United States Patent [19]
Karim

[11] **Patent Number: 4,589,112**
[45] **Date of Patent: May 13, 1986**

[54] SYSTEM FOR MULTIPLE ERROR DETECTION WITH SINGLE AND DOUBLE BIT ERROR CORRECTION

[75] Inventor: **Faraydon O. Karim,** Vestal, N.Y.

[73] Assignee: **International Business Machines Corporation,** Armonk, N.Y.

[21] Appl. No.: **574,221**

[22] Filed: **Jan. 26, 1984**

[51] **Int. Cl.**^{4} **G06F 11/10**
[52] **U.S. Cl.** **371/37;** 371/39
[58] **Field of Search** 371/37, 39, 40

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,622,982	11/1971	Clark, Jr. et al.	371/37
3,685,014	8/1972	Hsiao et al.	371/37
3,714,629	1/1973	Hong et al.	371/37 X
4,030,067	6/1977	Howell et al.	371/37
4,117,458	9/1978	Burghard et al.	371/37

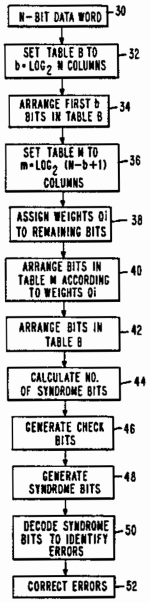
Primary Examiner—Charles E. Atkinson

Attorney, Agent, or Firm—Mark Levy

[57] ABSTRACT

A system for detecting multiple errors that may occur during transfer of data and for correcting up to two of these errors simultaneously. The system has a component for calculating a number of check bits associated with the data word. Also provided is a component for grouping all data bits into base groups and multiple groups, the sum of the number of base groups and multiple groups being equal to the number of check bits. Up to two weights are assigned for each data bit. The system distributes the data bits among the groups according to the weights assigned thereto. Also provided is a component for generating a check bit for each of the groups and for padding the data word with the check bits to form an appended data word. A generator creates a predetermined number of syndrome bits, the number being the number of check bits. Finally, a decoder is provided for decoding the syndrome bits to identify the erroneous bits in the data word.

5 Claims, 3 Drawing Figures


N-BIT DATA WORD
30
SET TABLE B TO b=LOG$_2$ N COLUMNS
32
ARRANGE FIRST b BITS IN TABLE B
34
SET TABLE M TO m=LOG$_2$ (N-b+1) COLUMNS
36
ASSIGN WEIGHTS Oi TO REMAINING BITS
38
ARRANGE BITS IN TABLE M ACCORDING TO WEIGHTS Oi
40
ARRANGE BITS IN TABLE B
42
CALCULATE NO. OF SYNDROME BITS
44
GENERATE CHECK BITS
46
GENERATE SYNDROME BITS
48
DECODE SYNDROME BITS TO IDENTIFY ERRORS
50
CORRECT ERRORS
52

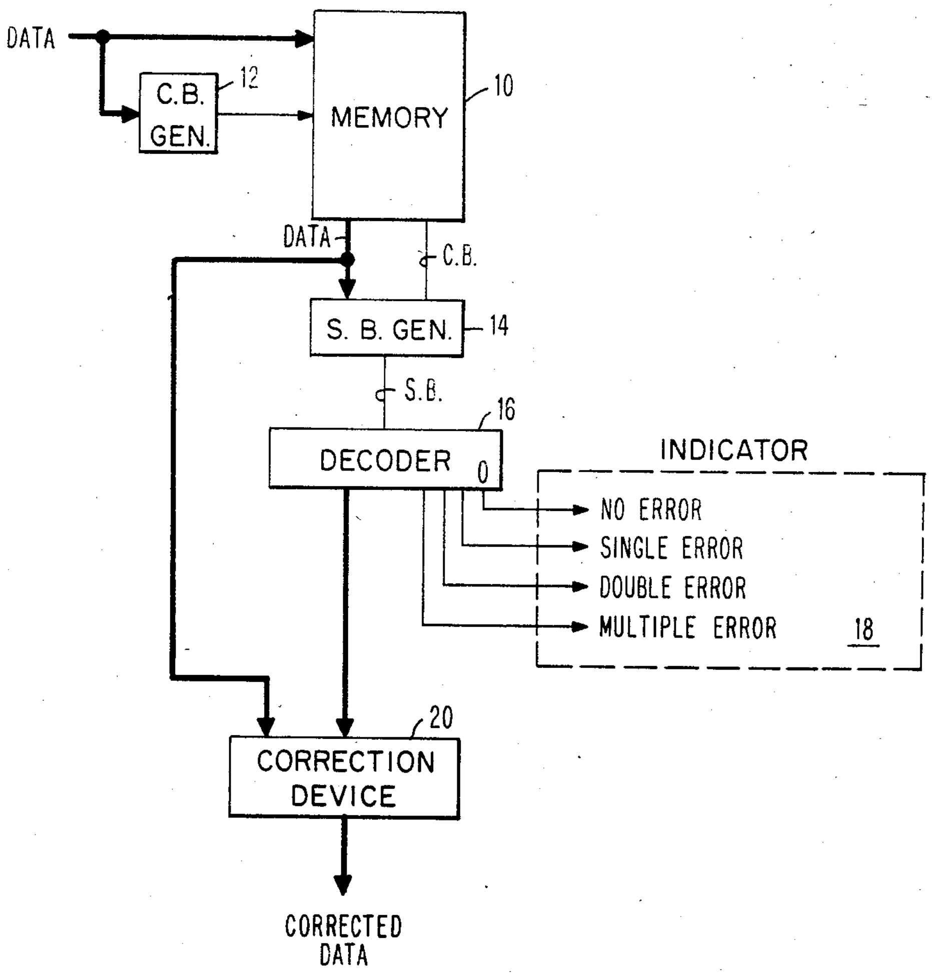

DATA
C.B.
GEN.
12
MEMORY
10
DATA
C.B.
S. B. GEN.
14
S.B.
16
DECODER
0
INDICATOR
NO ERROR
SINGLE ERROR
DOUBLE ERROR
MULTIPLE ERROR
18
20
CORRECTION
DEVICE
CORRECTED
DATA

FIG. 1

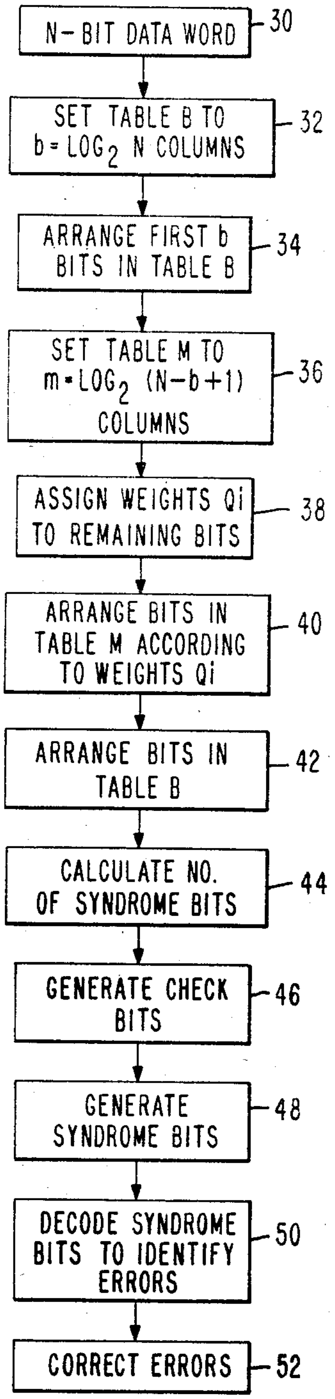

N-BIT DATA WORD
30
SET TABLE B TO
$b = LOG_2$ N COLUMNS
32
ARRANGE FIRST b
BITS IN TABLE B
34
SET TABLE M TO
$m = LOG_2 (N-b+1)$
COLUMNS
36
ASSIGN WEIGHTS Qi
TO REMAINING BITS
38
ARRANGE BITS IN
TABLE M ACCORDING
TO WEIGHTS Qi
40
ARRANGE BITS IN
TABLE B
42
CALCULATE NO.
OF SYNDROME BITS
44
GENERATE CHECK
BITS
46
GENERATE
SYNDROME BITS
48
DECODE SYNDROME
BITS TO IDENTIFY
ERRORS
50
CORRECT ERRORS
52

FIG. 2

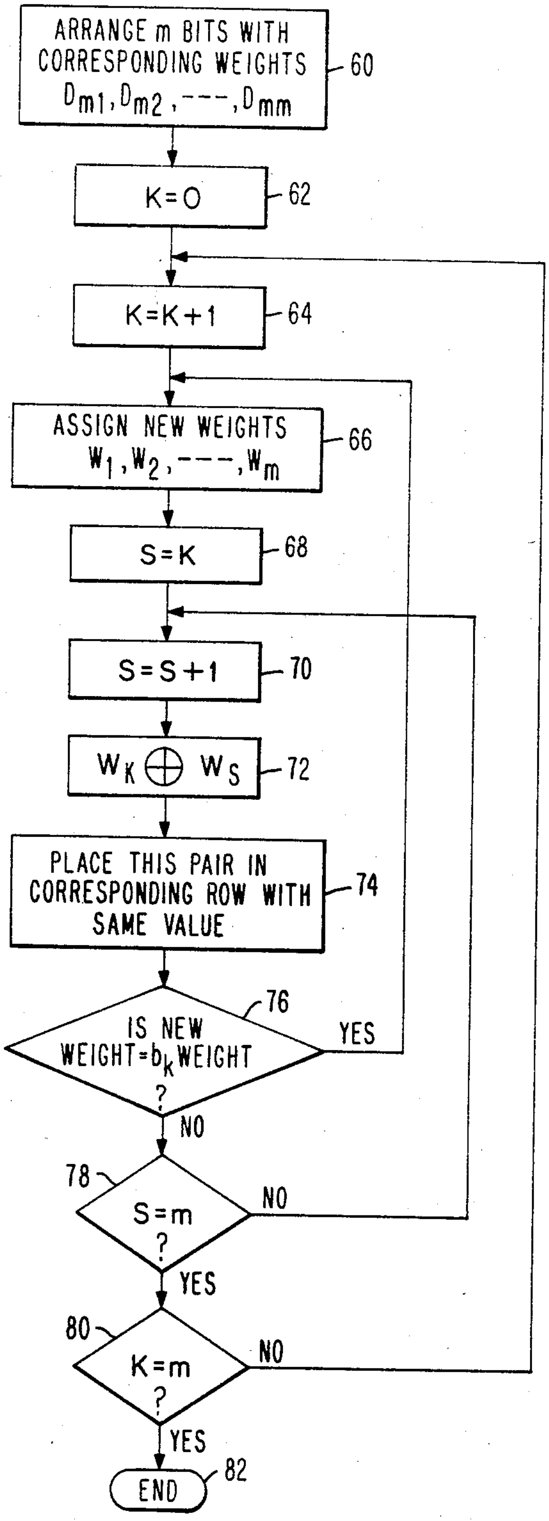

ARRANGE m BITS WITH CORRESPONDING WEIGHTS $D_{m1}, D_{m2}, ---, D_{mm}$
60
K=O
62
K=K+1
64
ASSIGN NEW WEIGHTS $W_1, W_2, ---, W_m$
66
S=K
68
S=S+1
70
$W_K \oplus W_S$
72
PLACE THIS PAIR IN CORRESPONDING ROW WITH SAME VALUE
74
IS NEW WEIGHT=b_KWEIGHT ?
76
YES
NO
78
S=m ?
NO
YES
80
K=m ?
NO
YES
END
82

FIG. 3

SYSTEM FOR MULTIPLE ERROR DETECTION WITH SINGLE AND DOUBLE BIT ERROR CORRECTION

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a system for detecting and correcting errors in data transfer and, more particularly, to a system for detecting single or multiple bit errors and for correcting single or double bit errors.

2. Description of the Prior Art

In any digital system where data is transmitted, one or more of the data bits in each data word or message may be received in error. This has been a problem from the time data processing systems were first invented.

As more sophisticated data processing operations are performed, involving more complex equipment, there is a greater need for systems to detect and correct multiple errors in data transfers. For example, operations such as merging of files, sorting of data within files, numerical/-statistical analyses, complex data handling procedures and word processing operations require increased reliability in data transfer. In the field of telecommunications and telemetry, error rates tend to increase when data is transmitted over analog lines at high baud rates. If data errors occur and are undetected, valuable information and system operation itself may be affected. Thus, error detecting and correcting features are not only advantageous, they are required to improve system integrity.

In response to the problem of error generation during data transfers, systems have been developed to detect such errors. One of the earliest methods for detecting errors was the parity check code. A binary code word has odd parity if an odd number of its digits are 1's. For example, the number 1011 has three 1 digits and therefore has odd parity. Similarly, the binary code word 1100 has an even number of 1 digits and therefore has even parity.

A single parity check code is characterized by an additional check bit that is added to each word to generate either odd or even parity. An error in a single digit or bit in a data word would be discernible since the parity check bit associated with that data word would then be reversed from what is expected. Typically, a parity generator adds the parity check bit to each word before transmission. This technique is called padding the data word. At the receiver, the digits in the word are tested and if the parity is incorrect, one of the bits in the data word is considered to be in error. When an error is detected at a receiver, a request for a repeat transmission can be given so that the error can be corrected. It should be noted that only errors in an odd number of digits can be detected with a single parity check, since an even number of errors results in the parity expected for a correct transmission. Moreover, it should be noted that the specific bit in error cannot be identified by the parity check procedure as hereinabove described.

A more sophisticated error detection system was later devised. Data words of a fixed length of bits were grouped into blocks of a fixed number of data words each. Parity checks were then performed between different data words as well as for each individual data word. The block parity code detected many patterns of errors and could be used not only for error detection, but also for error correction when an isolated error occurred in a given row and column of the matrix. While these geometric codes were an improvement over parity check bits per se, they still could not be used to detect errors that were even in number and symmetrical in two dimensions.

After parity check codes and geometric codes were devised, a code was invented by Hamming, after whom it is named. The Hamming code is a system of multiple parity checks that encodes data words in a logical manner so that single errors can be not only detected but also identified for correction. A transmitted data word used in the Hamming code consists of the original data word and parity check digits appended thereto. Each of the required parity checks is performed upon specific bit positions of the transmitted word. The system enables the isolation of an erroneous digit, whether it is in one of the original data word bits or in one of the added parity check bits.

If all the parity check operations are performed successfully, the data word is assumed to be error free. If one or more of the check operations is unsuccessful, however, the single bit in error is uniquely determined by decoding so-called syndrome bits, which are derived from the parity check bits. It should be noted once again that only single bit errors are detected and corrected by use of the conventional Hamming code. Double bit errors, although detectable by the Hamming code, are not correctable.

The Hamming code is only one of a number of codes, generically called error correcting codes (ECC's). Codes are usually described in mathematics as closed sets of values that comprise all the allowed number sequences in the code. In data communications, transmitted numbers are essentially random data patterns which are not related to any predetermined code set. The sequence of data, then, is forced into compliance with the code set by adding to it at the transmitter, as hereinabove mentioned. A scheme has heretofore been developed to determine what precise extra string to append to the original data stream to make the concatenation of transmitted data a valid code. There is a consistent way of extracting the original data from the code value at the receiver and to deliver the actual data to the location where it is ultimately used. For the code scheme to be effective, it must contain allowed values sufficiently different from one another so that expected errors do not alter an allowed value such that it becomes a different allowed value of the code.

A cyclic redundancy code (CRC) consists of strings of binary data evenly divisible by a generator polynomial, which is a selected number that results in a code set of values different enough from one another to achieve a low probability of an undetected error. To determine what to append to the string of original data, the original string is divided as it is being transmitted. When the last data bit is passed, the remainder from the division is the required string that is added since the string including the remainder is evenly divisible by the generator polynomial. Because the generator polynomial is of a known length, the remainder added to the original string is also of fixed length.

At the receiver, the incoming string is divided by the generator polynomial. If the incoming string does not divide evenly, an error is assumed to have occurred. If the incoming string is divided by the generator polynomial evenly, the data delivered to the ultimate destina-

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you’re on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm’s cloud-native docket research platform finds what other services can’t. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.