Microsoft

Broadcast Technologies White Paper

MSBDN Receiver Board Implementation

This paper contains requirements and suggestions for implementing hardware and software to support Microsoft Broadcast Data Network (MSBDN) streams emanating from a network device. The paper describes the equivalent of the media access control (MAC) layer of the network protocol stack.

Contents

Introduction

The audience for this paper is network card manufacturers and cryptographic equipment suppliers. While the details and diagrams in this paper describe aspects of an implementation to support the Digital Satellite System (DSS), a direct broadcast satellite network, some of these techniques are also appropriate for implementation on other types of networks, such as corporate Ethernet local area networks (LANs). The details of the other implementations are necessarily different from those used in the DSS implementation.

For a more general description of how the components described in this paper fit into a digital system of broadcast network clients, see Chapter 4, “Network Receiver and MPEG Display Subsystems,” in the Microsoft Broadcast Data Network (MSBDN) Device Driver Kit.

Data Flow Overview

The following figure shows how data flows through a MSBDN DSS satellite receiver card.

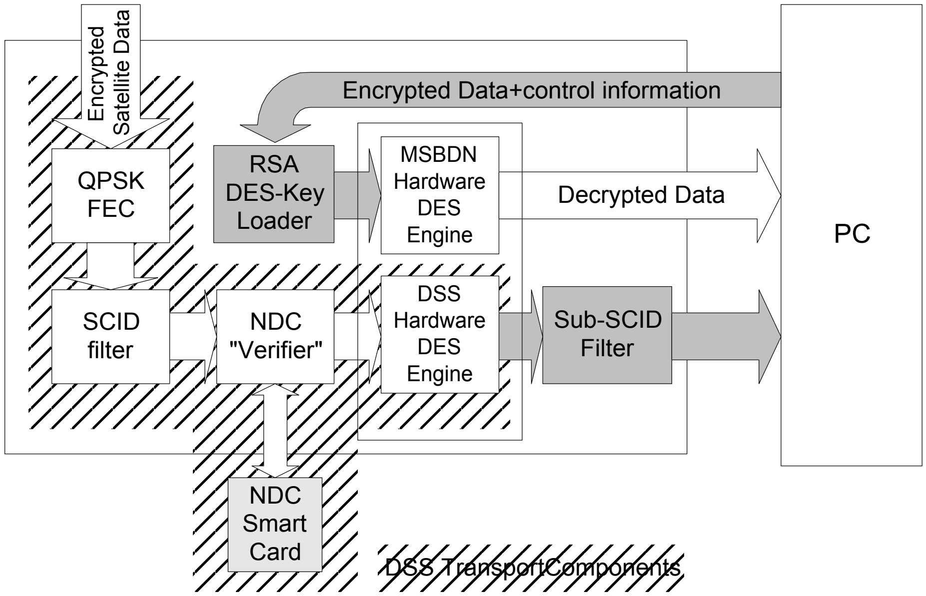

Figure 1. Data flow through the MSBDN DSS satellite receiver

A MSBDN receiver card consists of the normal circuitry necessary to tune, decode, demultiplex, error correct, and apply DSS access control to the received signal. The DSS transport filters and delivers the DSS service channel identifiers (SCIDs) to the input of the sub-SCID filter. The packets selected by the sub-SCID filter are passed to the computer through the computer bus interface (typically a PCI bus interface), where they are bus mastered into memory.

Note: Figure 1 does not show the optional ability to reassemble certain packet types in the receiver card, nor does it show received packets using Cyclic Redundancy Checks (CRCs).

The receiver card driver selectively programs bus mastering of packets through the MSBDN data encryption standard (DES) decryptor, which in some receiver designs is combined with the DSS DES decryptor. The MSBDN DES decryptor decrypts the packets and writes them back to memory.

Typical DSS Requirements

The DSS-specific information contained in this paper is insufficient to complete a DSS-compatible satellite receiver. For detailed information on the system requirements and internal workings of a DSS-compatible receiver, contact DIRECTV.

In the DSS environment, a transponder can send information streams at up to 30 megabits per second. This stream of information contains substreams, called *SCIDs*, analogous to packet identifiers (PIDs) found in the digital video broadcasting (DVB) system. The different SCIDs carry information such as audio, video, and data, which are separated to make decoding easier in a typical set-top device.

Bus and Computer Software Interface

The satellite receiver card should have bus mastering capability, and the receiver card driver should have programmatic access to various functions of the card through the bus interface.

In some designs, this interface may use I/O address mapping; in others, a memory buffer may be used to pass register contents between the computer CPU and the various chips on the receiver card. The choice depends largely on which system bus you design the adapter for, Industry Standard Architecture (ISA) or PCI.

You can achieve varying amounts of control over DSS reception through hardware on the receiver card, subject to overall design goals. Because a unique receiver card driver exists for each card design, a manufacturer can choose to decrease hardware functionality for a card, and increase the complexity of the software driver for that card.

Tuning

The satellite receiver card must perform all normal DSS functions, including tuning functions, QPSK demodulation, and forward error correction. The host CPU can control these various functions by programming transport registers through the computer bus interface. The receiver card driver can check status for these operations through the computer bus interface.

NDS Access Control

The satellite receiver card must perform all required News Data Systems (NDS) access control functions.

In typical receiver card designs, an on-board CPU handles the verification code for access control by communicating with the receiver card driver running on the computer CPU through the computer bus interface.

DIRECTV has mandated that data streams be encrypted using standard NDS access control methods. The NDS decryption process must occur before any CRC or sub-SCID filtering can take place.

The MSBDN access control method is completely independent from NDS access control and occurs after the data is written to computer memory.

SCID Filtering

The satellite receiver card must be able to filter at least five SCIDs. This figure is the minimum requirement and is merely adequate for receiving DSS signals. Because the personal computer is a more versatile device than a typical set-top device, the hardware design of the satellite receiver card should be capable of filtering more than five SCIDs.

The satellite receiver card must be able to support a total bandwidth into the computer of 30 megabits per second for the DSS network.

PCI Bus Mastering

Each SCID must be bus mastered into a separate buffer in computer memory. This rule includes data SCIDs, which may be subject to filtering, CRC checking, and other operations; for more information on these operations, see following sections in this paper.

Buffers must be capable of being at least as long as 64K bytes, and as small as 4K bytes. The minimum granularity required for buffer sizes is 128 bytes.

An important consideration in designing PCI bus mastering hardware is that DSS Motion Pictures Experts Group (MPEG) packets should not be formed as multiples of 4 bytes. Further, the packet transfers are unaligned and may begin on any physical memory address that is valid and page-locked.

When writing DSS MPEG packets to memory, only the 127 bytes of data should be written. No padding is allowed. All received bytes must be written when they arrive. After a set time-out determined by the receiver card driver running on the computer CPU, any partially filled large buffers must be made available to the computer, even if they are not completely filled. The resulting MPEG data is considered a stream by the computer and passed in large blocks directly to the MPEG decoder. Passing individual 127-byte blocks to the MPEG decoder causes unacceptable system performance degradation due to interrupt overhead.

Ring Buffers

When a data buffer fills, the satellite receiver card must generate an interrupt and switch to filling another prespecified buffer.

For each high-speed SCID, defined as streams that are 2 megabits per second or faster, you should provide a method to implement at least eight buffers. At any one time, the satellite receiver needs to process two high-speed SCIDs.

Note: It is best to support more than two SCIDs—over time, individual customers will likely want to use more than one high-speed data service. Because high-speed SCIDs are faster than 2 megabits per second, with current satellites it is possible to have up to 15 high-speed SCIDs per transponder.

For each low-speed SCID (that is, each SCID under 2 megabits per second), you should provide a method to implement at least three buffers. At any one time, the satellite receiver needs to be able to process five low-speed SCIDs.

High-speed SCIDs and low-speed SCIDs differ only in the expected sustained bandwidth. Burst bandwidth is identical—the full rate of the satellite. High- and low-speed SCIDs are distinguished in this paper only to ensure receiver boards can maintain expected rates in regards to interrupt latency, buffer setup, and on-board CPU usage. All SCIDs can be high-speed.

At minimum, the satellite receiver should support bus mastering of satellite data into ping-pong buffers (two active buffers) on a per-SCID basis. When either buffer fills, the satellite receiver must generate an interrupt. The host CPU must be able to change the address of the nonactive buffer while the active buffer is being filled without affecting the filling of the active buffer.

Ideally, the satellite receiver card should accept a buffer description list from the receiver card driver for each SCID that contains information about the location and size of an arbitrary number of buffers; the receiver card fills these buffers consecutively and generates interrupts when each is full. The receiver card driver should be able to change this buffer list on the fly during typical operation as more or fewer buffers become available.

Buffer Time-out

While filling any buffer for any specified SCID, the satellite receiver card must be capable of being forced to change to the next buffer. After the active buffer is changed, the satellite receiver must generate an interrupt (as if the buffer were full) and must allow queries on the number of packets that

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.