individual storage elements of the pages with more than two storage states, thereby storing more than one bit of data in the individual storage elements.

12. (Previously Presented) The method of claim 4, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

13. (Previously Presented) A method of operating a memory system of reprogrammable non-volatile charge storage elements organized in blocks of a minimum number of storage elements that are erasable together as a unit and in pages of storage elements within the blocks that have specified offset positions within their respective blocks and which are individually programmable as a unit, comprising:

as part of writing data into pages, data are written into the pages of individual blocks in sequence,

updating data previously written into one or more initial pages of one of original data blocks by writing the updated data into one or more update pages of an update data block and logically linking data of previously written pages with corresponding update data pages, wherein the updated data are caused to be writable into pages of the update data block having different offset positions as the pages of the original data block into which the logically linked data being updated was previously written, and

reading data from the pages of the original data block and update data block in an order that is a reverse of the sequence in which they were written and ignore data in any page that is logically linked with a page from which data have already been read.

14. (Previously Presented) The method of claim 13, wherein updating data previously written into one or more pages of one of the original data blocks includes updating data in a number of pages less than all of the pages of the one of the original data blocks while not updating data in a remaining one or more pages of the same original data block.

15. (Previously Presented) The method of claim 14, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

16. (Previously Presented) The method of claim 13, wherein as part of updating data into pages, logically linking the corresponding previously written pages and update data pages comprises writing logical addresses of updated data in the individual update pages in which the updated data are written.

17. (Previously Presented) The method of claim 13, wherein as part of writing data into pages, data are written in individual storage elements of the pages with more than two storage states, thereby storing more than one bit of data in the individual storage elements.

18. (Previously Presented) The method of claim 13, wherein the memory system in which the method is carried out utilizes electrically conductive floating gates as the charge storage elements.

19 – 25. (Cancelled)

26. (Previously Presented) The method of claim 4, wherein logically linking the corresponding previously written pages and update data pages comprises maintaining common logical addresses thereof.

27. (Previously Presented) The method of claim 26, wherein maintaining common logical addresses comprises storing the common logical addresses in the pages of the original data block and in the pages of the update data block along with the respective previously written and updated data therein.

28. (Previously Presented) The method of claim 26, wherein reading the data further comprises organizing the read data by the logical page addresses associated with the read data.

29. (Previously Presented) The method of claim 4, wherein data are written into individual pages of the original and update data blocks in a specified sequence.

30. (Previously Presented) The method of claim 4, wherein the original data previously written into one or more pages of one of original data blocks are written into pages of first and second blocks positioned in different ones of a plurality of units of the memory system, wherein the units are physically separate groupings of blocks of charge storage elements in which programming operations may be performed independently, and linking the first and second blocks to cause their charge storage elements to be erasable together and their pages to be programmable together in parallel.

31. (Previously Presented) The method of claim 4, additionally comprising:

updating data previously written into one or more pages of a second of the original data blocks by writing the updated data into one or more update pages of the update data block, and logically linking data of the corresponding previously written pages of the second of the original data blocks and the update data pages.

32. (Previously Presented) The method of claim 31, wherein updating data previously written into one or more pages of the one and of the second of the original data blocks includes updating data in a number of pages less than all of the pages of the individual original data block while not updating data in a remaining one or more pages of the same individual original data block.

33. (Previously Presented) The method of claim 4, additionally comprising:
subsequently updating for a second time at least some of the data previously written into one or more pages of one of the original data blocks that were previously updated and written into the update data block,
writing the data updated for a second time into the update data block as second updated data pages, and
logically linking data of the corresponding previously written pages and the second updated data pages.

34. (Previously Presented) The method of claim 33, wherein updating data previously written into one or more pages of one of the original data blocks and subsequently updating the data for the second time each includes updating data in a number of pages less than all of the pages of the individual original data block while not updating data in a remaining one or more pages of said individual original data block.

35. (Previously Presented) The method of claim 4, wherein the memory system in which the method is practiced is contained within an enclosed card having an electrical connector along one edge thereof for operably connecting with a host system.

36. (Previously Presented) The method of claim 13, wherein logically linking the corresponding previously written pages and update data pages comprises maintaining common logical addresses thereof.

37. (Previously Presented) The method of claim 36, wherein maintaining common logical addresses comprises storing the common logical addresses in the pages of the original data block and in the pages of the update data block along with the respective previously written and updated data therein.

38. (Previously Presented) The method of claim 36, wherein reading the data further comprise organizing the read data by the logical page addresses associated with the read data.

39. (Previously Presented) The method of claim 13, wherein data are written into individual pages of the blocks in a specified sequence.

40. (Previously Presented) The method of claim 13, wherein the data written into one or more initial pages of one of the original data are written into pages of first and second blocks positioned in different ones of a plurality of units of the memory system, wherein the units are physically separate groupings of blocks of charge storage elements in which programming operations may be performed independently, and linking the first and second blocks to cause their charge storage elements to be erasable together and their pages to be programmable together in parallel.

41. (Previously Presented) The method of claim 13, additionally comprising:
updating data previously written into one or more pages of a second of the original data blocks by writing the updated data into one or more update pages of the update data block, and
logically linking data of the corresponding previously written pages of the second of the original data blocks and the update data pages.

42. (Previously Presented) The method of claim 41, wherein updating data previously written into one or more pages of the one and of the second of the original data blocks includes updating data in a number of pages less than all of the pages of the individual original data block while not updating data in a remaining one or more pages of the same individual original data block.

43. (Previously Presented) The method of claim 13, additionally comprising:
subsequently updating for a second time at least some of the data previously written into one or more pages of one of the original data blocks that were previously updated and written into the update data block,

writing the data updated for a second time into the update data block as second updated data pages, and

logically linking data of the corresponding previously written pages and the second updated data pages.

44. (Previously Presented) The method of claim 43, wherein updating data previously written into one or more pages of one of the original data blocks and subsequently updating the data for the second time each includes updating data in a number of pages less than all of the pages of the individual original data block while not updating data in a remaining one or more pages of the same individual original data block.

45. (Previously Presented) The method of claim 13, wherein the memory system in which the method is practiced is contained within an enclosed card having an electrical connector along one edge thereof for operably connecting with a host system.

46. (Previously Presented) In a re-programmable non-volatile semiconductor memory system having a plurality of blocks of a minimum number of memory charge storage elements that are erasable together as a unit, the plurality of blocks individually being divided into a plurality of a given number of pages of memory storage elements that are individually

programmable as a unit and which have specified offset positions within their respective blocks, a method of operating the memory system, comprising:

programming original data into individual ones of a first plurality of pages in at least a first block, the pages of original data having logical addresses associated therewith,

thereafter programming, into individual ones of a second plurality of pages in a second block, an updated version of less than the given number of pages of the original data programmed into the first plurality of pages, the pages of the updated version of the original data having logical addresses associated therewith, wherein the logical addresses associated with the pages of the updated version of the original data are the same as the logical addresses associated with the pages of original data,

wherein programming the second plurality of pages additionally comprises programming the updated version of the original data in those of the second plurality of pages that have different offset positions within the second block than the offset positions of the first plurality of pages within said at least the first block that contain pages of original data with the same associated logical addresses,

thereafter reading data from the first and second plurality of pages, and organizing pages of the read data by their associated logical addresses.

47. (Previously Presented) The method of claim 46, wherein reading data and organizing pages of the read data by their associated logical addresses comprises, for the pages of read data having the same logical addresses associated therewith, utilizing the pages of the updated version of the original data and omitting use of the pages of original data.

48. (Previously Presented) The method of claim 47, wherein reading data and organizing pages of the read data by their associated logical addresses additionally comprises utilizing the pages of original data that have not been updated.

49. (Previously Presented) The method of claim 48, wherein programming original data into individual ones of the first plurality of pages and programming an updated version of the original data into individual ones of the second plurality

of pages additionally comprises programming the individual pages with an indication of a relative time of programming the data therein, and

reading data and organizing pages of the read data additionally comprise, for pages of data having the same logical addresses associated therewith, selecting the updated data from the read pages having the most recent time indications and omitting use of the data from others of the pages.

50. (Previously Presented) The method of claim 46, wherein programming data into the first plurality of pages in at least the first block and programming data into the second plurality of pages in the second block each comprise programming data into individual pages of the individual blocks in a specified sequence.

51. (Previously Presented) The method of claim 50, wherein reading data and organizing pages of the read data comprise reading the first and second plurality of pages in an order that is reverse to said specified sequence, and ignoring data read from pages having logical addresses that are the same as logical addresses associated with other pages of data that have already been read.

52. (Previously Presented) The method of claim 46, wherein the logical addresses associated with the pages of original data individually includes a logical block number and a logical page offset.

53. (Previously Presented) The method of claim 46, wherein programming original data into the first plurality of pages and programming the updated version of the original data into the second plurality of pages each comprise programming the logical addresses in individual pages along with the data with which the programmed logical addresses are associated.

54. (Previously Presented) The method of claim 46, wherein programming original data into individual ones of the first plurality of pages in at least a first block includes programming the original data into a first plurality of pages in at least the first block and a third

one of the blocks of charge storage elements located in different ones of a plurality of units of the memory system, wherein the plurality of units are physically separate groupings of blocks of charge storage elements in which programming operations may be performed independently, and linking the first and third blocks together as a metablock to cause their charge storage elements to be erasable together and their pages to be programmable together in parallel.

55. (Previously Presented) The method of any one of claims 46 – 54, additionally comprising:

subsequently programming, into individual ones of the second plurality of pages, a further updated version of at least some of the original data programmed into the first plurality of pages that have previously been updated and the updated version programmed into the second plurality of pages,

wherein the logical addresses associated with the pages of the further updated version of the original data are the same as the logical addresses associated with the pages of original data and the pages of the previously updated version of the original data.

56. (Previously Presented) The method of claim 46, wherein the charge storage elements of the memory system in which the method is carried out comprise electrically conductive floating gates.

57. (Previously Presented) The method of any one of claims 46 – 54, additionally comprising operating the individual memory system charge storage elements with more than two storage states, thereby storing more than one bit of data in each storage element, wherein programming the pages includes programming the individual memory storage elements into more than two storage states and reading data includes reading the more than two storage states from the individual memory storage elements.

58. (Previously Presented) The method of any one of claims 46 – 54, wherein the memory system in which the method is practiced is contained within an enclosed card having an

electrical connector along one edge thereof for operably connecting the memory system with a host system.

59. (Previously Presented) In a re-programmable non-volatile semiconductor memory system having a plurality of blocks of memory charge storage elements that are erasable together as a unit, the plurality of blocks individually being divided into a plurality of a given number of pages of memory storage elements that are programmable together, a method of operating the memory system, comprising:

programming individual ones of a first plurality of pages in at least a first block with original data, the pages of original data having logical addresses associated therewith,
thereafter programming individual ones of a second plurality of a total number of pages less than said given number in a second block with updated data and a logical address associated with the page of updated data, wherein the logical addresses associated with the pages of updated data programmed into the second plurality of pages are the same as those associated with the pages of original data programmed into the first plurality of pages,
wherein programming the second plurality of pages additionally comprises programming the updated version of the original data in those of the second plurality of pages that have different offset positions within the second block than the offset positions of the first plurality of pages within said at least the first block that contain pages of original data with the same logical addresses associated therewith,
thereafter reading data from the first and second plurality of pages, and
organizing pages of the read data by the logical addresses associated therewith.

60. (Previously Presented) The method of claim 59, wherein reading data and organizing pages of the read data by the logical addresses associated therewith comprises, for a plurality of pages of read data having the same logical addresses associated therewith, utilizing the pages of the updated version of the original data and omitting use of the pages of original data.

61. (Previously Presented) The method of claim 60, wherein reading data and organizing pages of the read data by the logical addresses associated therewith additionally comprises utilizing the pages of original data that have not been updated.

62. (Previously Presented) The method of claim 59, wherein programming data into the first plurality of pages in at least the first block and programming data into the second plurality of pages in the second block each comprise programming data into individual pages of the individual blocks in a specified sequence.

63. (Previously Presented) The method of claim 59, wherein the logical address associated with the page of original data includes a logical block number and a logical page offset.

64. (Previously Presented) The method of claim 59, wherein programming original data into the first plurality of pages additionally comprises programming in the pages the logical addresses associated with the data programmed therein.

65. (Previously Presented) The method of claim 59, wherein programming original data into individual ones of the first plurality of pages in at least a first block includes programming the original data into a first plurality of pages in at least the first and a third one of the blocks of charge storage elements located in different ones of a plurality of units of the memory system, wherein the units are physically separate groupings of blocks of charge storage elements in which programming operations may be performed independently, and linking the first and third blocks together as a metablock to cause their charge storage elements to be erasable together and pages thereof to be programmable together in parallel.

66. (Previously Presented) The method of claim 59, additionally comprising: subsequently programming, into individual ones of the second plurality of pages, a further updated version of at least some of the original data programmed into the first plurality of

pages that have previously been updated and the updated version programmed into the second plurality of pages,

wherein the logical addresses associated with the further updated version of the original data are the same as the logical addresses associated with the original data and the previously updated version of the original data.

67. (Previously Presented) The method of claim 59, wherein the addresses associated with pages of original and updated data are individually expressed as a logical block number and a logical page offset.

68. (Previously Presented) A method of operating a memory system having an array of reprogrammable non-volatile charge storage elements organized in blocks of storage elements that are erasable together and in pages of storage elements within the blocks that are individually programmable as a unit, comprising:

maintaining an indication of a time separately for individual blocks,

as part of writing data into any one or more of the pages of one of the blocks, updating the indication of a time maintained for the one block to a current time,

when updating data previously written into one or more initial pages, writing the updated data into one or more update pages and identifying the data written into the initial and update pages by common logical addresses, and

as part of reading data having common logical addresses from two or more pages of two or more blocks, reading the indications of the times from the two or more blocks and using the data in the two or more blocks having more recent indications of time without using data in the two or more blocks having older indications of time, and reading data from pages within each of the two or more blocks according to a reverse order in physical address to that by which the pages were written, and ignoring data from any page having the same logical address as data of a page from which data have already been read.

69. (Previously Presented) A method of operating a non-volatile memory system having a plurality of blocks of memory storage elements that are individually erasable together,

wherein the individual blocks are divided into a plurality of pages of storage elements that are programmable together, the blocks being organized in at least two separate units in which programming may be performed independently, comprising:

linking at least one block from individual ones of said at least two units to form a metablock wherein the storage elements of its component blocks are erased together, and updating one or more pages of original data within any of the metablock component blocks less than all the pages within the block by programming replacement data into another at least one block in only a designated one of the units regardless of which unit the data being updated are stored.

70. (Previously Presented) The method of claim 69, wherein storing the original and replacement data comprises:

identifying the original and replacement data by the same logical address to the memory system, and

distinguishing the replacement data from the original data by determining the relative order in time in which the original and replacement data have been programmed in their respective pages of the memory.

71. (Previously Presented) The method of any one of claims 46 – 54, which additionally comprises maintaining a data structure of the logical addresses associated with the pages of original data and the logical addresses associated with pages of the updated version of the original data that links (a) physical addresses of a plurality of blocks of memory storage elements that are storing data having common logical addresses with (b) those common logical addresses.

72. (Previously Presented) The method of claim 59, which additionally comprises maintaining a data structure of the logical addresses associated with the pages of original data and the logical addresses associated with pages of the updated version of the original data that links (a) physical addresses of a plurality of blocks of memory storage elements that are storing data having common logical addresses with (b) those common logical addresses.

73. (Previously Presented) The method of any one of claims 46 – 54, wherein organizing pages of the read data comprises writing the pages of read data into a volatile memory within the memory system.

74. (Previously Presented) The method of claim 59, wherein organizing pages of the read data comprises writing the pages of read data into a volatile memory within the memory system.

75. (Previously Presented) The method of claim 55, which additionally comprises maintaining a data structure of the logical addresses associated with the pages of original data and the logical addresses associated with pages of the updated version of the original data that links (a) physical addresses of a plurality of blocks of memory storage elements that are storing data having common logical addresses with (b) those common logical addresses.

76. (Previously Presented) The method of claim 55, wherein organizing pages of the read data comprises writing the pages of read data into a volatile memory within the memory system.

77. (Previously Presented) The method of claim 55, additionally comprising operating the individual memory system charge storage elements with more than two storage states, thereby storing more than one bit of data in each storage element, wherein programming the pages includes programming the individual memory storage elements into more than two storage states and reading data includes reading the more than two storage states from the individual memory storage elements.

78. (Previously Presented) The method of claim 65, additionally comprising: subsequently programming, into individual ones of the second plurality of pages, a further updated version of at least some of the original data programmed into the first plurality of

pages that have previously been updated and the updated version programmed into the second plurality of pages,

wherein the logical addresses associated with the further updated version of the original data are the same as the logical addresses associated with the original data and the previously updated version of the original data.

79. (Previously Presented) The method of claim 78, wherein the addresses associated with pages of original and updated data are individually expressed as a logical block number and a logical page offset.

80. (Previously Presented) The method of claim 69, wherein updating one or more pages of original data further includes programming a logical address associated with both the original and replacement data.

81. (Previously Presented) The method of claim 80, wherein programming a logical address associated with both the original and replacement data includes programming both a logical block number and a logical page offset.

82. (Previously Presented) The method of claim 69, wherein the memory storage elements of the non-volatile memory system in which the method is carried out comprise electrically conductive floating gates.

83. (Previously Presented) The method of claim 69, additionally comprising programming the original and replacement data into the individual memory storage elements with more than two storage states, thereby storing more than one bit of data the individual memory storage elements.

84. (Previously Presented) The method of claim 69, wherein the non-volatile memory system in which the method is practiced is contained within an enclosed card having an

electrical connector along one edge thereof for operably connecting the memory system with a host system.

85. (New) In a re-programmable non-volatile memory system having a plurality of blocks of memory storage elements that are erasable together as a unit, the plurality of blocks individually being divided into a plurality of a given number of pages of memory storage elements that are programmable together, a method of operating the memory system, comprising:

in response to receiving pages of original user data and logical addresses associated therewith, programming individual ones of a first plurality of pages of memory storage elements in at least a first block with the received pages of original user data,

in response to subsequently receiving one or more pages of updated user data and logical addresses associated therewith that are common with the logical addresses associated with at least some of the programmed pages of original user data, programming the one or more pages of updated user data into a second plurality of a total number of pages of memory storage elements less than said given number in at least a second block of memory storage elements, wherein the one or more pages of updated user data are programmable into pages of said at least the second block of memory storage elements having different offset positions therein than the offset positions of pages within said at least the first block of memory storage elements in which pages of original user data were programmed that have logical addresses associated therewith that are common with the logical addresses associated with the pages of updated user data, and

thereafter reading at least pages of original user data that have not been updated from the first plurality of pages of memory storage elements and pages of updated user data from the second plurality of pages of memory storage elements.

86. (New) The method of claim 85, additionally comprising programming the logical addresses associated with the programmed pages of updated user data.

87. (New) The method of claim 86, wherein programming the logical addresses associated with the pages of updated user data includes programming the logical addresses

associated with the pages of updated data within the pages of memory storage elements in which the pages of updated user data with which the logical addresses are associated are programmed.

88. (New) The method of claim 87, additionally comprising reading from the pages of updated user data the logical addresses associated with the read pages of updated user data.

89. (New) The method of any one of claims 86-88, wherein the programmed logical addresses common with those associated with the pages of updated user data individually include a logical block number and a logical page offset.

90. (New) The method of any one of claims 85-87, additionally comprising programming, into those of the first plurality of pages of memory storage elements in which the received pages of original user data are programmed, logical addresses common with the received logical addresses associated with the pages of original user data programmed therein.

91. (New) The method of claim 90, additionally comprising reading from the pages of original user data the logical addresses associated with the read pages of original user data.

92. (New) The method of claim 90, wherein the programmed logical addresses common with those associated with the pages of original user data individually include a logical block number and a logical page offset.

93. (New) The method of claim 85, wherein programming the received pages of original user data includes programming the received pages of original user data in the pages of memory storage elements of at least the first block having offsets therein that correspond to bits of the logical addresses associated with the received pages of original user data.

94. (New) The method of claim 85, additionally comprising organizing the read pages of original user data that have not been updated and the read pages of updated user data in an order of their associated logical addresses.

95. (New) The method of claim 94, wherein organizing the read pages of original and updated user data includes temporarily storing within the memory system at least some of the read pages of user data in the organized order.

96. (New) The method of claim 85, additionally comprising transferring out of the memory system the read pages of original user data that have not been updated and the read pages of updated user data.

97. (New) The method of claim 85, additionally comprising, in response to receiving pages of further updated user data having logical addresses common with the logical addresses associated with at least some of the programmed pages of original user data that have previously been updated, programming the received pages of further updated user data into the second plurality of pages of memory storage elements.

98. (New) The method of any one of claims 94-97, additionally comprising programming, into those of the second plurality of pages of memory storage elements in which the received one or more pages of updated user data are programmed, the logical addresses associated with the pages of updated user data programmed therein.

99. (New) The method of any one of claims 85-88 and 97, wherein the method is practiced by operating a memory system wherein the plurality of blocks of memory storage elements are organized in a plurality of units of physically separate groupings of blocks of memory storage elements in which programming operations may be performed independently, wherein the at least a first block includes the first block in one of the units and at least a third block in another one of the units, the method additionally comprising:

linking the first and third blocks together as a metablock to cause their memory storage elements to be erasable together and their pages to be programmable together in parallel, and

wherein programming the received pages of original user data includes programming the pages of original user data in parallel into a first plurality of pages of memory storage elements in at least the first and third blocks.

100. (New) The method of claim 99, wherein programming the pages of updated user data into the second plurality of pages of memory storage elements includes programming the pages of updated user data into the second block of memory storage elements that is located in one of the plurality of units of the memory system.

101. (New) The method of any one of claims 85-88 and 97, additionally comprising operating the individual memory system memory storage elements with more than two storage states, thereby storing more than one bit of user data in each storage element, wherein programming the pages of memory storage elements includes programming the individual memory storage elements into more than two storage states, and wherein reading user data includes reading the more than two storage states from the individual memory storage elements.

102. (New) The method of claim 85, wherein the memory storage elements of the memory system in which the method is carried out comprise electrically conductive floating gates.

103. (New) The method of claim 85, wherein the memory system in which the method is practiced is contained within an enclosed card having externally accessible electrical contacts connected with the memory system that are connectable with a host system from which the original and updated user data are received.

104. (New) The method of claim 85, wherein programming pages of user data into the first plurality of pages of memory storage elements in at least the first block and programming pages of updated user data into the second plurality of pages in the second block each comprise programming user data into individual pages of the individual blocks in a specified sequence.

105. (New) The method of claim 104, wherein reading user data include reading the first and second plurality of pages of user data in an order that is reverse to said specified sequence, and ignoring user data read from pages having logical addresses that are common with logical addresses associated with other pages of data that have already been read.

106. (New) In a re-programmable non-volatile semiconductor memory system having a plurality of sub-arrays of memory storage elements in which programming operations may be performed independently, the sub-arrays individually being divided into a plurality of blocks of memory storage elements that are erasable together as a unit, the plurality of blocks individually being divided into a plurality of pages of memory storage elements that are programmable together, a method of operating the memory system, comprising:

linking together blocks within the plurality of sub-arrays to form a plurality of metablocks whose memory storage elements are erasable together and whose pages of memory storage elements within the linked blocks are programmable together in parallel,

programming pages of original data of a file into individual ones of an erased first plurality of pages in a first plurality of blocks forming a first metablock, the pages of original data having logical addresses associated therewith,

thereafter programming one or more pages of updated data of the file into individual ones of an erased second plurality of pages in at least a second block, the pages of updated data having logical addresses associated therewith, wherein the logical addresses associated with the pages of updated data programmed into the second plurality of pages are common with those associated with the pages of original data programmed into the first plurality of pages,

wherein programming the second plurality of pages additionally allows programming the updated version of the original data in those of the second plurality of pages of memory storage elements that have different offset positions within the at least the second block than the offset positions of the first plurality of pages within the first plurality of blocks that contain pages of original data with common logical addresses associated therewith, and

thereafter reading data of the file from the first and second plurality of pages.

107. (New) The method of claim 106, wherein programming the updated data includes programming the one or more pages of updated data of the file into individual ones of the erased second plurality of pages in the at least the second block that includes a second plurality of blocks forming a second metablock.

108. (New) The method of claim 106, wherein programming the updated data includes programming the one or more pages of updated data of the file into individual ones of the erased second plurality of pages in only the second block.

109. (New) The method of claim 106, wherein the memory system in which the method is carried out is characterized by the pages of memory storage elements being programmable in a preset order within the individual blocks, and wherein programming the pages of original data includes programming the pages of original data of the file in an order of their associated logical addresses.

110. (New) The method of any one of claims 106-109, which additionally comprises programming the logical addresses associated with the programmed pages of the updated data within the pages of memory storage elements in which the pages of updated data with which the logical addresses are associated are programmed.

111. (New) The method of claim 110, wherein the programmed logical addresses common with those associated with the pages of updated data individually include a logical block number and a logical page offset.

112. (New) The method of any one of claims 106-109, which additionally comprises programming the logical addresses associated with the programmed pages of the original data within the pages of memory storage elements in which the pages of original data with which the logical addresses are associated are programmed.

113. (New) The method of claim 112, wherein the programmed logical addresses common with those associated with the pages of original data individually include a logical block number and a logical page offset.

114. (New) The method of claim 106, additionally comprising programming pages of further updated data into a third plurality of pages, the pages of further updated data having logical addresses common with the logical addresses associated with at least some of the programmed pages of original data that have previously been updated.

115. (New) The method of claim 114, additionally comprising programming the logical addresses associated with the programmed pages of the updated data and the further updated data within the pages of memory storage elements in which the pages of updated data with which the logical addresses are associated are programmed.

116. (New) The method of any one of claims 106-109, wherein reading data of the file includes reading the pages of original data of the file that have not been updated and the pages of updated data of the file, and wherein the method additionally comprises organizing the read pages of original data of the file that have not been updated and the pages of updated data of the file in an order of their associated logical addresses.

117. (New) The method of claim 106, additionally comprising transferring out of the memory system the read pages of original data that have not been updated and the read pages of updated data.

118. (New) The method of any one of claims 106-109 and 114-115, additionally comprising operating the individual memory system memory storage elements with more than two storage states, thereby storing more than one bit of data in each of the memory storage elements, wherein programming the pages of memory storage elements includes programming the individual memory storage elements into more than two storage states, and wherein reading

the data includes reading the more than two storage states from the individual memory storage elements.

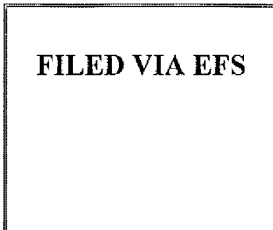
REMARKS

Continued examination of the present application is being requested in order to have the foregoing new claims and information provided by an accompanying Supplemental Information Disclosure Statement considered by the Examiner.


By this Amendment, new independent method claims 85 and 106 are being added which are generally directed to subject matter similar to that of allowed independent claims 46 and 59 but with a different scope. Claims 85 and 106 are believed to be patentable for at least the same reasons as given for the allowance of claims 46 and 59, and the claims dependent on claims 85 and 106 are therefore also believed to be patentable.

The Supplemental Information Disclosure Statement being filed herewith includes the public version of an Initial Determination by the Administrative Law Judge in Investigation No. 337-TA-619 of the United States International Trade Commission (ITC) of alleged infringements of grandparent patent no. 6,763,424, assigned to SanDisk Corporation. This Initial Determination includes the Judge's opinion as to the validity, infringement and enforceability of claims 17, 20, 24 and 30 of patent no. 6,763,424.

It is submitted that the present application remains allowable after this amendment, and an early formal allowance is earnestly solicited. But if the Examiner has any further matters that need to be resolved or any questions about this Supplemental Amendment, please contact James Hsue at 415-276-6541 (direct line).



Respectfully submitted,

 June 8, 2009

Gerald P. Parsons Date
Reg. No. 24,486

Davis Wright Tremaine LLP
505 Montgomery Street, Suite 800
San Francisco, CA 94111-6533
(415) 276-6500 (main)
(415) 276-6534 (direct)
(415) 276-6599 (fax)
Email: geraldparsons@dwt.com

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875					Application or Docket Number 11/250,238		Filing Date 10/13/2005		<input type="checkbox"/> To be Mailed														
APPLICATION AS FILED – PART I																							
(Column 1)			(Column 2)			SMALL ENTITY <input type="checkbox"/>		OR			OTHER THAN SMALL ENTITY												
FOR		NUMBER FILED		NUMBER EXTRA		RATE (\$)		FEE (\$)		OR		RATE (\$)		FEE (\$)									
<input type="checkbox"/> BASIC FEE (37 CFR 1.16(a), (b), or (c))		N/A		N/A		N/A				OR		N/A											
<input type="checkbox"/> SEARCH FEE (37 CFR 1.16(k), (l), or (m))		N/A		N/A		N/A				OR		N/A											
<input type="checkbox"/> EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))		N/A		N/A		N/A				OR		N/A											
TOTAL CLAIMS (37 CFR 1.16(i))		minus 20 =		*		X \$ =				OR		X \$ =											
INDEPENDENT CLAIMS (37 CFR 1.16(h))		minus 3 =		*		X \$ =				OR		X \$ =											
<input type="checkbox"/> APPLICATION SIZE FEE (37 CFR 1.16(s))		If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$250 (\$125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).																					
<input type="checkbox"/> MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))																							
* If the difference in column 1 is less than zero, enter "0" in column 2.												TOTAL				TOTAL							
APPLICATION AS AMENDED – PART II																							
(Column 1)			(Column 2)			(Column 3)			SMALL ENTITY		OR		OTHER THAN SMALL ENTITY										
AMENDMENT	06/08/2009		CLAIMS REMAINING AFTER AMENDMENT				HIGHEST NUMBER PREVIOUSLY PAID FOR		PRESENT EXTRA		RATE (\$)		ADDITIONAL FEE (\$)		OR		RATE (\$)		ADDITIONAL FEE (\$)				
	Total (37 CFR 1.16(o))		* 183		Minus		** 97		= 86		X \$ =				OR		X \$2=		4472				
	Independent (37 CFR 1.16(h))		* 8		Minus		***6		= 2		X \$ =				OR		X \$220=		440				
	<input type="checkbox"/> Application Size Fee (37 CFR 1.16(s))																						
	<input type="checkbox"/> FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))																						
												TOTAL ADD'L FEE		OR		TOTAL ADD'L FEE		4912					
AMENDMENT			CLAIMS REMAINING AFTER AMENDMENT				HIGHEST NUMBER PREVIOUSLY PAID FOR		PRESENT EXTRA		RATE (\$)		ADDITIONAL FEE (\$)		OR		RATE (\$)		ADDITIONAL FEE (\$)				
	Total (37 CFR 1.16(o))		*		Minus		**		=		X \$ =				OR		X \$ =						
	Independent (37 CFR 1.16(h))		*		Minus		***		=		X \$ =				OR		X \$ =						
	<input type="checkbox"/> Application Size Fee (37 CFR 1.16(s))																						
	<input type="checkbox"/> FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))																						
												TOTAL ADD'L FEE		OR		TOTAL ADD'L FEE							
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.																							
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".																							
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".																							
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.																							
This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.												Legal Instrument Examiner: /FREDERICK E. BRISCOE/											
If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.																							



EIC 2100 SEARCH REQUEST

Today's Date

Name <input type="text" value="Mano Padmanabhan"/>	Priority App. Filing Date
AU/Org. <input type="text" value="IC 2100"/> Examiner # <input type="text" value="76556"/>	Case/App. #
Bld.&Rm.# <input type="text" value="RND/2A79"/> Phone <input type="text" value="517-272-4210"/>	Format for Search Results EMAIL <input checked="" type="checkbox"/> PAPER <input type="checkbox"/>

If this is a Board of Appeals case, check here

Synonyms

Describe this invention in your own words.
.....
.....

Terms to avoid

Additional Comments

Please do a litigation search on Patent # 6763424. If there is any litigation found, please obtain full details.
thanks
mano

Please submit completed form to your EIC. SPE Signature here indicates Rush

STIC USE ONLY

Searcher	Date Completed
Phone	Sources

1 of 1 DOCUMENT

UNITED STATES PATENT AND TRADEMARK OFFICE GRANTED PATENT

6763424

[Link to Claims Section](#)

July 13, 2004

Partial block data programming and reading operations in a non-volatile memory

INVENTOR: Conley, Kevin M. - San Jose, California, United States (US)

APPL-NO: 766436 (09)

FILED-DATE: January 19, 2001

GRANTED-DATE: July 13, 2004

ASSIGNEE-PRE-ISSUE: January 19, 2001 - ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS)., SANDISK CORPORATION 140 CASPIAN COURTSUNNYVALE, CALIFORNIA, 94089, Reel and Frame Number: 011487/0823

ASSIGNEE-AT-ISSUE: SanDisk Corporation, Sunnyvale, California, United States (US), United States company or corporation (02)

REL-DATA:

Prior Publication 20020099904, July 25, 2002, PENDING

CORE TERMS: memory, updated, stored, logical, cell, superceded, programmed, programming, controller, flash ...

ENGLISH-ABST:

Data in less than all of the pages of a non-volatile memory block are updated by programming the new data in unused pages of either the same or another block. In order to prevent having to copy unchanged pages of data into the new block, or to program flags into superceded pages of data, the pages of new data are identified by the same logical address as the pages of data which they superceded and a time stamp is added to note when each page was written. When reading the data, the most recent pages of data are used and the older superceded pages of data are ignored. This technique is also applied to metablocks that include one block from each of several different units of a memory array, by directing all page updates to a single unused block in one of the units.

1 of 1 DOCUMENT

Sandisk Corp. v. Phison Elecs. Corp.

3:07-cv-00605-bbc, 3:07-cv-00607-bbc

**UNITED STATES DISTRICT COURT FOR THE WESTERN DISTRICT OF
WISCONSIN**

538 F. Supp. 2d 1060; 2008 U.S. Dist. LEXIS 6576

January 28, 2008, Entered

SUBSEQUENT HISTORY: Motion granted by, Motion denied by Sandisk Corp. v. Phison Elecs. Corp., 2008 U.S. Dist. LEXIS 70693 (W.D. Wis., Sept. 17, 2008)

CORE TERMS: patent, technology, infringement, memory, flash, hardship, mandatory, consolidation, consolidate, discretionary ...

OPINION

... **[*1063]** **[**9]** *U.S. Patent Nos. 6,149,316 (the '316 patent) and 6,757,842 (the '842 patent)*. Case No. 607 alleges infringement of *U.S. Patent Nos. 6,426,893 (the '893 patent), 5,719,808 (the '808 patent), 6,763,424 (the '424 patent), 6,947,332 (the '332 patent) and 7,137,011* **[*1064]** *(the '011 patent)*. Both cases assert that a variety of defendants manufactured or distributed products known ...

No Documents Found

No documents were found for your search terms

"6763424 or 6,763,424"

Click "Save this search as an Alert" to schedule your search to run in the future.

- OR -

Click "Edit Search" to return to the search form and modify your search.

Suggestions:

- Check for spelling errors .
 - Remove some search terms.
 - Use more common search terms, such as those listed in "Suggested Words and Concepts"
 - Use a less restrictive date range.
-

Save this Search as an Alert

Edit Search



LexisNexis®

[About LexisNexis](#) | [Terms & Conditions](#) | [Contact Us](#)

Copyright © 2009 LexisNexis, a division of Reed Elsevier Inc. All rights reserved.



1 of 4 DOCUMENTS

Copyright 2009 Warren Communications News, Inc
All Rights Reserved
CONSUMER ELECTRONICS DAILY

April 14, 2009 Tuesday

SECTION: COURTS

LENGTH: 239 words

HEADLINE: COURTS

BODY:

SanDisk is "disappointed" that ITC Administrative Law Judge Charles Bullock found in a ruling Friday that competitors' flash-memory products didn't infringe the two SanDisk U.S. patents still at issue in a patent lawsuit, SanDisk said. "We will continue to vigorously pursue actions against companies that use SanDisk's patented technology without a license," said Earle Thompson, the company's chief intellectual property counsel.

The company "expects that the initial determination will not adversely impact existing licensing agreements or the royalties expected from those agreements," he said. SanDisk sued in October 2007 on allegations that five of its patents were violated by 25 companies that manufacture, sell and import USB flash drives, CompactFlash cards, MultiMedia cards, MP3/media players and other removable flash storage products. SanDisk has won judgments against a few. Other companies, including PNY Technologies, signed settlement and licensing deals with SanDisk. Companies including Buffalo Technology and Corsair Memory signed settlement agreements that included consent orders requiring them to limit imports of specified products into the U.S. to SanDisk-licensed products, it said. But Bullock ruled Friday that Imation's "accused products do not infringe claim 8" of U.S. Patent No. 7,137,011, and named products from Phison, SMI and Skmedi "do not infringe claims 17,24 or 30" of U.S. Patent No. 6,763,424.

LOAD-DATE: April 13, 2009



2 of 4 DOCUMENTS

Copyright 2008 Federal Information and News Dispatch, Inc.
U.S. International Trade Commission Documents and Publications

October 10, 2008

SECTION: REGULATORY DOCUMENTS

LENGTH: 615 words

HEADLINE: In the Matter of: Certain Flash Memory Controllers, Drives, Memory Cards, and Media Players and Products Containing Same; Notice of Commission Decision Not To Review an Initial Determination Granting-in-Part Complainant's Motion To Amend the Complaint and Amending the Notice of Investigation

BODY:

SUMMARY: Notice is hereby given that the U.S. International Trade Commission has determined not to review an initial determination ("ID") (Order No. 40) issued by the presiding administrative law judge ("ALJ") in the above-referenced investigation granting-in-part complainant's motion to amend the complaint and amending the notice of investigation.

FOR FURTHER INFORMATION CONTACT: Michelle Walters, Office of the General Counsel, U.S. International Trade Commission, 500 E Street, SW., Washington, DC 20436, telephone (202) 708-5468. Copies of non-confidential documents filed in connection with this investigation are or will be available for inspection during official business hours (8:45 a.m. to 5:15 p.m.) in the Office of the Secretary, U.S. International Trade Commission, 500 E Street, SW., Washington, DC 20436, telephone (202) 205-2000. General information concerning the Commission may also be obtained by accessing its Internet server at <http://www.usitc.gov>. The public record for this investigation may be viewed on the Commission's electronic docket (EDIS) at <http://edis.usitc.gov>. Hearing-impaired persons are advised that information on this matter can be obtained by contacting the Commission's TDD terminal on (202) 205-1810.

SUPPLEMENTARY INFORMATION: The Commission instituted this investigation on December 12, 2007, based on a complaint filed by SanDisk Corporation ("SanDisk"). The complaint alleged violations of section 337 of the Tariff Act of 1930 (19 U.S.C. **1337) in the importation into the United States, the sale for importation, and the sale within the United States after importation of certain flash memory controllers, drives, memory cards, media players, and products containing the same by reason of infringement of various claims of five United States patents. The complaint named nearly fifty respondents.

On July 31, 2008, SanDisk moved to amend the complaint to (1) add Verbatim Americas, LLC as a respondent to reflect respondent Verbatim Corporation's corporate restructuring; (2) add as respondents Zhubai Chipsbank Microelectronics Co., Ltd. ("Zhubai") and Chipsbrand Technologies (HK) Co., Ltd. ("Chipsbrand"), both of which are wholly-owned subsidiaries of respondent Chipsbank Technologies (Shenzhen) Co., Ltd. ("Chipsbank"); (3) clarify that claims 12, 14, 17, and 58 of U.S. Patent No. 6,426,893 are asserted against respondent Afa Technologies, Inc.; (4) assert claim 8 of U.S. Patent No. 7,137,011 against respondents Transcend Information, Inc. (Taiwan), Transcend Information, Inc. (California), and Transcend Information Maryland, Inc.; and (5) assert claims 24 and 30 of U.S. Patent

In the Matter of: Certain Flash Memory Controllers, Drives, Memory Cards, and Media Players and Products
Containing Same; Notice of Commission Decision Not To Review an Initial Determination Granting-

No. 6,763,424 against respondent Chipsbank and proposed respondents Zhubai and Chipsbrand.

On September 12, 2008, the ALJ issued Order No. 40, granting SanDisk's motion with regard to items (1) and (2) and also so amending the notice of investigation, but denying the motion or finding it moot with regard to items (3)-(5). Only those portions of Order No. 40 granting SanDisk's motion and amending the notice of investigation constitute an initial determination subject to potential Commission review. No petitions for review of this ID were filed.

The Commission has determined not to review the ALJ's ID.

The authority for the Commission's determination is contained in section 337 of the Tariff Act of 1930, as amended (19 U.S.C. 1337), and in section 210.42 of the Commission's Rules of Practice and Procedure (19 CFR 210.42).

Issued: October 6, 2008.

By order of the Commission.

Marilyn R. Abbott,

Secretary to the Commission.

Notice.

Citation: "73 FR 60322"

Document Number: "Investigation No. 337-TA-619"

Federal Register Page Number: "60322"

"Notices"

LOAD-DATE: October 10, 2008



3 of 4 DOCUMENTS

Copyright 2007 Federal Information and News Dispatch, Inc.
U.S. International Trade Commission Documents and Publications

December 12, 2007

SECTION: REGULATORY DOCUMENTS

LENGTH: 1697 words

HEADLINE: In the Matter of: Certain Flash Memory Controllers, Drives, Memory Cards, and Media Players and Products Containing Same; Notice of Investigation;
Federal Register Extracts

BODY:

SUMMARY: Notice is hereby given that a complaint was filed with the U.S. International Trade Commission on October 24, 2007, under section 337 of the Tariff Act of 1930, as amended, 19 U.S.C. 1337, on behalf of SanDisk Corporation of Milpitas, California. A supplement to the complaint was filed on November 7, 2007. The complaint, as supplemented, alleges violations of section 337 in the importation into the United States, the sale for importation, and the sale within the United States after importation of certain flash memory controllers, drives, memory cards, and media players and products containing same by reason of infringement of certain claims of U.S. Patent Nos. 6,426,893, 6,763,424, 5,719,808, 6,947,332, and 7,137,011. The complaint, as supplemented, further alleges that an industry in the United States exists as required by subsection (a)(2) of section 337.

ADDRESSES: The complaint, as supplemented, except for any confidential information contained therein, is available for inspection during official business hours (8:45 a.m. to 5:15 p.m.) in the Office of the Secretary, U.S. International Trade Commission, 500 E Street, SW., Room 112, Washington, DC 20436, telephone 202-205-2000. Hearing impaired individuals are advised that information on this matter can be obtained by contacting the Commission's TDD terminal on 202-205-1810. Persons with mobility impairments who will need special assistance in gaining access to the Commission should contact the Office of the Secretary at 202-205-2000. General information concerning the Commission may also be obtained by accessing its internet server at <http://www.usitc.gov>. The public record for this investigation may be viewed on the Commission's electronic docket (EDIS) at <http://edis.usitc.gov>.

FOR FURTHER INFORMATION CONTACT: Christopher G. Paulraj, Esq., Office of Unfair Import Investigations, U.S. International Trade Commission, telephone (202) 205-3052.

Authority: The authority for institution of this investigation is contained in section 337 of the Tariff Act of 1930, as amended, and in section 210.10 of the Commission's Rules of Practice and Procedure, 19 CFR 210.10 (2007).

Scope of Investigation: Having considered the complaint, the U.S. International Trade Commission, on December 4, 2007, ordered that --

(1) Pursuant to subsection (b) of section 337 of the Tariff Act of 1930, as amended, an investigation be instituted to determine whether there is a violation of subsection (a)(1)(B) of section 337 in the importation into the United States,

In the Matter of: Certain Flash Memory Controllers, Drives, Memory Cards, and Media Players and Products
Containing Same; Notice of Investigation; Federal Register Extracts U.S. International Trade Co

the sale for importation, or the sale within the United States after importation of certain flash memory controllers, drives, memory cards, and media players and products containing same by reason of infringement of one or more of claims 12-14, 17, 25, 27, 30, 36, 37, 39, 41, and 58 of U.S. Patent No. 6,426,893; claims 17, 18, 24, and 30 of U.S. Patent No. ~~6,763,424~~; claims 11, 14-17, 20, and 21 of U.S. Patent No. 5,719,808; claims 5 and 10 of U.S. Patent No. 6,947,332; and claim 8 of U.S. Patent No. 7,137,011, and whether an industry in the United States exists as required by subsection (a)(2) of section 337;

(2) For the purpose of the investigation so instituted, the following are hereby named as parties upon which this notice of investigation shall be served:

(a) The complainant is--

SanDisk Corporation, 601 McCarthy Boulevard, Milpitas, California 95035.

(b) The respondents are the following entities alleged to be in violation of section 337, and are the parties upon which the complaint is to be served:

Phison Electronics Corporation, 2F, No. 669, Sec. 4, Zhongxing Road, Zhudong Town, Hsinchu County, Taiwan.

Silicon Motion Technology Corporation, Silicon Motion Inc., No. 8F-1, No. 36, Taiyuan Street, Zhubei City, Hsinchu County, Taiwan.

Silicon Motion, Inc., Silicon Motion International, Inc., 1591 McCarthy Blvd., Milpitas, California 95035.

USBest Technology, Inc., 7F, No. 1, Jinshan 8th Street, East District, Hsinchu City, Taiwan.

Skymedi Corporation, 5F, No. 6, Dusing 1st Road, Hsinchu Science Park, Hsinchu, 300, Taiwan.

Chipsbrand Microelectronics (HK) Co., Ltd., 31/F The Landmark Gloucester Road, 11 Pedder St., Central District, Hong Kong Island, Hong Kong, Chipsbank Technology (Shenzhen) Co., Ltd.

Chipsbank Microelectronics Co., Ltd., No. 201-205, 2/F, Bldg. No. 4, Keji Central Road 2, Software Park, South Area High-Tech Industrial Park, Shenzhen, China 518057.

Zotek Electronic Co., Ltd., DbA Zodata Technology Limited, Rm 2502, 25/F, EW International Tower, 120 Texaco Road, Tsuen Wan, Hong Kong.

Infotech Logistic, LLC, DbA Supertron Memory, c/o USA Corporate Services Inc., 46 State Street, 3rd Floor, Albany, New York 12207.

Power Quotient International Co., Ltd., 14F, No. 16, Jian 8th Road, Zhonghe City, Taipei County, Taiwan.

Power Quotient International (HK) Co., Ltd., Flat F. 4/F, Yeung Yiu Chung (No. 8) Industrial, Building, 20 Wang Hoi Road, Kowloon Bay, Kowloon, Hong Kong.

Syscom Development Co., Ltd., c/o Insigner Corporation Services (BVI) Ltd., Palm Grow Service House, PO Box 438, Road Town, Tortola, British Virgin Islands.

PQI Corporation, 46539 Fremont Blvd., Fremont, California 94538.

PNY Technologies, Inc., 299 Webro Road #2, Parsippany, New Jersey 07054-0218.

Kingston Technology Company, Inc., Kingston Technology Corporation, 17600 Newhope St., Fountain Valley, California 92708.

In the Matter of: Certain Flash Memory Controllers, Drives, Memory Cards, and Media Players and Products
Containing Same; Notice of Investigation; Federal Register Extracts U.S. International Trade Co

Payton Technology Corporation, 17600 Newhope St., Ste. B, Fountain Valley, California 92708.

MemoSun, Inc., 17600 Newhope St., Fountain Valley, California 92708.

Melco Holdings, Inc., 4-11-50, Osu, Naka-Ku, 460-0011 Nagoya, Aichi, Japan.

Buffalo, Inc., 15, Shibata hondori 4-chome, Minami-ku, Nagoya, 457-8520, Japan.

Buffalo Technology (USA), Inc., 11100 Metric Blvd., Suite 750, Austin, Texas 78758.

Verbatim Corporation, 1200 West W.T. Harris Blvd., Charlotte, North Carolina 28262.

Transcend Information Inc., No. 70, Xing Zhong Road, Nei Hu Dist., Taipei 11494, Taiwan.

Transcend Information Inc., 1645 North Brian St., Orange, California 92867.

Transcend Information Maryland, Inc., Suites Q and R, 514 Progress Drive, Linthicum, Maryland 21090.

Imation Corp., Imation Enterprises Corp., 1 Imation Place, Oakdale, Minnesota 55128.

Memorex Products, Inc., Imation Consumer Division, 17777 Center Court Drive, Suite 800, Cerritos, California 90703.

Add-On Computer Peripherals, Inc., Add-On Computer Peripherals, LLC, DbA Acp-Ep Memory, DbA Ep Memory, 34 Mauchly, Suite A, Irvine, California 92618.

Add-On Technology Co., 1F, No. 11, Lane 206, Da-An, Road Sec. 1, Taipei, Taiwan.

A-Data Technology Co., Ltd., 18F, No. 25, Liancheng Road, Zhonghe City, Taipei County, Taiwan.

A-Data Technology (USA) Co., Ltd., 3149 Skyway Court, Fremont, California 94539.

Acer, Inc., 8F, 88, Sec. 1, Xintai 5th Road, Xizhi City, Taipei County, Taiwan.

Apacer Technology Inc., 9F, 100, Sec. 1, Xintai 5th Road, Xizhi City, Taipei County, Taiwan.

Apacer Memory America, Inc., 380 Fairview Way, Milpitas, California 95035.

Behavior Tech Computer Corp., 20F-B, No. 98, Sec. 1, Xintai 5th Road., Xizhi City, Taipei County, Taiwan.

Emprex Technologies Corp., 20F, 108 Xintai 5th Road, Sec. 1, Xizhi City, Taipei County, Taiwan.

Behavior Tech Computer (USA) Corp., DbA BTC USA, 4180 Business Center Dr., Fremont, California 94538.

Corsair Memory, Inc., 46221 Landing Parkway, Fremont, California 94538.

Dane-Elec Memory S.A., 149-165 Avenue Gallieni, 93171 Bagnolet, France.

Deantusaiocht Dane-Elec TEO, DbA Dane-Elec Manufacturing, Spiddal Industrial Estate, Spiddal, Galway, Ireland.

Dane Elec Corp. USA, DbA Intervalle Corporation, DbA Dane-Elec Manufacturing USA, 15770 Laguna Canyon Road, #100, Irvine, California 92618.

EDGE Tech Corporation, DbA Peripheral Enhancements Corporation, 1310 North Hills Center, Ada, Oklahoma 74820.

In the Matter of: Certain Flash Memory Controllers, Drives, Memory Cards, and Media Players and Products
Containing Same; Notice of Investigation; Federal Register Extracts U.S. International Trade Co

Interactive Media Corp., Dba Kanguru Solutions, 3 Christina Center, 120 Jeffrey Ave, Holliston, Massachusetts
01746.

Kaser Corporation, 46711 Fremont Blvd., Fremont, California 94538.

LG Electronics, Inc., LG Twin Towers, 20 Yeouido-dong, Yeongdeungpo-gu, Seoul, Seoul 150875, Republic of
Korea.

LG Electronics U.S.A., Inc., 1000 Sylvan Ave., Englewood Cliffs, New Jersey 07632.

TSR Silicon Resources Inc., 16 West 30th Street, New York, New York 10001.

Welldone Company, 1F., No. 181, Anmei Street, Neihu District, Taipei City, Taiwan.

(c) The Commission investigative attorney, party to this investigation, is Christopher G. Paulraj, Esq., Office of
Unfair Import Investigations, U.S. International Trade Commission, 500 E Street, SW., Room 401P, Washington, DC
20436; and

(3) For the investigation so instituted, the Honorable Charles E. Bullock is designated as the presiding
administrative law judge.

Responses to the complaint and the notice of investigation must be submitted by the named respondents in
accordance with section 210.13 of the Commission's Rules of Practice and Procedure, 19 CFR 210.13. Pursuant to 19
CFR 201.16(d) and 210.13(a), such responses will be considered by the Commission if received not later than 20 days
after the date of service by the Commission of the complaint and the notice of investigation. Extensions of time for
submitting responses to the complaint and the notice of investigation will not be granted unless good cause therefor is
shown.

Failure of a respondent to file a timely response to each allegation in the complaint and in this notice may be
deemed to constitute a waiver of the right to appear and contest the allegations of the complaint and this notice, and to
authorize the administrative law judge and the Commission, without further notice to the respondent, to find the facts to
be as alleged in the complaint and this notice and to enter an initial determination and a final determination containing
such findings, and may result in the issuance of an exclusion order or cease and desist order or both directed against a
respondent.

By order of the Commission.

Issued: December 6, 2007.

Marilyn R. Abbott,

--This is a summary of a Federal Register article originally published on the page number listed below--

Institution of investigation pursuant to 19 U.S.C. 1337.

Citation: "72 FR 70610"

Document Number: "Inv. No. 337-TA-619"

Federal Register Page Number: "70610"

"Notices"

In the Matter of: Certain Flash Memory Controllers, Drives, Memory Cards, and Media Players and Products
Containing Same; Notice of Investigation; Federal Register Extracts U.S. International Trade Co

LOAD-DATE: December 12, 2007

US District Court Civil Docket

**U.S. District - Wisconsin Western
(Madison)**

3:07cv607

Sandisk Corp v. Phison Electronics Corp et al

This case was retrieved from the court on Thursday, August 20, 2009

Date Filed: 10/24/2007	Class Code: CONSOLIDATED_CASE, MG, STAYED
Assigned To: Chief Judge Barbara B Crabb	Closed: No
Referred To: Magistrate Judge Stephen L Crocker	Statute: 35:271
Nature of suit: Patent (830)	Jury Demand: Plaintiff
Cause: Patent Infringement	Demand Amount: \$0
Lead Docket: 3:07-cv-00605-bbc	NOS Description: Patent
Other Docket: None	
Jurisdiction: Federal Question	

Litigants

Sandisk Corp
Plaintiff

Attorneys

Allen Arntsen
[COR LD NTC]
Foley & Lardner LLP
Post Office Box 1497
Madison , WI 53701
USA
608-258-4293
Fax: 608-258-4258
Email: Aarntsen@foley.com

David Adam Caine
[COR LD NTC]
Wilson Sonsini Goodrich & Rosati
650 Page Mill Road
Palo Alto , CA 94304-1050
USA
650-493-9300x3025
Fax: 650-565-5100
Email: DCAINE@WSGR.COM

James C Yoon
[COR LD NTC]
Wilson Sonsini Goodrich & Rosati
650 Page Mill Road
Palo Alto , CA 94304-1050
USA

Jose C Villarreal
[COR LD NTC]
Wilson Sonsini Goodrch & Rosati
8911 Capital of Texas Highway North
Suite 3350
Austin , TX 78759
USA

Julie Holloway
[COR LD NTC]

Wilson, Sonsini, Goodrich & Rosati
650 Page Mill Road
Palo Alto , CA 94304
USA
650-493-9300x4562
Email: Jholloway@wsgr.com

Lisa Kim ANH Nguyen
[COR LD NTC]
Wilson Sonsini Goodrich & Rosati
650 Page Mill Road
Palo Alto , CA 94304-1050
USA
650-493-9300x3755
Fax: 650-565-5100
Email: Lknguyen@wsgr.com

Matthew J BYE
[COR LD NTC]
Wilson Sonsini Goodrich & Rosati, PC
1700 K Street, NW
Washington , DC 20006
USA

Michael A Ladra
[COR LD NTC]
Wilson Sonsini Goodrich & Rosati
650 Page Mill Road
Palo Alto , CA 94304-1050
USA
650-493-9300x4869
Fax: 650-493-6811
Email: Mladra@wsgr.com

Monica Mucchetti ENO
[COR LD NTC]
Wilson Sonsini Goodrich & Rosati
650 Page Mill Road
Palo Alto , CA 94304-1050
USA
650-493-9300
Fax: 650-565-5100
Email: Meno@wsgr.com

Nicole W Stafford
[COR LD NTC]
Wilson Sonsini Goodrich & Rosati
900 South Capital of Texas Highway
Las Cimas IV, Fifth Floor
Austin , TX 78746-5546
USA
512-338-5402
Fax:
Email: NSTAFFORD@WSGR.COM

Olga V Kotlyarevskaya
[COR LD NTC]
Wilson Sonsini Goodrich & Rosati
650 Page Mill Road
Palo Alto , CA 94304
USA

Ron E Shulman
[COR LD NTC]
Wilson Sonsini Goodrich & Rosati
650 Page Mill Road
Palo Alto , CA 94304-1050

USA
(650) 493-9300

Jeffrey A Simmons
[COR LD NTC]
Foley & Lardner
150 East Gilman St
PO Box 1497
Madison , WI 53701
USA
608-257-5035x4267
Fax: 608-258-4258
Email: Jsimmons@foley.com

Phison Electronics Corp
Defendant

Anthony J Sievert
[COR LD NTC]
Whyte Hirschboeck Dudek
1 East Main Street, Suite 300
Madison , WI 53703
USA
608-255-4440
Fax: 608-258-7138
Email: Asievert@whdlaw.com

David M Barkan
[COR LD NTC]
Fish & Richardson PC
500 Arguello Street, Suite 500
Redwood City , CA 94063
USA
650-839-5065
Fax: 650-839-5071
Email: BARKAN@FR.COM

Eugenia G Carter
[COR LD NTC]
Whyte Hirschboeck Dudek
1 East Main Street, Suite 300
Madison , WI 53703
USA
608-234-6058
Fax: 608-258-7138
Email: Gcarter@whdlaw.com

Silicon Motion Technology Corp
Defendant

Silicon Motion, Inc (Taiwan)
Defendant

Silicon Motion, Inc (California)
Defendant

Silicon Motion International, Inc
Defendant

Synergistic Sales, Inc
Defendant
[Term: 01/17/2008]

Usbest Technology, Inc
Defendant

David J Harth
[COR LD NTC]
[Term: 03/06/2009]
Perkins Coie LLP
One East Main Street, Ste 201

Madison , WI 53703
USA
608-663-7470
Fax: 608-663-7499
Email: DHARTH@PERKINSCOIE.COM

Maureen Frances Browne
[COR LD NTC]
Covington & Burling LLP
1201 Pennsylvania Avenue, NW
Washington , DC 20004
USA
202-662-5038
Fax: 202-778-5038
Email: MBROWNE@COV.COM

Catherine Cetrangolo
[COR LD NTC]
Boardman, Suhr, Curry & Field
One South Pinckney St, 4TH Floor
PO Box 927
Madison , WI 53701
USA
608-257-9521
Email: Ccetrangolo@boardmanlawfirm.com

Skymedi Corp
Defendant

Chipsbrand Microelectronics (Hk) Co, Ltd
Defendant

Chipsbank Technology (Shenzhen) Co, Ltd
Defendant

Chipsbank Microelectronics Co, Ltd
Defendant

Zotek Electronic Co, Ltd
Defendant

Zodata Technology Ltd
Defendant

Infotech Logistic Llc
Defendant
[Term: 01/23/2008]

David J Harth
[COR LD NTC]
[Term: 03/06/2009]
Perkins Coie LLP
One East Main Street, Ste 201
Madison , WI 53703
USA
608-663-7470
Fax: 608-663-7499
Email: DHARTH@PERKINSCOIE.COM

Melinda Hein
[COR LD NTC]
[Term: 01/11/2008]
Godfrey, Braun & Frazier, LLP
735 North Water St
Milwaukee , WI 53202
USA
414-278-8500x123
Fax: 414-278-0421

Power Quotient International Co, Ltd

David J Harth

Defendant

[COR LD NTC]
[Term: 03/06/2009]
Perkins Coie LLP
One East Main Street, Ste 201
Madison , WI 53703
USA
608-663-7470
Fax: 608-663-7499
Email: DHARTH@PERKINSCOIE.COM

Maureen Frances Browne
[COR LD NTC]
Covington & Burling LLP
1201 Pennsylvania Avenue, NW
Washington , DC 20004
USA
202-662-5038
Fax: 202-778-5038
Email: MBROWNE@COV.COM

Catherine Cetrangolo
[COR LD NTC]
Boardman, Suhr, Curry & Field
One South Pinckney St, 4TH Floor
PO Box 927
Madison , WI 53701
USA
608-257-9521
Email: Ccetrangolo@boardmanlawfirm.com

Pqi Corp
Defendant

David J Harth
[COR LD NTC]
[Term: 03/06/2009]
Perkins Coie LLP
One East Main Street, Ste 201
Madison , WI 53703
USA
608-663-7470
Fax: 608-663-7499
Email: DHARTH@PERKINSCOIE.COM

Maureen Frances Browne
[COR LD NTC]
Covington & Burling LLP
1201 Pennsylvania Avenue, NW
Washington , DC 20004
USA
202-662-5038
Fax: 202-778-5038
Email: MBROWNE@COV.COM

Catherine Cetrangolo
[COR LD NTC]
Boardman, Suhr, Curry & Field
One South Pinckney St, 4TH Floor
PO Box 927
Madison , WI 53701
USA
608-257-9521
Email: Ccetrangolo@boardmanlawfirm.com

Pny Technologies, Inc
Defendant
[Term: 01/10/2008]

David J Harth
[COR LD NTC]
[Term: 03/06/2009]
Perkins Coie LLP

One East Main Street, Ste 201
Madison , WI 53703
USA
608-663-7470
Fax: 608-663-7499
Email: DHARTH@PERKINSCOIE.COM

William J Heller
[COR LD NTC]
McCarter & English, LLP
Four Gateway Center
100 Mulberry Street
Newark , NJ 07102
USA
973-639-6973

Kingston Technology Co, Inc
Defendant

Christine Yang
[COR LD NTC]
PREV

Anthony J Sievert
[COR LD NTC]
Whyte Hirschboeck Dudek
1 East Main Street, Suite 300
Madison , WI 53703
USA
608-255-4440
Fax: 608-258-7138
Email: Asievert@whdlaw.com

David M Barkan
[COR LD NTC]
Fish & Richardson PC
500 Arguello Street, Suite 500
Redwood City , CA 94063
USA
650-839-5065
Fax: 650-839-5071
Email: BARKAN@FR.COM

Eugenia G Carter
[COR LD NTC]
Whyte Hirschboeck Dudek
1 East Main Street, Suite 300
Madison , WI 53703
USA
608-234-6058
Fax: 608-258-7138
Email: Gcarter@whdlaw.com

Kingston Technology Corp
Defendant

Christine Yang
[COR LD NTC]
PREV

Anthony J Sievert
[COR LD NTC]
Whyte Hirschboeck Dudek
1 East Main Street, Suite 300
Madison , WI 53703
USA
608-255-4440
Fax: 608-258-7138
Email: Asievert@whdlaw.com

Payton Technology Corp
Defendant

David M Barkan
[COR LD NTC]
Fish & Richardson PC
500 Arguello Street, Suite 500
Redwood City , CA 94063
USA
650-839-5065
Fax: 650-839-5071
Email: BARKAN@FR.COM

Eugenia G Carter
[COR LD NTC]
Whyte Hirschboeck Dudek
1 East Main Street, Suite 300
Madison , WI 53703
USA
608-234-6058
Fax: 608-258-7138
Email: Gcarter@whdlaw.com

Christine Yang
[COR LD NTC]
PREV

Anthony J Sievert
[COR LD NTC]
Whyte Hirschboeck Dudek
1 East Main Street, Suite 300
Madison , WI 53703
USA
608-255-4440
Fax: 608-258-7138
Email: Asievert@whdlaw.com

David M Barkan
[COR LD NTC]
Fish & Richardson PC
500 Arguello Street, Suite 500
Redwood City , CA 94063
USA
650-839-5065
Fax: 650-839-5071
Email: BARKAN@FR.COM

Eugenia G Carter
[COR LD NTC]
Whyte Hirschboeck Dudek
1 East Main Street, Suite 300
Madison , WI 53703
USA
608-234-6058
Fax: 608-258-7138
Email: Gcarter@whdlaw.com

Memosun, Inc
Defendant

Christine Yang
[COR LD NTC]
PREV

Anthony J Sievert
[COR LD NTC]
Whyte Hirschboeck Dudek
1 East Main Street, Suite 300

Buffalo, Inc
Defendant

Melco Holdings, Inc
Defendant

Madison , WI 53703
USA
608-255-4440
Fax: 608-258-7138
Email: Asievert@whdlaw.com

David M Barkan
[COR LD NTC]
Fish & Richardson PC
500 Arguello Street, Suite 500
Redwood City , CA 94063
USA
650-839-5065
Fax: 650-839-5071
Email: BARKAN@FR.COM

Eugenia G Carter
[COR LD NTC]
Whyte Hirschboeck Dudek
1 East Main Street, Suite 300
Madison , WI 53703
USA
608-234-6058
Fax: 608-258-7138
Email: Gcarter@whdlaw.com

Frank J West
[COR LD NTC]
Oblon, Spivak, McClelland, Maier & Neustadt, PC
1940 Duke St
Alexandria , VA 22314
USA
703-412-7049
Fax: 703-413-2220
Email: FWEST@OBLON.COM

Richard D Kelly
[COR LD NTC]
Oblon, Spivak, McClelland, Maier & Neustadt, PC
1940 Duke Street
Alexandria , VA 22314
USA

Robert C Mattson
[COR LD NTC]
Oblon, Spivak, McClelland, Maier & Neustadt, PC
1940 Duke St
Alexandria , VA 22314
USA
703-412-6466
Fax:
Email: RMATTSON@OBLON.COM

Takahiro Miura
[COR LD NTC]
Oblon, Spivak, McClelland, Maier & Neustadt PC
1940 Duke St
Alexandria , VA 22314
USA
703-412-6217
Fax:
Email: TMIURA@OBLON.COM

Frank J West
[COR LD NTC]

Oblon, Spivak, McClelland, Maier & Neustadt, PC
1940 Duke St
Alexandria , VA 22314
USA
703-412-7049
Fax: 703-413-2220
Email: FWEST@OBLON.COM

Richard D Kelly
[COR LD NTC]
Oblon, Spivak, McClelland, Maier & Neustadt, PC
1940 Duke Street
Alexandria , VA 22314
USA

Robert C Mattson
[COR LD NTC]
Oblon, Spivak, McClelland, Maier & Neustadt, PC
1940 Duke St
Alexandria , VA 22314
USA
703-412-6466
Fax:
Email: RMATTSON@OBLON.COM

Takahiro Miura
[COR LD NTC]
Oblon, Spivak, McClelland, Maier & Neustadt PC
1940 Duke St
Alexandria , VA 22314
USA
703-412-6217
Fax:
Email: TMIURA@OBLON.COM

Buffalo Technology (Usa) Inc
Defendant

Frank J West
[COR LD NTC]
Oblon, Spivak, McClelland, Maier & Neustadt, PC
1940 Duke St
Alexandria , VA 22314
USA
703-412-7049
Fax: 703-413-2220
Email: FWEST@OBLON.COM

Richard D Kelly
[COR LD NTC]
Oblon, Spivak, McClelland, Maier & Neustadt, PC
1940 Duke Street
Alexandria , VA 22314
USA

Robert C Mattson
[COR LD NTC]
Oblon, Spivak, McClelland, Maier & Neustadt, PC
1940 Duke St
Alexandria , VA 22314
USA
703-412-6466
Fax:
Email: RMATTSON@OBLON.COM

Takahiro Miura
[COR LD NTC]
Oblon, Spivak, McClelland, Maier & Neustadt PC
1940 Duke St

Verbatim Corp
Defendant

Alexandria , VA 22314
USA
703-412-6217
Fax:
Email: TMIURA@OBLON.COM

Perry Clark
[COR LD NTC]
Kirkland & Ellis LLP
555 California Street
Suite 2700
San Francisco , CA 94104
USA

Brian C Kwok
[COR LD NTC]
[Term: 03/24/2009]
Kirkland & Ellis LLP
555 California Street
Suite 2700
San Francisco , CA 94104
USA
415-439-1851
Email: Bkwok@kirkland.com

Transcend Information Inc (Taiwan)
Defendant

Anup Tikku
[COR LD NTC]
K&L Gates LLP
55 Second Street
Suite 1700
San Francisco , CA 94105
USA
415-882-8200

Chien-Wei Chou
[COR LD NTC]
K&L Gates
55 - 2ND Street
Suite 1700
San Francisco , CA 94105
USA
415-882-8200x8000
Fax: 415-882-8220
Email: CHRIS.CHOU@KLGATES.COM

Jane C Schlicht
[COR LD NTC]
Cook & Franke SC
660 E Mason Street
Milwaukee , WI 53202
USA
414-271-5900x1291
Fax: 414-271-2002
Email: SCHLICHT@CF-Law.com

Jong Ho Lee
[COR LD NTC]
K&L Gates
55 - 2ND Street
Suite 1700
San Francisco , CA 94105
USA
415-882-8036
Fax: 415-882-8220
Email: HANK.LEE@KLGATES.COM

Transcend Information Inc (California, U.S.A.)
Defendant

Michael J Bettinger
[COR LD NTC]
K&L Gates
55 Second Street
Suite 1700
San Francisco , CA 94105
USA
(415)882-8200

Rummel Mor Bautista
[COR LD NTC]
The Ryan Firm
1100 N Tustin Avenue
Suite 200
Anaheim , CA 92807
USA
714-666-1362

Anup Tikku
[COR LD NTC]
K&L Gates LLP
55 Second Street
Suite 1700
San Francisco , CA 94105
USA
415-882-8200

Chien-Wei Chou
[COR LD NTC]
K&L Gates
55 - 2ND Street
Suite 1700
San Francisco , CA 94105
USA
415-882-8200x8000
Fax: 415-882-8220
Email: CHRIS.CHOU@KLGATES.COM

Jane C Schlicht
[COR LD NTC]
Cook & Franke SC
660 E Mason Street
Milwaukee , WI 53202
USA
414-271-5900x1291
Fax: 414-271-2002
Email: SCHLICHT@CF-Law.com

Jong Ho Lee
[COR LD NTC]
K&L Gates
55 - 2ND Street
Suite 1700
San Francisco , CA 94105
USA
415-882-8036
Fax: 415-882-8220
Email: HANK.LEE@KLGATES.COM

Michael J Bettinger
[COR LD NTC]
K&L Gates
55 Second Street
Suite 1700
San Francisco , CA 94105

Transcend Information Maryland, Inc
Defendant

USA
(415)882-8200

Rummel Mor Bautista
[COR LD NTC]
The Ryan Firm
1100 N Tustin Avenue
Suite 200
Anaheim , CA 92807
USA
714-666-1362

Anup Tikku
[COR LD NTC]
K&L Gates LLP
55 Second Street
Suite 1700
San Francisco , CA 94105
USA
415-882-8200

Chien-Wei Chou
[COR LD NTC]
K&L Gates
55 - 2ND Street
Suite 1700
San Francisco , CA 94105
USA
415-882-8200x8000
Fax: 415-882-8220
Email: CHRIS.CHOU@KLGATES.COM

Jane C Schlicht
[COR LD NTC]
Cook & Franke SC
660 E Mason Street
Milwaukee , WI 53202
USA
414-271-5900x1291
Fax: 414-271-2002
Email: SCHLICHT@CF-Law.com

Jong Ho Lee
[COR LD NTC]
K&L Gates
55 - 2ND Street
Suite 1700
San Francisco , CA 94105
USA
415-882-8036
Fax: 415-882-8220
Email: HANK.LEE@KLGATES.COM

Michael J Bettinger
[COR LD NTC]
K&L Gates
55 Second Street
Suite 1700
San Francisco , CA 94105
USA
(415)882-8200

Rummel Mor Bautista
[COR LD NTC]
The Ryan Firm
1100 N Tustin Avenue

Imation Corp
Defendant

Suite 200
Anaheim , CA 92807
USA
714-666-1362

Allen A Slaughter , Jr
[COR LD NTC]
Robins, Kaplan, Miller & Ciresi, LLP
2800 Lasalle Plaza
800 Lasalle Avenue
Minneapolis , MN 55402
USA
(612) 349-8500
Fax: (612) 339-4181

Byron Todd Jones
[COR LD NTC]
Robins, Kaplan, Miller & Ciresi LLP
2800 Lasalle Plaza
800 Lasalle Avenue
Minneapolis , MN 55402
USA
612-349-8545
Fax: 612-339-4181
Email: Btjones@rkmc.com

Jennifer L McKenna
[COR LD NTC]
Robins, Kaplan, Miller & Ciresi LLP
2800 Lasalle Plaza
800 Lasalle Avenue
Minneapolis , MN 55402-2015
USA
612-349-8211
Fax: 612-339-4181
Email: Jlmckenna@rkmc.com

Lester A Pines
[COR LD NTC]
Cullen Weston Pines & Bach LLP
122 W Washington Ave
Suite 900
Madison , WI 53703
USA
608-251-0101
Fax: 608-251-2883
Email: Pines@cwpb.com

Ronald J Schutz
[COR LD NTC]
Robins, Kaplan, Miller & Ciresi, LLP
2800 Lasalle Plaza
800 Lasalle Avenue
Minneapolis , MN 55402
USA
(612) 349-8500
Fax: (612) 339-4181

Imation Enterprises Corp
Defendant

Allen A Slaughter , Jr
[COR LD NTC]
Robins, Kaplan, Miller & Ciresi, LLP
2800 Lasalle Plaza
800 Lasalle Avenue
Minneapolis , MN 55402
USA

(612) 349-8500
Fax: (612) 339-4181

Byron Todd Jones
[COR LD NTC]
Robins, Kaplan, Miller & Ciresi LLP
2800 Lasalle Plaza
800 Lasalle Avenue
Minneapolis , MN 55402
USA
612-349-8545
Fax: 612-339-4181
Email: Btjones@rkmc.com

Jennifer L McKenna
[COR LD NTC]
Robins, Kaplan, Miller & Ciresi LLP
2800 Lasalle Plaza
800 Lasalle Avenue
Minneapolis , MN 55402-2015
USA
612-349-8211
Fax: 612-339-4181
Email: Jlmckenna@rkmc.com

Lester A Pines
[COR LD NTC]
Cullen Weston Pines & Bach LLP
122 W Washington Ave
Suite 900
Madison , WI 53703
USA
608-251-0101
Fax: 608-251-2883
Email: Pines@cwpb.com

Ronald J Schutz
[COR LD NTC]
Robins, Kaplan, Miller & Ciresi, LLP
2800 Lasalle Plaza
800 Lasalle Avenue
Minneapolis , MN 55402
USA
(612) 349-8500
Fax: (612) 339-4181

Memorex Products, Inc
Defendant

Allen A Slaughter , Jr
[COR LD NTC]
Robins, Kaplan, Miller & Ciresi, LLP
2800 Lasalle Plaza
800 Lasalle Avenue
Minneapolis , MN 55402
USA
(612) 349-8500
Fax: (612) 339-4181

Byron Todd Jones
[COR LD NTC]
Robins, Kaplan, Miller & Ciresi LLP
2800 Lasalle Plaza
800 Lasalle Avenue
Minneapolis , MN 55402
USA
612-349-8545
Fax: 612-339-4181
Email: Btjones@rkmc.com

Jennifer L McKenna
[COR LD NTC]
Robins, Kaplan, Miller & Ciresi LLP
2800 Lasalle Plaza
800 Lasalle Avenue
Minneapolis, MN 55402-2015
USA
612-349-8211
Fax: 612-339-4181
Email: Jlmckenna@rkmc.com

Lester A Pines
[COR LD NTC]
Cullen Weston Pines & Bach LLP
122 W Washington Ave
Suite 900
Madison, WI 53703
USA
608-251-0101
Fax: 608-251-2883
Email: Pines@cwpb.com

Ronald J Schutz
[COR LD NTC]
Robins, Kaplan, Miller & Ciresi, LLP
2800 Lasalle Plaza
800 Lasalle Avenue
Minneapolis, MN 55402
USA
(612) 349-8500
Fax: (612) 339-4181

Add-On Computer Peripherals, Inc
Defendant
[Term: 01/28/2008]

Add-On Peripherals, Llc
Defendant
[Term: 01/28/2008]

Add-On Technology Co
Defendant

A-Data Technology Co, Ltd
Defendant

David J Harth
[COR LD NTC]
[Term: 03/06/2009]
Perkins Coie LLP
One East Main Street, Ste 201
Madison, WI 53703
USA
608-663-7470
Fax: 608-663-7499
Email: DHARTH@PERKINSCOIE.COM

Maureen Frances Browne
[COR LD NTC]
Covington & Burling LLP
1201 Pennsylvania Avenue, NW
Washington, DC 20004
USA
202-662-5038
Fax: 202-778-5038
Email: MBROWNE@COV.COM

Catherine Cetrangolo

[COR LD NTC]
Boardman, Suhr, Curry & Field
One South Pinckney St, 4TH Floor
PO Box 927
Madison , WI 53701
USA
608-257-9521
Email: Ccetrangolo@boardmanlawfirm.com

David L Anstaett
[COR LD NTC]
[Term: 03/06/2009]
Perkins Coie LLP
One East Main St
Ste 201
Madison , WI 53703-5118
USA
608-663-5408
Fax: 608-663-7499
Email: DANSTAETT@PERKINSCOIE.COM

A-Data Technology (Usa) Co, Ltd
Defendant

David J Harth
[COR LD NTC]
[Term: 03/06/2009]
Perkins Coie LLP
One East Main Street, Ste 201
Madison , WI 53703
USA
608-663-7470
Fax: 608-663-7499
Email: DHARTH@PERKINSCOIE.COM

Maureen Frances Browne
[COR LD NTC]
Covington & Burling LLP
1201 Pennsylvania Avenue, NW
Washington , DC 20004
USA
202-662-5038
Fax: 202-778-5038
Email: MBROWNE@COV.COM

Catherine Cetrangolo
[COR LD NTC]
Boardman, Suhr, Curry & Field
One South Pinckney St, 4TH Floor
PO Box 927
Madison , WI 53701
USA
608-257-9521
Email: Ccetrangolo@boardmanlawfirm.com

David L Anstaett
[COR LD NTC]
[Term: 03/06/2009]
Perkins Coie LLP
One East Main St
Ste 201
Madison , WI 53703-5118
USA
608-663-5408
Fax: 608-663-7499
Email: DANSTAETT@PERKINSCOIE.COM

Apacer Technology Inc

Jane C Schlicht

Defendant

[COR LD NTC]
Cook & Franke SC
660 E Mason Street
Milwaukee , WI 53202
USA
(414) 271-5900 X1291
Fax: (414) 271-2002
Email: SCHLICHT@CF-Law.com

William David Shenk
[COR LD NTC]
Whyte Hirschboeck Dudek
33 East Main Street
Suite 300
Madison , WI 53703
USA
608-255-4440x6059
Fax: 608-258-7138
Email: Dshenk@whdlaw.com

Anthony J Sievert
[COR LD NTC]
Whyte Hirschboeck Dudek
1 East Main Street, Suite 300
Madison , WI 53703
USA
608-255-4440
Fax: 608-258-7138
Email: Asievert@whdlaw.com

Eugenia G Carter
[COR LD NTC]
Whyte Hirschboeck Dudek
1 East Main Street, Suite 300
Madison , WI 53703
USA
608-234-6058
Fax: 608-258-7138
Email: Gcarter@whdlaw.com

Apacer Memory America, Inc
Defendant

Jane C Schlicht
[COR LD NTC]
Cook & Franke SC
660 E Mason Street
Milwaukee , WI 53202
USA
414-271-5900x1291
Fax: 414-271-2002
Email: SCHLICHT@CF-Law.com

William David Shenk
[COR LD NTC]
Whyte Hirschboeck Dudek
33 East Main Street
Suite 300
Madison , WI 53703
USA
608-255-4440x6059
Fax: 608-258-7138
Email: Dshenk@whdlaw.com

Anthony J Sievert
[COR LD NTC]
Whyte Hirschboeck Dudek
1 East Main Street, Suite 300
Madison , WI 53703

USA
608-255-4440
Fax: 608-258-7138
Email: Asievert@whdlaw.com

Eugenia G Carter
[COR LD NTC]
Whyte Hirschboeck Dudek
1 East Main Street, Suite 300
Madison , WI 53703
USA
608-234-6058
Fax: 608-258-7138
Email: Gcarter@whdlaw.com

Acer, Inc
Defendant
[Term: 01/07/2008]

Behavior Tech Computer Corp
Defendant

Alfredo Bismonte
[COR LD NTC]
Beck, Ross, Bismonte & Finley, LLP Fairmont Plaza
50 W San Fernando St
Suite 1300
San Jose , CA 95113
USA
408-938-7900
Fax: 408-938-0790
Email: ABISMONTE@BECKROSS.COM

Behavior Tech Computer USA Corp
Defendant

Alfredo Bismonte
[COR LD NTC]
Beck, Ross, Bismonte & Finley, LLP Fairmont Plaza
50 W San Fernando St
Suite 1300
San Jose , CA 95113
USA
408-938-7900
Fax: 408-938-0790
Email: ABISMONTE@BECKROSS.COM

Corsair Memory, Inc
Defendant
[Term: 09/05/2008]

David J Harth
[COR LD NTC]
[Term: 03/06/2009]
Perkins Coie LLP
One East Main Street, Ste 201
Madison , WI 53703
USA
608-663-7470
Fax: 608-663-7499
Email: DHARTH@PERKINSCOIE.COM

Dane-Elec Memory, SA
Defendant

Dane-Elec Corp USA
Defendant

J Thomas Haley
[COR LD NTC]
Haley Palmersheim, SC
1424 North High Point Road
PO Box 628005
Middleton , WI 53562
USA
608-836-6400

Fax: 608-836-9697
Email: Haley@hplawoffice.com

Kevin Palmersheim
[COR LD NTC]
Haley Palmersheim, SC
1424 North High Point Road
PO Box 628005
Middleton, WI 53562
USA
608-836-6400
Fax: 608-836-9697
Email: PALMERSHEIM@HPLAWOFFICE.COM

Edge Tech Corp
Defendant
[Term: 01/28/2008]

Cassandra L Wilkinson
[COR LD NTC]
Head, Johnson & Kachigian
228 West 17TH Place
Tulsa, OK 74119
USA
918-587-2000
Email: Cwilkinson@hjkllaw.com

Interactive Media Corp
Defendant
[Term: 01/28/2008]

Lg Electronics, Inc
Defendant

Lg Electronics USA, Inc
Defendant

Tsr Silicon Resources Inc
Defendant
[Term: 01/23/2008]

Welldone Co
Defendant
[Term: 01/18/2008]

Leodis C Matthews
[COR LD NTC]
Matthews & Partners
4322 Wilshire Blvd, Suite 200
Los Angeles, CA 90010
USA
323-930-5690
Fax: 323-930-5693
Email: LMATTHEWS@LMATTYS.COM

Date	#	Proceeding Text
10/24/2007	1	COMPLAINT against all defendants (Filing fee \$ 350 receipt number 64354.), filed by all plaintiffs. (Attachments: # 1 Exhibit Patent 5,719,808, # 2 Exhibit Patent 6,763,424, # 3 Exhibit Patent 6,426,893, # 4 Exhibit Patent 6,947,332, # 5 Exhibit Patent 7,137,011, # 6 JS-44 Civil Cover Sheet) (rep) (Entered: 10/26/2007)
10/24/2007	3	Corporate Disclosure Statement by Plaintiff SanDisk Corp.. (rep) (Entered: 10/26/2007)
10/26/2007	2	Standard attachments for Judge Barbara B. Crabb sent. (Attachments: # 1 BBC Order Governing Filing of Dispositive Motions # 2 Briefing Guidelines) (rep) (Entered: 10/26/2007)
10/30/2007	4	SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendant Synergistic Sales, Inc. Synergistic Sales, Inc. served on 10/26/2007, answer due 11/15/2007. (krj) (Entered: 10/31/2007)
10/30/2007	5	SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendant Synergistic Sales, Inc. (krj) (Entered: 10/31/2007)
11/01/2007	--	Summons Issued as to Imation Corp., Imation Enterprises Corp. (elc) (Entered: 11/02/2007)

- 11/02/2007 6 STIPULATION for Extension of time for Deft. Edge Tech to Answer or Otherwise Plead by Plaintiff SanDisk Corp., Defendant EDGE Tech Corp.. (Attachments: # 1 Proposed Order) (elc) (Entered: 11/05/2007)
- 11/02/2007 7 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendant Imation Corp. served on 10/26/2007, answer due 11/15/2007. (elc) (Entered: 11/05/2007)
- 11/02/2007 8 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendant Imation Enterprises Corp. served on 10/26/2007, answer due 11/15/2007. (elc) (Entered: 11/05/2007)
- 11/02/2007 -- Summons Issued as to Silicon Motion Technology Corp. (elc) (Entered: 11/05/2007)
- 11/05/2007 -- Summons Issued as to Infotech Logistic LLC. (elc) (Entered: 11/05/2007)
- 11/05/2007 9 ORDER granting 6 STIPULATION extending of time for deft. EDGE Tech. to respond to complaint to 12/15/2007. Signed by Magistrate Judge Stephen L Crocker on 11/5/2007. (lj) (Entered: 11/05/2007)
- 11/05/2007 10 NOTICE of Appearance by Lester A. Pines on behalf of Defendants Imation Corp., Imation Enterprises Corp., Memorex Products, Inc. (elc) (Entered: 11/06/2007)
- 11/06/2007 11 **DOCKETING ERROR - NO ENTRY** (krj) (Entered: 11/07/2007)
- 11/07/2007 12 STIPULATION for Extension of time to answer complaint by Defendants Buffalo, Inc., Melco Holdings, Inc., Buffalo Technology (USA) Inc., Plaintiff SanDisk Corp.. (elc) (Entered: 11/08/2007)
- 11/08/2007 -- Summons Issued as to Imation Corp., Imation Enterprises Corp.; Returned to Plaintiff for service. (elc) (Entered: 11/08/2007)
- 11/08/2007 13 MOTION to Admit Ronald J. Schutz, B. Todd Jones, Jennifer L. McKenna and Allen A. Slaughter, Jr. Pro Hac Vice by Defendants Imation Corp., Imation Enterprises Corp., Memorex Products, Inc. Motions referred to Magistrate Judge Stephen L Crocker. (lj) (Entered: 11/14/2007)
- 11/08/2007 14 AFFIDAVIT of Lester A. Pines filed by Defendants Imation Corp., Imation Enterprises Corp., Memorex Products, Inc. re: 13 MOTION to Admit Ronald J. Schutz, B. Todd Jones, Jennifer L. McKenna and Allen A. Slaughter, Jr. Pro Hac Vice filed by Memorex Products, Inc., Imation Corp., Imation Enterprises Corp. (lj) (Entered: 11/14/2007)
- 11/09/2007 15 STIPULATION AND ORDER extension of time to answer complaint; answer due 1/22/08. Signed by Magistrate Judge Stephen L Crocker on 11/8/07. (krj) (Entered: 11/14/2007)
- 11/09/2007 16 MOTION to Admit James C. Yoon Pro Hac Vice by Plaintiff SanDisk Corp. Motions referred to Magistrate Judge Stephen L Crocker. (Attachments: # 1 Cover Letter) (krj) (Entered: 11/14/2007)
- 11/09/2007 17 AFFIDAVIT of Hannah L. Renfro re: 16 MOTION to Admit James C. Yoon Pro Hac Vice filed by SanDisk Corp. (krj) (Entered: 11/14/2007)
- 11/09/2007 18 MOTION to Admit David Caine Pro Hac Vice by Plaintiff SanDisk Corp. Motions referred to Magistrate Judge Stephen L Crocker. (krj) (Entered: 11/14/2007)
- 11/09/2007 19 AFFIDAVIT of Hannah L. Renfro re: 18 MOTION to Admit David Caine Pro Hac Vice filed by SanDisk Corp. (krj) (Entered: 11/14/2007)
- 11/09/2007 20 MOTION to Admit Julie M. Holloway Pro Hac Vice by Plaintiff SanDisk Corp. Motions referred to Magistrate Judge Stephen L Crocker. (krj) (Entered: 11/14/2007)
- 11/09/2007 21 AFFIDAVIT of Hannah L. Renfro re: 20 MOTION to Admit Julie M. Holloway Pro Hac Vice filed by SanDisk Corp. (krj) (Entered: 11/14/2007)
- 11/09/2007 22 MOTION to Admit Olga V. Kotlyarevskaya Pro Hac Vice by Plaintiff SanDisk Corp. Motions referred to Magistrate Judge Stephen L Crocker. (krj) (Entered: 11/14/2007)
- 11/09/2007 23 AFFIDAVIT of Hannah L. Renfro re: 22 MOTION to Admit Olga V. Kotlyarevskaya Pro Hac Vice filed by SanDisk Corp. (krj) (Entered: 11/14/2007)
- 11/09/2007 24 MOTION to Admit Michael A. Ladra Pro Hac Vice by Plaintiff SanDisk Corp. Motions referred to Magistrate Judge Stephen L Crocker. (krj) (Entered: 11/14/2007)
- 11/09/2007 25 AFFIDAVIT of Hannah L. Renfro re: 24 MOTION to Admit Michael A. Ladra Pro Hac Vice filed by SanDisk Corp. (krj) (Entered: 11/14/2007)
- 11/09/2007 26 MOTION to Admit Lisa K. Nguyen Pro Hac Vice by Plaintiff SanDisk Corp. Motions referred to Magistrate Judge Stephen L Crocker. (krj) (Entered: 11/14/2007)
- 11/09/2007 27 AFFIDAVIT of Hannah L. Renfro re: 26 MOTION to Admit Lisa K. Nguyen Pro Hac Vice filed by SanDisk Corp. (krj) (Entered: 11/14/2007)

- 11/09/2007 28 MOTION to Admit Jose C. Villarreal Pro Hac Vice by Plaintiff SanDisk Corp. Motions referred to Magistrate Judge Stephen L Crocker. (krj) (Entered: 11/14/2007)
- 11/09/2007 29 AFFIDAVIT of Hannah L. Renfro re: 28 MOTION to Admit Jose C. Villarreal Pro Hac Vice filed by SanDisk Corp. (krj) (Entered: 11/14/2007)
- 11/09/2007 30 MOTION (ex parte) to appoint Crowe Foreign Services as international process server by Plaintiff SanDisk Corp. Motions referred to Magistrate Judge Stephen L Crocker. (Attachments: # 1 Text of Proposed Order) (krj) (Entered: 11/15/2007)
- 11/09/2007 31 AFFIDAVIT of Celeste Ingalls re: 30 MOTION to appoint Crowe Foreign Services as an international process server filed by SanDisk Corp. (krj) (Entered: 11/15/2007)
- 11/09/2007 32 STIPULATION for extension of time to file responsive pleading by Defendant Verbatim Corp., Plaintiff SanDisk Corp. (krj) (Entered: 11/15/2007)
- 11/09/2007 33 STIPULATION for extension of time to answer complaint by Plaintiff SanDisk Corp., Defendant Interactive Media Corp.. (krj) (Entered: 11/15/2007)
- 11/09/2007 35 STIPULATION for extension of time to file responsive pleading by Plaintiff SanDisk Corp., Defendants Dane-Elec Memory, S.A., Dane-Elec Corp. USA. (krj) (Entered: 11/15/2007)
- 11/12/2007 34 STIPULATION for extension of time to file responsive pleading by Defendant PNY Technologies, Inc., Plaintiff SanDisk Corp.. (krj) (Entered: 11/15/2007)
- 11/12/2007 36 MOTION to Admit Nicole W. Stafford Pro Hac Vice by Plaintiff SanDisk Corp. Motions referred to Magistrate Judge Stephen L Crocker. (krj) (Entered: 11/15/2007)
- 11/12/2007 37 AFFIDAVIT of Hannah L. Renfro re: 36 MOTION to Admit Nicole W. Stafford Pro Hac Vice filed by SanDisk Corp. (krj) (Entered: 11/15/2007)
- 11/13/2007 38 STIPULATION for extension of time to file responsive pleading by Plaintiff SanDisk Corp., Defendant Add-On Computer Peripherals, Inc. (krj) (Entered: 11/15/2007)
- 11/13/2007 -- Summons Issued as to Behavior Tech Computer USA Corp.; delivered to Plaintiff for service. (krj) (Entered: 11/15/2007)
- 11/14/2007 39 STIPULATION for extension of time to file responsive pleading by Defendants Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc., Plaintiff SanDisk Corp. (krj) (Entered: 11/15/2007)
- 11/14/2007 40 STIPULATION for extension of time to file responsive pleading by Plaintiff SanDisk Corp., Defendants Imation Corp., Imation Enterprises Corp., Memorex Products, Inc. (krj) (Entered: 11/15/2007)
- 11/14/2007 41 STIPULATION for extension of time to file responsive pleading by Plaintiff SanDisk Corp., Defendants Transcend Information Inc. (Taiwan), Transcend Information Inc. (California, U.S.A.), Transcend Information Maryland, Inc. (krj) (Entered: 11/15/2007)
- 11/14/2007 42 Report on Filing of Patent or Trademark. (Attachments: # 1 Cover Letter) (krj) (Entered: 11/15/2007)
- 11/14/2007 43 STIPULATION for extension of time to answer complaint by Plaintiff SanDisk Corp., Defendants Silicon Motion Technology Corp., Silicon Motion, Inc. (Taiwan), Silicon Motion, Inc. (California), Silicon Motion International, Inc. (krj) (Entered: 11/15/2007)
- 11/15/2007 44 STIPULATION to waive the Hague Convention and extend time to respond to complaint by Plaintiff SanDisk Corp., Defendants LG ELECTRONICS, INC., LG Electronics U.S.A., Inc. (krj) (Entered: 11/15/2007)
- 11/15/2007 -- Summons Issued as to Add-On Computer Peripherals, Inc.. (elc) (Entered: 11/16/2007)
- 11/15/2007 45 STIPULATION for to extend Responsive Pleading Date for defendant A-Data Technology Co., Ltd and defendnat A-Data Technology (USA) Co., Ltd by Plaintiff SanDisk Corp.. (elc) (Entered: 11/19/2007)
- 11/19/2007 46 STIPULATION for Extension of time for Deft. Corsair to file responsive pleading by Plaintiff SanDisk Corp., Defendant Corsair Memory, Inc.. (elc) (Entered: 11/20/2007)
- 11/19/2007 47 STIPULATION for Extension of time for Deft. Infotech Logistic LLC to file responsive pleading by Plaintiff SanDisk Corp.. (elc) (Entered: 11/20/2007)
- 11/19/2007 48 STIPULATION for Extension of time for defendants Behavior Tech Computer Corp and Behavior Tech Computer (USA)Corp. by Plaintiff SanDisk Corp. (elc) (Entered: 11/20/2007)
- 11/20/2007 49 MOTION to Admit Perry Clark Pro Hac Vice by Defendant Verbatim Corp.. Motions referred to Magistrate Judge Stephen L Crocker. (Attachments: # 1 Certificate of Service) (elc) (Entered: 11/21/2007)

- 11/20/2007 50 AFFIDAVIT of Brian Kwok re: 49 MOTION to Admit Perry Clark Pro Hac Vice filed by Verbatim Corp. (elc) (Entered: 11/21/2007)
- 11/21/2007 52 STIPULATION for Extension of time for defendant Phison Electronics Corp. to respond to complaint. (elc) (Entered: 11/23/2007)
- 11/21/2007 53 NOTICE of Appearance by David Barkan, Anthony J. Sievert, Eugenia G. Carter Defendant Phison Electronics Corp. (elc) (Entered: 11/23/2007)
- 11/21/2007 54 NOTICE of Appearance by David Barkan, Eugenia G. Carter, Anthony J. Sievert Defendants Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc. (elc) (Entered: 11/23/2007)
- 11/21/2007 55 MOTION to Consolidate and Stay or in the alternative to stay by Defendants Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc.. Motions referred to Magistrate Judge Stephen L Crocker. (elc) (Entered: 11/23/2007)
- 11/21/2007 56 BRIEF in Support re: 55 MOTION to Consolidate and Stay or in the alternative to Stay filed by MemoSun, Inc., Kingston Technology Corp., Kingston Technology Co., Inc., Payton Technology Corp. (Attachments: # 1 Certificate of Service) (elc) (Entered: 11/23/2007)
- 11/21/2007 57 MOTION to Stay by Defendants Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc.. Motions referred to Magistrate Judge Stephen L Crocker. (elc) (Entered: 11/23/2007)
- 11/21/2007 58 BRIEF in Support by Defendants Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc. re: 57 MOTION to Stay filed by MemoSun, Inc., Kingston Technology Corp., Kingston Technology Co., Inc., Payton Technology Corp. (elc) (Entered: 11/23/2007)
- 11/23/2007 51 ORDER granting 30 Motion Appointing of International Process Server. Signed by Magistrate Judge Stephen L Crocker on 11/20/07. (elc) (Entered: 11/23/2007)
- 11/23/2007 59 ** TEXT ONLY ORDER ** Order Granting 49 MOTION to Admit Perry Clark Pro Hac Vice filed by Verbatim Corp.. Perry Clark for Verbatim Corp. admitted pro hac vice. Signed by Magistrate Judge Stephen L Crocker on 11/23/07. (elc) (Entered: 11/23/2007)
- 11/23/2007 60 ** TEXT ONLY ORDER ** Order Granting 36 MOTION to Admit Nicole W. Stafford Pro Hac Vice filed by SanDisk Corp.. Nicole W. Stafford for SanDisk Corp. admitted pro hac vice. Signed by Magistrate Judge Stephen L Crocker on 11/20/07. (elc) (Entered: 11/23/2007)
- 11/23/2007 61 ** TEXT ONLY ORDER ** Order Granting 28 MOTION to Admit Jose C. Villarreal Pro Hac Vice filed by SanDisk Corp.. Jose C. Villarreal for SanDisk Corp. admitted pro hac vice. Signed by Magistrate Judge Stephen L Crocker on 11/20/07. (elc) (Entered: 11/23/2007)
- 11/23/2007 62 ** TEXT ONLY ORDER ** Order Granting 26 MOTION to Admit Lisa K. Nguyen Pro Hac Vice filed by SanDisk Corp.. Lisa K. Nguyen for SanDisk Corp. admitted pro hac vice. Signed by Magistrate Judge Stephen L Crocker on 11/20/07. (elc) (Entered: 11/23/2007)
- 11/23/2007 63 ** TEXT ONLY ORDER ** Order Granting 24 MOTION to Admit Michael A. Ladra Pro Hac Vice filed by SanDisk Corp.. Michael A. Ladra for SanDisk Corp. admitted pro hac vice. Signed by Magistrate Judge Stephen L Crocker on 11/20/07. (elc) (Entered: 11/23/2007)
- 11/23/2007 64 ** TEXT ONLY ORDER ** Order Granting 22 MOTION to Admit Olga V. Kotlyarevskaya Pro Hac Vice filed by SanDisk Corp.. Olga V. Kotlyarevskaya for SanDisk Corp. admitted pro hac vice. Signed by Magistrate Judge Stephen L Crocker on 11/20/07. (elc) (Entered: 11/23/2007)
- 11/23/2007 65 ** TEXT ONLY ORDER ** Order Granting 20 MOTION to Admit Julie M. Holloway Pro Hac Vice filed by SanDisk Corp.. Julie Holloway for SanDisk Corp. admitted pro hac vice. Signed by Magistrate Judge Stephen L Crocker on 11/20/07. (elc) (Entered: 11/23/2007)
- 11/23/2007 66 ** TEXT ONLY ORDER ** Order Granting 18 MOTION to Admit David Caine Pro Hac Vice filed by SanDisk Corp.. David Caine for SanDisk Corp. admitted pro hac vice. Signed by Magistrate Judge Stephen L Crocker on 11/20/07. (elc) (Entered: 11/23/2007)
- 11/23/2007 67 ** TEXT ONLY ORDER ** Order Granting 16 MOTION to Admit James C. Yoon Pro Hac Vice filed by SanDisk Corp.. James C. Yoon for SanDisk Corp. admitted pro hac vice. Signed by Magistrate Judge Stephen L Crocker on 11/20/07. (elc) (Entered: 11/23/2007)
- 11/26/2007 68 ORDER granting 52 STIPULATION for Extension of time until 1/18/08 for Defendant Phison Electronics Corp. to respond to complaint. Signed by Magistrate Judge Stephen L Crocker on 11/26/07. (elc) (Entered: 11/26/2007)
- 11/26/2007 69 ** TEXT ONLY ORDER ** Order Granting 48 STIPULATION to Extend responsive pleading date for defendants Behavior Tech Computer Corp and Behavior Tech Computer (USA)Corp. Answer due by 12/18/2007. Signed by Magistrate Judge Stephen L Crocker on 11/26/07. (elc)

- (Entered: 11/26/2007)
- 11/26/2007 70 ** TEXT ONLY ORDER ** Order Granting 47 STIPULATION for Extension of time for Deft. Infotech Logistic LLC to file responsive pleading. Answer due by 12/5/2007. Signed by Magistrate Judge Stephen L Crocker on 11/26/07. (elc) (Entered: 11/26/2007)
- 11/26/2007 71 ** TEXT ONLY ORDER ** Order Granting 46 STIPULATION for Extension of time for Deft. Corsair to file responsive pleading. Answer due by 12/18/2007. Signed by Magistrate Judge Stephen L Crocker on 11/26/07. (elc) (Entered: 11/26/2007)
- 11/26/2007 72 AFFIDAVIT of Jonathan M. Fritz re: 57 MOTION to Stay filed by MemoSun, Inc., Kingston Technology Corp., Kingston Technology Co., Inc., Payton Technology Corp. (elc) (Entered: 11/27/2007)
- 11/26/2007 73 STIPULATION for Extension of time for defendant TSR Silicon Resources to respond to complaint. (elc) (Entered: 11/27/2007)
- 11/27/2007 74 ** TEXT ONLY ORDER ** Order Accepting and Granting 45 STIPULATION to extend A-Data Defendants Responsive Pleading Date. Answer due by 1/14/2008. Signed by Magistrate Judge Stephen L Crocker on 11/26/07. (elc) (Entered: 11/27/2007)
- 11/27/2007 75 ** TEXT ONLY ORDER ** Order Granting 44 STIPULATION for to waive the Hague Convention and extend time to respond to complaint by Plaintiff SanDisk Corp., Defendants LG ELECTRONICS, INC., LG Electronics U.S.A., Inc. Signed by Magistrate Judge Stephen L Crocker on 11/26/07. (elc) (Entered: 11/27/2007)
- 11/27/2007 76 ** TEXT ONLY ORDER ** Order Granting 43 STIPULATION for extension of time to answer complaint by Defendants Silicon Motion Technology Corp., Silicon Motion, Inc. (Taiwan), Silicon Motion, Inc. (California), Silicon Motion International, Inc. Answer due by 1/22/2008. Signed by Magistrate Judge Stephen L Crocker on 11/26/07. (elc) (Entered: 11/27/2007)
- 11/27/2007 77 ** TEXT ONLY ORDER ** Order Accepting and Granting 41 STIPULATION for extension of time to file responsive pleading by Defendants Transcend Information Inc. (Taiwan), Transcend Information Inc. (California, U.S.A.), Transcend Information Maryland, Inc. Answer due by 1/14/2008. Signed by Magistrate Judge Stephen L Crocker on 11/26/07. (elc) (Entered: 11/27/2007)
- 11/27/2007 78 ** TEXT ONLY ORDER ** Order Accepting and Granting 40 STIPULATION for extension of time to file responsive pleading by Defendants Imation Corp., Imation Enterprises Corp., Memorex Products, Inc. Answer due by 1/1/2008. Signed by Magistrate Judge Stephen L Crocker on 11/26/07. (elc) (Entered: 11/27/2007)
- 11/27/2007 79 ** TEXT ONLY ORDER ** Order Accepting and Granting 39 STIPULATION for extension of time to file responsive pleading by Defendants Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc., Plaintiff SanDisk Corp. Answer due by 12/17/2007. Signed by Magistrate Judge Stephen L Crocker on 11/26/07. (elc) (Entered: 11/27/2007)
- 11/27/2007 80 ** TEXT ONLY ORDER ** Order Granting 38 STIPULATION for extension of time to file responsive pleading by Defendant Add-On Computer Peripherals, Inc. Answer due by 12/17/2007. Signed by Magistrate Judge Stephen L Crocker on 11/26/07. (elc) (Entered: 11/27/2007)
- 11/27/2007 81 ** TEXT ONLY ORDER ** Order Accepting and Granting 35 STIPULATION for extension of time to file responsive pleading by Defendants Dane-Elec Memory, S.A., Dane-Elec Corp. USA. Answer due by 12/31/2007. Signed by Magistrate Judge Stephen L Crocker on 11/26/07. (elc) (Entered: 11/27/2007)
- 11/27/2007 82 ** TEXT ONLY ORDER ** Order Accepting and Granting 34 STIPULATION for extension of time to file responsive pleading by Defendant PNY Technologies, Inc. Answer due by 12/17/2007. Signed by Magistrate Judge Stephen L Crocker on 11/26/07. (elc) (Entered: 11/27/2007)
- 11/27/2007 83 ** TEXT ONLY ORDER ** Order Granting 32 STIPULATION for extension of time to file responsive pleading by Defendant Verbatim Corp. Answer due by 12/17/2007. Signed by Magistrate Judge Stephen L Crocker on 11/26/07. (elc) (Entered: 11/27/2007)
- 11/27/2007 84 ** TEXT ONLY ORDER ** Order Accepting and Granting 33 STIPULATION for extension of time to answer complaint Defendant Interactive Media Corp. Answer due by 12/17/2007. Signed by Magistrate Judge Stephen L Crocker on 11/26/07. (elc) (Entered: 11/27/2007)
- 11/27/2007 85 ** TEXT ONLY ORDER ** Order Accepting and Granting 12 STIPULATION for Extension of time to answer complaint by Defendants Buffalo, Inc., Melco Holdings, Inc., Buffalo Technology (USA) Inc. Signed by Magistrate Judge Stephen L Crocker on 11/26/07. (elc) (Entered: 11/27/2007)
- 11/28/2007 86 MOTION for clarification regarding briefing schedule of motion to Consolidate and stay by

- Plaintiff SanDisk Corp.. Motions referred to Magistrate Judge Stephen L Crocker. (elc) (Entered: 11/28/2007)
- 11/28/2007 87 Response Stating No Opposition to 57 MOTION to Stay filed by MemoSun, Inc., Kingston Technology Corp., Kingston Technology Co., Inc., Payton Technology Corp. (elc) (Entered: 11/29/2007)
- 11/29/2007 88 ** TEXT ONLY ORDER ** Order Accepting and Granting 73 STIPULATION for Extension of time for defendant TSR Silicon Resources to respond to complaint. Answer due by 12/28/2007. Signed by Magistrate Judge Stephen L Crocker on 11/29/07. (elc) (Entered: 11/29/2007)
- 12/03/2007 89 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendants A-Data Technology Co., Ltd., A-Data Technology (USA) Co., Ltd., Acer, Inc. (krj) (Entered: 12/06/2007)
- 12/03/2007 90 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendants Add-On Computer Peripherals, Inc., Add-On Peripherals, LLC, Add-On Technology Co., Apacer Technology Inc., Apacer Memory America, Inc. (krj) (Entered: 12/06/2007)
- 12/03/2007 91 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendants Behavior Tech Computer Corp., Behavior Tech Computer USA Corp. (krj) (Entered: 12/06/2007)
- 12/03/2007 92 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendants Chipsbrand Microelectronics (HK) Co., Ltd., Buffalo, Inc., Buffalo Technology (USA) Inc., Corsair Memory, Inc. (krj) (Entered: 12/06/2007)
- 12/03/2007 93 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendants Imation Corp., Dane-Elec Memory, S.A., Dane-Elec Corp. USA, EDGE Tech Corp. (krj) (Entered: 12/06/2007)
- 12/03/2007 94 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendants Infotech Logistic LLC, Imation Enterprises Corp., Interactive Media Corp. (krj) (Entered: 12/06/2007)
- 12/03/2007 95 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendants Kingston Technology Co., Inc., Kingston Technology Corp., Melco Holdings, Inc., LG ELECTRONICS, INC., LG Electronics U.S.A., Inc. (krj) (Entered: 12/06/2007)
- 12/03/2007 96 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendants PNY Technologies, Inc., Payton Technology Corp., MemoSun, Inc., Memorex Products, Inc., Phison Electronics Corp. (krj) (Entered: 12/06/2007)
- 12/03/2007 97 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendants Power Quotient International Co., Ltd., PQI Corp., Silicon Motion, Inc. (Taiwan), Silicon Motion, Inc. (California). (krj) (Entered: 12/06/2007)
- 12/03/2007 98 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendants Skymedi Corp., Silicon Motion Technology Corp., Silicon Motion International, Inc. (krj) (Entered: 12/06/2007)
- 12/03/2007 99 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendants Synergistic Sales, Inc., Transcend Information Inc. (California, U.S.A.). (krj) (Entered: 12/06/2007)
- 12/03/2007 100 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendants USBest Technology, Inc., Transcend Information Inc. (Taiwan), Transcend Information Maryland, Inc., TSR Silicon Resources Inc. (krj) (Entered: 12/06/2007)
- 12/03/2007 101 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendants Zodata Technology Ltd., Verbatim Corp., Welldone Co. (krj) (Entered: 12/06/2007)
- 12/05/2007 102 STIPULATION for extension of time to file responsive pleading by Defendant USBest Technology, Inc., Plaintiff SanDisk Corp.. (krj) (Entered: 12/06/2007)
- 12/05/2007 103 STIPULATION for extension of time to file responsive pleading by Defendant Infotech Logistic LLC, Plaintiff SanDisk Corp. (krj) (Entered: 12/06/2007)
- 12/06/2007 105 MOTION for Joinder to unopposed motion to stay by Defendant Phison Electronics Corp. Motions referred to Magistrate Judge Stephen L Crocker. (Attachments: # 1 Certificate of Service) (krj) (Entered: 12/10/2007)
- 12/06/2007 106 STIPULATION for extension of time to file responsive pleading by Plaintiff SanDisk Corp., Defendant Welldone Co. (krj) (Entered: 12/10/2007)
- 12/07/2007 107 ORDER that each defendant has until 1/22/08 to answer or otherwise respond to the complaint. No further extensions. Signed by Magistrate Judge Stephen L Crocker on 12/4/07. (elc) (Entered: 12/10/2007)
- 12/10/2007 104 STIPULATION for extension of time to file responsive pleading by Plaintiff SanDisk Corp., Defendant Acer, Inc. (krj) (Entered: 12/10/2007)
- 12/11/2007 109 MOTION to Admit William Heller Pro Hac Vice by Defendant PNY Technologies, Inc. Motions referred to Magistrate Judge Stephen L Crocker. (Attachments: # 1 Certificate of Service) (elc)

- (Entered: 12/13/2007)
- 12/11/2007 110 AFFIDAVIT of Eugenia G. Carter re: 57 MOTION to Stay filed by MemoSun, Inc., Kingston Technology Corp., Kingston Technology Co., Inc., Payton Technology Corp. (Attachments: # 1 Exhibit # 2 Certificate of Service) (elc) (Entered: 12/13/2007)
- 12/12/2007 108 ** TEXT ONLY ORDER ** Order Granting 105 MOTION for Joinder filed by Phison Electronics Corp.. Signed by Magistrate Judge Stephen L Crocker on 12/11/07. (krj) (Entered: 12/12/2007)
- 12/12/2007 -- Set/Reset Deadlines as to 57 MOTION to Stay. Brief in Opposition due by 12/19/2007. Brief in Reply due by 12/31/2007. (krj) (Entered: 12/12/2007)
- 12/13/2007 111 ** TEXT ONLY ORDER ** Order Granting 109 MOTION to Admit William Heller Pro Hac Vice filed by PNY Technologies, Inc. William J. Heller for PNY Technologies, Inc. admitted pro hac vice. Signed by Magistrate Judge Stephen L Crocker on 12/13/07. (krj) (Entered: 12/13/2007)
- 12/14/2007 112 Unopposed MOTION to Stay by Defendants USBest Technology, Inc., Infotech Logistic LLC, Power Quotient International Co., Ltd., PQI Corp., PNY Technologies, Inc., A-Data Technology Co., Ltd., A-Data Technology (USA) Co., Ltd., Corsair Memory, Inc.. Motions referred to Magistrate Judge Stephen L Crocker. Response due by 12/21/2007. (Attachments: # 1 Certificate of Service) (elc) (Entered: 12/17/2007)
- 12/14/2007 113 MOTION to join defts. Kingston, Payton and Memosun unopposed Motion to stay by by Defendants Imation Corp., Imation Enterprises Corp., Memorex Products, Inc.. Motions referred to Magistrate Judge Stephen L Crocker. Response due by 12/21/2007. (elc) (Entered: 12/17/2007)
- 12/17/2007 114 ** TEXT ONLY ORDER ** Order Granting 13 MOTION to Admit Ronald J. Schutz, B. Todd Jones, Jennifer L. McKenna and Allen A. Slaughter, Jr. Pro Hac Vice filed by Memorex Products, Inc., Imation Corp., Imation Enterprises Corp.. Signed by Magistrate Judge Stephen L Crocker on 12/13/07. (elc) (Entered: 12/17/2007)
- 12/17/2007 115 NOTICE of Joinder by Defendants Buffalo, Inc., Melco Holdings, Inc., Buffalo Technology (USA) Inc. re 55 MOTION to Consolidate and Stay or in the Alternative to Stay. (elc) (Entered: 12/18/2007)
- 12/18/2007 116 ** TEXT ONLY ORDER ** Order Granting 113 MOTION for Joinder filed by Memorex Products, Inc., Imation Corp., Imation Enterprises Corp. Signed by Magistrate Judge Stephen L Crocker on 12/18/07. (krj) (Entered: 12/18/2007)
- 12/18/2007 119 BRIEF in Opposition (non-opposition) by Plaintiff SanDisk Corp. to Motion to Stay. (krj) (Entered: 12/19/2007)
- 12/18/2007 120 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendants Power Quotient International Co., Ltd., Apacer Technology Inc., Acer, Inc., Behavior Tech Computer Corp., Silicon Motion, Inc. (Taiwan). (Attachments: # 1 # 2 # 3 # 4) (krj) (Entered: 12/19/2007)
- 12/18/2007 121 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendants USBest Technology, Inc., Skymedi Corp., Transcend Information Inc. (Taiwan), Welldone Co., Phison Electronics Corp., Silicon Motion, Inc. (Taiwan). (Attachments: # 1 # 2 # 3 # 4 # 5) (krj) (Entered: 12/19/2007)
- 12/19/2007 117 ORDER granting motion to join defendants Kingston, Payton and Memosun unopposed motion to stay by Imation Corp., Imation Enterprises Corp. and Memorex Products, Inc. Signed by Judge Barbara B Crabb on 12/18/07. (krj) (Entered: 12/19/2007)
- 12/19/2007 118 ORDER granting 112 Motion to Stay; counsel for moving defendants to advise court on progress of ITC proceeding no later than 3/1/08, or earlier, if commission issues determination. Signed by Judge Barbara B Crabb on 12/18/07. (krj) (Entered: 12/19/2007)
- 12/19/2007 122 MOTION to file documents under seal by Plaintiff SanDisk Corp. Motions referred to Magistrate Judge Stephen L Crocker. (krj) Additional attachment(s) added on 12/20/2007 (Jacobson, Kris). (Entered: 12/20/2007)
- 12/19/2007 123 BRIEF in Opposition by Plaintiff SanDisk Corp. re: 55 MOTION to Consolidate Cases filed by MemoSun, Inc., Kingston Technology Corp., Kingston Technology Co., Inc., Payton Technology Corp. (Redacted Version) (krj) (Entered: 12/20/2007)
- 12/19/2007 124 AFFIDAVIT of Allen A. Arntsen filed by Plaintiff SanDisk Corp. re: 55 MOTION to Consolidate Cases filed by MemoSun, Inc., Kingston Technology Corp., Kingston Technology Co., Inc., Payton Technology Corp. (krj) (Entered: 12/20/2007)
- 12/19/2007 125 Letter from Defendants requesting clarification of briefing schedules.(krj),(ps) Additional attachment(s) added on 12/28/2007 (Plender, Ryan). (Entered: 12/20/2007)
- 12/19/2007 126 BRIEF in Opposition by Plaintiff SanDisk Corp. re: 55 MOTION to Consolidate Cases and Stay filed by MemoSun, Inc., Kingston Technology Corp., Kingston Technology Co., Inc., Payton

- Technology Corp. (Sealed Document) (krj) (Entered: 12/20/2007)
- 12/19/2007 127 AFFIDAVIT of Allen A. Arntsen filed by Plaintiff SanDisk Corp. re: 112 MOTION to Stay filed by PQI Corp., Power Quotient International Co., Ltd., USBest Technology, Inc., Corsair Memory, Inc., PNY Technologies, Inc., Infotech Logistic LLC, A-Data Technology (USA) Co., Ltd., A-Data Technology Co., Ltd. (krj) (Entered: 12/20/2007)
- 12/19/2007 128 AFFIDAVIT of Monica Mucchetti Eno filed by Plaintiff SanDisk Corp. re: 112 MOTION to Stay filed by PQI Corp., Power Quotient International Co., Ltd., USBest Technology, Inc., Corsair Memory, Inc., PNY Technologies, Inc., Infotech Logistic LLC, A-Data Technology (USA) Co., Ltd., A-Data Technology Co., Ltd. (Attachments: # 1 Exhibit 1-3# 2 Exhibit 4-10) (krj) (Entered: 12/20/2007)
- 12/19/2007 129 EXHIBIT A to Affidavit of Greg Rhine by Plaintiff SanDisk Corp.. (Sealed Document) (krj) (Entered: 12/21/2007)
- 12/21/2007 131 MOTION to Admit Frank J. West, Richard D. Kelly, Robert C. Mattson and Takahiro Miura Pro Hac Vice by Defendants Buffalo, Inc., Melco Holdings, Inc., Buffalo Technology (USA) Inc. Motions referred to Magistrate Judge Stephen L Crocker. (krj) (Entered: 12/28/2007)
- 12/26/2007 132 MOTION to Admit Christine Yang Pro Hac Vice by Defendants Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc. Motions referred to Magistrate Judge Stephen L Crocker. (krj) (Entered: 12/28/2007)
- 12/26/2007 133 AFFIDAVIT of Christine Yang re: 132 MOTION to Admit Christine Yang Pro Hac Vice filed by MemoSun, Inc., Kingston Technology Corp., Kingston Technology Co., Inc., Payton Technology Corp. (krj) (Entered: 12/28/2007)
- 12/27/2007 130 ORDER granting request for clarification of briefing schedules on motions to consolidate, to stay, and/or to join; previous briefing schedules stricken; additional motions due 12/31/07; plaintiff response due 1/7/08; replies due 1/11/08. Signed by Magistrate Judge Stephen L Crocker on 12/21/07. (krj) (Entered: 12/27/2007)
- 12/27/2007 -- Set/Reset Deadlines as to 112 MOTION to Stay, 55 MOTION to Consolidate Cases, 57 MOTION to Stay. Brief in Opposition due by 1/7/2008. Brief in Reply due by 1/11/2008. (krj) (Entered: 12/27/2007)
- 12/27/2007 134 BRIEF in Reply in Support re: 55 MOTION to Consolidate Cases filed by MemoSun, Inc., Kingston Technology Corp., Kingston Technology Co., Inc., Payton Technology Corp. (Attachments: # 1 Certificate of Service) (elc) (Entered: 12/31/2007)
- 12/27/2007 135 AFFIDAVIT of Eugenia G. Carter re: 55 MOTION to Consolidate Cases filed by MemoSun, Inc., Kingston Technology Corp., Kingston Technology Co., Inc., Payton Technology Corp. (Attachments: # 1 Exhibit) (elc) (Entered: 12/31/2007)
- 12/28/2007 136 MOTION for Joinder by Defendant LG ELECTRONICS, INC. to Unopposed Motions to Stay the case. Motions referred to Magistrate Judge Stephen L Crocker. (Attachments: # 1 Proposed Order) (elc) (Entered: 12/31/2007)
- 12/28/2007 137 SUMMONS Returned Executed by Plaintiff SanDisk Corp. Acer, Inc. (Taiwan) served on 12/27/2007. (elc) (Entered: 12/31/2007)
- 12/28/2007 138 SUMMONS Returned Executed by Plaintiff SanDisk Corp. USBest Technology, Inc. served on 12/27/2007. (elc) (Entered: 12/31/2007)
- 12/28/2007 139 NOTICE of Appearance by W. David Shenk, Eugenia G. Carter, Anthony J. Sievert for Defendants Apacer Technology Inc., Apacer Memory America, Inc. (elc) (Entered: 12/31/2007)
- 12/28/2007 140 MOTION to join Motions to Consolidate and Stay by Defendants Apacer Technology Inc., Apacer Memory America, Inc.. Motions referred to Magistrate Judge Stephen L Crocker. (Attachments: # 1 Certificate of Service) (elc) (Entered: 12/31/2007)
- 01/02/2008 141 ** TEXT ONLY ORDER ** Order Granting 131 MOTION to Admit Frank J. West, Richard D. Kelly, Robert C. Mattson and Takahiro Miura Pro Hac Vice filed by Buffalo, Inc., Melco Holdings, Inc., Buffalo Technology (USA) Inc. Signed by Magistrate Judge Stephen L Crocker on 1/2/08. (krj) (Entered: 01/02/2008)
- 01/02/2008 142 ** TEXT ONLY ORDER ** Order Granting 132 MOTION to Admit Christine Yang Pro Hac Vice filed by MemoSun, Inc., Kingston Technology Corp., Kingston Technology Co., Inc., Payton Technology Corp. Signed by Magistrate Judge Stephen L Crocker on 1/2/08. (krj) (Entered: 01/02/2008)
- 01/02/2008 143 MOTION for Joinder in unopposed motions to stay by Defendant EDGE Tech Corp. Motions referred to Magistrate Judge Stephen L Crocker. (elc) (Entered: 01/02/2008)
- 01/03/2008 144 NOTICE of Appearance by Michael J. Bettinger, Chien-Wei Chou, Anup Tikku, Jong H. Lee, Jane

- C. Schlicht and Jeffrey S. Sokol for Defendants Transcend Information Inc. (Taiwan), Transcend Information Inc. (California, U.S.A.), Transcend Information Maryland, Inc. (Attachments: # 1 Certificate of Service) (elc) (Entered: 01/04/2008)
- 01/03/2008 145 MOTION to Admit Michael J. Bettinger, Chien-Wei Chou, Anup Tikku and Jong H. Lee Pro Hac Vice by Defendants Transcend Information Inc. (Taiwan), Transcend Information Inc. (California, U.S.A.), Transcend Information Maryland, Inc. Motions referred to Magistrate Judge Stephen L Crocker. (elc) (Entered: 01/04/2008)
- 01/03/2008 146 AFFIDAVIT of Michael J. Bettinger re: 145 MOTION to Admit Michael J. Bettinger, Chien-Wei Chou, Anup Tikku and Jong H. Lee Pro Hac Vice filed by Transcend Information Maryland, Inc., Transcend Information Inc. (Taiwan), Transcend Information Inc. (California, U.S.A.) (elc) (Entered: 01/04/2008)
- 01/03/2008 147 AFFIDAVIT of Chien-Wei Chou re: 145 MOTION to Admit Michael J. Bettinger, Chien-Wei Chou, Anup Tikku and Jong H. Lee Pro Hac Vice filed by Transcend Information Maryland, Inc., Transcend Information Inc. (Taiwan), Transcend Information Inc. (California, U.S.A.) (elc) (Entered: 01/04/2008)
- 01/03/2008 148 AFFIDAVIT of Jong H. Lee re: 145 MOTION to Admit Michael J. Bettinger, Chien-Wei Chou, Anup Tikku and Jong H. Lee Pro Hac Vice filed by Transcend Information Maryland, Inc., Transcend Information Inc. (Taiwan), Transcend Information Inc. (California, U.S.A.) (elc) (Entered: 01/04/2008)
- 01/03/2008 149 AFFIDAVIT of Anup Tikku re: 145 MOTION to Admit Michael J. Bettinger, Chien-Wei Chou, Anup Tikku and Jong H. Lee Pro Hac Vice filed by Transcend Information Maryland, Inc., Transcend Information Inc. (Taiwan), Transcend Information Inc. (California, U.S.A.) (elc) (Entered: 01/04/2008)
- 01/03/2008 150 NOTICE of joinder by Defendants Transcend Information Inc. (Taiwan), Transcend Information Inc. (California, U.S.A.), Transcend Information Maryland, Inc. re 55 MOTION to Consolidate Cases (elc) (Entered: 01/04/2008)
- 01/04/2008 151 SUMMONS Returned Executed by Plaintiff SanDisk Corp., Defendant WellDone Co.. WellDone Co. served on 12/28/2007. (elc) (Entered: 01/04/2008)
- 01/07/2008 153 NOTICE of Voluntary Dismissal without prejudice of Acer, Inc. by Plaintiff SanDisk Corp. (Attachments: # 1 Certificate of Service) (krj) (Entered: 01/08/2008)
- 01/07/2008 154 MOTION to Admit Monica Mucchetti Eno and Matthew J. Bye Pro Hac Vice by Plaintiff SanDisk Corp. Motions referred to Magistrate Judge Stephen L Crocker. (Attachments: # 1 Certificate of Service) (krj) (Entered: 01/08/2008)
- 01/07/2008 155 AFFIDAVIT of Monica Mucchetti Eno re: 154 MOTION to Admit Monica Mucchetti Eno and Matthew J. Bye Pro Hac Vice filed by SanDisk Corp. (krj) (Entered: 01/08/2008)
- 01/07/2008 156 AFFIDAVIT of Matthew J. Bye re: 154 MOTION to Admit Monica Mucchetti Eno and Matthew J. Bye Pro Hac Vice filed by SanDisk Corp. (krj) (Entered: 01/08/2008)
- 01/08/2008 152 ** TEXT ONLY ORDER ** Order Granting 145 MOTION to Admit Michael J. Bettinger, Chien-Wei Chou, Anup Tikku and Jong H. Lee Pro Hac Vice. Michael J. Bettinger for Transcend Information Inc. (Taiwan) and Transcend Information Inc. (California, U.S.A.), Chien-Wei Chou for Transcend Information Inc. (Taiwan), Transcend Information Inc. (California, U.S.A.), Transcend Information Maryland, Inc., Anup Tikku for Transcend Information Inc. (Taiwan), Transcend Information Inc. (California, U.S.A.), Transcend Information Maryland, Inc., Jong H. Lee for Transcend Information Inc. (Taiwan), Transcend Information Inc. (California, U.S.A.), Transcend Information Maryland, Inc. admitted pro hac vice. Signed by Magistrate Judge Stephen L Crocker on 1/7/08. (elc) (Entered: 01/08/2008)
- 01/08/2008 157 ** TEXT ONLY ORDER ** Order Granting 154 MOTION to Admit Monica Mucchetti Eno and Matthew J. Bye Pro Hac Vice filed by SanDisk Corp.. Monica Mucchetti Eno for SanDisk Corp., Matthew J. Bye for SanDisk Corp. admitted pro hac vice. Signed by Magistrate Judge Stephen L Crocker on 1/8/08. (krj) (Entered: 01/08/2008)
- 01/09/2008 158 STIPULATION for Substitution of Counsel by Defendant Infotech Logistic LLC. (Attachments: # 1 Certificate of Service) (elc) (Entered: 01/10/2008)
- 01/10/2008 159 NOTICE of Voluntary Dismissal of defendant PNY Technologies, Inc. with prejudice and without costs. (elc) (Entered: 01/10/2008)
- 01/11/2008 160 ** TEXT ONLY ORDER ** Order Granting 158 STIPULATION for Substitution of Counsel. Heller Ehrman, LLP is substituted as counsel for defendant Infotech Logistics, LLC in place of Godfrey, Braun & Frazier LLP. Signed by Magistrate Judge Stephen L Crocker on 1/10/08. (elc) (Entered: 01/11/2008)
- 01/11/2008 161 MOTION to Voluntarily Dismiss defendants Add-On Computer Peripherals, Inc. and Add-On

- Computer Peripherals, LLC by Plaintiff SanDisk Corp. (Attachments: # 1 Certificate of Service) (elc) (Entered: 01/14/2008)
- 01/14/2008 162 MOTION to Voluntarily Dismiss defendant Infotech Logistic, LLC by Plaintiff SanDisk Corp. (elc) (Entered: 01/14/2008)
- 01/14/2008 163 MOTION to Voluntarily Dismiss Defendant TSR Silicon Resources Inc. by Plaintiff SanDisk Corp.. (elc) (Entered: 01/14/2008)
- 01/14/2008 164 ORDER STAYING CASE for all parties. Signed by Magistrate Judge Stephen L Crocker on 1/14/08. (elc) (Entered: 01/15/2008)
- 01/15/2008 165 STIPULATION AND ORDER of dismissal with prejudice and without costs re: Defendant PNY Technologies, Inc. Signed by Judge Barbara B Crabb on 1/11/08. (krj) (Entered: 01/15/2008)
- 01/15/2008 166 ORDER approving notice of dismissal of Acer, Inc. without prejudice. Signed by Judge Barbara B Crabb on 1/11/08. (krj) (Entered: 01/15/2008)
- 01/17/2008 167 MOTION to Voluntarily Dismiss by Plaintiff SanDisk Corp., Defendant Interactive Media Corp. (Attachments: #(1) Certificate of Service) (krj) (Entered: 01/17/2008)
- 01/17/2008 168 CERTIFICATE OF SERVICE re: 163 motion to voluntarily dismiss TSR Silicon Resources Inc. by Plaintiff SanDisk Corp. (krj) (Entered: 01/17/2008)
- 01/17/2008 169 NOTICE of Voluntary Dismissal without prejudice of defendant Synergistic Sales, Inc. by Plaintiff SanDisk Corp. (elc) (Entered: 01/18/2008)
- 01/18/2008 170 MOTION to Voluntarily Dismiss Defendant Edge Tech Corporation with prejudice and without costs by Plaintiff SanDisk Corp. (Attachments: # 1 Certificate of Service) (elc) (Entered: 01/18/2008)
- 01/18/2008 171 NOTICE of Voluntary Dismissal by Plaintiff SanDisk Corp., Defendant Welldone Co. (Attachments: # 1 Certificate of Service) (krj) (Entered: 01/18/2008)
- 01/22/2008 172 ANSWER to Complaint with Jury Demand by Defendants Dane-Elec Memory, S.A., Dane-Elec Corp. USA. (Attachments: # 1 Exhibit Signature Page) (Palmersheim, Kevin) ATTORNEY CONTACTED; ASKED TO FILE CORRECTED DOCUMENT TO INCLUDE S/FULL NAME AND MATCH FILING ATTORNEY NAME. Modified on 1/23/2008 (Jensen, Lori). (Entered: 01/22/2008)
- 01/22/2008 173 Corporate Disclosure Statement by Defendants Dane-Elec Memory, S.A., Dane-Elec Corp. USA. (Palmersheim, Kevin) ATTORNEY CONTACTED; ASKED TO FILE CORRECTED DOCUMENT TO INCLUDE S/FULL NAME AND MATCH FILING ATTORNEY NAME. Modified on 1/23/2008 (Jensen, Lori). (Entered: 01/22/2008)
- 01/23/2008 174 ORDER granting 162 Motion to Voluntarily Dismiss defendant Infotech Logistic LLC w/prejudice and w/o costs. Signed by Judge Barbara B Crabb on 1/22/2008. (ljl) Additional attachment(s) added on 1/23/2008 (Jensen, Lori). (Entered: 01/23/2008)
- 01/23/2008 175 ORDER granting 163 Motion to Voluntarily Dismiss defendant TSR Silicon Resources, Inc. Signed by Judge Barbara B Crabb on 1/22/2008. (ljl) Additional attachment(s) added on 1/23/2008 (Jensen, Lori). (Entered: 01/23/2008)
- 01/23/2008 176 ANSWER to Complaint with Jury Demand by Defendant Dane-Elec Corp. USA. (Attachments: # 1 Replacement page 61, signature page) (Palmersheim, Kevin) (Entered: 01/23/2008)
- 01/23/2008 177 Corporate Disclosure Statement by Defendant Dane-Elec Corp. USA. (Attachments: # 1 Replacement page 3, signature page.) (Palmersheim, Kevin) (Entered: 01/23/2008)
- 01/23/2008 178 MOTION to Dismiss for Lack of Jurisdiction by Defendant Dane-Elec Memory, S.A.. Brief in Opposition due by 2/13/2008. Brief in Reply due by 2/25/2008. (Attachments: # 1 Motion replacement page 3, signature page# 2 Supplement Supporting Memorandum to Motion to Dismiss# 3 Memorandum replacement page 7, signature page# 4 Supplement Declaration of David Haccoun in Support of Motion to Dismiss) (Palmersheim, Kevin) ATTORNEY CONTACTED; ASKED TO RE-FILE DOCUMENTS USING SEPARATE EVENTS. Modified on 1/24/2008 (Jensen, Lori). (Entered: 01/23/2008)
- 01/24/2008 179 AFFIDAVIT of Service for Summons, Civil Cover Sheet, Complaint, Corporate Disclosure Statement served on A-Data Technology Co., Ltd. on November 16, 2007, filed by Plaintiff SanDisk Corp.. (Arntsen, Allen) (Entered: 01/24/2008)
- 01/28/2008 180 ORDER FOR DISMISSAL on notice of pltf. re: deft. Synergistic Sales, Inc. Signed by Judge Barbara B Crabb on 1/21/2008. (ljl) (Entered: 01/29/2008)
- 01/28/2008 181 ORDER FOR DISMISSAL of deft. Interactive Media Corporation per notice of voluntary dismissal by pltf. Signed by Judge Barbara B Crabb on 1/23/2008. (ljl) (Entered: 01/29/2008)
- 01/28/2008 182 ORDER FOR DISMISSAL of deft. Edge Tech Corporation per notice of voluntary dismissal by pltf.

- Signed by Judge Barbara B Crabb on 1/24/2008. (ljl) (Entered: 01/29/2008)
- 01/28/2008 183 ORDER FOR DISMISSAL of deflt. Welldone Co. per notice of voluntary dismissal by pltf. Signed by Judge Barbara B Crabb on 1/24/2008. (ljl) (Entered: 01/29/2008)
- 01/28/2008 184 ORDER FOR DISMISSAL of defts. Add-On Computer Peripherals, Inc., Add-On Computer Peripherals, LLC, per notice of voluntary dismissal by pltf. Signed by Judge Barbara B Crabb on 1/28/2008. (ljl) (Entered: 01/29/2008)
- 01/29/2008 185 ORDER granting defts. motions to consolidate this with stayed Case No. 07-cv-605; counsel defendants to provide status of International Trade Commission proceedings by 3/1/08 and every other month thereafter until commission issues a determination. Signed by Judge Barbara B Crabb on 1/28/08. (elc) Modified docket text on 2/4/2008 (Jensen, Lori). (Entered: 02/01/2008)
- 02/01/2008 -- Create association to 3:07-cv-00605-bbc. All further docketing directed to be in lead case #07-cv-605-bbc. (elc) Modified docket text on 2/4/2008 (Jensen, Lori). (Entered: 02/01/2008)
- 02/07/2008 186 AFFIDAVIT of Service by Plaintiffs SanDisk Corp., SanDisk Corp., Defendant A-Data Technology Co., Ltd.. A-Data Technology Co., Ltd. served on 11/16/2007, answer due 12/6/2007. Associated Cases: 3:07-cv-00607-bbc, 3:07-cv-00605-bbc (Arntsen, Allen) DOCUMENT DELETED AT ATTORNEY REQUEST; FILING MADE IN ERROR. Modified on 2/7/2008 (Jensen, Lori). (Entered: 02/07/2008)
- 02/28/2008 187 STATUS REPORT by Defendants Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc., Phison Electronics Corp.. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Carter, Eugenia) (Entered: 02/28/2008)
- 02/29/2008 188 STATUS REPORT Regarding the Related International Trade Commission Proceedings by Defendants USBest Technology, Inc., Power Quotient International Co., Ltd., PQI Corp., A-Data Technology Co., Ltd., A-Data Technology (USA) Co., Ltd., USBest Technology, Inc., Chipsbrand Microelectronics (HK) Co., Ltd., Chipsbank Technology (Shenzhen) Co., Ltd., Chipsbank Microelectronics Co., Ltd., Power Quotient International Co., Ltd., PQI Corp., A-Data Technology Co., Ltd., A-Data Technology (USA) Co., Ltd., Corsair Memory, Inc.. (Attachments: # 1 Certificate of Service) Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Anstaett, David) (Entered: 02/29/2008)
- 04/30/2008 189 STATUS REPORT by Defendants Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc., Phison Electronics Corp., Counter Claimant Apacer Memory America, Inc.. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Carter, Eugenia) (Entered: 04/30/2008)
- 05/01/2008 190 STATUS REPORT Regarding Related International Trade Commission Proceedings by Defendants USBest Technology, Inc., Power Quotient International Co., Ltd., PQI Corp., A-Data Technology Co., Ltd., A-Data Technology (USA) Co., Ltd., USBest Technology, Inc., Chipsbrand Microelectronics (HK) Co., Ltd., Chipsbank Technology (Shenzhen) Co., Ltd., Chipsbank Microelectronics Co., Ltd., Power Quotient International Co., Ltd., PQI Corp., A-Data Technology Co., Ltd., A-Data Technology (USA) Co., Ltd., Corsair Memory, Inc.. (Attachments: # 1 Certificate of Service) Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Anstaett, David) (Entered: 05/01/2008)
- 07/01/2008 191 STATUS REPORT - Third - Regarding Related International Trade Commission Proceedings by Defendants USBest Technology, Inc., Power Quotient International Co., Ltd., PQI Corp., A-Data Technology Co., Ltd., A-Data Technology (USA) Co., Ltd., USBest Technology, Inc., Chipsbrand Microelectronics (HK) Co., Ltd., Chipsbank Technology (Shenzhen) Co., Ltd., Chipsbank Microelectronics Co., Ltd., Power Quotient International Co., Ltd., PQI Corp., A-Data Technology Co., Ltd., A-Data Technology (USA) Co., Ltd., Corsair Memory, Inc.. (Attachments: # 1 Certificate of Service) Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Anstaett, David) (Entered: 07/01/2008)
- 07/01/2008 192 STATUS REPORT by Defendants Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc., Phison Electronics Corp.. (Attachments: # 1 Certificate of Service 5 pages) Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Carter, Eugenia) (Entered: 07/01/2008)
- 07/31/2008 193 MOTION to Lift Stay for the Limited Purpose of Seeking Issuance of Letters Rogatory to Effect Service of Process on Defendants in Taiwan by Plaintiff SanDisk Corp., Counter Defendant SanDisk Corp. Motion referred to Magistrate Judge Stephen L Crocker. Response due 8/7/2008. (Attachments: # 1 Certificate of Service) Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Arntsen, Allen) Modified docket text on 8/1/2008 (Wiseman, Andrew). (Entered: 07/31/2008)
- 07/31/2008 194 MOTION for Issuance of Letters Rogatory to Effect Service of Process on Defendants in Taiwan re (217 in 3:07-cv-00605-bbc) MOTION to Lift Stay for the Limited Purpose of Seeking Issuance

- of Letters Rogatory to Effect Service of Process on Defendants in Taiwan by Plaintiff SanDisk Corp., Counter Defendant SanDisk Corp. Motion referred to Magistrate Judge Stephen L Crocker. Response due 8/7/2008. (Attachments: # 1 Exhibit A-K) Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Arntsen, Allen) Modified docket text on 8/1/2008 (Wiseman, Andrew). (Entered: 07/31/2008)
- 07/31/2008 195 AFFIDAVIT of Chen Szu-Chien filed by Plaintiffs SanDisk Corp., SanDisk Corp., Counter Defendants SanDisk Corp., SanDisk Corp. re: (194 in 3:07-cv-00607-bbc) MOTION filed by SanDisk Corp. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Arntsen, Allen) (Entered: 07/31/2008)
- 07/31/2008 196 AFFIDAVIT of Monica Mucchetti Eno filed by Plaintiffs SanDisk Corp., SanDisk Corp., Counter Defendants SanDisk Corp., SanDisk Corp. re: (194 in 3:07-cv-00607-bbc) MOTION filed by SanDisk Corp. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Arntsen, Allen) (Entered: 07/31/2008)
- 07/31/2008 197 Exhibit to (196 in 3:07-cv-00607-bbc) Affidavit, filed by SanDisk Corp. A-K. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Arntsen, Allen) (Entered: 07/31/2008)
- 07/31/2008 198 Exhibit to (196 in 3:07-cv-00607-bbc) Affidavit, filed by SanDisk Corp. L-N. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Arntsen, Allen) (Entered: 07/31/2008)
- 07/31/2008 199 Exhibit to (196 in 3:07-cv-00607-bbc) Affidavit, filed by SanDisk Corp. O-Q. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Arntsen, Allen) (Entered: 07/31/2008)
- 07/31/2008 200 Exhibit to (196 in 3:07-cv-00607-bbc) Affidavit, filed by SanDisk Corp. R-V. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Arntsen, Allen) (Entered: 07/31/2008)
- 08/06/2008 201 Stipulated MOTION for Extension of Time by Defendants A-Data Technology Co., Ltd., A-Data Technology (USA) Co., Ltd. Motion referred to Magistrate Judge Stephen L Crocker. Response due by 8/13/2008. (Attachments: # 1 Certificate of Service) Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Anstaett, David) (Entered: 08/06/2008)
- 08/06/2008 202 STIPULATION for Substitution of Counsel by Defendants Apacer Technology Inc., Apacer Memory America, Inc., Counter Claimants Apacer Technology Inc., Apacer Memory America, Inc. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Schlicht, Jane) (Entered: 08/06/2008)
- 08/06/2008 203 NOTICE of Appearance by Jane C. Schlicht for Defendants Apacer Technology Inc., Apacer Memory America, Inc., Counter Claimants Apacer Technology Inc., Apacer Memory America, Inc. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Schlicht, Jane) (Entered: 08/06/2008)
- 08/06/2008 204 CERTIFICATE OF SERVICE by Defendants Apacer Technology Inc., Apacer Memory America, Inc., Counter Claimants Apacer Technology Inc., Apacer Memory America, Inc. re (202 in 3:07-cv-00607-bbc) STIPULATION for, (203 in 3:07-cv-00607-bbc) Notice of Appearance, Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Schlicht, Jane) (Entered: 08/06/2008)
- 08/07/2008 205 ** TEXT ONLY ORDER ** Order Granting (202 in 3:07-cv-00607-bbc, 226 in 3:07-cv-00605-bbc) STIPULATION for substitution of counsel. Atty. Jane Schlicht and firm of Cook & Franke substituted as counsel for defendants Apacer Technology, Inc. and Apacer Memory American, Inc. in place of Eugenia Carter and firm of Whyte Hirschboeck. Signed by Magistrate Judge Stephen L Crocker on 8/7/2008. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Ilj) (Entered: 08/07/2008)
- 08/12/2008 206 ** TEXT ONLY ORDER ** Order Granting (201 in 3:07-cv-00607-bbc, 225 in 3:07-cv-00605-bbc) MOTION for Extension of Time filed by A-Data Technology (USA) Co., Ltd. Brief in Opposition due by 8/19/2008. Signed by Magistrate Judge Stephen L Crocker on 8/12/08. (elc) (Entered: 08/12/2008)
- 08/18/2008 207 BRIEF in Opposition by Defendants Behavior Tech Computer Corp., Behavior Tech Computer USA Corp., Behavior Tech Computer Corp., Behavior Tech Computer USA Corp. re: (217 in 3:07-cv-00605-bbc) MOTION to Lift Stay for the Limited Purpose of Seeking Issuance of Letters Rogatory to Effect Service of Process on Defendants in Taiwan filed by SanDisk Corp. (Attachments: # 1 Certificate of Service 5 pages) Associated Cases: 3:07-cv-00607-bbc, 3:07-cv-00605-bbc (Carter, Eugenia) (Entered: 08/18/2008)
- 08/18/2008 208 AFFIDAVIT of Jeremy Duggan filed by Defendants Behavior Tech Computer Corp., Behavior Tech Computer USA Corp., Behavior Tech Computer Corp., Behavior Tech Computer USA Corp. re: (217 in 3:07-cv-00605-bbc) MOTION to Lift Stay for the Limited Purpose of Seeking Issuance of Letters Rogatory to Effect Service of Process on Defendants in Taiwan filed by SanDisk Corp. (Attachments: # 1 Exhibit 1) Associated Cases: 3:07-cv-00607-bbc, 3:07-cv-00605-bbc (Carter, Eugenia) (Entered: 08/18/2008)
- 08/19/2008 209 BRIEF in Opposition by Defendants A-Data Technology Co., Ltd., A-Data Technology (USA) Co.,

- Ltd., A-Data Technology Co., Ltd., A-Data Technology (USA) Co., Ltd. re: (193 in 3:07-cv-00607-bbc, 217 in 3:07-cv-00605-bbc) MOTION to Lift Stay filed by SanDisk Corp. (Attachments: # 1 Certificate of Service) Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Anstaett, David) (Entered: 08/19/2008)
- 08/19/2008 210 BRIEF in Opposition by Defendants Transcend Information Inc. (Taiwan), Apacer Technology Inc., Silicon Motion, Inc. (Taiwan), Counter Claimants Transcend Information Inc. (Taiwan), Apacer Technology Inc. re: (193 in 3:07-cv-00607-bbc, 217 in 3:07-cv-00605-bbc) MOTION to Lift Stay filed by SanDisk Corp., (218 in 3:07-cv-00605-bbc, 194 in 3:07-cv-00607-bbc) MOTION SanDisk's Motion for Issuance of Letters Rogatory to Effect Service of Process on Defendants in Taiwan re (217 in 3:07-cv-00605-bbc) MOTION to Lift Stay for the Limited Purpose of Seeking Issuance of Letters Rogatory to Effect Service o MOTION SanDisk's Motion for Issuance of Letters Rogatory to Effect Service of Process on Defendants in Taiwan re (217 in 3:07-cv-00605-bbc) MOTION to Lift Stay for the Limited Purpose of Seeking Issuance of Letters Rogatory to Effect Service of Process on Defendants in Taiwan re (217 in 3:07-cv-00605-bbc) MOTION to Lift Stay for the Limited Purpose of Seeking Issuance of Letters Rogatory to Effect Service of Process on Defendants in Taiwan re (217 in 3:07-cv-00605-bbc) MOTION to Lift Stay for the Limited Purpose of Seeking Issuance of Letters Rogatory to Effect Service o filed by SanDisk Corp. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Schlicht, Jane) (Entered: 08/19/2008)
- 08/19/2008 211 CERTIFICATE OF SERVICE by Defendants Transcend Information Inc. (Taiwan), Apacer Technology Inc., Silicon Motion, Inc. (Taiwan), Counter Claimants Transcend Information Inc. (Taiwan), Apacer Technology Inc. re: (210 in 3:07-cv-00607-bbc) Brief in Opposition,,,, Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Schlicht, Jane) (Entered: 08/19/2008)
- 08/19/2008 212 BRIEF in Opposition by Defendants Transcend Information Inc. (Taiwan), Apacer Technology Inc., Silicon Motion, Inc. (Taiwan) re: (218 in 3:07-cv-00605-bbc) MOTION SanDisk's Motion for Issuance of Letters Rogatory to Effect Service of Process on Defendants in Taiwan re (217 in 3:07-cv-00605-bbc) MOTION to Lift Stay for the Limited Purpose of Seeking Issuance of Letters Rogatory to Effect Service o MOTION SanDisk's Motion for Issuance of Letters Rogatory to Effect Service of Process on Defendants in Taiwan re (217 in 3:07-cv-00605-bbc) MOTION to Lift Stay for the Limited Purpose of Seeking Issuance of Letters Rogatory to Effect Service o MOTION SanDisk's Motion for Issuance of Letters Rogatory to Effect Service of Process on Defendants in Taiwan re (217 in 3:07-cv-00605-bbc) MOTION to Lift Stay for the Limited Purpose of Seeking Issuance of Letters Rogatory to Effect Service of Process on Defendants in Taiwan re (217 in 3:07-cv-00605-bbc) MOTION to Lift Stay for the Limited Purpose of Seeking Issuance of Letters Rogatory to Effect Service o filed by SanDisk Corp. Associated Cases: 3:07-cv-00607-bbc, 3:07-cv-00605-bbc (Schlicht, Jane) (Entered: 08/19/2008)
- 08/19/2008 213 CERTIFICATE OF SERVICE by Defendants Transcend Information Inc. (Taiwan), Apacer Technology Inc., Silicon Motion, Inc. (Taiwan) re: (236 in 3:07-cv-00605-bbc) Brief in Opposition,,,, Associated Cases: 3:07-cv-00607-bbc, 3:07-cv-00605-bbc (Schlicht, Jane) (Entered: 08/19/2008)
- 08/22/2008 214 MOTION for Leave to File Reply Brief in Support of Motion to Lift Stay for the Limited Purpose of Seeking Issuance of Letters Rogatory by Plaintiffs SanDisk Corp., SanDisk Corp.. Motions referred to Magistrate Judge Stephen L Crocker. (Attachments: # 1 Reply In Support of Sandisk's Motion to Lift Stay for the Limited Purpose of Seeking Issuance of Letters Rogatory# 2 Certificate of Service) Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Simmons, Jeffrey) (Entered: 08/22/2008)
- 08/28/2008 215 STATUS REPORT by Defendants Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc., Phison Electronics Corp., Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc., Phison Electronics Corp.. (Attachments: # 1 Certificate of Service 4 Pages) Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Carter, Eugenia) (Entered: 08/28/2008)
- 09/02/2008 216 STATUS REPORT - Fourth - Regarding Related International Trade Commission Proceedings by Defendants A-Data Technology Co., Ltd., A-Data Technology (USA) Co., Ltd., A-Data Technology Co., Ltd., A-Data Technology (USA) Co., Ltd.. (Attachments: # 1 Certificate of Service) Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Anstaett, David) (Entered: 09/02/2008)
- 09/04/2008 217 STIPULATION of Dismissal With Prejudice and Without Costs of Complaint Against Corsair Memory, Inc.. (Attachments: # 1 Certificate of Service) Associated Cases: 3:07-cv-00607-bbc, 3:07-cv-00605-bbc (Arntsen, Allen) (Entered: 09/04/2008)
- 09/05/2008 218 ** TEXT ONLY ORDER ** Order Approving 217 STIPULATION of Dismissal With Prejudice and Without Costs of Complaint Against Corsair Memory, Inc.. Signed by Judge Barbara B Crabb on 9/5/08. (elc) (Entered: 09/05/2008)

- 09/17/2008 219 ORDER denying 193 Motion to Lift Stay ; denying 194 Motion for issuance of letters rogatory ; granting 214 Motion for Leave to File reply brief. Signed by Judge Barbara B Crabb on 9/17/08. (krj) (Entered: 09/17/2008)
- 10/31/2008 220 STATUS REPORT by Defendants Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc., Phison Electronics Corp., Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc., Phison Electronics Corp.. (Attachments: # 1 Certificate of Service) Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Carter, Eugenia) (Entered: 10/31/2008)
- 12/31/2008 221 STATUS REPORT by Defendants Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc., Phison Electronics Corp., Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc., Phison Electronics Corp.. Associated Cases: 3:07-cv-00607-bbc, 3:07-cv-00605-bbc (Carter, Eugenia) (Entered: 12/31/2008)
- 01/23/2009 222 MOTION to Admit Maureen F. Browne Pro Hac Vice by Defendants. Motions referred to Magistrate Judge Stephen L Crocker. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Cetrangolo, Catherine) (Entered: 01/23/2009)
- 01/23/2009 223 DECLARATION of Maureen F. Browne filed by Defendants. re: (222 in 3:07-cv-00607-bbc) MOTION to Admit Maureen F. Browne Pro Hac Vice filed by A-Data Technology (USA) Co., Ltd., PQI Corp., Power Quotient International Co., Ltd., USBest Technology, Inc., A-Data Technology Co., Ltd. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Cetrangolo, Catherine) (Entered: 01/23/2009)
- 01/27/2009 224 ** TEXT ONLY ORDER ** ORDER granting 222 Motion to Admit Maureen Browne Pro Hac Vice. Signed by Magistrate Judge Stephen L Crocker on 1/23/09. (elc) (Entered: 01/27/2009)
- 02/27/2009 225 STATUS REPORT by Defendants Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc., Phison Electronics Corp., Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc., Phison Electronics Corp.. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Carter, Eugenia) (Entered: 02/27/2009)
- 03/04/2009 226 MOTION to Withdraw as Attorney David J. Harth and David L. Anstaett by Defendants USBest Technology, Inc., USBest Technology, Inc..Response due by 3/11/2009. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Anstaett, David) (Entered: 03/04/2009)
- 03/06/2009 227 ** TEXT ONLY ORDER ** ORDER granting (250) Motion to Withdraw as Attorney. Attorney David L. Anstaett and David J. Harth terminated in case 3:07-cv-00605-bbc; granting (226) Motion to Withdraw as Attorney. Attorney David L. Anstaett and David J. Harth terminated in case 3:07-cv-00607-bbc Signed by Magistrate Judge Stephen L Crocker on 3/6/09. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (krj) (Entered: 03/06/2009)
- 03/18/2009 228 MOTION to Withdraw as Attorney Brian Kwok by Defendants Verbatim Corp., Verbatim Corp. (Kwok, Brian) Modified from a notice to a motion on 3/19/2009 (jef). (Entered: 03/18/2009)
- 03/19/2009 229 CERTIFICATE OF SERVICE by Defendants Verbatim Corp., Verbatim Corp. (Kwok, Brian) (Entered: 03/19/2009)
- 03/24/2009 230 ** TEXT ONLY ORDER ** ORDER granting (252) Motion to Withdraw as Attorney. Attorney Brian C. Kwok terminated in case 3:07-cv-00605-bbc; granting (228) Motion to Withdraw as Attorney. Attorney Brian C. Kwok terminated in case 3:07-cv-00607-bbc. Signed by Magistrate Judge Stephen L Crocker on 3/24/09. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (krj) (Entered: 03/24/2009)
- 04/23/2009 231 STATUS REPORT by Defendants Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc., Phison Electronics Corp., Kingston Technology Co., Inc., Kingston Technology Corp., Payton Technology Corp., MemoSun, Inc., Phison Electronics Corp.. (Attachments: # 1 Notice of Final Issuance) Associated Cases: 3:07-cv-00607-bbc, 3:07-cv-00605-bbc (Carter, Eugenia) (Entered: 04/23/2009)
- 07/01/2009 232 STATUS REPORT by Plaintiffs SanDisk Corp., SanDisk Corp.. Associated Cases: 3:07-cv-00605-bbc, 3:07-cv-00607-bbc (Carter, Eugenia) (Entered: 07/01/2009)

Copyright © 2009 LexisNexis CourtLink, Inc. All rights reserved.
*** THIS DATA IS FOR INFORMATIONAL PURPOSES ONLY ***

Search: (US6763424)/PN/XPN

1 / 1

Patent Number: US2002099904 A1 20020725

Date	Action Taken	
Family member: KR20030070119 A		
No legal status information available for this member.		
Family member: KR20090006217 A		
No legal status information available for this member.		
Family member: JP2008226254 A		
Family member: JP2008004117 A		
Family member: ES2262782 T3		
No legal status information available for this member.		
Family member: CN1924831 A		
Status: Alive		
20090506	CN/C14-A [POS; PIF]	GRANTED
20070502	CN/C10-A [EXM]	REQUEST OF EXAMINATION AS TO SUBSTANCE
20070307	CN-A [POS; EXM]	Unexamined application for a patent for inv. CN1924831 A 20070307 [CN1924831]
20070307	CN/C06-A [POS; PIF]	PUBLICATION
20020107	CN-API [POS; EXM]	FILING DETAILS CN200610142359 20020107 [2006CN-0142359]
Family member: CN1924830 A		
Status: Alive		
20090506	CN/C14-A [POS; PIF]	GRANTED
20070502	CN/C10-A [EXM]	REQUEST OF EXAMINATION AS TO SUBSTANCE
20070307	CN-A [POS; EXM]	Unexamined application for a patent for inv. CN1924830 A 20070307 [CN1924830]
20070307	CN/C06-A [POS; PIF]	PUBLICATION
20020107	CN-API [POS; EXM]	FILING DETAILS CN200610142358 20020107 [2006CN-0142358]
Family member: AT327556 T		
Status: Dead		
20061115	AT/RER-T [NEG; NIF]	CEASED AS TO PARAGRAPH 5 LIT. 3 LAW INTRODUCING PATENT TREATIES ERLOSCHEN GEM. PAR. 5 ABS. 3 PATVEG.
20060615	AT-T [POS; PIF]	EP Patent valid in AT AT327556 T 20060615 [ATE327556]
20020107	AT-API [POS; EXM]	FILING DETAILS AT02703078T 2002010 [2002AT-0703078]
Family member: DE60211653 D1		
Status: Alive		
20080424	DE/8363-A [NEG; OPP]	OPPOSITION AGAINST THE PATENT EINSPRUCH GEGEN DAS PATENT ERHOBEN
20070621	DE/8327-A [NMC]	CHANGE IN THE PERSON/NAME/ADDRESS OF THE PATENT OWNER AENDERUNG IN PERSON, NAMEN ODER WOHNORT DES PATENTINHABERS OWNER: SANDISK CORP., MILPITAS, CALIF., US
20070412	DE-T2 [POS; EXM]	Trans. of EP patent DE60211653 T2 20070412 [DE60211653]
20060629	DE-D1 [POS; EXM]	Granted EP number in Bulletin DE60211653 D1 20060629 [DE60211653]
20020107	DE-API [POS; EXM]	FILING DETAILS DE60211653T 20020107 [2002DE-6011653]

Family member: EP1653323 A2		
Status: Alive		
20060503	EP/17P-A [POS; EXM]	REQUEST FOR EXAMINATION FILED PRUEFUNGSANTRAG GESTELLT EFFECTIVE DATE: 20060117
20060503	EP/AX-A [POS; ADM]	EXTENSION OF THE EUROPEAN PATENT TO ERSTRECKUNG DES EUROPAEISCHEN PATENTS AUF AL LT LV MK RO SI
20060503	EP-A2 [POS; EXM]	Application published without search report EP1653323 A2 20060503 [EP1653323]
20060503	EP/AK-A [POS; ADM]	DESIGNATED CONTRACTING STATES: BENANNTE VERTRAGSSTAATEN AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
20060503	EP/AC-A [ADM]	DIVISIONAL APPLICATION (ART. 76) OF: TEILANMELDUNG (ART. 76) AUS: Corresponding Pat: EP 1352394 0 [EP1352394]
20020107	EP-API [POS; EXM]	FILING DETAILS EP06075106 20020107 [2006EP-0075106]
Family member: EP1645964 A2		
Status: Alive		
20060412	EP/17P-A [POS; EXM]	REQUEST FOR EXAMINATION FILED PRUEFUNGSANTRAG GESTELLT EFFECTIVE DATE: 20051220
20060412	EP/AX-A [POS; ADM]	EXTENSION OF THE EUROPEAN PATENT TO ERSTRECKUNG DES EUROPAEISCHEN PATENTS AUF AL LT LV MK RO SI
20060412	EP-A2 [POS; EXM]	Application published without search report EP1645964 A2 20060412 [EP1645964]
20060412	EP/AK-A [POS; ADM]	DESIGNATED CONTRACTING STATES: BENANNTE VERTRAGSSTAATEN AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
20060412	EP/AC-A [ADM]	DIVISIONAL APPLICATION (ART. 76) OF: TEILANMELDUNG (ART. 76) AUS: Corresponding Pat: EP 1352394 0 [EP1352394]
20020107	EP-API [POS; EXM]	FILING DETAILS EP05077929 20020107 [2005EP-0077929]
Family member: TW221217 B		
No legal status information available for this member.		
Family member: CN1514971 A		
Status: Alive		
20061213	CN-C [POS; PIF]	Granted patent for invention CN1290021 C 20061213 [CN1290021C]
20061213	CN/C14-A [POS; PIF]	GRANTED
20040929	CN/C10-A [EXM]	REQUEST OF EXAMINATION AS TO SUBSTANCE
20040721	CN-A [POS; EXM]	Unexamined application for a patent for inv. CN1514971 A 20040721 [CN1514971]
20040721	CN/C06-A [POS; PIF]	PUBLICATION
20020107	CN-API [POS; EXM]	FILING DETAILS CN02803882 20020107 [2002CN-0803882]
Family member: JP2004533029 T		
Family member: US2004210708 A1		
No legal status information available for this member.		
Family member: EP1352394 A2		
Status: Alive		
20060503	EP/17P-A [POS; EXM]	REQUEST FOR EXAMINATION FILED

		PRUEFUNGSANTRAG GESTELLT EFFECTIVE DATE: 20060117
20060503	EP/AX-A [POS; ADM]	EXTENSION OF THE EUROPEAN PATENT TO ERSTRECKUNG DES EUROPAEISCHEN PATENTS AUF AL LT LV MK RO SI
20060503	EP-A2 [POS; EXM]	Application published without search report EP1653323 A2 20060503 [EP1653323]
20060503	EP/AK-A [POS; ADM]	DESIGNATED CONTRACTING STATES: BENANNTTE VERTRAGSSTAATEN AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
20060503	EP/AC-A [ADM]	DIVISIONAL APPLICATION (ART. 76) OF: TEILANMELDUNG (ART. 76) AUS: Corresponding Pat: EP 1352394 0 [EP1352394]
20020107	EP-API [POS; EXM]	FILING DETAILS EP06075106 20020107 [2006EP-0075106]
20060412	EP/17P-A [POS; EXM]	REQUEST FOR EXAMINATION FILED PRUEFUNGSANTRAG GESTELLT EFFECTIVE DATE: 20051220
20060412	EP/AX-A [POS; ADM]	EXTENSION OF THE EUROPEAN PATENT TO ERSTRECKUNG DES EUROPAEISCHEN PATENTS AUF AL LT LV MK RO SI
20060412	EP-A2 [POS; EXM]	Application published without search report EP1645964 A2 20060412 [EP1645964]
20060412	EP/AK-A [POS; ADM]	DESIGNATED CONTRACTING STATES: BENANNTTE VERTRAGSSTAATEN AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
20060412	EP/AC-A [ADM]	DIVISIONAL APPLICATION (ART. 76) OF: TEILANMELDUNG (ART. 76) AUS: Corresponding Pat: EP 1352394 0 [EP1352394]
20020107	EP-API [POS; EXM]	FILING DETAILS EP05077929 20020107 [2005EP-0077929]
20090630	EP/PGFP-A [POS; PIF]	POSTGRANT: ANNUAL FEES PAID TO NATIONAL OFFICE Corresponding cc: GB PAYMENT DATE: 20090107; PAYMENT YEAR: 08
20090529	EP/PGFP-A [POS; PIF]	POSTGRANT: ANNUAL FEES PAID TO NATIONAL OFFICE Corresponding cc: NL PAYMENT DATE: 20090104; PAYMENT YEAR: 08
20090529	EP/PGFP-A [POS; PIF]	POSTGRANT: ANNUAL FEES PAID TO NATIONAL OFFICE Corresponding cc: DE PAYMENT DATE: 20090102; PAYMENT YEAR: 08
20090430	EP/PGFP-A [POS; PIF]	POSTGRANT: ANNUAL FEES PAID TO NATIONAL OFFICE Corresponding cc: ES PAYMENT DATE: 20090218; PAYMENT YEAR: 08
20080731	EP/PGFP-A [POS; PIF]	POSTGRANT: ANNUAL FEES PAID TO NATIONAL OFFICE Corresponding cc: FR PAYMENT DATE: 20080117; PAYMENT YEAR: 07
20080731	EP/PGFP-A [POS; PIF]	POSTGRANT: ANNUAL FEES PAID TO NATIONAL OFFICE Corresponding cc: DE

		PAYMENT DATE: 20080229; PAYMENT YEAR: 07
20080530	EP/PGFP-A [POS; PIF]	POSTGRANT: ANNUAL FEES PAID TO NATIONAL OFFICE Corresponding cc: NL PAYMENT DATE: 20080124; PAYMENT YEAR: 07
20080530	EP/PGFP-A [POS; PIF]	POSTGRANT: ANNUAL FEES PAID TO NATIONAL OFFICE Corresponding cc: IT PAYMENT DATE: 20080128; PAYMENT YEAR: 07
20080530	EP/PGFP-A [POS; PIF]	POSTGRANT: ANNUAL FEES PAID TO NATIONAL OFFICE Corresponding cc: GB PAYMENT DATE: 20080129; PAYMENT YEAR: 07
20080501	EP/NLR2-A [OPP]	NL: DECISION OF OPPOSITION NL: BESLISSINGEN OVER OPPOSITIES, INGESTELT TEGEN EP OCTROOIEN EFFECTIVE DATE: 20071104
20080430	EP/PG25-A [NEG; RLW]	LAPSED IN A CONTRACTING STATE ANNOUNCED VIA POSTGRANT INFORM. FROM NAT. OFFICE TO EPO Corresponding cc: GR EFFECTIVE DATE: 20060825 LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATION OF THE DESCRIPTION OR TO PAY THE FEE WITHIN THE PRESCRIBED TIME-LIMIT
20080430	EP/PGFP-A [POS; PIF]	POSTGRANT: ANNUAL FEES PAID TO NATIONAL OFFICE Corresponding cc: FR PAYMENT DATE: 20070117; PAYMENT YEAR: 06
20080430	EP/PGFP-A [POS; PIF]	POSTGRANT: ANNUAL FEES PAID TO NATIONAL OFFICE Corresponding cc: ES PAYMENT DATE: 20080128; PAYMENT YEAR: 07
20080312	EP/27C-A [POS; OPP]	TERMINATION OF OPPOSITION PROCEDURE EINSTELLUNG DES EINSPRUCHSVERFAHRENS EFFECTIVE DATE: 20071104
20080102	EP/PGFP-A [POS; PIF]	POSTGRANT: ANNUAL FEES PAID TO NATIONAL OFFICE Corresponding cc: IT PAYMENT DATE: 20070517; PAYMENT YEAR: 06
20070601	EP/NLR1-A [NEG; OPP]	NL: OPPOSITION HAS BEEN FILED WITH THE EPO NL: EUROPESE OCTROOIEN, WAARTEGEN OPPOSITIE IS INGESTELD OPPONENT: S.I.SV.EL. S.P.A.
20070511	EP/REG-A; FR/CA [NMC]	REFERENCE TO A NATIONAL CODE FR: CHANGE OF ADDRESS FR: CHANGEMENT D'ADRESSE Corresponding cc: FR
20070404	EP/26-A [NEG; OPP]	OPPOSITION FILED EINSPRUCH EINGELEGT OPPONENT: S.I.SV.EL. S.P.A.; EFFECTIVE DATE: 20070226
20070301	EP/NLT2-A [NMC]	NL: MODIFICATIONS (OF NAMES), TAKEN FROM THE EUROPEAN PATENT PATENT

		BULLETIN NL: (NAAMS)WIJZIGINGEN, DIE ZIJN OVERGENOMEN UIT HET EP OCTROOIBLAD OWNER: SANDISK CORPORATION; EFFECTIVE DATE: 20070110
20070228	EP/PGFP-A [POS; PIF]	POSTGRANT: ANNUAL FEES PAID TO NATIONAL OFFICE Corresponding cc: DE PAYMENT DATE: 20070228; PAYMENT YEAR: 06
20070131	EP/PG25-A [NEG; RLW]	LAPSED IN A CONTRACTING STATE ANNOUNCED VIA POSTGRANT INFORM. FROM NAT. OFFICE TO EPO Corresponding cc: MC EFFECTIVE DATE: 20070131 LAPSE BECAUSE OF NON-PAYMENT OF DUE FEES
20070126	EP/PGFP-A [POS; PIF]	POSTGRANT: ANNUAL FEES PAID TO NATIONAL OFFICE Corresponding cc: ES PAYMENT DATE: 20070126; PAYMENT YEAR: 06
20070125	EP/PGFP-A [POS; PIF]	POSTGRANT: ANNUAL FEES PAID TO NATIONAL OFFICE Corresponding cc: GB PAYMENT DATE: 20070125; PAYMENT YEAR: 06
20070124	EP/PGFP-A [POS; PIF]	POSTGRANT: ANNUAL FEES PAID TO NATIONAL OFFICE Corresponding cc: NL PAYMENT DATE: 20070124; PAYMENT YEAR: 06
20070110	EP/RAP2-A [NMC]	PATENT OWNER REASSIGNMENT (CORRECTION) PATENTINHABER UEBERTRAGUNG (KORR.) OWNER: SANDISK CORPORATION
20070108	EP/PG25-A [NEG; RLW]	LAPSED IN A CONTRACTING STATE ANNOUNCED VIA POSTGRANT INFORM. FROM NAT. OFFICE TO EPO Corresponding cc: IE EFFECTIVE DATE: 20070108 LAPSE BECAUSE OF NON-PAYMENT OF DUE FEES
20061222	EP/ET-A [POS; ENP]	FR: TRANSLATION FILED FR: TRADUCTION A ETE REMISE
20061201	EP/REG-A; ES/FG2A [ENP]	REFERENCE TO A NATIONAL CODE ES: DEFINITIVE PROTECTION ES: PROTECCION DEFINITIVA Corresponding Pat: ES 2262782 0 [ES2262782]
20061130	EP/REG-A; CH/PL [NEG; RLW]	REFERENCE TO A NATIONAL CODE CH: PATENT CEASED CH: LOESCHUNG/RADIATION/RADIAZION Corresponding cc: CH
20061024	EP/PG25-A [NEG; RLW]	LAPSED IN A CONTRACTING STATE ANNOUNCED VIA POSTGRANT INFORM. FROM NAT. OFFICE TO EPO Corresponding cc: PT EFFECTIVE DATE: 20061024 LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATION OF THE DESCRIPTION OR TO PAY THE FEE WITHIN THE PRESCRIBED TIME-LIMIT
20060824	EP/PG25-A [NEG; RLW]	LAPSED IN A CONTRACTING STATE ANNOUNCED VIA POSTGRANT INFORM.

		FROM NAT. OFFICE TO EPO Corresponding cc: SE EFFECTIVE DATE: 20060824 LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATION OF THE DESCRIPTION OR TO PAY THE FEE WITHIN THE PRESCRIBED TIME-LIMIT
20060824	EP/PG25-A [NEG; RLW]	LAPSED IN A CONTRACTING STATE ANNOUNCED VIA POSTGRANT INFORM. FROM NAT. OFFICE TO EPO Corresponding cc: DK EFFECTIVE DATE: 20060824 LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATION OF THE DESCRIPTION OR TO PAY THE FEE WITHIN THE PRESCRIBED TIME-LIMIT
20060629	EP/REF-A [ENP]	CORRESPONDS TO: ENTSPRICHT Corresponding Pat: DE 60211653 20060629 [DE60211653]
20060628	EP/REG-A; IE/FG4D [POS; PIF]	REFERENCE TO A NATIONAL CODE IE: EUROPEAN PATENTS GRANTED DESIGNATING IRELAND Corresponding cc: IE
20060531	EP/REG-A; CH/EP [POS; ENP]	REFERENCE TO A NATIONAL CODE CH: ENTRY IN THE NATIONAL PHASE CH: EINTRITT IN DIE NATIONALE PHASE Corresponding cc: CH
20060524	EP/REG-A; GB/FG4D [POS; PIF]	REFERENCE TO A NATIONAL CODE GB: EUROPEAN PATENT GRANTED Corresponding cc: GB
20060524	EP/PG25-A [NEG; RLW]	LAPSED IN A CONTRACTING STATE ANNOUNCED VIA POSTGRANT INFORM. FROM NAT. OFFICE TO EPO Corresponding cc: BE EFFECTIVE DATE: 20060524 LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATION OF THE DESCRIPTION OR TO PAY THE FEE WITHIN THE PRESCRIBED TIME-LIMIT
20060524	EP/PG25-A [NEG; RLW]	LAPSED IN A CONTRACTING STATE ANNOUNCED VIA POSTGRANT INFORM. FROM NAT. OFFICE TO EPO Corresponding cc: LI EFFECTIVE DATE: 20060524 LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATION OF THE DESCRIPTION OR TO PAY THE FEE WITHIN THE PRESCRIBED TIME-LIMIT
20060524	EP/PG25-A [NEG; RLW]	LAPSED IN A CONTRACTING STATE ANNOUNCED VIA POSTGRANT INFORM. FROM NAT. OFFICE TO EPO Corresponding cc: IT EFFECTIVE DATE: 20060524 LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATION OF THE DESCRIPTION OR TO PAY THE FEE WITHIN THE PRESCRIBED TIME-LIMIT
20060524	EP/PG25-A [NEG; RLW]	LAPSED IN A CONTRACTING STATE ANNOUNCED VIA POSTGRANT INFORM. FROM NAT. OFFICE TO EPO Corresponding cc: CH EFFECTIVE DATE: 20060524 LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATION OF THE DESCRIPTION OR TO

		PAY THE FEE WITHIN THE PRESCRIBED TIME-LIMIT
20060524	EP/PG25-A [NEG; RLW]	LAPSED IN A CONTRACTING STATE ANNOUNCED VIA POSTGRANT INFORM. FROM NAT. OFFICE TO EPO Corresponding cc: FI EFFECTIVE DATE: 20060524 LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATION OF THE DESCRIPTION OR TO PAY THE FEE WITHIN THE PRESCRIBED TIME-LIMIT
20060524	EP/PG25-A [NEG; RLW]	LAPSED IN A CONTRACTING STATE ANNOUNCED VIA POSTGRANT INFORM. FROM NAT. OFFICE TO EPO. Corresponding cc: AT EFFECTIVE DATE: 20060524 LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATION OF THE DESCRIPTION OR TO PAY THE FEE WITHIN THE PRESCRIBED TIME-LIMIT
20060524	EP/AK-A [POS; ADM]	DESIGNATED CONTRACTING STATES: BENANNTE VERTRAGSSTAATEN AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
20060524	EP-B1 [POS; PIF]	Patent specification EP1352394 B1 20060524 [EP1352394]
20040107	EP/17Q-A [POS; EXM]	FIRST EXAMINATION REPORT ERSTER PRUEFUNGSBESCHEID EFFECTIVE DATE: 20031120
20031015	EP/17P-A [POS; EXM]	REQUEST FOR EXAMINATION FILED PRUEFUNGSANTRAG GESTELLT EFFECTIVE DATE: 20030715
20031015	EP-A2 [POS; EXM]	Application published without search report EP1352394 A2 20031015 [EP1352394]
20031015	EP/AX-A [POS; ADM]	EXTENSION OF THE EUROPEAN PATENT TO ERSTRECKUNG DES EUROPAEISCHEN PATENTS AUF AL LT LV MK RO SI
20031015	EP/AK-A [POS; ADM]	DESIGNATED CONTRACTING STATES: BENANNTE VERTRAGSSTAATEN AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
20020107	EP-API [POS; EXM]	FILING DETAILS EPO2703078 20020107 [2002EP-0703078]
Family member: WO02058074 A2		
Status: Alive		
20060524	WO/WWG [POS; ADM]	WIPO INFORMATION: GRANT IN NATIONAL OFFICE Corresponding Appl: EP 2002703078 [2002EP-0703078]
20031127	WO/REG; DE/8642 [NEG; ENP; RLW]	REFERENCE TO NATIONAL CODE DE: IMPACT ABOLISHED FOR DE - I.E. PCT APPL. NOT ENT. GERMAN PHASE DE: WIRKUNG WEGGEFALLEN FUER DE Corresponding cc: DE
20031015	WO/WWP [POS; ADM]	WIPO INFORMATION: PUBLISHED IN NATIONAL OFFICE Corresponding Appl: EP 2002703078 [2002EP-0703078]
20030827	WO/WWP [POS; ADM]	WIPO INFORMATION: PUBLISHED IN NATIONAL OFFICE Corresponding Appl: KR 20037009551 [2003KR-7009551]

20030718	WO/WWE [POS; ENP]	WIPO INFORMATION: ENTRY INTO NATIONAL PHASE Corresponding Appl: KR 20037009551 [2003KR-7009551]
20030718	WO/WWE [POS; ENP]	WIPO INFORMATION: ENTRY INTO NATIONAL PHASE Corresponding Appl: CN 028038827 [2002CN-8038827]
20030718	WO/WWE [POS; ENP]	WIPO INFORMATION: ENTRY INTO NATIONAL PHASE Corresponding Appl: JP 2002558275 [2002JP-0558275]
20030715	WO/WWE [POS; ENP]	WIPO INFORMATION: ENTRY INTO NATIONAL PHASE Corresponding Appl: EP 2002703078 [2002EP-0703078]
20030710	WO-A3 [POS; EXM]	International search report WO02058074 A3 20030710 [WO200258074]
20021003	WO/COP [COR]	CORRECTED VERSION OF PAMPHLET PAGES 1/7-7/7, DRAWINGS, REPLACED BY NEW PAGES 1/9-9/9; DUE TO LATE TRANSMITTAL BY THE RECEIVING OFFICE
20021003	WO/AL [POS; ADM]	DESIGNATED COUNTRIES FOR REGIONAL PATENTS CITED IN A SUBSEQUENTLY PUBLISHED SEARCH REPORT GH GM KE LS MW MZ SD SL SZ TZ UG ZM ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR BF BJ CF CG CI CM GA GN GQ GW ML MR NE SN TD TG
20021003	WO/AK [POS; ADM]	DESIGNATED STATES CITED IN A SUBSEQUENTLY PUBLISHED SEARCH REPORT AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PH PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW
20021003	WO-A9 [POS; EXM]	Complete corrected document WO02058074 A9 20021003 [WO200258074]
20020918	WO/121 [ENP]	EP: THE EPO HAS BEEN INFORMED BY WIPO THAT EP WAS DESIGNATED IN THIS APPLICATION
20020725	WO-A2 [POS; EXM]	International publication without international search report WO02058074 A2 20020725 [WO200258074]
20020725	WO/AL [POS; ADM]	DESIGNATED COUNTRIES FOR REGIONAL PATENTS CITED IN A SUBSEQUENTLY PUBLISHED SEARCH REPORT GH GM KE LS MW MZ SD SL SZ TZ UG ZM ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR BF BJ CF CG CI CM GA GN GQ GW ML MR NE SN TD TG
20020725	WO/AK [POS; ADM]	DESIGNATED STATES CITED IN A SUBSEQUENTLY PUBLISHED SEARCH REPORT AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PH PL PT RO

		RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW
20020107	WO-API [POS; EXM]	FILING DETAILS WOUS0200366 20020107 [2002WO-US00366]
Family member: US2002099904 A1		
Status: Alive		
20040713	US-B2 [POS; PIF]	Granted patent as second publication US6763424 B2 20040713 [US6763424]
20020725	US-A1 [POS; EXM]	First published patent application US2002099904 A1 20020725 [US2002099904]
20010119	US/AS-A [NMC]	ASSIGNMENT OWNER: SANDISK CORPORATION, CALIFORNIA; EFFECTIVE DATE: 20010118 ASSIGNMENT OF ASSIGNORS INTEREST;ASSIGNOR:CONLEY, KEVIN M.;REEL/FRAME:011487/0823
20010119	US-API [POS; EXM]	FILING DETAILS US76643601 20010119 [2001US-0766436]
20010119	US/AS-A [NMC]	ASSIGNMENT OWNER: SANDISK CORPORATION 140 CASPIAN COURTSUNNYVALE, CA; EFFECTIVE DATE: 20010118 ASSIGNMENT OF ASSIGNORS INTEREST;ASSIGNOR:CONLEY, KEVIN M. /AR;REEL/FRAME:011487/0823

© Questel-Orbit



September 14, 2009

Commissioner For Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Customer No. 66785

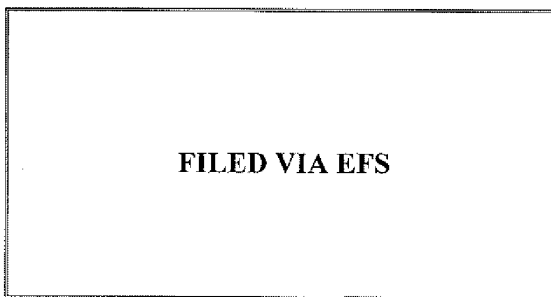
Re: Applicant: Kevin M. Conley
Title: Partial Block Data Programming And Reading Operations In A
Non-Volatile Memory
Application No.: 11/250,238 Filing Date: October 13, 2005
Examiner: Dinh, Ngoc V. Group Art Unit: 2188
Docket No.: 0084567-156US2 Conf. No.: 7727

Dear Sir:

Transmitted herewith are the following documents in the above-identified application:

- (1) This Transmittal Letter; and
- (2) Power of Attorney by Assignee of Entire Interest (2 pages).

No additional fee is required.



Respectfully submitted,

A handwritten signature in black ink, appearing to read "Philip Yau".

Philip Yau
Reg. No. 32,892

Anchorage
Bellevue
Los Angeles

New York
Portland
San Francisco

Seattle
Shanghai
Washington, D.C.

www.dwt.com

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Kevin M. Conley		
Title:	Partial Block Data Programming And Reading Operations In A Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	Ocotber 13, 2005
Examiner:	Dinh, Ngoc V	Group Art Unit:	2188
Docket No.:	0084567-156US2	Conf. No.:	7727

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

POWER OF ATTORNEY BY ASSIGNEE OF ENTIRE INTEREST

Sir:

SanDisk Corporation, a Delaware corporation, hereby revokes all powers of attorney previously given and appoints the registered patent practitioners associated with **Customer No. 66785** specifically and solely to prosecute the above-identified application and to transact all business in the United States Patent and Trademark Office in connection therewith.

Please direct all communications in connection with the above-identified patent application to:

Customer Number: 66785
Davis Wright Tremaine LLP
505 Montgomery Street, Suite 800
San Francisco, CA 94111
Telephone No.: (415) 276-6500
Fax No.: (415) 276-6599


Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

ASSIGNEE CERTIFICATION UNDER 37 CFR 3.73(B)

The undersigned hereby certifies that SanDisk Corporation is the assignee of the entire right, title and interest in the above-identified patent application by virtue of a chain of title from the inventor to SanDisk Corporation, as shown by the Assignment from the inventor, which was separately recorded in the United States Patent and Trademark Office at Reel 011487, Frame 0823.

The undersigned is empowered to sign this certificate on behalf of SanDisk Corporation.

9/11/09
Date _____

Name: E. Earle Thompson
Title: Vice President and Chief IP Counsel

Electronic Acknowledgement Receipt

EFS ID:	6064557
Application Number:	11250238
International Application Number:	
Confirmation Number:	7727
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY
First Named Inventor/Applicant Name:	Kevin M. Conley
Customer Number:	66785
Filer:	Philip Yau/Svetlana Stellmach
Filer Authorized By:	Philip Yau
Attorney Docket Number:	0084567-156US2
Receipt Date:	14-SEP-2009
Filing Date:	13-OCT-2005
Time Stamp:	13:32:29
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no
------------------------	----

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		SNDK156US2- Trans_POA_9-14-09.pdf	61537 <small>b9b442f0cd3d93fbf87552c0628cd8ed85f4eb5b</small>	yes	3

Multipart Description/PDF files in .zip description		
Document Description	Start	End
Miscellaneous Incoming Letter	1	1
Power of Attorney	2	3
Warnings:		
Information:		
Total Files Size (in bytes):		61537
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>		

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.		Application No.				
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		0084567-156US2		11/250,238				
		Applicants		Conf. No.				
(Use several sheets if necessary)		Kevin M. Conley		7727				
(Form PTO-1449)		Filing Date		Art Group				
		October 13, 2005		2188				
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
	1	6,839,285	1/4/2005	Zink et al.				
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
	2	USPTO, "Notice of Allowance and Fee(s) Due," mailed in related U.S. Patent Application No. 12/371,460 on November 16, 2009, 24 pages.						
Examiner			Date Considered					
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

Electronic Acknowledgement Receipt

EFS ID:	6521956
Application Number:	11250238
International Application Number:	
Confirmation Number:	7727
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY
First Named Inventor/Applicant Name:	Kevin M. Conley
Customer Number:	66785
Filer:	Philip Yau/Svetlana Stellmach
Filer Authorized By:	Philip Yau
Attorney Docket Number:	0084567-156US2
Receipt Date:	24-NOV-2009
Filing Date:	13-OCT-2005
Time Stamp:	19:03:12
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no
------------------------	----

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		SNDK156US2_Suppl_IDS_11-24-09.pdf	73317 <small>2aca27171fcdb151ef3753dd33285b248104ab4a</small>	yes	3

Multipart Description/PDF files in .zip description					
Document Description			Start	End	
Transmittal Letter			1	2	
Information Disclosure Statement (IDS) Filed (SB/08)			3	3	
Warnings:					
Information:					
2	NPL Documents	Reference_2_SNDK156US3_NO A_dtd-11-16-09.pdf	1038701	no	24
			a6fe0587c1f1b641abb05530051df5b93c00 22a7		
Warnings:					
Information:					
Total Files Size (in bytes):			1112018		
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Kevin M. Conley		
Title:	Block Data Programming and Reading Operations in a Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	October 13, 2005
Examiner:	Ngoc V. Dinh	Group Art Unit:	2188
Docket No.:	0084567-156US2	Conf. No.:	7727

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Dear Sir:

Pursuant to 37 C.F.R. §§ 1.56, 1.97 and 1.98, Applicant calls the documents listed on the enclosed Form PTO-1449 to the Examiner's attention in this patent application.

According to 37 C.F.R. 1.98(2)(ii), copies of the U.S. Patents and U.S. Published Patent Applications documents are not required and are therefore not enclosed. A copy of the listed Other Art is enclosed.

Citation of these documents shall not be construed as (1) an admission that the documents are prior art with respect to the invention or inventions claimed in this application, (2) a representation that a search has been made (other than as indicated by any cited document), or (3) an admission that the cited information is, or is considered to be, material to patentability as defined in § 1.56(b).

Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.		Application No.				
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		0084567-156US2		11/250,238				
		Applicants		Conf. No.				
(Use several sheets if necessary)		Kevin M. Conley		7727				
(Form PTO-1449)		Filing Date		Art Group				
		October 13, 2005		2188				
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
	1	7,657,702 B2	2/2/2010	Conley				
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
	2	Korean Patent Office, "Preliminary Rejection," corresponding Korean Patent Application No. 2008-7028861, mailed on January 28, 2010, 11 pages (including translation.)						
Examiner			Date Considered					
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

FILED VIA EFS
Sheet 1 of 1

Electronic Acknowledgement Receipt

EFS ID:	7031162
Application Number:	11250238
International Application Number:	
Confirmation Number:	7727
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY
First Named Inventor/Applicant Name:	Kevin M. Conley
Customer Number:	66785
Filer:	Philip Yau/Svetlana Stellmach
Filer Authorized By:	Philip Yau
Attorney Docket Number:	0084567-156US2
Receipt Date:	17-FEB-2010
Filing Date:	13-OCT-2005
Time Stamp:	17:03:17
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no
------------------------	----

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		SNDK156US2_Suppl_IDS_2-17-10.pdf	287870 <small>50f19230910874774fec582ad2c915da929295a0</small>	yes	3

Multipart Description/PDF files in .zip description			
Document Description	Start	End	
Transmittal Letter	1	2	
Information Disclosure Statement (IDS) Filed (SB/08)	3	3	

Warnings:

Information:

2	NPL Documents	Reference_SNDK156KR1_Prelim-Rejection_1-28-10.pdf	1820791 93de96586f30eab4c5e46b2a328fe6d0bc685e5c	no	11
---	---------------	---	---	----	----

Warnings:

Information:

Total Files Size (in bytes):	2108661
-------------------------------------	---------

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Kevin M. Conley		
Title:	Block Data Programming and Reading Operations in a Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	October 13, 2005
Examiner:	Ngoc V. Dinh	Group Art Unit:	2188
Docket No.:	0084567-156US2	Conf. No.:	7727

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Dear Sir:

Pursuant to 37 C.F.R. §§ 1.56, 1.97 and 1.98, Applicant calls the documents listed on the enclosed Form PTO-1449 to the Examiner's attention in this patent application.

A copy of the listed Other Art is enclosed.

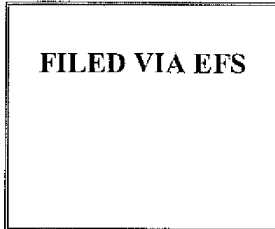
Citation of these documents shall not be construed as (1) an admission that the documents are prior art with respect to the invention or inventions claimed in this application, (2) a representation that a search has been made (other than as indicated by any cited document), or (3) an admission that the cited information is, or is considered to be, material to patentability as defined in § 1.56(b).

This information disclosure statement is submitted under 37 C.F.R. § 1.97(b) and consequently no fee should be required. The Commissioner is authorized, however, to charge

Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

any fee that may be required, or to credit any overpayment, against Deposit Account No. 04-0258.



Respectfully submitted,

Philip Yau  2/17/2010
Reg. No. 32,892 Date

Davis Wright Tremain LLP
505 Montgomery Street, Suite 800
San Francisco, CA 94111-6533
(415) 276-6500 (main)
(415) 276-6543 (direct)
(415) 276-6599 (fax)
Email: PhilipYau@dwt.com

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.		Application No.				
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		0084567-156US2		11/250,238				
		Applicants		Conf. No.				
(Use several sheets if necessary)		Kevin M. Conley		7727				
(Form PTO-1449)		Filing Date		Art Group				
		October 13, 2005		2188				
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
	1	Korean Patent Office, "Preliminary Rejection," corresponding Korean Patent Application No. 2008-7028861, mailed on January 28, 2010, 12 pages (including translation.)						
Examiner			Date Considered					
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

Electronic Acknowledgement Receipt

EFS ID:	7079837
Application Number:	11250238
International Application Number:	
Confirmation Number:	7727
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY
First Named Inventor/Applicant Name:	Kevin M. Conley
Customer Number:	66785
Filer:	Philip Yau/Svetlana Stellmach
Filer Authorized By:	Philip Yau
Attorney Docket Number:	0084567-156US2
Receipt Date:	24-FEB-2010
Filing Date:	13-OCT-2005
Time Stamp:	16:23:11
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no
------------------------	----

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		SNDK156US2_Suppl_IDS_2-24-10.pdf	62327 f0881e53dfb12ed164c5fbef004e980eb15d1ad1	yes	3

Multipart Description/PDF files in .zip description			
Document Description	Start	End	
Transmittal Letter	1	2	
Information Disclosure Statement (IDS) Filed (SB/08)	3	3	

Warnings:

Information:

2	NPL Documents	Reference_SNDK156KR1_Copy- OA_1-28-10_translation.pdf	1407961	no	12
			c093e63d2bd6c5d4f2cb9d64d8ea032a381 cce13		

Warnings:

Information:

Total Files Size (in bytes):	1470288
-------------------------------------	---------

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Kevin M. Conley		
Title:	Block Data Programming and Reading Operations in a Non-Volatile Memory		
Application No.:	11/250,238	Filing Date:	October 13, 2005
Examiner:	Ngoc V. Dinh	Group Art Unit:	2188
Docket No.:	0084567-156US2	Conf. No.:	7727

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Dear Sir:

Pursuant to 37 C.F.R. §§ 1.56, 1.97 and 1.98, Applicant calls the documents listed on the enclosed Form PTO-1449 to the Examiner's attention in this patent application.

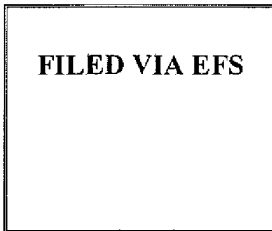
A copy of the listed Other Art is enclosed.

Citation of these documents shall not be construed as (1) an admission that the documents are prior art with respect to the invention or inventions claimed in this application, (2) a representation that a search has been made (other than as indicated by any cited document), or (3) an admission that the cited information is, or is considered to be, material to patentability as defined in § 1.56(b).

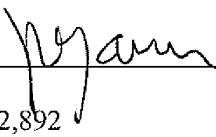
Attorney Docket No.: SNDK.156US2
FILED VIA EFS

Application No.: 11/250,238

This information disclosure statement is submitted under 37 C.F.R. § 1.97(b) and consequently no fee should be required. The Commissioner is authorized, however, to charge any fee that may be required, or to credit any overpayment, against Deposit Account No. 04-0258.



Respectfully submitted,



Philip Yau
Reg. No. 32,892

February 24, 2010

Date

Davis Wright Tremaine LLP
505 Montgomery Street, Suite 800
San Francisco, CA 94111-6533
(415) 276-6500 (main)
(415) 276-6543 (direct)
(415) 276-6599 (fax)
Email: PhilipYau@dwt.com

PLUS Search Results for S/N 11250238, Searched Tue Mar 09 10:40:31 EST 2010
The Patent Linguistics Utility System (PLUS) is a USPTO automated search system for U.S. Patents from 1971 to the present PLUS is a query-by-example search system which produces a list of patents that are most closely related linguistically to the application searched. This search was prepared by the staff of the Scientific and Technical Information Center, SIRA.

5177796 99	6106572 96
4287558 98	
5359706 98	
5956743 98	
4504860 97	
4879720 97	
5717945 97	
5761706 97	
4292674 97	
4758951 97	
4853779 97	
5500685 97	
5602995 97	
5611067 97	
5668968 97	
5832235 97	
5873089 97	
5875455 97	
5890188 97	
5983319 97	
6141722 97	
6222775 97	
6236593 97	
4394746 96	
4445191 96	
4573074 96	
4581638 96	
4817090 96	
4926322 96	
4939598 96	
4941043 96	
5267264 96	
5280612 96	
5412781 96	
5490260 96	
5579503 96	
5587948 96	
5619594 96	
5675387 96	
5704022 96	
5809548 96	
5835959 96	
5861892 96	
5898620 96	
5910821 96	
5982442 96	
6000006 96	
6023423 96	
6094708 96	



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

66785 7590 06/01/2010

DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION
505 MONTGOMERY STREET
SUITE 800
SAN FRANCISCO, CA 94111

EXAMINER

DINH, NGOC V

ART UNIT PAPER NUMBER

2185

DATE MAILED: 06/01/2010

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.

11/250,238 10/13/2005 Kevin M. Conley 0084567-156US2 7727

TITLE OF INVENTION: PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY

Table with 7 columns: APPLN. TYPE, SMALL ENTITY, ISSUE FEE DUE, PUBLICATION FEE DUE, PREV. PAID ISSUE FEE, TOTAL FEE(S) DUE, DATE DUE

nonprovisional NO \$1510 \$300 \$0 \$1810 09/01/2010

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

- A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.
B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

- A. Pay TOTAL FEE(S) DUE shown above, or
B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PART B - FEE(S) TRANSMITTAL

**Complete and send this form, together with applicable fee(s), to: Mail Mail Stop ISSUE FEE
 Commissioner for Patents
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 or Fax (571)-273-2885**

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

66785 7590 06/01/2010

DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION
 505 MONTGOMERY STREET
 SUITE 800
 SAN FRANCISCO, CA 94111

Certificate of Mailing or Transmission
 I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

_____ (Depositor's name)
_____ (Signature)
_____ (Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/250,238	10/13/2005	Kevin M. Conley	0084567-156US2	7727

TITLE OF INVENTION: PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	\$1510	\$300	\$0	\$1810	09/01/2010

EXAMINER	ART UNIT	CLASS-SUBCLASS
DINH, NGOC V	2185	711-103000

<p>1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).</p> <p><input type="checkbox"/> Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.</p> <p><input type="checkbox"/> "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. Use of a Customer Number is required.</p>	<p>2. For printing on the patent front page, list</p> <p>(1) the names of up to 3 registered patent attorneys or agents OR, alternatively, _____ 1</p> <p>(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed. _____ 2</p> <p>_____ 3</p>
---	---

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE _____ (B) RESIDENCE: (CITY and STATE OR COUNTRY) _____

Please check the appropriate assignee category or categories (will not be printed on the patent): Individual Corporation or other private group entity Government

<p>4a. The following fee(s) are submitted:</p> <p><input type="checkbox"/> Issue Fee</p> <p><input type="checkbox"/> Publication Fee (No small entity discount permitted)</p> <p><input type="checkbox"/> Advance Order - # of Copies _____</p>	<p>4b. Payment of Fee(s): (Please first reapply any previously paid issue fee shown above)</p> <p><input type="checkbox"/> A check is enclosed.</p> <p><input type="checkbox"/> Payment by credit card. Form PTO-2038 is attached.</p> <p><input type="checkbox"/> The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).</p>
---	--

5. Change in Entity Status (from status indicated above)

a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27. b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.
Row 1: 11/250,238, 10/13/2005, Kevin M. Conley, 0084567-156US2, 7727
Row 2: 66785, 7590, 06/01/2010, [EXAMINER], [DINH, NGOC V]
Row 3: [ART UNIT], [PAPER NUMBER], 2185, DATE MAILED: 06/01/2010

DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION
505 MONTGOMERY STREET
SUITE 800
SAN FRANCISCO, CA 94111

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 0 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 0 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

Notice of Allowability	Application No.	Applicant(s)	
	11/250,238	CONLEY, KEVIN M.	
	Examiner	Art Unit	
	NGOC V. DINH	2185	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. This communication is responsive to 06/08/09.
2. The allowed claim(s) is/are 4, 7-18, 26-118 (renumbered as 1-106).
3. Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some* c) None of the:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____ .
 3. Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
5. CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
 - (a) including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
 - 1) hereto or 2) to Paper No./Mail Date _____.
 - (b) including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

- | | |
|--|---|
| <ol style="list-style-type: none"> 1. <input type="checkbox"/> Notice of References Cited (PTO-892) 2. <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) 3. <input checked="" type="checkbox"/> Information Disclosure Statements (PTO/SB/08),
Paper No./Mail Date <u>06/08/09, 11/24/09, 2/17/10, and 2/24/10</u> 4. <input type="checkbox"/> Examiner's Comment Regarding Requirement for Deposit of Biological Material | <ol style="list-style-type: none"> 5. <input type="checkbox"/> Notice of Informal Patent Application 6. <input type="checkbox"/> Interview Summary (PTO-413),
Paper No./Mail Date _____ . 7. <input type="checkbox"/> Examiner's Amendment/Comment 8. <input checked="" type="checkbox"/> Examiner's Statement of Reasons for Allowance 9. <input type="checkbox"/> Other _____. |
|--|---|

--	--

DETAILED ACTION

This office action is a response to the RCE filed on 06/08/09.

INFORMATION DISCLOSURE STATEMENT

The Applicant's submission of the IDS filed 06/08/09, 11/24/09, 2/17/10, and 2/24/10 have been considered. As required by M.P.E.P. 609 C(2), copies of the PTOL-1449 is attached to the instant office action.

REASON FOR ALLOWANCE

The primary reasons for allowance of claims 4, 13, 46, 59, 68, 69, 85, 106 in the instant application is the combination with the inclusion of at least the limitations set forth in lines 8-13 of claims 4, 13; lines 15-19 of claim 46; lines 13-17 of claim 59; lines 11-17 of claim 68; lines 6-11 of claim 69; lines 14-18 of claim 85; lines 18-22 of claim 106.

Because claims 7-12, 26-35; 14-18, 36-45; 47-58, 71, 73, 75-77; 60-67, 72, 74, 78-79, 70, 80-84, 86-105, 107-118 depend directly or indirectly on claims 4, 13, 46, 59, 69, 85, 106. These claims are considered allowable for at least the same reasons noted above.

CONCLUSION

Any response to this action should be mailed to:

Under Secretary of Commerce for intellectual Property and Director of the
United States Patent and Trademark Office

PO Box 1450

Alexandria, VA 22313-1450

or faxed to:

(571) 273-8300, (for Official communications intended for entry)

Under Secretary of Commerce for intellectual Property and Director of the
United States Patent and Trademark Office PO Box 1450 Alexandria, VA 22313-1450 or
faxed to: (571) 273-8300, (for Official communications intended for entry) Information

Art Unit: 2185

regarding the status of an application may be obtained from the Patent Application Information Retrieval (PMR) system. Status information for published Applications may be obtained from either Private PMR or Public PMR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pak-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ngoc Dinh whose telephone number is (571) 272-4191. The examiner can normally be reached on Monday-Friday 8:30 AM-5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor Sanjiv Shah can be reached on (571) 272-4098.

/N. V. D./

Examiner, Art Unit 2185

/Hong Kim/

Primary Examiner, Art Unit 2185

EAST Search History

EAST Search History (Prior Art)


Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	853095	(flash eeprom eeprom non\$volatile)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/09/09 14:10
L2	596	((updat\$3 programm \$3) with offset) same 1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/09/09 14:11
L3	1	"6023423".pn.	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:11
L4	655770	(eeprom flash non \$volatile)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:11
L5	18818	(programming program\$4 programing update updat\$3) same (time \$stamp (time adj stamp))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:11
L6	448	L4 same L5	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:11
L7	7	(most near2 recent near3 (page block sector)) same L6	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:11
L8	942580	prom eeprom flash rom non\$volatile raid	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:11
L9	322	(journal journal\$3) same ((time adj stamp) time\$stamp)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:11
L10	219	L8 and L9	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:11
L11	312	((most least) near3 (recently) near3 (block page sector)) with (time timing (time\$stamp))	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:12

L12	169	L11 and (flash eeprom rom non \$volatile)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:12
L13	2392	(eeprom eprom flash non\$volatile) with separate with (unit portion section module)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:12
L14	63	((program\$4 write witting written) with (offset)) and L13	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:12
L15	8209	711/103 711/200 711/203	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:12
L16	1922	("same" identical common) near5 (logical adj2 address)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:12
L17	461	L15 and L16	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:12
L18	71	((updat\$3 programm \$3) with offset) and 17	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/09/09 14:13
L19	8209	711/103 711/200 711/203	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:13
L20	50	2 and 19	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:13
L21	25	2 and 16	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:13
L22	37614	(logical physical) with (updat\$3 programm\$3)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:14
L23	41	2 same 22	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:14
L24	2392	(eeprom eprom flash non\$volatile) with separate with (unit portion section module)	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:15

L25	63	((program\$4 write writting written) with (offset)) and L24	US-PGPUB; USPAT; EPO; IBM_TDB	OR	ON	2009/09/09 14:15
-----	----	--	-------------------------------------	----	----	---------------------

9/ 9/ 2009 2:15:30 PM

C:\ Documents and Settings\ ndinh\ My Documents\ EAST\ Workspaces\ 11250238.wsp

Issue Classification 	Application/Control No. 11250238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.
	Examiner NGOC V DINH	Art Unit 2185

<input type="checkbox"/> Claims renumbered in the same order as presented by applicant								<input type="checkbox"/> CPA		<input checked="" type="checkbox"/> T.D.		<input type="checkbox"/> R.1.47			
22	17	35	47	46	77	95	107								
23	18	36	48	59	78	96	108								
	19	37	49	60	79	97	109								
	20	38	50	68	80	98	110								
	21	39	51	69	81	99	111								
	22	40	52	70	82	100	112								
	23	41	53	71	83	101	113								
	24	42	54	72	84	102	114								
	25	43	55	73	85	103	115								
8	26	48	56	74	86	104	116								
9	27	47	57	75	87	105	117								
10	28	49	58	76	88	106	118								
11	29	52	59	77	89										
12	30	53	60	78	90										

/NGOC V DINH/ Examiner.Art Unit 2185 (Assistant Examiner)	09/21/09 (Date)	Total Claims Allowed: 106	
/Hong Kim/ Primary Examiner.Art Unit 2185 (Primary Examiner)	09/28/2009 (Date)	O.G. Print Claim(s) 1	O.G. Print Figure 14



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
 United States Patent and Trademark Office
 Address: COMMISSIONER FOR PATENTS
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 7727

SERIAL NUMBER 11/250,238	FILING or 371(c) DATE 10/13/2005 RULE	CLASS 711	GROUP ART UNIT 2185	ATTORNEY DOCKET NO. 0084567-156US2		
APPLICANTS Kevin M. Conley, San Jose, CA;						
** CONTINUING DATA ***** This application is a CON of 10/841,388 05/07/2004 PAT 6,968,421 which is a CON of 09/766,436 01/19/2001 PAT 6,763,424						
** FOREIGN APPLICATIONS *****						
** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ** 11/03/2005						
Foreign Priority claimed <input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	35 USC 119(a-d) conditions met <input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	<input type="checkbox"/> Met after Allowance Initials _____	STATE OR COUNTRY CA	SHEETS DRAWINGS 9	TOTAL CLAIMS 3	INDEPENDENT CLAIMS 2
ADDRESS DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION 505 MONTGOMERY STREET SUITE 800 SAN FRANCISCO, CA 94111 UNITED STATES						
TITLE PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY						
FILING FEE RECEIVED 10966	FEES: Authority has been given in Paper No. _____ to charge/credit DEPOSIT ACCOUNT No. _____ for following:		<input type="checkbox"/> All Fees <input type="checkbox"/> 1.16 Fees (Filing) <input type="checkbox"/> 1.17 Fees (Processing Ext. of time) <input type="checkbox"/> 1.18 Fees (Issue) <input type="checkbox"/> Other _____ <input type="checkbox"/> Credit			

11/24/2009

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.		Application No.		
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		0084567-156US2		11/250,238		
(Use several sheets if necessary)		Applicants		Conf. No.		
(Form PTO-1449)		Kevin M. Conley		7727		
		Filing Date		Art Group		
		October 13, 2005		2188		
U.S. Patent Documents						
*Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate
<i>Idc</i>	1	6,839,285	1/4/2005	Zink et al.		
U.S. Published Patent Application Documents						
*Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate
Foreign Patent Documents						
						Translation
	Document	Date	Country	Class	Subclass	Yes No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)						
<i>Wh</i>	2	USPTO, "Notice of Allowance and Fee(s) Due," mailed in related U.S. Patent Application No. 12/371,460 on November 16, 2009. 24 pages.				
Examiner	<i>H. Kim</i>		Date Considered <i>5/27/10</i>			
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.						

2/24/2010


U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.		Application No.				
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		0084567-156US2		11/250,238				
		Applicants		Conf. No.				
(Use several sheets if necessary)		Kevin M. Conley		7727				
(Form PTO-1449)		Filing Date		Art Group				
		October 13, 2005		2188				
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
						Translation		
		Document	Date	Country	Class	Subclass	Yes	No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
10/2	1	Korean Patent Office, "Preliminary Rejection," corresponding Korean Patent Application No. 2008-7028861, mailed on January 28, 2010, 12 pages (including translation.)						
Examiner	H. Kim		Date Considered 5/27/10					
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

FILED VIA EFS
Sheet 1 of 1

2/17/2010

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.		Application No.				
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		0084567-156US2		11/250,238				
		Applicants		Conf. No.				
(Use several sheets if necessary)		Kevin M. Conley		7727				
(Form PTO-1449)		Filing Date		Art Group				
		October 13, 2005		2188				
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
<i>idk</i>	1	7,657,702 B2	2/2/2010	Conley				
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
<i>idk</i>	2	Korean Patent Office, "Preliminary Rejection," corresponding Korean Patent Application No. 2008-7028861, mailed on January 28, 2010, 11 pages (including translation.)						
Examiner	<i>H. Kim</i>		Date Considered	<i>5/27/10</i>				
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

U.S. Department of Commerce, Patent and Trademark		Atty. Docket No.		Application No.				
INFORMATION DISCLOSURE STATEMENT BY APPLICANT		0084567-156US2		11/250,238				
		Applicants		Conf. No.				
(Use several sheets if necessary)		Kevin M. Conley		7727				
(Form PTO-1449)		Filing Date		Art Group				
		October 13, 2005		2188				
U.S. Patent Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
U.S. Published Patent Application Documents								
*Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate	
Foreign Patent Documents								
							Translation	
		Document	Date	Country	Class	Subclass	Yes	No
OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)								
/N.D./	1	United States International Trade Commission, "In the Matter of Certain Flash Memory Controllers, Drives, Memory Cards and Media Players, and Products Containing Same," Inv. No. 337-TA-619, Initial Determination on Violation of Section 337 and Recommended Determination on Remedy and Bond, Administrative Law Judge Charles E. Bullock, Public Version, May 5, 2009, 398 pages.						
Examiner	/Ngoc Dinh/		Date Considered	09/09/2009				
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with your communication to applicant.								

Index of Claims 	Application/Control No. 11250238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.
	Examiner NGOC V DINH	Art Unit 2185

✓	Rejected
=	Allowed


-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

Claims renumbered in the same order as presented by applicant
 CPA
 T.D.
 R.1.47

CLAIM		DATE							
Final	Original	09/14/2009							
	1	-							
	2	-							
	3	-							
1	4	=							
	5	-							
	6	-							
2	7	=							
3	8	=							
4	9	=							
5	10	=							
6	11	=							
7	12	=							
18	13	=							
19	14	=							
20	15	=							
21	16	=							
22	17	=							
23	18	=							
	19	-							
	20	-							
	21	-							
	22	-							
	23	-							
	24	-							
	25	-							
8	26	=							
9	27	=							
10	28	=							
11	29	=							
12	30	=							
13	31	=							
14	32	=							
15	33	=							
16	34	=							
17	35	=							
24	36	=							

Index of Claims 	Application/Control No. 11250238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.
	Examiner NGOC V DINH	Art Unit 2185

✓	Rejected
=	Allowed


-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

Claims renumbered in the same order as presented by applicant
 CPA
 T.D.
 R.1.47


CLAIM		DATE							
Final	Original	09/14/2009							
25	37	=							
26	38	=							
27	39	=							
28	40	=							
29	41	=							
30	42	=							
31	43	=							
32	44	=							
33	45	=							
34	46	=							
35	47	=							
36	48	=							
37	49	=							
38	50	=							
39	51	=							
40	52	=							
41	53	=							
42	54	=							
43	55	=							
48	56	=							
47	57	=							
49	58	=							
52	59	=							
53	60	=							
54	61	=							
55	62	=							
56	63	=							
57	64	=							
58	65	=							
61	66	=							
62	67	=							
65	68	=							
66	69	=							
67	70	=							
50	71	=							
63	72	=							

Index of Claims 	Application/Control No. 11250238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.
	Examiner NGOC V DINH	Art Unit 2185

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claims renumbered in the same order as presented by applicant
 CPA
 T.D.
 R.1.47

CLAIM		DATE									
Final	Original	09/14/2009									
51	73	=									
64	74	=									
44	75	=									
45	76	=									
46	77	=									
59	78	=									
60	79	=									
68	80	=									
69	81	=									
70	82	=									
71	83	=									
72	84	=									
73	85	=									
74	86	=									
75	87	=									
76	88	=									
77	89	=									
78	90	=									
79	91	=									
80	92	=									
81	93	=									
82	94	=									
83	95	=									
84	96	=									
85	97	=									
86	98	=									
87	99	=									
88	100	=									
89	101	=									
90	102	=									
91	103	=									
92	104	=									
93	105	=									
94	106	=									
95	107	=									
96	108	=									

<i>Index of Claims</i> 	Application/Control No. 11250238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.
	Examiner NGOC V DINH	Art Unit 2185

✓	Rejected
=	Allowed


-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

Claims renumbered in the same order as presented by applicant
 CPA
 T.D.
 R.1.47

CLAIM		DATE								
Final	Original	09/14/2009								
97	109	=								
98	110	=								
99	111	=								
100	112	=								
101	113	=								
102	114	=								
103	115	=								
104	116	=								
105	117	=								
106	118	=								

Search Notes 	Application/Control No. 11250238	Applicant(s)/Patent Under Reexamination CONLEY, KEVIN M.
	Examiner NGOC V DINH	Art Unit 2185

SEARCHED			
Class	Subclass	Date	Examiner
711	103	9/14/2009	ND

SEARCH NOTES		
Search Notes	Date	Examiner
Limited classified search of Class/subclass. East text search w/o classified/search. See printout.	9/14/2009	ND
Consulted Mano Padmanabhan (101)	08/20/09	ND
Consulted Kevin Verbrugge (AU 2189).	10/23/08	ND
Consulted Kevin Ellis.	10/29/08	ND
Consulted Kevin Ellis.	9/24/2009	ND

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner
711/103, 711/113, 711/115	Intereference search EAST.	9/14/2009	ND

--	--

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: **Mail** **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
 or **Fax** **(571)-273-2885**

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

66785 7590 06/01/2010

DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION
 505 MONTGOMERY STREET
 SUITE 800
 SAN FRANCISCO, CA 94111

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

SVETLANA STELLMACH	(Depositor's name)
FILED VIA EFS	(Signature)
AUGUST 31, 2010	(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/250,238	10/13/2005	Kevin M. Conley	0084567-156US2	7727

TITLE OF INVENTION: PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	\$1510	\$300	\$0	\$1810	09/01/2010

EXAMINER	ART UNIT	CLASS-SUBCLASS
DINH, NGOC V	2185	711-103000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
 Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
 "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. Use of a Customer Number is required.

2. For printing on the patent front page, list
 (1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
 (2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1. DAVIS WRIGHT
2. TREMAINE LLP
3. _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY AND STATE OR COUNTRY)

SANDISK CORPORATION

MILPITAS, CA

Please check the appropriate assignee category or categories (will not be printed on the patent): Individual Corporation or other private group entity Government

4a. The following fee(s) are submitted:

- Issue Fee
- Publication Fee (No small entity discount permitted)
- Advance Order - # of Copies _____

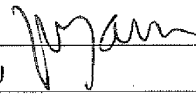
4b. Payment of Fee(s): (Please first reapply any previously paid issue fee shown above)

- A check is enclosed.
- Payment by credit card. Form PTO-2038 is attached. via EFS
- The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number 04-0258 (enclose an extra copy of this form).

5. Change in Entity Status (from status indicated above)

- a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
- b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature 
 Typed or printed name PHILIP YAU

Date AUGUST 31, 2010
 Registration No. 32,892

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Electronic Patent Application Fee Transmittal

Application Number:	11250238			
Filing Date:	13-Oct-2005			
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY			
First Named Inventor/Applicant Name:	Kevin M. Conley			
Filer:	Philip Yau/Svetlana Stellmach			
Attorney Docket Number:	0084567-156US2			
Filed as Large Entity				
Utility under 35 USC 111(a) Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Utility Appl issue fee	1501	1	1510	1510
Publ. Fee- early, voluntary, or normal	1504	1	300	300

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				1810

Electronic Acknowledgement Receipt

EFS ID:	8324974
Application Number:	11250238
International Application Number:	
Confirmation Number:	7727
Title of Invention:	PARTIAL BLOCK DATA PROGRAMMING AND READING OPERATIONS IN A NON-VOLATILE MEMORY
First Named Inventor/Applicant Name:	Kevin M. Conley
Customer Number:	66785
Filer:	Philip Yau/Svetlana Stellmach
Filer Authorized By:	Philip Yau
Attorney Docket Number:	0084567-156US2
Receipt Date:	31-AUG-2010
Filing Date:	13-OCT-2005
Time Stamp:	11:34:02
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$1810
RAM confirmation Number	8508
Deposit Account	040258
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
-----------------	----------------------	-----------	-------------------------------------	------------------	------------------

1	Issue Fee Payment (PTO-85B)	SNDK156US2_Issue_Fee_Transmittal_8-31-10.pdf	85197 918992b1b0f39e08547b40f6568b720f4380271	no	1
Warnings:					
Information:					
2	Fee Worksheet (PTO-875)	fee-info.pdf	32075 4d091ed4c0b036b6bf899d98b94399a443fc4a	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			117272		
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P. O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/250,238	10/19/2010	7818490	0084567-156US2	7727

66785 7590 09/29/2010
DAVIS WRIGHT TREMAINE LLP - SANDISK CORPORATION
505 MONTGOMERY STREET
SUITE 800
SAN FRANCISCO, CA 94111

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment is 0 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (<http://pair.uspto.gov>).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Application Assistance Unit (AAU) of the Office of Data Management (ODM) at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site <http://pair.uspto.gov> for additional applicants):

Kevin M. Conley, San Jose, CA;