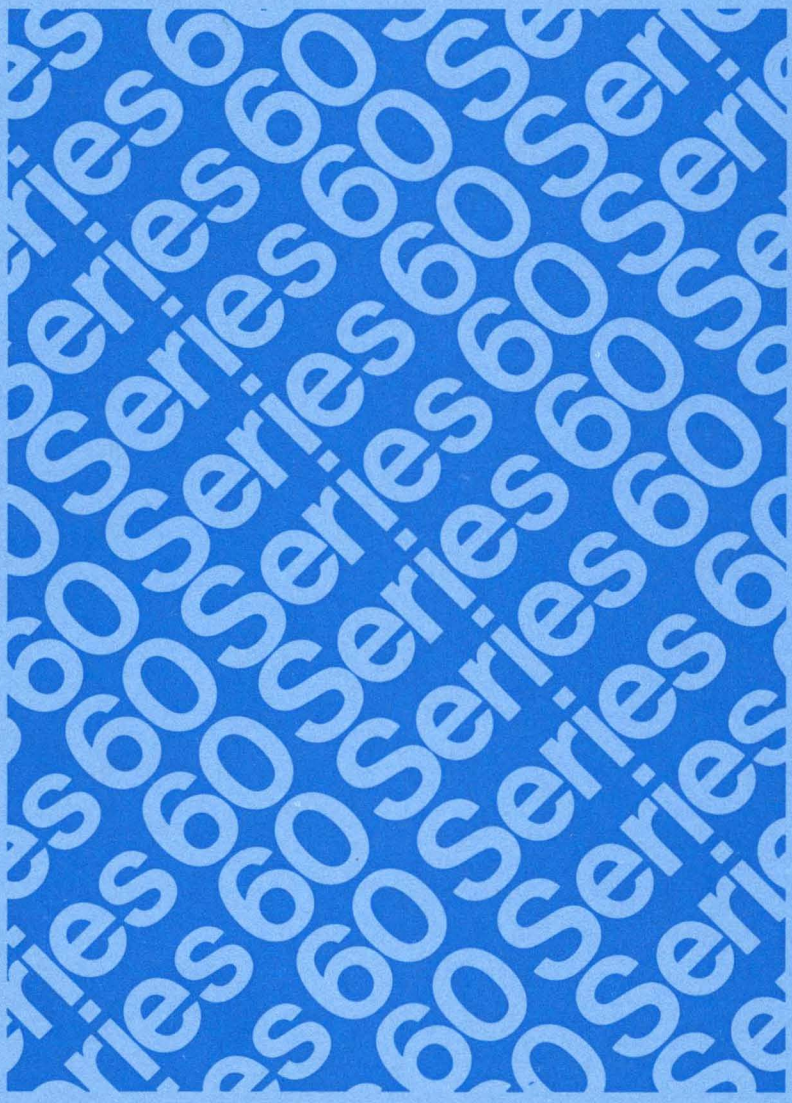
Honeywell

HONEYWELL CONFIDENTIAL & PROPRIETARY



LEVEL 6
HARDWARE
TYPE MSC9102 STORAGE MODULE ADAPTER MANUAL

SERIES 60 (LEVEL 6)

TYPE MSC9102
STORAGE MODULE ADAPTER MANUAL

HONEYWELL CONFIDENTIAL & PROPRIETARY

SUBJECT

Functional Operation of the Type MSC9102 Storage Module Adapter, the Adapter/Device Software, and the Medium Formats

SPECIAL INSTRUCTIONS

This manual has been revised to the -200 level. It supersedes all previous issues.

ORDER NUMBER

FN80, Rev. 1

August 1978

Honeywell

RECORD OF REVISIONS

REVISION	DATE	AUTHORITY	AFFECTED PAGES
-100	March 1977	--	--
-200	March 1978	BLCD87349	Entire manual updated

Hardware Publications, M&TO, Billerica, MA 01821

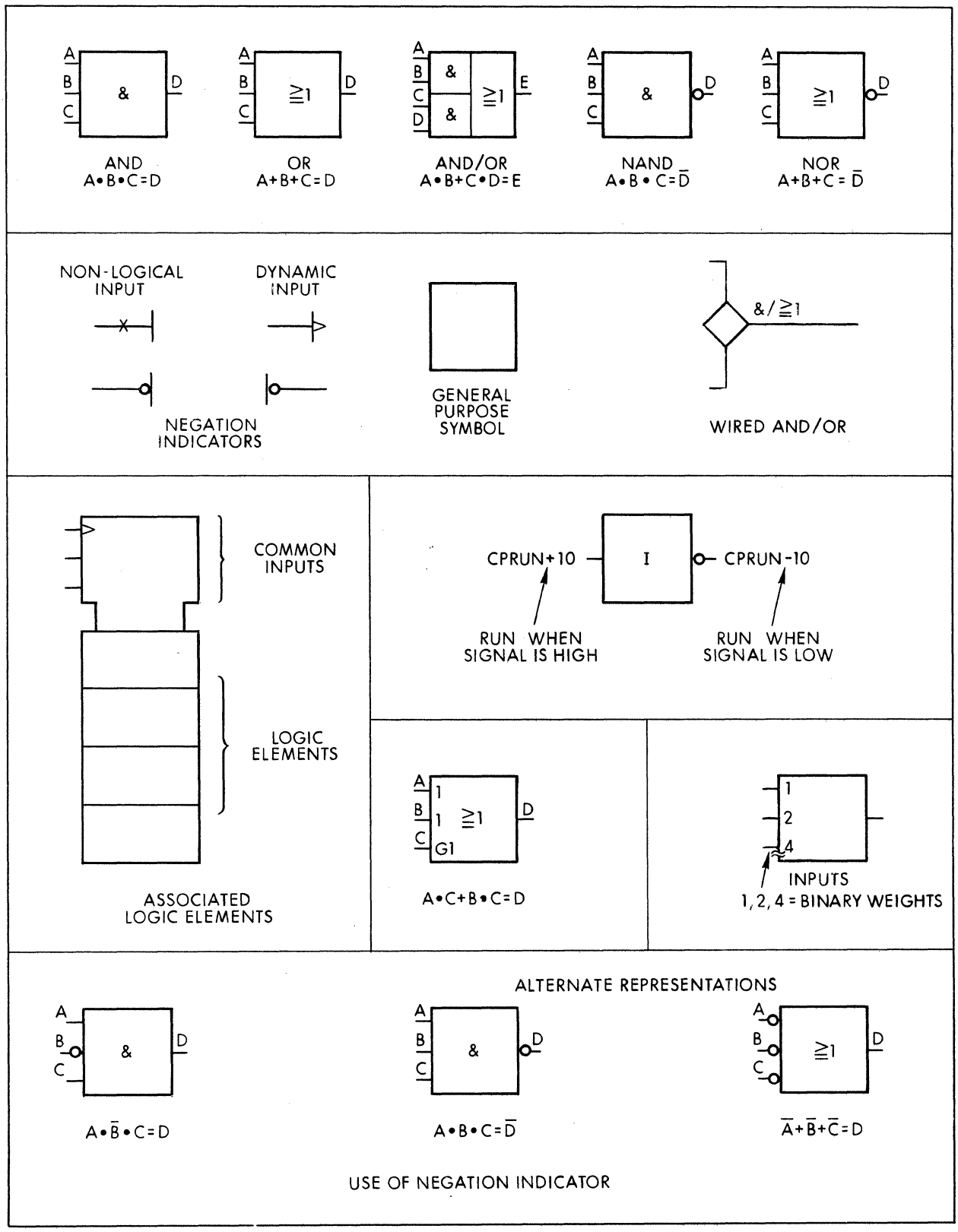
0429/FN80

File No.: 71010429-200

CONTENTS

ILLUSTRATIONS

TABLES


A
B
C
D
&
≧1
E
AND
A•B•C=D
OR
A+B+C=D
AND/OR
A•B+C•D=E
NAND
A•B•C=D̄
NOR
A+B+C=D̄
NON-LOGICAL INPUT
DYNAMIC INPUT
NEGATION INDICATORS
GENERAL PURPOSE SYMBOL
&/≧1
WIRED AND/OR
COMMON INPUTS
LOGIC ELEMENTS
ASSOCIATED LOGIC ELEMENTS
CPRUN+10
I
CPRUN-10
RUN WHEN SIGNAL IS HIGH
RUN WHEN SIGNAL IS LOW
1
G1
A•C+B•C=D
2
4
INPUTS
1, 2, 4 = BINARY WEIGHTS
ALTERNATE REPRESENTATIONS
A•B̄•C=D
A•B•C=D̄
Ā+B̄+C̄=D
USE OF NEGATION INDICATOR

Section 1
Introduction

This product manual describes (1) the functionality and operation of the Type MSC9102 Storage Module Adapter (SMA), (2) the adapter/device software, and (3) the medium formats. Designed for use with the Type MSC9101/9102 Medium Performance Disk Controller (MPDC), the SMA/MPDC combination has a unique firmware load which resides within a separate Read Only Store (ROS) adapter board (BD2DT1). For this reason, firmware for both the SMA and the MPDC is contained in this manual. Operational theory contained in this manual is designed to acquaint the reader with the major functional logic areas of the SMA at the major and intermediate levels. Refer to the Storage Module Adapter Reference Manual (Order No. FN81) for detailed logic block diagrams and physical location information.

1.1 GENERAL DESCRIPTION

The Type MSC9102 Storage Module Adapter is a solid-state module (board BX4DSM) used with the Type MSC9101 Medium Performance Disk Controller (MPDC) to operate up to four MPI Storage Module Devices (see Table 1-1) in the Model 33 or 43 configurations of the Series 60 Level 6 computer system. Figure 1-1 illustrates the attachment configuration of the storage module subsystem. Note that only one Type MSC9102 Storage Module Adapter (hereafter called the adapter) can be connected to the MPDC.

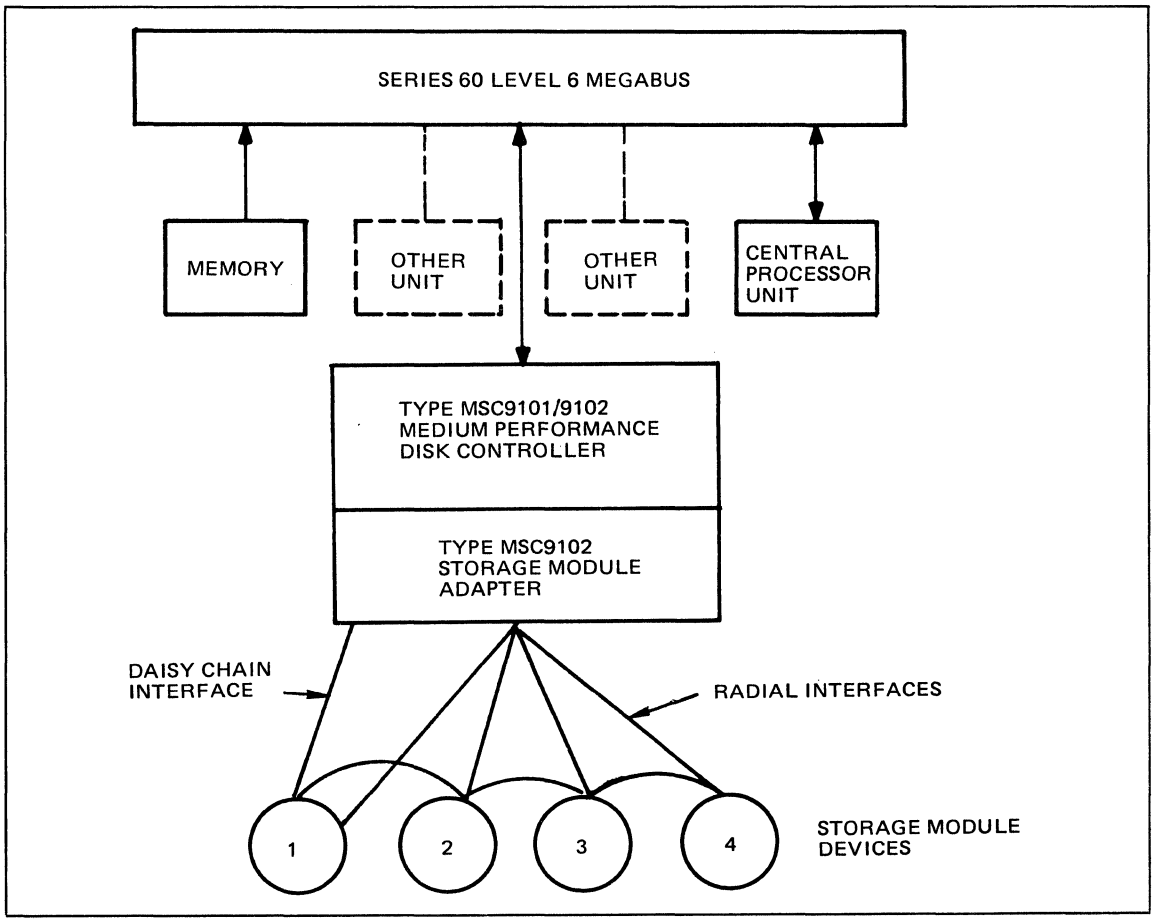


Figure 1-1 Storage Module Subsystem Configuration

The adapter consists of dual-in-line packages (DIPs) mounted on a triple-sized Series 60 Level 6 package (manufactured using printed wiring assembly techniques). The adapter is mounted on the MPDC package by its two 25-pin in-line connectors. The storage module disk device (hereafter called the disk device) is cabled to the adapter and contains its own source of device power.

Table 1-1 Attachable Storage Module Devices

TYPE	STORAGE CAPACITY (UNFORMATTED)	DESCRIPTION
MSU9101	40M Bytes	Storage module device with cabinet (includes room for an additional device drawer).
MSU9102	80M Bytes	Storage module device with cabinet (includes room for an additional device drawer).
MSU9103	150M Bytes	Storage module device with cabinet (capable of housing one device only).
MSU9104	300M Bytes	Storage module device with cabinet (capable of housing one device only).
MSU9105	40M Bytes	Storage module device without cabinet (drawer for add-on to MSU9101 or MSU9102).
MSU9106	80M Bytes	Storage module device without cabinet (drawer for add-on to MSU9101 or MSU9102).

1.2 FUNCTIONAL DESCRIPTION

The adapter provides the device-specific hardware which, in conjunction with the adapter/MPDC firmware (located on the MPDC board), enables MPDC control of the positioning, head selection, reading and writing functions performed by the disk device.

1.2.1 Medium Performance Disk Controller (MPDC)

The MPDC performs the following general purpose control functions:

- Execution of the Series 60 Level 6 Megabus network sequences
- Command decoding
- Status and control register storage
- Direction of the general flow of command execution.

1.2.2 Storage Module Adapter

The adapter performs the following device-specific functions:

- Device interface dialog control
- Data recovery not performed by the device
- EDAC check generation/verification
- Cyclic check generation/verification
- Sync byte detection
- Device state monitoring
- Serial-to-parallel and parallel-to-serial conversions.

1.2.3 Storage Module Device

The device performs the following basic functions:

- Receives and generates control signals for device operation
- Positions the device read/write heads on designated cylinders
- Writes data received in serial form from the adapter on the disk surface
- Reads data from the disk surface and sends it to the adapter in serial form
- Performs all digital-to-analog and analog-to-digital conversions required for read/write operations
- Monitors device operations and reports status and detected faults to the MPDC via the adapter
- Address mark generation/detection.

1.3 OPERATIONAL SUMMARY

Devices attached to the MPDC/adapter are software addressable via channel numbers. Each individual device has two channel numbers assigned, differing from each other only in the low order bit position, called the direction bit. When an IOLD instruction is issued to a storage module device, the direction bit of the channel number of the output address specifies whether the command is an input or an output data transfer. For all other commands the direction bit is ignored. Figure 1-2 shows the composition of the channel number. Bits 0 - 6 are assigned at system installation via the setting of configuration switches located on the MPDC board (see MPDC manual for switch settings).

Software visibility of the attached devices is such that they are generally independent of each other. For example, operations on one disk are independent of any activity on another disk except that the MPDC initiation of a command sequence addressed to one device may be stalled (the command sequence is accepted but not initiated) while the MPDC is busy servicing another device.

The MPDC provides a single level of simultaneity. Because of disk subsystem command sequencing, a nonbusy channel can accept any command addressed to it, and the MPDC can accept any command addressed to a device while another device is executing a data transfer. The accepted command is not invoked until the present data transfer is completed. Also, after completion of each data transfer operation, any stored control commands are initiated prior to the execution of any subsequent data transfer commands. To avoid having one device dominate adapter usage, devices (channels) are serviced on a rotating priority basis. If fewer than four devices are attached to the adapter, it responds only to channel numbers associated with the installed devices.

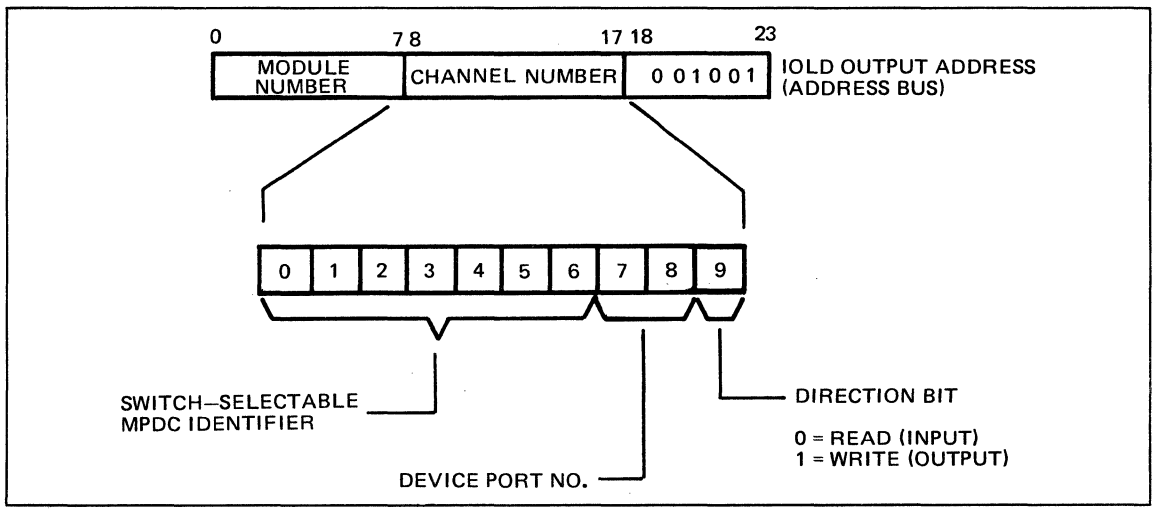


Figure 1-2 Channel Number Composition

The MPDC contains a set of software-loadable scratch pad registers for each device; these registers contain the parameters required for device operation (see Section III for details). In addition to range and address registers, there are two configuration registers which contain record location and identification information, and a task register which contains command codes. To perform a specific operation, software first loads the address, range, and configuration registers. The task register is loaded last and specifies the operation to be performed. Although commands addressed to a nonbusy device are always accepted, their execution may be delayed as previously described. All commands addressed to a busy device are rejected (NAK response on Megabus) except Output Control Word.

The location of a desired record on the disk is defined by configuration words A and B which are stored in the scratch pad memory. These words contain an image of the sector ID field on which an operation is to be performed. Configuration word A contains the cylinder number, while configuration word B contains the track number and sector number. These are used as the firmware seek argument for seek operations. The sector number field of configuration word B is incremented by one at the end of each error-free data sector that is written on or read from the disk surface. This enables software to perform sequential read or write operations on the same track without reloading the configuration words for every operation. Note that during search-and-read or search-and-write operations, records are processed in numerical order, and not necessarily the order in which they are physically recorded. An extended operation can, therefore, be used to advantage on files that are recorded using a sector identifier (ID) interleaving technique as well as on files that are recorded in numerical order.

Multisector (extended) operation is automatically continued on the next consecutive track if the last logical record on the track was transferred and the range has not expired. In this case the configuration words are modified to reflect sector zero of the next track. If cylinder completion is also detected, the configuration words are modified and the MPDC initiates a seek to track zero, sector 0 of the next consecutive cylinder. The MPDC can service other channels during the seek latency period. Interleaving cannot be used in conjunction with Rotational Position Sensing (RPS).

When multitracking and/or cylinder linking is used, the assigned sector numbers must be 0 to 63 (for 256 byte sectors). Note that the ID fields are software selectable and do not have to be related to the physical track or sector, even though they commonly are.

If, during a read data operation, an uncorrectable read error is encountered in a data field, the sector number field of configuration register B is not incremented. The command is terminated and the sector number field references the sector in which the error was encountered.

During output operations, the adapter accepts 8-bit bytes of information from the MPDC and serializes the information for transfer to an addressed, attached device via the bidirectional read/write data line. Clock synchronization is provided by clock signals from the device. The adapter generates Error Detection Codes (EDC) (cyclic redundancy check characters) for the sector ID fields, and Error Detection and Correction (EDAC) codes for data fields. These error codes are checked during read operations to ensure data integrity.

During input operations the adapter accepts serial read information from the device (together with synchronizing clock signals) and assembles it in 8-bit bytes for subsequent transfer to the MPDC. The adapter is capable of recognizing address marks and sync bytes, and of comparing sector IDs being read with a sought-for sector ID (previously loaded in the adapter by firmware). Error detection codes (EDC and EDAC) which were generated during the output operation are checked, and detected errors are reported to the MPDC for further processing.

During both input and output operations, the adapter stores both device-specific and adapter-specific commands. Device-specific commands are subsequently transferred to the device for decoding and execution, while adapter-specific commands are decoded within the adapter and employed to control adapter operations. Error detection logic monitors adapter operation and reports fault conditions to the MPDC. Cycle flow within the adapter is controlled by the adapter firmware, located on the MPDC board. See Section III of this manual for a detailed description of adapter cycle flow.

1.4 MEDIUM FORMATS

1.4.1 General Recording Characteristics

Each track on the disk contains 64 or 8 equal length sectors of 256 or 2304 bytes, respectively. There are 411 or 823 cylinders per surface, and 5 or 19 tracks (not counting a servo track) per cylinder. The last cylinder (411 or 823) is reserved for T&V usage. The total formatted device capacity varies from 33,669,120 to 288,221,184 bytes depending upon the device and format. The data encoding scheme is modified frequency modulation (MFM) recording.

1.4.2 Track Format

The track format for the recording medium is illustrated in Figure 1-3. Each sector is preceded by an address mark, while each sector ID and each data field is preceded by a sync word. The sector ID field is followed by Error Detection Code (EDC) bytes, and the data fields are followed by Error Detection and Correction (EDAC) bytes. Sector ID fields are software programmable and do not have to be numerically sequential (except when the Rotational Position Sensing (RPS) function is used, in which case the fourth byte of the sector ID field must be physical). For multilinking, the last sector must be 63 or greater.

When a track is formatted, 60 or 8 sectors are written beginning at the sector following the index mark. The sector IDs are specified by software and are extracted from memory during the format operation. Data fields are Zero-filled during formatting.

1.4.2.1 Gap 0

The field begins at the trailing edge of the index mark and consists of 16 bytes of Zeros.

1.4.2.2 Gap 1

The field begins a sector and is comprised of 30 bytes as follows:

- 16 bytes of Zeros
- 3 bytes of address mark
- 10 bytes of Zeros
- 1 sync byte consisting of the hex character 19.

1.4.2.3 Sector Identifier

This is a 4-byte field formatted as shown in Figure 1-3. The cylinder number is a 10-bit binary number (right-justified in bytes 1 and 2) which represents the logical cylinder number from 000 for the outermost cylinder to 410 or 822 for the innermost cylinder. The track number is a 5-bit binary number (right-justified in byte 3) which represents one of 5 or 19 possible data surfaces. The sector number is an 8-bit binary number (right-justified in byte 4) which represent the sector address.

1.4.2.4 Error Detection Code (EDC)

The two EDC bytes are generated within the adapter on the basis of a half-add algorithm implemented via the CRC logic. The EDC is generated for sector ID and sync bytes only and is written in the field following the sector ID field.

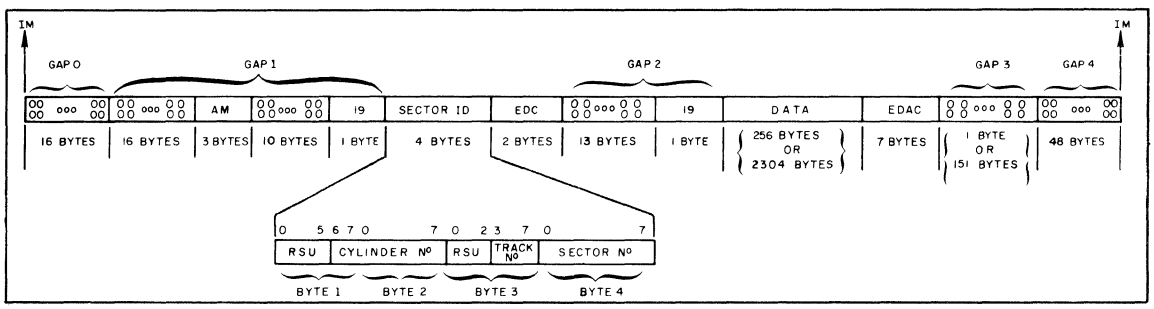


Figure 1-3 Track Format

1.4.2.5 Gap 2

The Gap 2 field begins following the last byte of the EDC field and is composed of 14 bytes as follows:

- 13 bytes of Zeros (including a 1-byte end-of-field pad)
- 1 sync byte consisting of the hex character 19.

1.4.2.6 Data Field

Data fields are either 256 or 2304 bytes long. Any write operation that does not write a complete data field results in the remainder of the field being Zero-filled. The Format Write command causes all the data fields on a track to be Zero-filled.

1.4.2.7 Error Detection and Correction (EDAC) Field

The seven EDAC bytes are generated within the adapter. The EDAC code is generated for data fields and sync bytes only.

1.4.2.8 Gap 3

The Gap 3 field begins following the last byte of the EDAC field and is composed of 1 byte (for 256-byte sectors) or 151 bytes (for a 2304-byte sector) extending to the next sector mark. This gap includes a 1-byte end-of-field pad.

1.4.2.9 Gap 4

The Gap 4 field begins following the last sector on the track and continues to the index mark, and consists of 48 bytes of Zeros.

1.4.3 Defective Track Handling

New disks may have a minimal number of defective areas. These areas, together with areas that may become defective during the life of the media, are accounted for by software. Software has the ability to recognize defective areas and reallocate read/ write space around the defective areas.

1.5 REFERENCE DOCUMENTS

The documents listed in Table 1-2 contain information that can aid in understanding the storage module adapter and the subsystem of which it is a part.

Table 1-2 Reference Documents

TITLE	PART NUMBER	ORDER NUMBER
Model 34/36 System Manual	71010200	FL35
Circuit Descriptions Reference Manual	71010206	FL47
Type MSC9101/9102 Medium Performance Disk Controller Manual	71010423	FM54
Type MSC9101/9102 Medium Performance Disk Controller Reference Manual	71010241	FM55
Power System Manual	71010290	FL34
Series 60 Level 6 Peripherals Manual	N/A	AT04
Series 60 Level 6 Minicomputer Handbook	N/A	AS22
Type MSC9102 Storage Module Adapter Reference Manual	71010430	FN81

Section 2
Theory of Operation

2.1 SOFTWARE

Software interface between the disk subsystem and the Megabus is accomplished via input, output, and diagnostic instructions. This manual describes only those instructions having specific interpretation for, or effect upon, the adapter and/or attached disk device. See the Medium Performance Disk Controller Manual (Order Number FM54) for a detailed description of all software instructions.

In order to accomplish read, write, and control operations, the specific sector on the disk surface must be accessed, the command code for the desired operation must be delivered to the adapter, and the direction of data transfer must be specified. These software-controlled parameters are placed in the MPDC scratch pad memory via the Output Configuration Word (A and B), Output Task Word, and the IOLD Output Address instructions, as described in the following subsections.

2.1.1 Configuration Words

Figure 2-1 illustrates the two configuration words required for control of the device. Together, these words specify the cylinder, track and sector for the desired record. These words are loaded into the scratch pad memory prior to execution of a seek or data transfer command. Under firmware control, the cylinder, track and sector information is subsequently transferred to the adapter and device.

2.1.2 Transfer Direction Control

The direction of data transfer is determined by bit 17 of the IOLD instruction (bit 17 is the direction bit of the channel number field). Adapter firmware examines this bit to determine whether the device command specified by the task word is to be executed as an input or an output operation. Firmware then enters the appropriate routine for command execution.

2.1.3 Output Task Word (Figure 2-2)

Disk device operations are primarily controlled by the adapter, which is in turn controlled by adapter firmware routines (Section III). The device operations performed are specified by the content of the command code field of the Output Task Word instruction. This instruction loads task words into scratch pad memory for access by adapter firmware. The channel enters

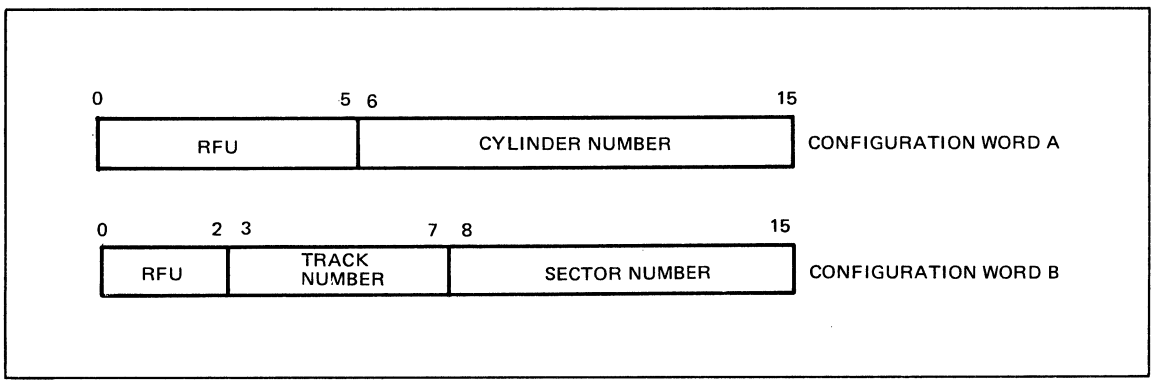


Figure 2-1 Adapter Configuration Words

the busy state when this instruction is accepted, and the indicated task is either initiated or stacked. The direction of data transfer is determined by bit 17 of the IOLD instruction as described in subsection 2.1.2. The Output Task Word instruction causes detailed status indicators in the device to be reset for all device commands except the Tag Code In/Out commands.

Bit 8 of the task word has specific meaning for Format Read and Format Read ID commands. When bit 8 is a logic One for these commands, the EDC bytes are not checked. Bits 8 - 15 of the task word contain tag information as described in subsection 2.3.1.

Bits 2 through 4 of the command code have specific meaning for all media data transfers as described below.

2.1.3.1 Automatic Seek - Bit 2

If this bit is a logic Zero, the data transfer is initiated based on the current cylinder position of the device. If this bit is a logic One, a seek cylinder operation, based on the current content of configuration word A, is initiated. At seek completion, the specified data transfer operation is initiated immediately (i.e., no interrupt is required). The MPDC can service other channels during the seek latency period.

2.1.3.2 Sector Size - Bit 3

If this bit is a logic Zero, the data transfer operation assumes a track format of 256 byte sectors. If this bit is a logic One, the data transfer operation assumes a track format of 2304 byte sectors.

2.1.3.3 Automatic RPS - Bit 4

If this bit is a logic Zero, initiation of the data transfer operation causes a search for the sector ID specified in configuration words A and B (or the index mark in the case of format operations). The MPDC cannot service other channels until the search and any subsequent data transfer has been completed.

If this bit is a logic One, initiation of the data transfer operation is preceded by an RPS command to the device based on the current sector number of configuration word B (bits 10 - 15 only; bits 8 and 9 are ignored). The MPDC can service other channels after initiation of the RPS command. The data transfer operation begins when the sector ID in configuration words A and B (or the index mark in the case of format operations) has been found following the RPS on indication from the device. Rotational position sensing requires that all sectors on a track be numbered sequentially.

2.1.4 Automatic Track Switching

The storage module disk subsystem can effect multisector data transfers (read or write) without software intervention via an automatic track/cylinder switching capability. Track switching is not associated with index marks from the device, but rather occurs whenever the last logical sector on a track (63 or 7) has been completed and the range has not expired. When the last track of the cylinder has been completed and track switching is attempted, firmware initiates a seek to the next consecutive cylinder, selects track zero, and initiates a search for sector zero.

Configuration words are updated for each track and/or cylinder completion as long as the range is nonzero. Any attempt to switch tracks from the last track of the last cylinder causes the illegal seek bit of the status word to be set.

2.1.5 Device-Specific Commands

The device-specific commands illustrated in Figure 2-2 are described in the following subsections.

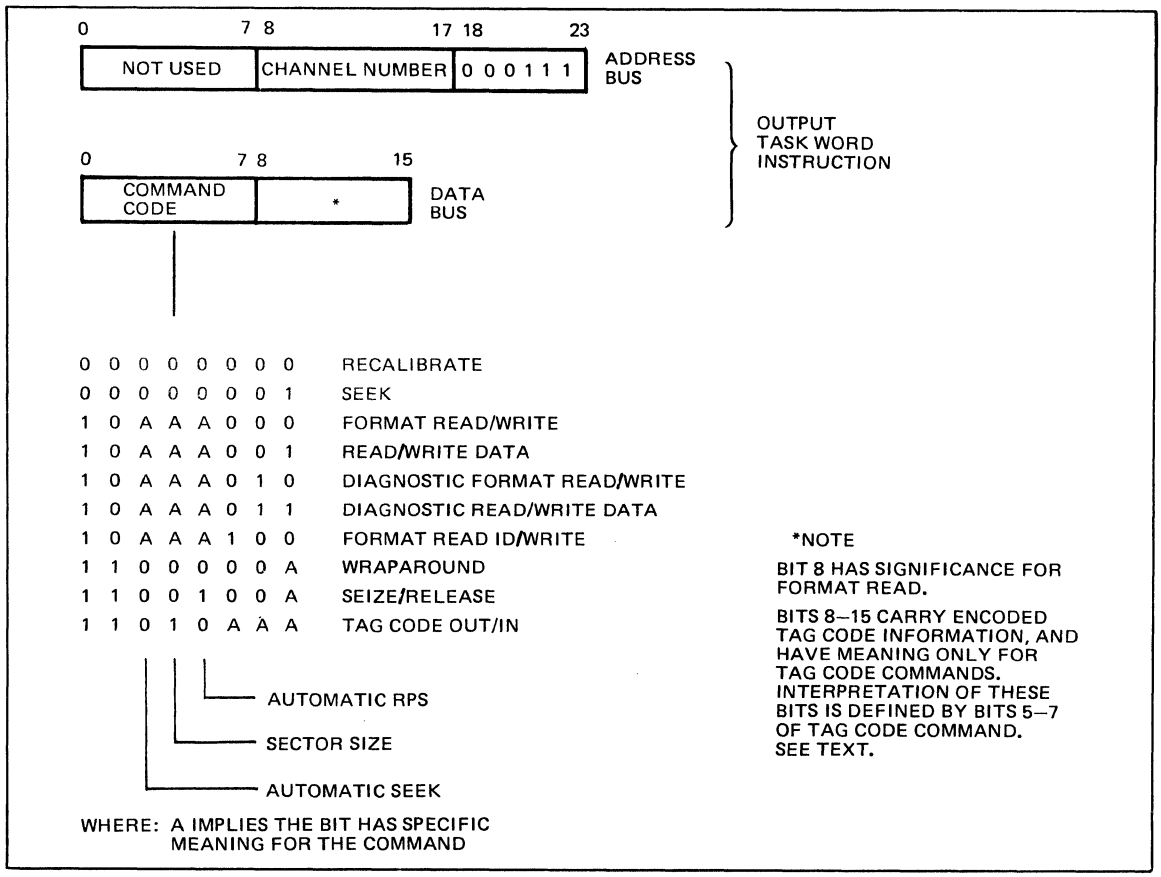


Figure 2-2 Output Task Word

2.1.5.1 Recalibrate Command

The Recalibrate command causes the device to move its positioner to track zero of cylinder zero, resets the Seek End line on the adapter/device interface, and clears all device fault indications. Completion of the recalibration operation results in the generation of an interrupt.

2.1.5.2 Seek Command

The Seek command causes the device to move its positioner to the cylinder indicated in configuration word A. If the cylinder specified is greater than 410 or 822 (depending on the device), or if an error occurs during positioner movement, an error bit is set in the status word. An interrupt is generated upon completion of a positioning operation, except for a seek completion during an automatic seek.

2.1.5.3 Format Read Command

The Format Read command causes the channel to sequentially read all ID and data fields on a track beginning with the first sector after index, with all data being transferred to memory. If bit 8 of the task word is a logic One, the EDC bytes associated with the sector ID fields are ignored. If bit 8 is a logic Zero the EDC bytes are checked in the adapter. Data is transferred until an uncorrectable read error occurs (except as noted for bit 8), the range is satisfied, or the entire track is read.

If an uncorrectable read error occurs in any field (except as noted for bit 8 above), the operation is terminated, the read error bit is set in the status word, and the sector ID field of configuration word B retains the address of the sector in error. The normal range for this command (to read one complete track) is:

- R = (4 bytes + 256 bytes) x 64 sectors = 16640 bytes (for 256 byte sectors)
- R = (4 bytes + 2304 bytes) x 8 sectors = 18464 bytes (for 2304 byte sectors)

2.1.5.4 Format Read ID Command

The Format Read ID command is identical to the Format Read command except that only the sector ID field of each sector is transferred to memory. The normal range for this command (to read one complete track) is:

- R = 4 bytes x 64 sectors = 256 bytes (for 256 byte sectors)
- R = 4 bytes x 8 sectors = 32 bytes (for 2304 byte sectors)

2.1.5.5 Format Write Command

The Format Write command causes the channel to format the track that is positioned under the selected read/write head with 8 or 64 equal length sectors (as specified by bit 3 of the task word) starting with the first sector after index. Sector ID fields read from memory are transferred to the adapter and written on the media. Firmware-supplied data field sync bytes are also written on the media, and all data fields are Zero-filled by the MPDC. The range to format one complete track is:

- R = 4 bytes x 64 sectors = 256 bytes (for 256 byte sectors)
- R = 4 bytes x 8 sectors = 32 bytes (for 2304 byte sectors)

2.1.5.6 Read Data Command

The Read Data command causes the device to locate the sector defined by configuration words A and B and to begin transfer of the data field of that sector to memory. Data transfer continues until the range is satisfied. When transfer of the first data field is complete (without errors), the sector number field of configuration word B is incremented. If the initial range (plus offset) is greater than 256 (or 2304 if 2304 byte sectors are specified), data transfer continues with the data field of the next sector as specified by the updated contents of configuration words A and B. This operation continues until the range is satisfied, an uncorrectable read error occurs, or the specified record cannot be found (when adapter logic detects two index pulses without a successful comparison between the sought-for ID and the IDs being read from the device). If the record cannot be located, the unsuccessful search bit is set in the status word. Track and cylinder switching can occur during this command as long as the range is not exhausted when the last sector of a track/cylinder has been completed (see subsection 2.1.4).

If an uncorrectable read error occurs in a data field, the operation is terminated, the read error bit is set in the status word, and the sector number field of configuration word B retains the address of the record in error. If a read error occurs in a sector ID field, a miscompare is assumed and the search continues. The read error bit of the status word is set, however, so that if the desired record is not found, the operation will be terminated with both the unsuccessful search and the read error bits posted in the status word. The read error is reset if the desired record is found.

2.1.5.7 Write Data Command

The Write Data command causes the device to locate the sector defined by configuration words A and B, and to write a sync byte followed by the data field of (at least) that sector. Data from memory is transferred to the media via the MPDC and the adapter. The sector number field of configuration word B is incremented when the transfer of data for the specified sector is completed.

If the range is less than 256 or 2304, the remainder of the data field is Zero-filled. If the range is greater than 256 or 2304, the sector represented by the updated contents of configuration words A and B is located and the data field and associated sync byte are written. This operation continues until the range is satisfied or until the specified record cannot be found on the track (in which case the unsuccessful search bit is set in the status word). Detection of a read error in the sector ID field is described in subsection 2.1.5.6.

Track and cylinder switching can occur during this command as described in subsection 2.1.4.

2.1.5.8 Diagnostic Write Data Command

The Diagnostic Write Data command causes the channel to perform as if the Write Data command was specified except that the EDAC field associated with the data field is written from memory (not from adapter hardware). Only one sector can be written by this command.

2.1.5.9 Diagnostic Read Data Command

The Diagnostic Read Data command causes the channel to perform as if the Read Data command was specified except that the EDAC field associated with the data field is also transferred to memory (and error detection and correction is not performed by adapter logic). Only one sector can be read by this command.

2.1.5.10 Diagnostic Format Write Command

The Diagnostic Format Write command causes the channel to perform as if the Format Write command was specified except that invalid EDC characters (all Zeros) are written after each sector ID field.

2.1.5.11 Diagnostic Format Read Command

The Diagnostic Format Read command causes the channel to read everything on a track, beginning with the first sector ID after the next address mark detected. Meaningful operation of this command can only be guaranteed on tracks which have not been updated since the last format operation (updated records may contain write splice discontinuities). No EDC or EDAC checks are performed by the adapter during this command.

2.1.5.12 Wrap-around Read/Write Commands

Two wrap-around levels are available in the Storage Module Disk subsystem. If bit 7 of the task word is a logic Zero, wraparound is at the MPDC level; if bit 7 is a logic One, wrap-around is at the adapter level.

During a Wrap-around Write command, the channel transfers up to eight words from memory to either the MPDC or the adapter FIFO buffer. During a Wrap-around Read command (which must occur immediately following a Wrap-around Write command), the bytes previously loaded into the FIFO buffer are returned to memory.

The range for these commands is from 1 to 8 words, with the same range used for both read and write wrap-around operations. Associated write and read wrap-around commands must start and end from the same memory boundary.

2.1.5.13 Seize/Release Commands

These commands are for dual-port devices only. Definitions are presently not available.

2.1.5.14 Tag Code In/Out Commands

These commands provide the capability of sending and receiving all of the tag codes described in subsection 2.3.1. Bits 8 through 15 of the task word contain the tag code information. Interpretation of these bits is determined by the tag code itself, which is contained in bits 5 through 7 of the task word command code field. For output transfers, the tag code and associated information are placed in registers in the adapter for subsequent transfer to the device. If the range specifies more than one byte, the tag code is incremented so that consecutive tag code data bytes can be stored and transferred. Multiple byte transfers begin with the tag code specified and end with expiration of the range or on tag code 7. For input transfers, the tag code information is transferred from the device through the adapter and is stored in the scratch pad memory for subsequent transfer to memory.

2.1.6 Software Status Words (Figures 2-3 and 2-4)

The MPDC sends status information to the Megabus data lines from two status words located in its scratch pad memory. Status bits that are adapter/device-specific are described in Tables 2-1 and 2-2. Refer to the MPDC manual for a description of the remaining MPDC status bits.

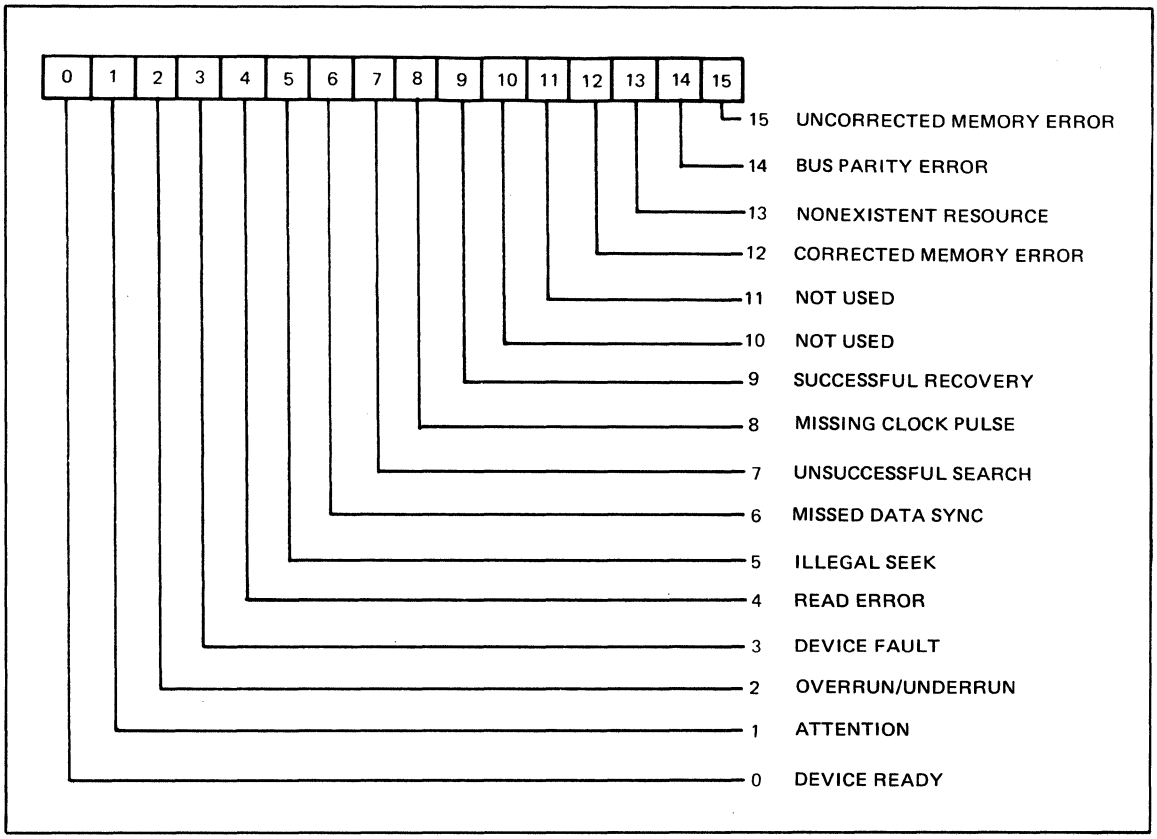


Figure 2-3 Status Word 1

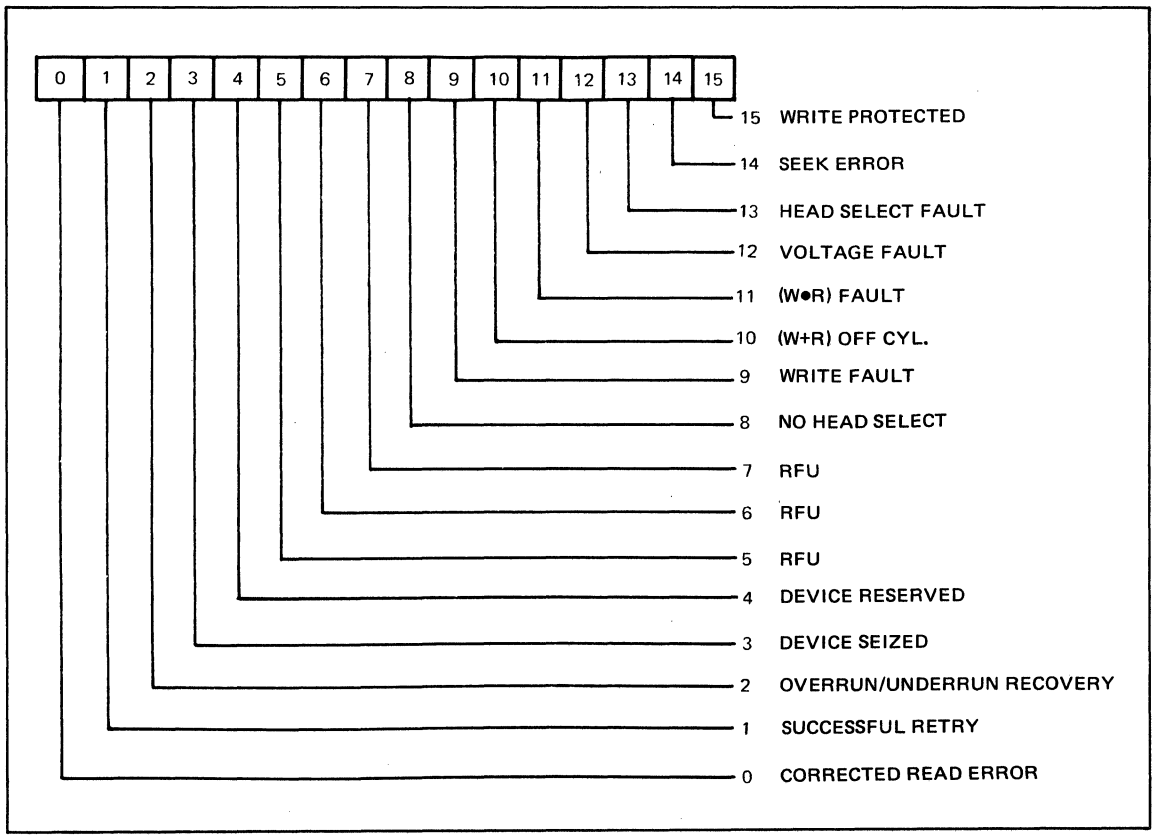


Figure 2-4 Status Word 2

Table 2-1 Status Word 1 Definitions

STATUS WORD BIT	DEFINITION
0 Device Ready	Indicates that the device is on-line with the medium loaded and that no further manual intervention is required to place it under program control. A change in the state of this bit causes the Attention bit to be set resulting in an interrupt (if the interrupt level is nonzero).
1 Attention	Indicates to software any change in the operational state of the device. Sets whenever the Device Ready bit changes state. When set, an interrupt is attempted (if the interrupt level is nonzero). This interrupt serves as notification of the device state change. The attention bit is reset by initialize, an Output Task Word command, an Input Status Word command, or a master clear.
2 Overrun/Underrun	Sets during a read or write operation if the data transfer rate to/from main memory is not maintained at a high enough rate (600 KW/sec. nominal during data field transfers in the word mode). Either data was lost on input because of failure to keep up with device demands, or data was unavailable on output when required by the device. For search and write or search and read, the operation is retried; if successful, this bit is reset and bit 9 of status word 1 and bit 2 of status word 2, are set. Also, bit 2 of status word 1 can be set during the format operation. If such an error occurs, no retry is attempted. This bit is reset by an initialize, an Output Task word command, or a master clear.
3 Device Fault	Sets if the Check Diagnostic bit (bit 7 of Tag Code In 111) is set, indicating a device fault. Bits 8-15 of status word 2 reflect the specific fault. This bit is reset by an initialize, an Output Task Word command, or a master clear.
4 Read Error	Sets during any read operation if the EDAC word at the end of a field indicates that either a correctable or uncorrectable error has occurred within the data field. Also sets for an error in any ID field and is reset if there is a successful search. This bit is reset by an initialize, an Output Task Word command or a master clear.

Table 2-1 Status Word 1 Definitions (Cont.)

STATUS WORD BIT	DEFINITION
5 Illegal Seek	Sets if bits 6 - 15 of configuration word A exceed the maximum cylinder number for a device during a Seek command execution or a multisector operation. This bit is reset by an initialize, an Output Task Word command, or a master clear.
6 Missed Data Sync	Sets if, after a sector ID has been detected during any read operation, the corresponding data field is not detected. This bit is reset by an initialize, an Output Task Word command, or a master clear.
7 Unsuccessful Search	Sets during a nonformat read or write operation for which the sector ID specified in configuration words A and B cannot be located on the track. This bit also sets if index mark is detected during a format write operation, in which case the memory address and range registers are invalid. This bit is reset by an initialize, an Output Task Word command, or a master clear.
8 Missing Clock Pulse	Sets if the MPDC detects a missing clock pulse during read or write operations. This bit is reset by an initialize, an Output Task Word command, or a master clear. Note that a missing clock pulse during a read operation also sets the Read Error status bit.
9 Successful Recovery	Sets when a recoverable error condition was successfully recovered during the previous operation. Status word 2 specifies the error condition which occurred. This bit is reset by an initialize, an Output Task Word command, or a master clear.
10, 11	Not used.
12	See MPDC manual.

Table 2-2 Status Word 2 Definitions

STATUS WORD BIT	DEFINITION
0 Corrected Read Error	Indicates when set that a correctable read error occurred during the previous read operation, and that correction was performed successfully by the MPDC. When this bit is set, it causes bit 9 of status word 1 to be set.
1 Successful Retry	Indicates when set that an uncorrectable data read error was successfully retried during the previous operation. When this bit is set, it causes bit 9 of status word 1 to be set.

Table 2-2 Status Word 2 Definitions (Cont.)

STATUS WORD BIT	DEFINITION
2 Overrun/Underrun	Indicates when set that an underrun/overrun error has been successfully retried during the previous operation. When the bit is set, it causes bit 9 of status word 1 to be set.
3 Device Seized	Indicates when set that the addressed device has been seized by this channel. A seized device remains seized until released by a Release command, expiration of a timer in the device after a predetermined time following completion of the last device operation, or by initialization.
4 Device Reserved	Indicates when set that the addressed device is presently seized at its other port and is not accessible to this channel.
5 - 7	Reserved for future use.
8 No Head Select	Indicates when set that a Head Select command greater than 4 or 18 has been issued.
9 Write Fault	Indicates when set that the addressed device has sensed the absence of write current during a write operation.
10 Off Cylinder (Write or Read)	Indicates when set that a read or a write operation has been attempted while the device positioner is offset.
11 Write & Read Fault	Indicates when set that both the write gate and read gate signals to the addressed device are simultaneously true.
12 Voltage Fault	Indicates when set that the addressed device has detected a below-normal voltage level.
13 Head Select Fault	Indicates when set that more than one read/write head has been selected in the addressed device.
14 Seek Error	Indicates when set that a seek error has occurred in the addressed device.
15 Write Protected	Indicates when set that the addressed device is in the write protect state due to a fault which inhibits writing, to head alignment being performed, or to the write protect switch setting.

2.2 FIRMWARE

Firmware, a sequence of microinstructions resident in the microprogram control store on the PROM adapter board, is the primary control element of the adapter. The main function of the firmware is to interpret external and internal events or conditions and react in a prescribed manner (i.e., setting or resetting of hardware functions, etc.). Efficient data transfer is a result of firmware control of hardware components in the data path.

Firmware is divided into MPDC-specific routines and adapter-specific routines. Adapter-specific firmware is assigned an area of the MPDC scratch pad memory for temporary storage

of firmware operational and device-specific information, including configuration words (subsection 2.1.1), task words (subsection 2.1.3), and the status word (subsection 2.1.6). Firmware uses the configuration words as the seek argument and updates them as required during data transfer operations. Device-specific commands contained in the task word are not sent to the adapter/device but are analyzed by adapter firmware. Firmware then generates and sends to the adapter those constants necessary to set up adapter hardware to execute the desired operation. Firmware also monitors adapter/device status and updates the status word in the scratch pad memory as required.

Firmware effects the transfer of data, control information, and EDAC information to the adapter, and the transfer of data, status, I/D, and EDAC information to the MPDC via manipulation of strobe and control lines on the MPDC/adapter interface. Refer to Section III for a description of firmware operation.

2.3 HARDWARE OVERVIEW DESCRIPTION

Data flow through the major areas of logic in the adapter is shown in Figure 2-5. The command store and read/write control logic, together with the service request logic, controls the flow of data between the MPDC and the addressed device. Data bytes received from the

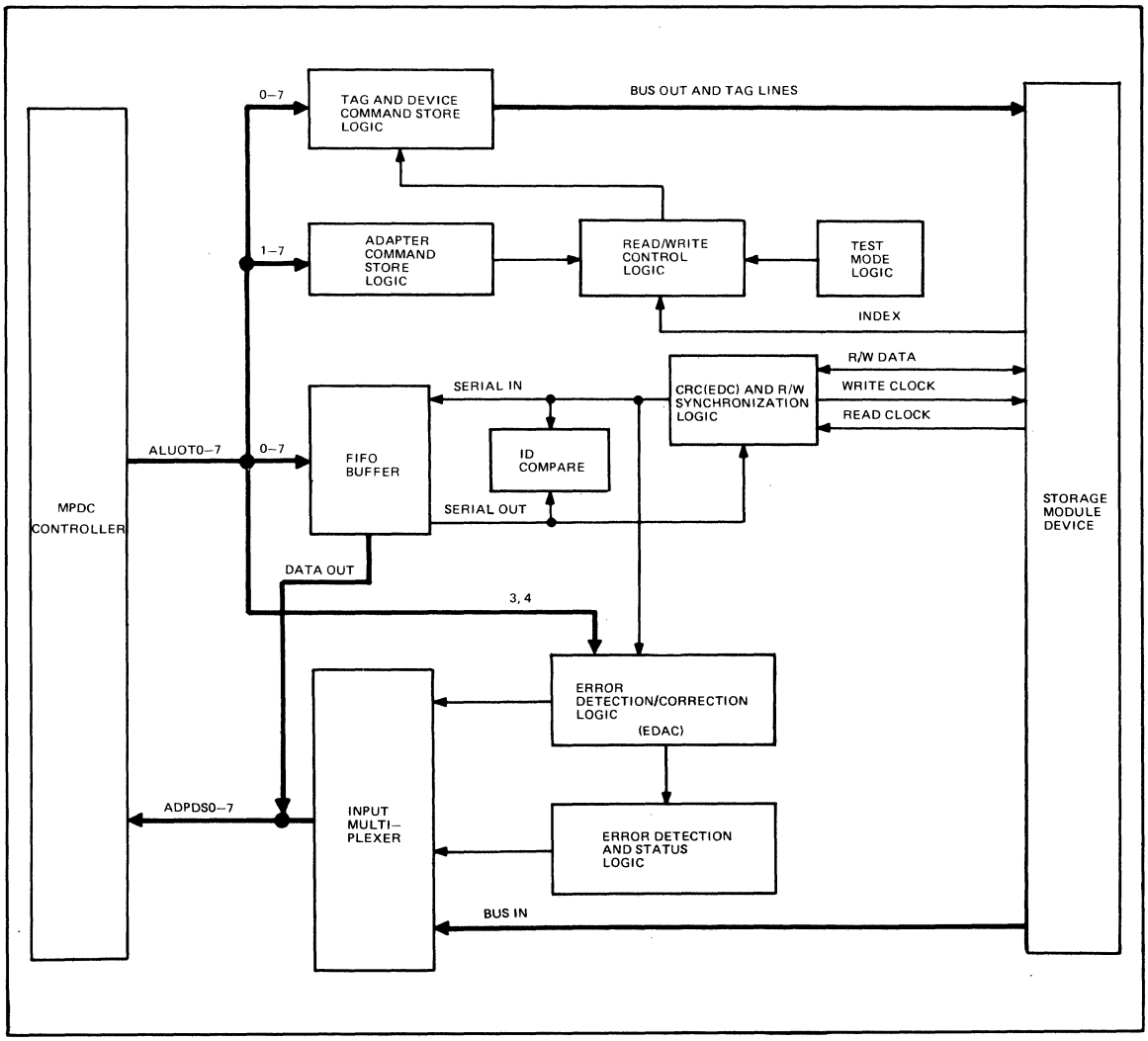


Figure 2-5 Adapter Hardware Major Block Diagram

MPDC are transformed into a serial data stream via the FIFO buffer and sent to the device via the bidirectional read/write data line. Write synchronization is ensured by a write clock (derived from the device servo clock) which accompanies write data to the device. During read operations, serial data from the device is assembled into bytes in the FIFO buffer for transfer to the MPDC. Read synchronization is ensured by a read clock from the device.

During search operations, the ID compare logic compares the sector ID being read with the sought-for sector ID, which has been placed in the FIFO buffer by firmware. An unsuccessful search is posted in the adapter status register for subsequent firmware interrogation. Firmware initiates the appropriate read or write operation when it ascertains that a search has been successful.

Data integrity is ensured by the Cyclic Redundancy Check (CRC) logic and the Error Detection and Correction (EDAC) logic. A two-byte Error Detection Code (EDC) is developed by the CRC logic and written on the media for sector ID fields during write operations. During read operations, the EDC code is developed again as the sector ID is being read and is compared with the recorded EDC code. An unequal comparison indicates a read error.

A 7-byte EDAC code is developed for data fields and written on the media during write operations. During read operations, this EDAC code is developed again as the data field is being read, and compared with the recorded EDAC code. An unequal comparison indicates a read error, in which case the EDAC code is acted upon by firmware in an attempt to effect correction of the data field information.

In addition to controlling data flow and developing error codes, the adapter also:

- Assists the MPDC in formatting data written on the media.
- Generates operational error signals for notifying the MPDC of adapter status.
- Generates service requests to the MPDC.
- Generates adapter status information for interrogation by firmware.

2.3.1 Interface Signal Lines

Interface signal lines between the adapter and the MPDC are shown in Figure 2-6. Interface signal lines between the adapter and the device are shown in Figure 2-7 and described in Tables 2-3 through 2-6. Signal pin location information (adapter/ device) is shown in Table 2-7. The device is attached to the adapter by both daisy chain and radial interface cables. The radial interface includes the clock, seize, reserve, and bidirectional read/write data lines for one device. The daisy chain interface includes the bus, tag and control lines for up to four attached devices.

2.3.2 Logic Elements

Major functional logic elements within the adapter are shown in Figure 2-8 and described in Table 2-8.

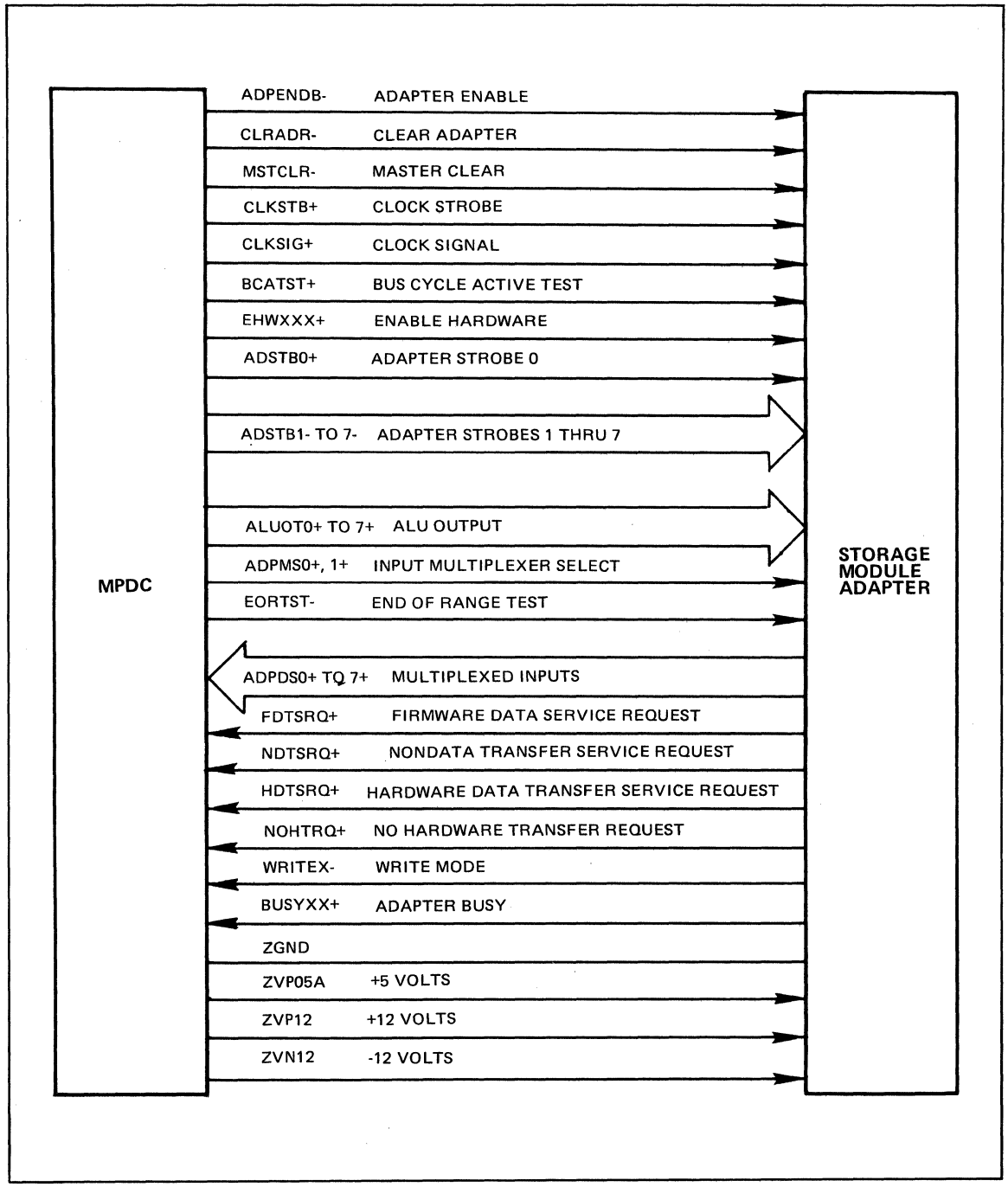


Figure 2-6 Adapter/MPDC Interface Signal Lines

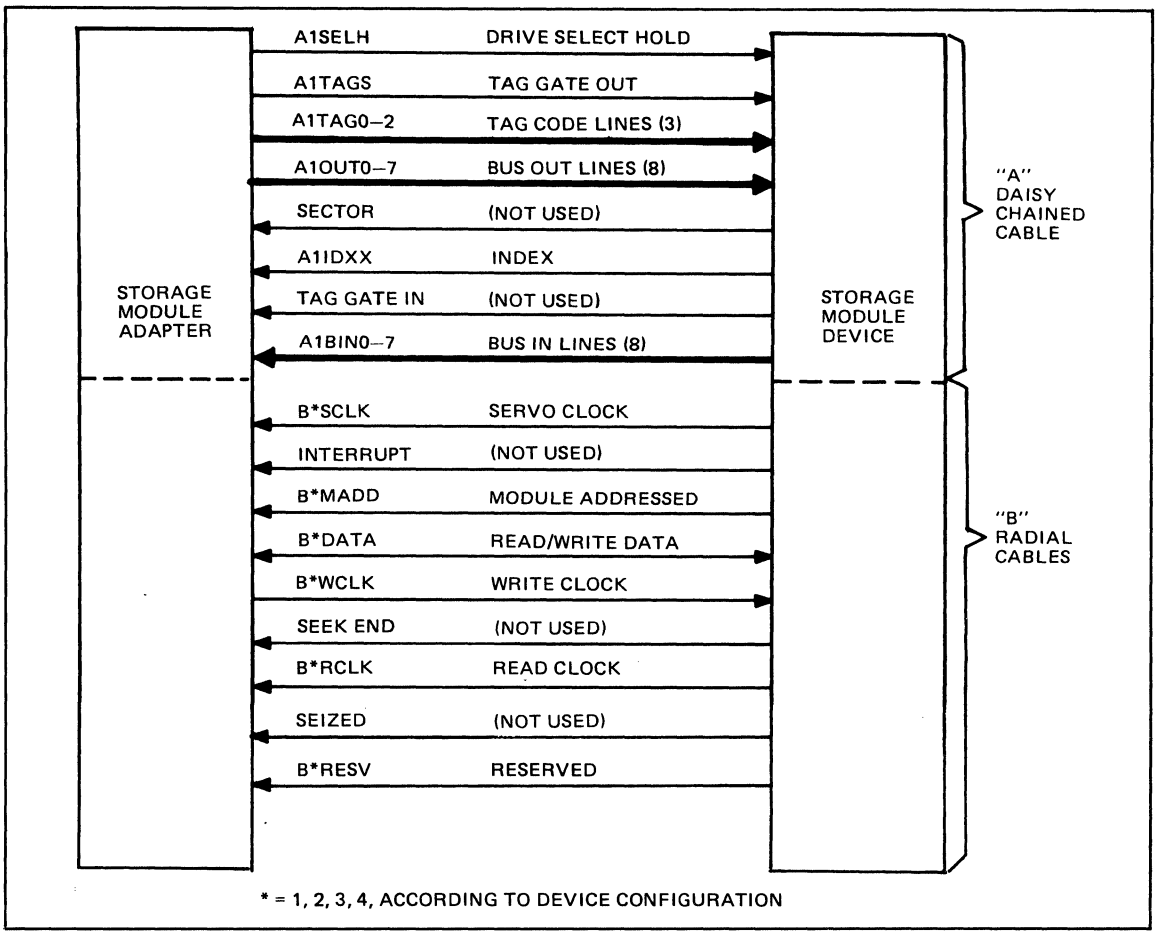


Figure 2-7 Adapter/Device Interface Signal Lines

Table 2-3 Adapter/Device Interface Signal Lines

SIGNAL LINE	DESCRIPTION
Drive Select	The signal on this line is set when MPDC firmware successfully completes its Basic Logic Test (BLT) routine and remains true unless reset by a master clear. Drive Select enables the device interface receivers.
Tag Gate Out	The signal on this line is controlled by adapter firmware and, when true, is used by the device to gate the tag code (Tag lines) and associated tag information (Bus Out lines). When false, this signal causes the device to ignore the information on the Bus Out lines.
Tag Lines (3)	Three lines which, when decoded by the device logic, define the tag function(s) encoded on the Bus Out lines if Tag Gate Out is true. Tag codes for both input and output operations are strobed by adapter firmware into the tag register in the adapter. Table 2-4 defines the tag-out functions, and Table 2-5 defines the tag-in functions. Tag-out and tag-in information is summarized in Table 2-6.

Table 2-3 Adapter/Device Interface Signal Lines (Cont.)

SIGNAL LINE	DESCRIPTION
Bus Out 0-7	Eight lines carrying tag information to the addressed device. Interpretation of the information encoded on these lines is performed by the device in accordance with the Tag Code. Bus Out line definitions for each tag code are described in Table 2-4 and summarized in Table 2-6.
Sector	Not used by the adapter.
Index	A signal of approximately 2.5 microseconds duration from the device signaling the beginning of track.
Bus In 0-7	Eight lines carrying information from the device to the adapter. Information on these lines has meaning according to the tag code that is stored in the adapter's tag code register at the time when the content of the Bus In lines is transferred to the adapter. This information is passed through the adapter's input multiplexer (under firmware control) to the MPDC where it is interpreted by firmware. Bus In line definitions for each tag code are described in Table 2-5 and summarized in Table 2-6.
Servo Clock	A 9.677-MHz clock signal derived from the device medium servo track.
Interrupt	Not used by the adapter.
Module Addressed	Indicates, when true, that the device was addressed during the last Tag000 operation.
Read/Write Data	A bidirectional line carrying NRZ data that has been read from or is to be recorded on the device media.
Write Clock	A signal, derived from the device-generated servo clock, that is used to synchronize NRZ write data on the Read/Write Data line.
Seek End	Not used by the adapter.
Read Clock	A signal, derived from the read data by a phase lock loop (PLL), synchronous with the read data, and defining a bit cell.
Seized	Not used by the adapter.
Reserve	Device port seized by other port.

Table 2-4 Tag Code Out Information

TAG CODE	BUS OUT LINE	DEFINITION
000 Select	0-3	Must be Zero.
	4-7	Address. Binary encoded device address. Note that address 15 is reserved for maintenance.
000 Release	0,2,3	Must be Zero.
	1	When true, a release command is sent to the drive selected by address bits 4-7. This command is sent to release a dual port device.
001 Error Recovery	0	Early Strobe. When true, causes the device data separation logic to strobe the data earlier than normal. This condition is cleared by any of the following: 1. A Recalibrate command 2. A Seek command 3. A Diagnostic Tag (Code 010) with Clear Error Recovery (bit 4) set 4. Another Tag (Code 001) with the Early Strobe (bit 0) reset
	1	Late Strobe. When true, causes the device data separation logic to strobe the data later than normal. This condition is cleared by another Tag (code 001) with the Late Strobe (bit 1) reset, or by items 1, 2, or 3 above.
	2	Positive Offset. When true, causes the device to offset its positioner from the nominal on-cylinder position in a direction away from the spindle. If a write operation is attempted while the positioner is offset, Check Diagnostic and Write Fault indications are set at the device. The offset condition is cleared by any of the following: 1. Another Error Recovery tag (code 001) with Positive Offset (bit 2) reset or with Negative Offset (bit 3) set 2. A Recalibrate command. 3. A Seek command. 4. A Diagnostic Tag (code 010) with Clear Error Recovery (bit 4) set.
	3	Negative Offset. When true, causes the device to offset its positioner from the nominal on-cylinder position

Table 2-4 Tag Code Out Information (Cont.)

TAG CODE	BUS OUT LINE	DEFINITION
		in a direction toward the spindle. If a write operation is attempted while the positioner is offset, Check Diagnostic and Write Fault indications are set in the device. The Negative Offset condition is cleared by another Error Recovery Tag (code 001) with Negative Offset (bit 3) reset or with Positive Offset (bit 2) set, or by items 2, 3, or 4 above.
	4-7	Must be Zero.
010 Diagnostic	0	Recalibrate. When true, causes the device to seek to track zero, clear its seek error, strobe and positioner offset conditions, and clear its cylinder and head address registers.
	1	Clear Attention. When true, clears the attention flip-flop in the device logic.
	2	Clear Check Diagnostic. When true, clears the Check Diagnostic indication in the device logic if the fault condition no longer exists.
	3	Clear Fault Status. When true, clears the individual fault latches, the fault flip-flop, and the fault indicators in the device.
	4	Clear Error Recovery. When true, clears the error recovery latches set by Tag Code 001.
	5	Clear RPS. When true, clears RPS Enable in the device logic so that an interrupt will not be sent to the adapter/MPDC for every revolution of the medium.
	6-7	Must be Zero.
011 Head Address	0-2	Must be Zero.
	3-7	Head Address. Five binary-encoded lines that carry the desired head address to the selected device.
100 High Cylinder	0-5	Must be Zero.
	6-7	High Cylinder Address. Two binary-encoded lines that carry the two high-order (most significant) bits of the cylinder address to the device. If these bits change from one operation to the next, they must be sent to

Table 2-4 Tag Code Out Information (Cont.)

TAG CODE	BUS OUT LINE	DEFINITION
		the device before the new low-order address bits are sent. If they do not change they need not be sent.
101 Target Register	0	Load Target Register. When true, causes the desired sector address (bits 2-7 of the Data Out lines) to be sent to the device's target register and enables Rotational Position Sensing (RPS).* When false, the present sector address will be displayed on the Bus In lines, and the old target will be destroyed. (Refer to Target Register in Table 2-5.)
	1	Must be Zero.
	2-7	Sector Address. See above.
110 Low Cylinder	0-7	Low Cylinder Address. Seven binary-encoded lines that carry the low-order (least significant) bits of the cylinder address to the device. If the high order bits of the cylinder address have changed since the previous operation, Low Cylinder address must be preceded by the new High Cylinder address.
111 Control	0	Transfer Sector Count. When true, causes the device to load its present sector address into the target register (thereby destroying the old target address), and sets the RPS mode.
	1	Write Gate. When true, enables the write driver in the device logic provided that no inhibiting fault condition exists.
	2	Must be Zero.
	3	Read Gate. When true, enables NRZ read data on the Read/Write Data interface line. The Read Gate must go false when the read/write head is passing a spliced area (except when searching for an address mark).
	4	Address Mark Enable. When true, and with Write Gate true, causes the device to perform an erase operation, thereby creating an address mark. When true, and with Read Gate true, enables recovery of recorded address marks.
	5-7	Must be Zero.

*Not used by this adapter. RPS accomplished by polling device sector counter.

Table 2-5 Tag Code In Information

TAG CODE	BUS IN LINE	DEFINITION
000 Select	0	Density. When true, indicates that an 823-cylinder device has been selected. When false, indicates that a 411-cylinder device has been selected.
	1	Tracks. When true, indicates that a 19-track device has been selected. When false, indicates that a 5-track device has been selected.
	2	Reserved for future use.
	3	Attention. When true, indicates that the read/write heads have been unloaded. The attention condition is cleared by a Diagnostic Tag Out (code 010) with the clear attention bit set.
	4-7	Device Address. Four binary-encoded lines that carry the address of the selected device.
001 Error Recovery	0-7	Bus Out Bits 0-7. Wraparound of Bus Out Lines to test drivers/receivers in device.
010 Diagnostic	0	No Head Select. When true, indicates that a head address greater than 4 or 18 (device dependent) has been sent to the device. This condition is cleared by sending a valid head address.
	1	Write Fault. When true, indicates absence of write current when Write Gate is true. Writing is inhibited in the device when this condition exists.
	2	Write or Read and Off Cylinder. When true, indicates that a write or a read operation has been attempted while the read/write heads are offset. Writing is inhibited in the device when this condition exists.
	3	Write and Read Fault. When true, indicates that Write Gate and Read Gate are true at the same time. Writing is inhibited in the device when this condition exists.
	4	Voltage Fault. When true, indicates that the device has detected a below-normal voltage condition. Writing is inhibited and the device performs an emergency positioner retraction when this condition exists.

Table 2-5 Tag Code In Information (Cont.)

TAG CODE	BUS IN LINE	DEFINITION
	5	Multi-Head Select Fault. When true, indicates that more than one head has been selected. Writing is inhibited when this condition exists.
	6	Seek Error. When true, indicates that a seek error has occurred in the device. This signal comes true when an invalid cylinder address has been sent to the device, when the positioner fails to complete its move within a specified time, or when the positioner has moved the heads outside the inner or outer cylinder limits. This condition can only be cleared by a recalibrate operation.
	7	Write Protected. When true, indicates that the selected device is in the write protect state. The write protect state may be entered due to a fault condition that causes inhibition of writing, or by manual actuation of the protect switch on the device control panel. If Write Gate becomes true while the device is in the write protect state, the device's Check Diagnostic indication is set. The write protect state is reset by actuation of the protect switch on the control panel.
001 Head Address	0	Start. When true, indicates that both the device start switch is set to ON, and a remote Start command has been sent to the drive.
	1	Speed. When true, indicates that an up-to-speed (3000 RPM) condition has been sensed.
		Load or Return to Zero. When true, indicates that the drive has received a command to perform a Load or a Return to Zero (RTZ) operation.
		Dibit Fault. When true, indicates that a loss of track servo dibits, under a loaded condition (for 350 ms), is sensed. This fault will cause the head to unload, and a check condition will be set. A Clear Fault Status or a Manual Fault Clear command is required to clear the fault and re-initiate a head load sequence.

Table 2-5 Tag Code In Information (Cont.)

TAG CODE	BUS IN LINE	DEFINITION
		Heads Loaded. When true, indicates that the transfer of the heads loaded switch to the loaded condition has been sensed.
		Slope. When true, indicates the slope of the fine servo position signal used for servoing on track based on the new programmed cylinder address.
		Fine. When true, indicates that the servo is in the fine mode. This mode signifies that the servo is now servoing on the detected track servo signal.
100		Not used by this adapter.
101 Target Register	0	Echo Read Target Register. When true, indicates that the Bus In lines (2-7) contain the contents of the target register after the transfer takes place. When false, the Bus In lines contain the contents of the present sector counter.
	1	Reserved for future use.
	2-7	Sector Count. Represents present sector counter value. Must send Tag Gate Out to strobe present sector into register which goes on bus bits 2-7. Used to poll device when RPS is active.
110 Low Cylinder	0	Address Mark Found. Indicates during a search, that the device has detected an address mark.
	1	Reserved for future use.
	2	On Cylinder. Indicates that the read/write heads are positioned on the desired cylinder.
	3	Unit Ready. Indicates that device is selected, heads are loaded, spindle is up to speed, and no faults exist.
	4-5	Reserved for future use.
	6	Offset Active. Indicates that the positioner is offset in either the positive or the negative direction.
	7	Check Diagnostic. A fault condition exists in the device.
111 Control	0-7	Same as Low Cylinder (Tag Code 110).

Table 2-6 Tag Out/In Decode

	Bus Lines															
	Bus Out Lines								Bus In Lines							
Tag Code	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Select* 000	MBZ	MBZ	MBZ	MBZ	2^3	2^2	2^1	2^0	823/411 Cyl.	19/5 Track	RFU	Atten-tion	2^3	2^2	2^1	2^0
Release 000	MBZ	1	MBZ	MBS	2^3	2^2	2^1	2^0	823/411 Cyl.	19/5 Tracks	RFU	Atten-tion	2^3	2^2	2^1	2^0
Error Recovery 010	Early Strobe	Late Strobe	+ Offset	− Offset	MBZ	MBZ	MBZ	MBZ	Bus Out Bit 0	Bus Out Bit 1	Bus Out Bit 2	Bus Out Bit 3	Bus Out Bit 4	Bus Out Bit 5	Bus Out Bit 6	Bus Out Bit 7
Diagnostic* 010	RTZ	Clear Atten-tion	Clear Check Diag.	Clear Fault Status	Clear Error Recovery	Clear RPS	MBZ	MBZ	No Head Select	Write Fault	(W+R) Off Cyl.	W.R. Fault	Voltage Fault	Multi Head Select Fault	Seek Error	Write Protected
Head Address 011	MBZ	MBZ	MBZ	2^4	2^3	2^2	2^1	2^0	Start	Speed	(Load + RTZ)	Dibit Fault	Heads Loaded	Slope	Fine	Not Used
High Cyl 100	MBZ	MBZ	MBZ	MBZ	MBZ	MBZ	2^9	2^8	Echo Read Target Register		2^5	2^4	2^3	2^2	2^1	2^0
Target* Register 101	Load Target Register	MBZ	2^5	2^4	2^3	2^2	2^1	2^0	Echo Read Target Register		2^5	2^4	2^3	2^2	2^1	2^0
Low Cyl 110	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	AM Found	0	On Cyl	Unit Ready	0	0	Offset Active	Check Diag.
Control* 111	Transfer Sector Count	Write Gate	MBZ	Read Gate	Address Mark Enable	MBZ	MBZ	MBZ	AM Found	0	On Cyl	Unit Ready	0	0	Offset Active	Check Diag.

*Detailed Status.

Table 2-7 Adapter/Devices Interface Signal Pin Information

SIGNAL NAME	DEVICE PLUG/ PIN NO. (-)		DEVICE PLUG/ PIN NO. (+)		ADAPTER CONNECTOR PIN NO. (-)		ADAPTER CONNECTOR PIN NO. (+)	
	Daisy Chain "A" Cable							
A1SELH	P3,	22	P4,	25	Y01	43	Y01	22
A1TAGS	P3,	1	P4,	4	Y01	25	Y01	54
A1TAG0	P3,	46	P4,	49	Y01	10	Y01	17
A1TAG1	P3,	48	P4,	51	Y01	7	Y01	13
A1TAG2	P3,	52	P4,	55	Y01	36	Y01	12
A1OUT0	P3,	23	P4,	26	Y01	16	Y01	44
A1OUT1	P3,	24	P4,	27	Y01	45	Y01	34
A1OUT2	P3,	28	P4,	31	Y01	15	Y01	14
A1OUT3	P3,	29	P4,	32	Y01	40	Y01	8
A1OUT4	P3,	30	P4,	33	Y01	32	Y01	31
A1OUT5	P3,	34	P4,	37	Y01	35	Y01	6
A1OUT6	P3,	35	P4,	38	Y01	41	Y01	49
A1OUT7	P3,	36	P4,	39	Y01	42	Y01	39
A1IDXX	P3,	10	P4,	13	Y01	26	Y01	51
A1BIN0	P3,	42	P4,	45	Y01	24	Y01	50
A1BIN1	P3,	3	P4,	7	Y01	52	Y01	53
A1BIN2	P3,	15	P4,	18	Y01	3	Y01	2
A1BIN3	P3,	17	P4,	21	Y01	27	Y01	55
A1BIN4	P3,	8	P4,	12	Y01	11	Y01	33
A1BIN5	P3,	16	P4,	20	Y01	4	Y01	5
A1BIN5	P3,	75	P4,	78	Y01	9	Y01	20
	Radial "B" Cables							
B1SCLK	P2,	M	P2,	N	Y03	42	Y03	41
B2SCLK	P2,	M	P2,	N	Y03	20	Y03	19
B3SCLK	P2,	M	P2,	N	Y02	42	Y02	41
B4SCLK	P2,	M	P2,	N	Y02	20	Y02	19
B1MADD	P2,	BB	P2,	DD	Y03	10	Y03	31
B2MADD	P2,	BB	P2,	DD	Y03	7	Y03	8
B3MADD	P2,	BB	P2,	DD	Y02	10	Y02	31
B4MADD	P2,	BB	P2,	DD	Y02	7	Y02	8
B1DATA	P2,	A	P2,	B	Y03	36	Y03	13
B2DATA	P2,	A	P2,	B	Y03	15	Y03	37
B3DATA	P2,	A	P2,	B	Y02	36	Y02	13
B4DATA	P2,	A	P2,	B	Y02	15	Y02	37
B1WCLK	P2,	H	P2,	H	Y03	30	Y03	9
B2WCLK	P2,	H	P2,	J	Y03	39	Y03	17
B3WCLK	P2,	H	P2,	H	Y02	30	Y02	9
B4WCLK	P2,	H	P2,	J	Y02	39	Y02	17
B1RCLK	P2,	W	P2,	X	Y03	18	Y03	40
B2RCLK	P2,	W	P2,	X	Y03	14	Y03	16
B3RCLK	P2,	W	P2,	X	Y02	18	Y02	40
B4RCLK	P2,	W	P2,	X	Y02	14	Y02	16

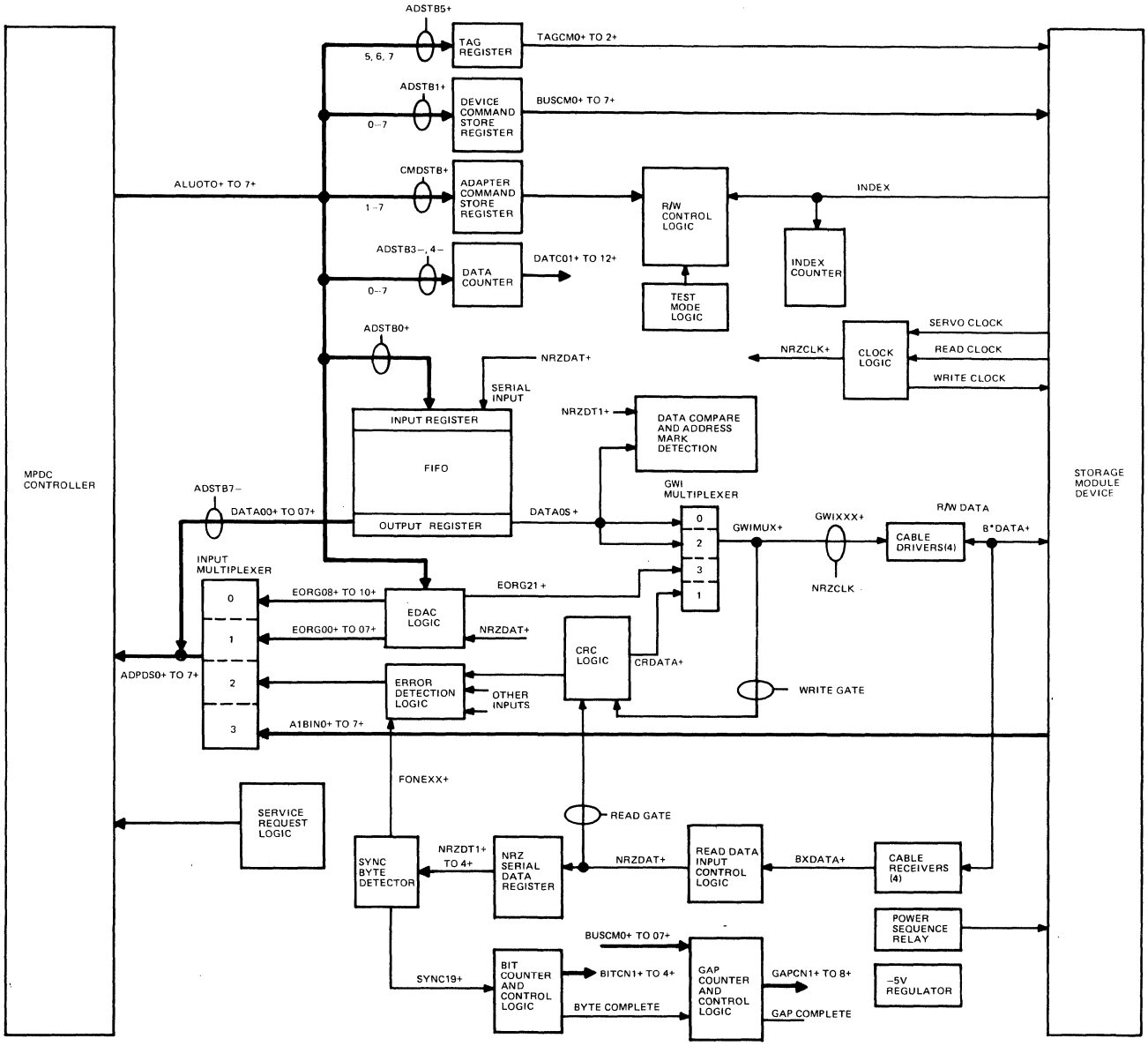


Figure 2-8 Adapter Hardware Intermediate Block Diagram

Table 2-8 Adapter Logic Elements

LOGIC ELEMENT	DESCRIPTION
Adapter Command Register	A collection of D-flops logically structured as a register and used to store commands for adapter operation. Adapter commands consist of constants generated by firmware as the result of analysis of device commands contained in bits 0-7 of the output task word.
Bit Counter	A four-position binary counter that counts bits sent to or received from the device. Preloaded to a count of eight, the bit counter increments by one for each read or write bit. When eight bits have been counted, the counter's byte complete function comes true to indicate the transfer of one byte. Other bit counter outputs are also used to control various adapter operations.
Clock Logic	Generates synchronizing clock signals for adapter operation (NRZ clock), for write operations (write clock), and for read operations (read clock). All clocks are derived from the device except for the test clock (4 MHz) used in QLT.
CRC Logic	The cyclic redundancy check (CRC) logic generates a 2-byte checksum for the sector ID field. The entire sector ID field is half-added and the 16-bit remainder (called the EDC code) is then written on the medium. When the sector ID is read from the media the CRC logic regenerates the EDC code and compares it with the recorded EDC code. An error signal is generated if the two EDC codes are not identical.
Data Compare and Address Mark Detection Logic	Data compare logic compares the sector ID being read from the device with a sector ID that has been placed in the FIFO buffer by firmware. Address mark detection logic monitors Bus In line A1BIN0+ during a search operation, waiting for the device to indicate that it is reading an address mark. When an address mark is found, this logic terminates the address mark search and informs firmware by setting a firmware data service request.

Table 2-8 Adapter Logic Elements (Cont.)

LOGIC ELEMENT	DESCRIPTION
Data Counter	A 12-bit, firmware-loadable counter that contains the length of either the sector ID or the data field. During write operations this counter is decremented each time firmware loads a new byte in the FIFO buffer. During read operations this counter is decremented once for each byte received from the device.
Device Command Register	Eight D-flops logically structured as a register and used to store tag information for transfer to the device via the Bus Out interface lines. The meaning of each bit in the register is defined by the concurrent tag code contained in the tag register (Table 2-6). The addressed device is responsible for decoding tag information. Also used as a backup register for the gap counter.
EDAC Logic	Four shift registers and associated logic configured as a polynomial generator to develop and check an error detection and correction (EDAC) code for data fields. EDAC codes are hardware generated. EDAC logic can be manipulated by firmware to determine whether an error is correctable. Firmware can correct a single burst of errors of 11 bits or fewer and correct the data in the main memory without re-reading the field in error.
Gap Counter	An 8-bit, firmware-loadable binary counter that counts the number of bytes of Zeros written during write gap cycles. Gap length varies with the device type attached, and according to the software-specified format. Firmware loads the gap size in this counter (via the device command register). The counter is then decremented once each time the bit counter indicates that eight bits have been sent to the device.
GWI Multiplexer	Selects either the serial FIFO buffer output (sync bytes, sector ID, or write data), the EDAC code, or the CRC logic (EDC code) as the gated write information to be sent to the device. Forced data field Zeros (during format write) are clocked from an empty FIFO

Table 2-8 Adapter Logic Elements (Cont.)

LOGIC ELEMENT	DESCRIPTION
	buffer and pass through the GWI multiplexer as Zero transitions. Gap field Zeros are sent to the device via the same multiplexer input but appears as a continuous low input because the FIFO serial data output clock is disabled during write gap cycles.
Index Counter	An incrementing register used to count index pulses from the device.
NRZ Serial Data Register	A four-position register that assembles serial read data from the device in four-bit groups for examination by the sync detector.
Power Sequence Relay	Applies a ground to pick and hold lines on the daisy-chain cable when +5 volts is valid in the adapter. This cycles up the devices in sequence when power is applied, and causes all devices to unload heads and stop spindle rotation when the +5 volts is lost in the adapter.
Status Logic	A logical collection of flip-flops distributed throughout the adapter and used to inform the MPDC of various conditions which exist within the adapter. See subsection 2.4.9 for a description of adapter status indications. Note that device status is reported to the MPDC via tag code responses on the Bus In lines (see Table 2-6).
Sync Byte Detector	A comparator chip that compares the content of the serial data register with a hard-wired binary value. This detector indicates the occurrence of a sync byte (hex 19).
Tag Register	Three D-flops logically structured as a register and used to store the current tag code for transfer to the device.
Test Mode Logic	A collection of logic functions that, when activated by firmware, defines the adapter as being in the test mode. In the test mode the FIFO buffer is logically split into two separately controllable buffers, and timing is supplied by a clock from the MPDC.
-5 Volt Regulator	Derives -5 volts from the -12 volts line from the MPDC for use by balanced line drivers and receivers.

2.4 HARDWARE INTERMEDIATE DESCRIPTION

This subsection describes the functionality of various hardware registers, counters, etc., as used during typical functional operations (seek, read, write, EDC code generation, etc.). The reader is referred to the supplemental timing diagrams contained in Appendix B and to the adapter reference manual for additional details.

2.4.1 Device Selection

Since software (via the MPDC and adapter) can service any one of four devices, a method is required to select the desired device. Device selection is as follows:

- Software (via the output task word) sends a Select command (000) to the adapter and devices. The encoded device address (also from software) is sent to the devices on the adapter/device Bus Out lines.
- Each device compares the bus address with its plug address.
- The addressed device responds to the adapter over the device/adapter interface receivers which are gated with the Module Addressed signal (B*MADD). Thus, the correct device receivers are turned on to connect the adapter to the selected drive.

2.4.2 Seek Operation (Figure 2-9)

Firmware loads the high-order cylinder address bits in the device command register and the high cylinder tag (code 100) in the tag register. Firmware then sets the tag strobe flip-flop and strobes this information to the device. The low-order cylinder address bits are sent to the device in the same manner with low cylinder tag (110). The device performs the seek operation and indicates seek complete by setting the on cylinder bit (Bus In line 2). Firmware checks for seek complete by selecting the Bus In lines through the input multiplexer and examining the on cylinder bit.

2.4.3 Recalibrate Operation (Figure 2-9)

Firmware loads the device command register with the recalibrate bit (BUSCM0+) set, and the tag code register with the diagnostic tag (Code 010). Firmware then strobes this information

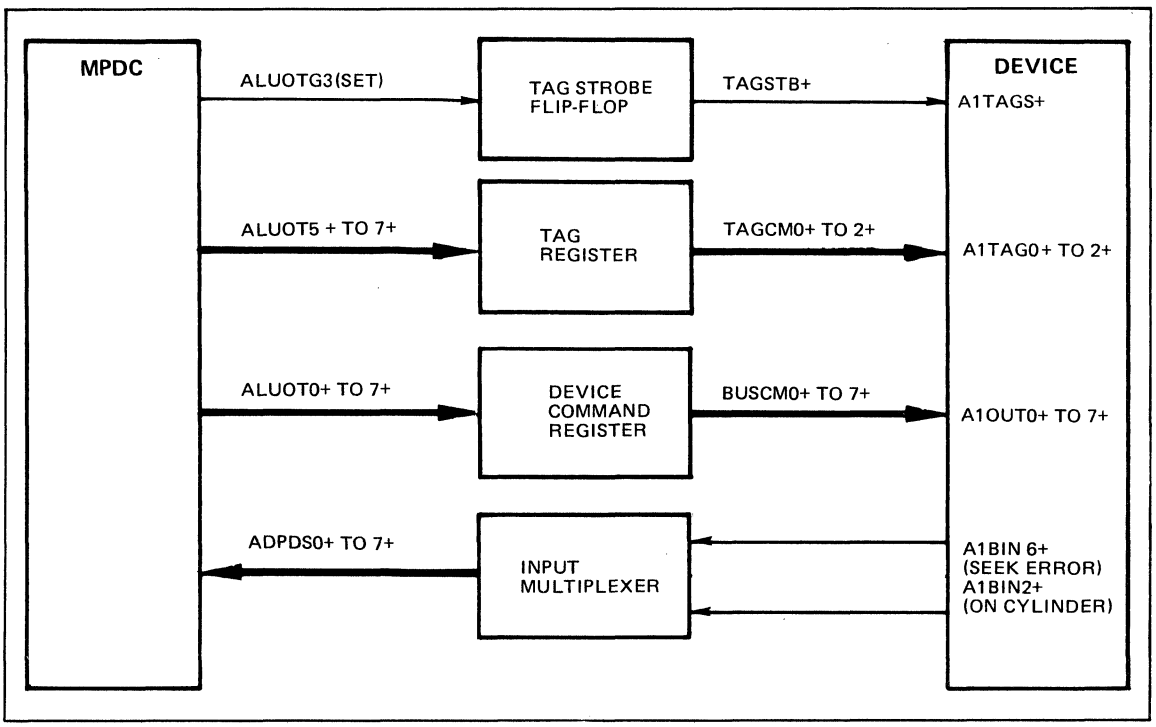


Figure 2-9 Seek and Recalibrate Operations

to the device. The device performs the recalibrate operation and indicates its completion by setting the on cylinder bit (Bus In line 2). Firmware checks for completion of the recalibrate operation by loading a control tag (code 111) in the tag register, selecting the Bus In lines through the input multiplexer, and examining the on cylinder bit.

2.4.4 Format Write Operation

This subsection describes adapter hardware used when the MPDC is performing a format write operation on the addressed device. Sync bytes and sector ID bytes originate in the MPDC, while gap and data field Zeros, address marks, EDC codes and EDAC codes are supplied by adapter hardware. Write data synchronization is supplied by NRZ Clock signals (NRZCLK+ used within the adapter) and by Write Clock signals (B*WCLK+ sent to the device). Both clocks are derived from the servo clock supplied by the address device.

The data stream sent to the device on the B*DATA+ interface line consists of data Ones and Zeros for sync, ID, EDC, EDAC, and data fields, accompanied by write clock signals on the B*WCLK+ interface line. For gap fields and address mark fields, the read/write data line is held false and only clock signals are sent to the device. When it recognizes that an address mark is to be written, adapter logic sets Bus Out line 4 (A1OUT4+). Since firmware has previously sent a control tag code (tag code 111) to the device, the device interprets this bit as an address mark enable function and starts erasing so that no data Ones or Zeros are recorded in the address mark field.

A CRC cycle counter in the adapter determines when the last bit of the field being written (ID field, EDC byte 1, EDC byte 2, data field, and EDAC field) has been reached. This counter supplies the signals to control the adapter operational cycles while a sector is being written and to initiate a firmware data service request at the end of each sector.

2.4.4.1 Format Write Command Timing

Refer to Appendix B, Adapter Timing Diagrams.

2.4.4.2 Adapter Setup for Format Write (Figure 2-10)

Execution of a Format Write command begins after firmware has set up the adapter and the device medium has rotated to the beginning of the track. Adapter setup begins when firmware clears the adapter FIFO buffer and status flip-flop by sending a constant of A0 (hex) on the ALUOT0 - 7 lines, together with adapter strobe ADSTB+2. Since bit zero (ALUOT0+) is a One, the adapter strobe signal ADPSTB+ goes true, causing the adapter to interpret the remaining ALU lines as adapter control gates. ALUOT2+ being true drives the Firmware Reset gate (FRESET-) low to reset the FIFO buffer and the status flip-flops.

Next, firmware sends adapter strobe ADSTB0+ and loads a sync byte (hex 19) into the FIFO buffer. Once the sync byte is loaded, firmware loads a value of 4 decimal in the data counter via strobes ADSTB3+ and ADSTB4+. This value corresponds to the number of bytes in the sector ID field.

Adapter setup continues with firmware loading a gap count of 44 in the device command register. This value is loaded into the gap counter on the next NRXCLK+ clock pulse (i.e., the gap counter is always active, with each clock pulse causing it to load if the Gap Counter Enable function CAPCNT+ is false, or count if CAPCNT+ is true). The count of 44 is one less than the number of bytes in gap 1 (i.e., 32 bytes of Zeros, 3 bytes of address mark, and 10 more bytes of Zeros = 45 bytes). Gap counter control logic is such that the counter decrements to zero when the last (45th) byte of the gap is transferred to the device.

After the gap counter is loaded, firmware loads a control tag code (tag code = 111) in the tag register. This code is used by the device to interpret the content of the Bus Out interface lines.

---

* = 1, 2, 3, 4 according to device configuration.

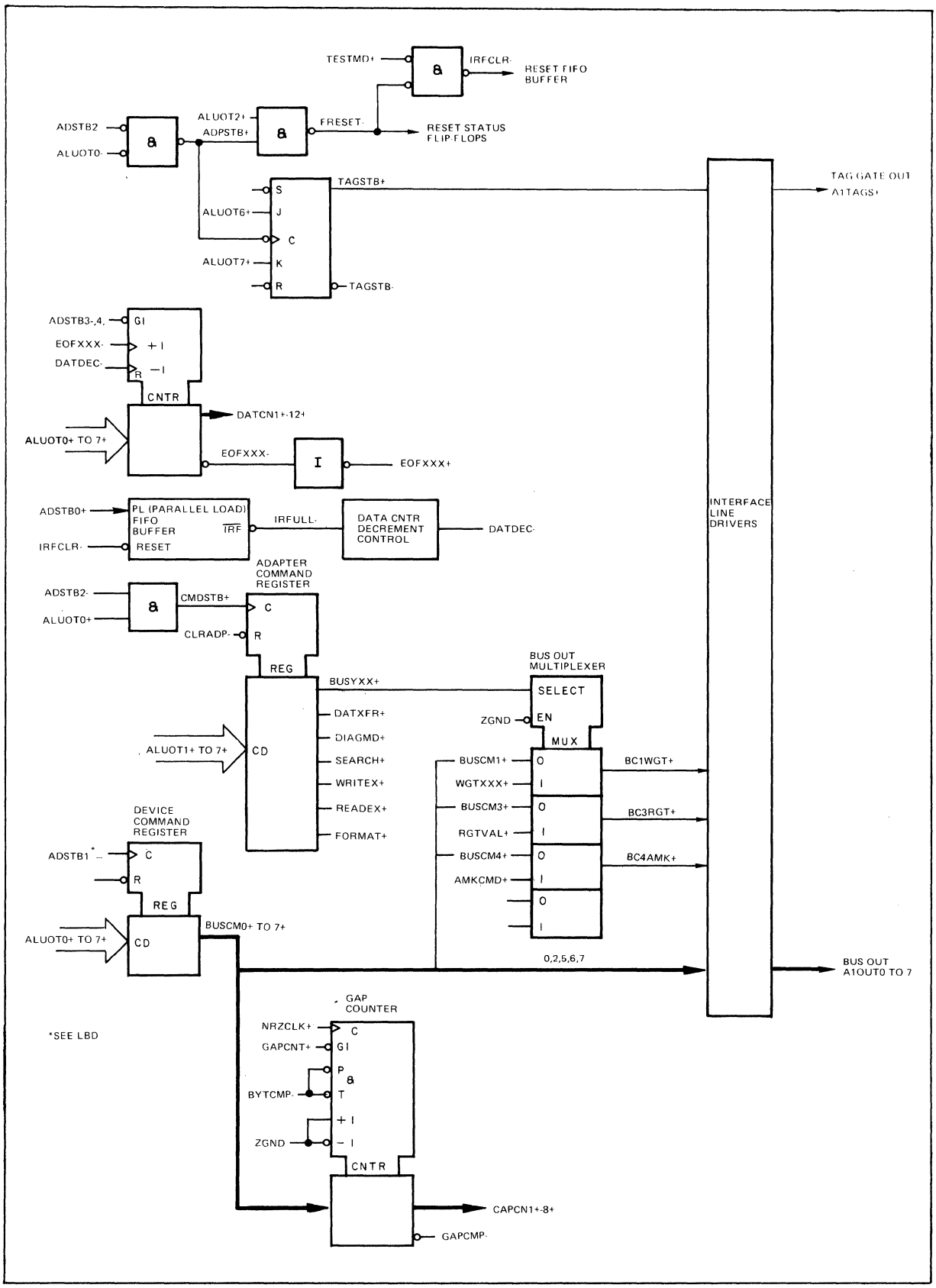


Figure 2-10 Adapter Setup for Format Write Command

Firmware next strobes (ADSTB2+) the format write command constant to the adapter via the ALU lines. Since bit zero (ALUOT0+) is now a logic Zero, the ALU lines are interpreted as an adapter command, and the constant is loaded into the adapter command register, setting the following functions:

- BUSYXX+ (Adapter Busy)
- DATXFR+ (Data Transfer)
- WRITEX+ (Write Operation)
- FORMAT+ (Format Operation)

With the adapter command register loaded, firmware sets the Tag Strobe flip-flop (TAGSTB+) via adapter strobe ADSTB2+, with ALUOT6+ set to One. This makes the Tag Gate Out (A1TAGS+) interface line true, and both the tag code and the tag out information on the Bus Out lines are strobed to the device.

Bus Out lines 0, 2, 5, 6, and 7 always reflect the content of the corresponding bits of the firmware-controlled device command register (BUSCM0+ - 7+). Bus Out lines 1, 3, and 4, which are taken from the bus out multiplexer, can reflect either the correponding bits of the device command register or the state of certain flip-flops in adapter hardware. Since the adapter has now been set busy, BUSYXX+ has conditioned the multiplexer to select the hardware conditions onto the Bus Out lines 1, 3, and 4. The tag out information strobed to the device therefore appears as follows:

Bus Out Lines A1OUTn+

0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0

└— Write Gate Bit (WGTXXX+)

The write gate bit WGTXXX+ will change to a logic One when the next index pulse is received from the device. At that time the device will enter its write mode.

Now conditioned by firmware, the adapter waits for an index pulse from the device before proceeding with the format write operation.

2.4.4.3 Write Data Serialization (Figure 2-11)

Data to be written on the selected device can reside in byte form in the FIFO buffer (sync bytes, section ID bytes, and data Zeros for Zero-filled data fields), or is generated by adapter logic (gap byte Zeros, address marks, EDC and EDAC characters). All write data is transferred to the device via the GWI Multiplexer (GWIMUX+) and the appropriate read/write data line (B*DATA+).

The FIFO buffer serial data output (DATAOS+) is selected through the multiplexer for the writing of gap Zeros. During gap cycles, serial shifting does not occur in the FIFO buffer because the Serial Data Out Clock (SDOCLK+) is blocked by the Data Cycle (DATCYC+) being low. DATAOS+ and GWIMUX+, therefore, remain false (low) for the duration of the gap cycle. Write synchronization is provided by the adapter clock (NRZCLK+) and the Write Clock (B*WCLK+), both derived from the device-generated servo clock. Because GWIMUX+ is held false, the NRZCLK+ is unable to toggle the GWI flip-flop (GWIXXX+), the read/write data line to the device remains low, and only write clock pulses are sent to the device. The device responds by writing data Zeros and Clock Ones on the medium surface.

During the address mark cycles, the same condition exists. The device writes nothing in the address mark field, however, because the adapter conditions the device to erase during the address mark cycle.

During the sync byte, sector ID and data fields DATCYC+ allows the clock (NRZCLK+) to generate Serial Data Out Clock signals (SDOCLK+) for the FIFO buffer. Serial data from the FIFO buffer is selected through the GWI multiplexer and clocked through the GWI data flip-flop by NRZCLK+. This serialized write data is sent through a line driver to become the serial write data (B*DATA+) that is sent to the device. (Note that the FIFO buffer is Zero-

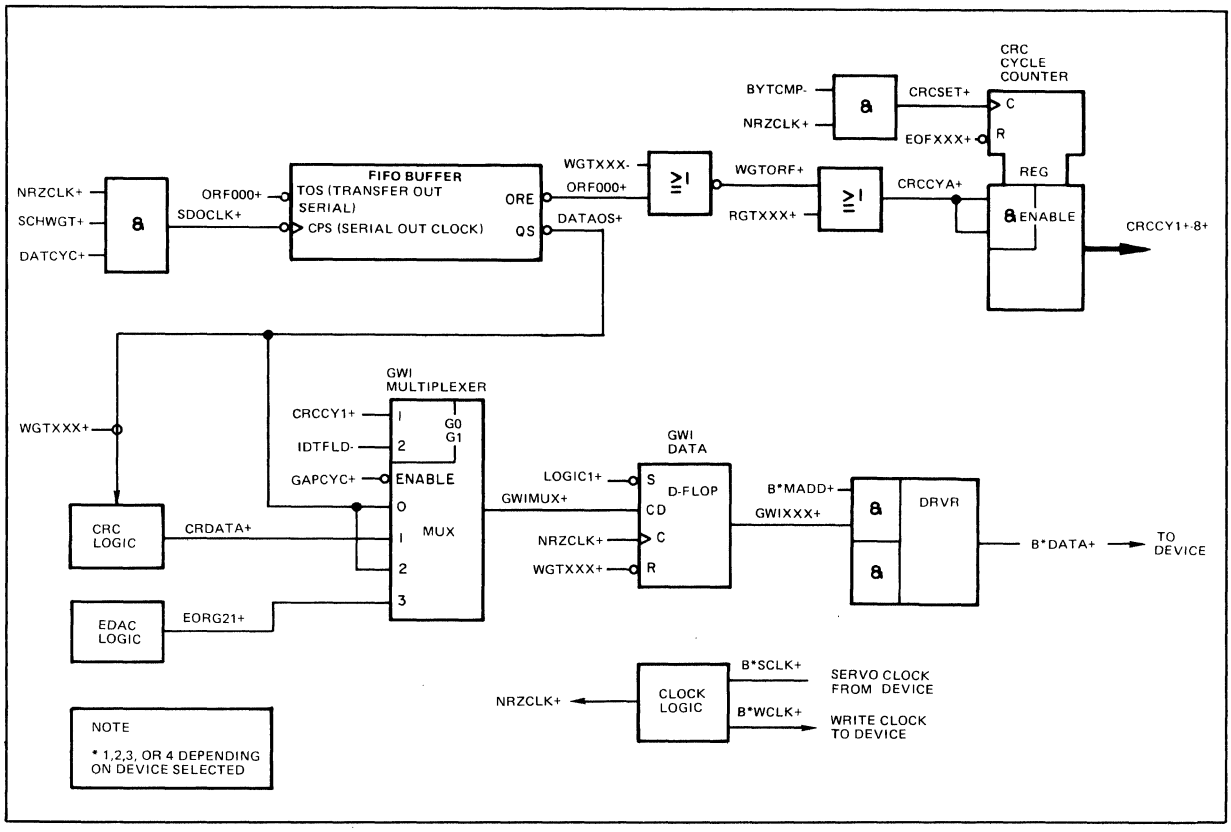


Figure 2-11 Write Data Serialization CRC Cycle Counter

loaded by shifting Zeros into the serial input of the FIFO during format write data fields, thereby forcing all Zeros to be written on the media.

Immediately after writing the sector ID field, the EDC code (CRDATA+) is selected for transfer through the GWI multiplexer to the device. Immediately after writing the data field, the EDAC code (EORG21+) is selected through the multiplexer for transfer to the device.

2.4.4.4 Write Gap Cycle (First Sector on Track) (Figure 2-12)

The following paragraphs describe adapter hardware used to write gap 0 and the first 16 bytes of gap 1. Gap 0 occurs only for the first sector on a track; all other sectors begin with a gap 1. The starting point for the first write gap cycle on a track is the index pulse received from the device.

Condition Device to Write

The leading edge of the index pulse (A1IDXX+) sets Write Gate (WGTXXX+), which is immediately reflected to the device via the bus out multiplexer (BC1WGT+) so that the Bus Out lines to the device now appear as:

Bus Out Lines A1OUTn+

0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0

└— Write Gate Bit (WGTXXX+)

The write gate signal to the device conditions it to perform a write operation.

Enable Gap Counter

Write gate coming true causes the Gap Cycle (GAPCYC+) and Gap Counter Enable (GAPCNT+) flip-flops to be set. GAPCNT+ conditions the gap counter to decrement (i.e., the gap counter cannot now be loaded from the device command register).

Load Count for ID Field-to-Data Field Gap (Gap 2)

Write gate coming true during a format write operation causes a Firmware Data Service Request (FDTSRQ+) by forcing a dummy address mark request as shown below. Firmware responds by loading a count of 12 (corresponding to the length of gap 2) in the device command register. This count is not loaded into the gap counter at this time because its enable function (GAPCNT+) is now true and the counter is conditioned to count, not to load.

Write Gap Zeros (Gap 0 and 1)

Gap Zeros and write clock pulses are sent to the device as described in subsection 2.4.4.3. The length of the gap field is controlled by the bit and gap counters as shown in Figure 2-12. For each bit sent to the device, the bit counter increments by one by counting the synchronizing clock pulse (NRZCLK+). For each group of 8 bits counted, the Byte Complete function (BYTCOMP-) comes true and, on the next clock pulse, the gap counter is decremented by one. Gap counter decrementing continues until 32 bytes of Zeros have been sent to the device (16 bytes for gap 0 and the first 16 bytes of gap 1).

2.4.4.5 Write Address Mark

When the first 32 bytes of Zeros have been written, the gap counter reaches a count of 12, causing the Address Mark Cycle flip-flop (AMKCYC+) to be set. AMKCYC+ causes the address mark enable signal to be sent to the device via Bus Out line A1OUT4+. Three bytes later, the gap counter reaches a count of 9 to reset the address mark cycle flip-flop and terminate the address mark enable signal to the device.

The device responds to the address mark enable signal by erasing for a period of time corresponding to 3 bytes (i.e., a recorded address mark consists of a 3-byte area in which no data Ones or Zeros are recorded). The address mark cycle initiates another firmware data service request, to which firmware responds by loading the device command register with a count of 12 (gap count). This is the same count loaded previously, so nothing is changed in the adapter logic.

2.4.4.6 Write Gap Zeros (Remainder of Gap 1)

With the address mark cycle finished, the adapter writes the remaining 10 bytes of Zeros in gap 1. When the gap counter reaches zero the Gap Complete function (GAPCMP-) comes true and sets the Data Cycle flip-flop (DATCYC+). This marks the end of the gap cycle and the beginning of the sector ID data cycle.

2.4.4.7 Write Sync Byte and Sector ID Fields

Data Cycle (DATCYC-) coming true causes the Gap Cycle (GAPCYC+) and the Gap Counter Enable (GAPCNT+) flip-flops to be reset, thereby terminating the gap cycle operation and initiating the data cycle operation. During the next clock pulse (NRZCLK+), the gap counter is loaded from the device command register with the previously stored count of 12 (Figure 2-10). This preconditions the gap counter for the next write gap cycle.

Data Cycle (DATCYC+) coming true enables the FIFO buffer Serial Data Out Clock (SDOCLK+) as shown in Figure 2-11. This clock shifts the 8 bits of the sync byte out of the FIFO buffer (DATAOS+) and through the GWI multiplexer. The serialized sync byte is sent to the device, together with read/write clock pulses. The 4 bytes of sector ID information are extracted from the FIFO buffer and sent to the device in the same manner.

When the last bit of the sector ID information has been shifted out, the FIFO buffer's Output Register Full function (ORF000+) goes false, thereby enabling the CRC Cycle counter to increment to CRCCY1+ when the data counters' byte complete function comes true with the next NRZCLK+ pulse. CRCCY1+ terminates the data cycle by resetting the Data Cycle flip-flop (DATCYC+).

2.4.4.8 Write EDC Field

CRCCY1+ selects the output of the CRC logic (CRCOUT+) for transfer through the GWI multiplexer to the device. The bit counter's Byte Complete function (BYTCMP-) advances the

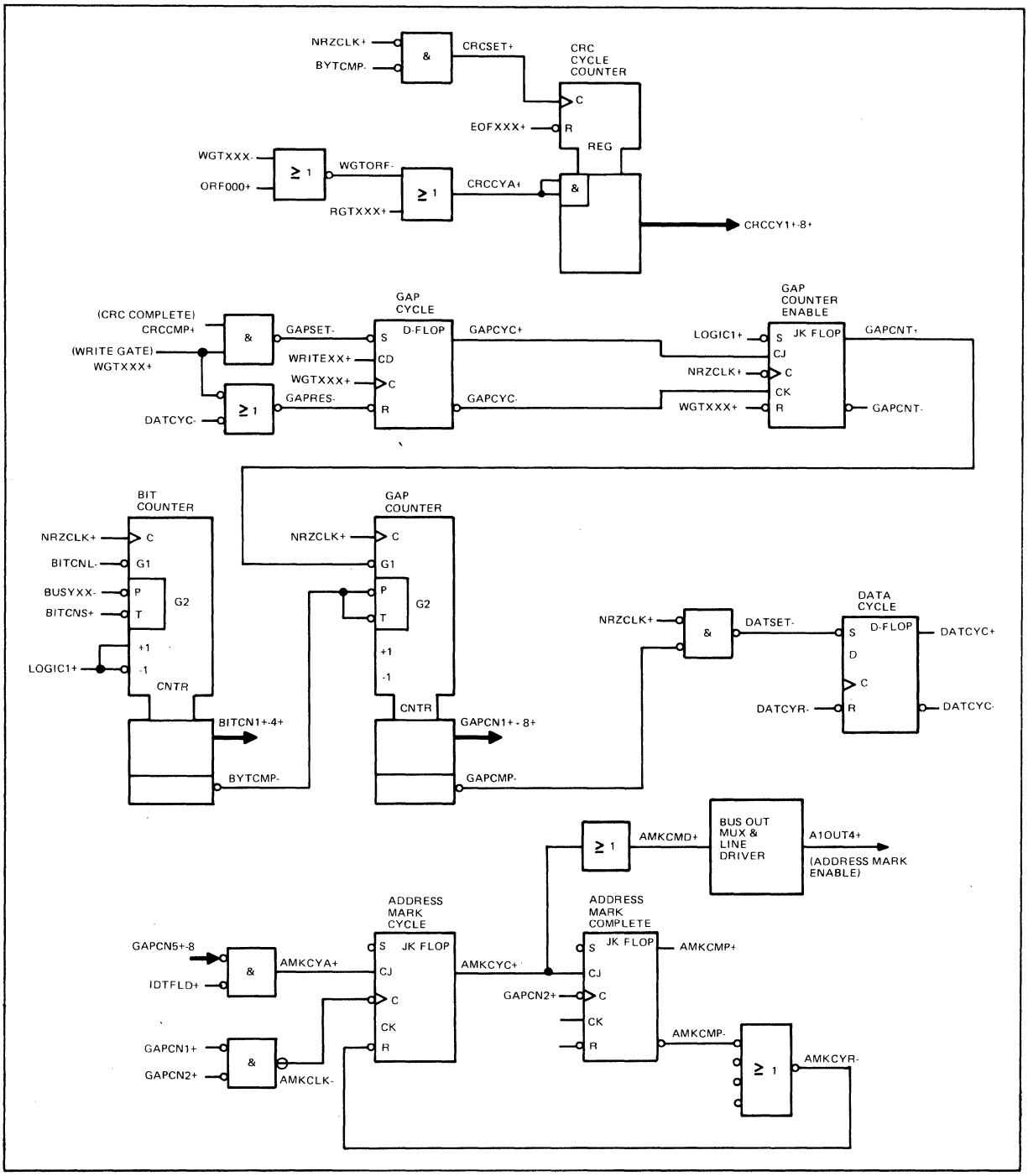


Figure 2-12 Write Gap Cycles/Enable Address Mark/Set Data Cycle

CRC cycle counter to CRCCY2+ and then to CRCCY3+ as it counts the first and second EDC bytes. CRCCY3+ marks the beginning of the next gap cycle (or the end of the CRC (EDC) field).

2.4.4.9 Write Gap Cycle (Gap 2)

CRC Complete (CRCCMP+) comes true with CRCCY3+ to set the gap cycle and gap counter enable flip-flops to initiate the next write gap cycle. CRCCY3+ also initiates another firmware data service request, to which firmware responds by loading the count for the next gap (actually, the count for gap 3 plus gap 1 of the next sector) in the device command register. This count will be loaded into the gap counter during the next write data cycle.

The adapter sends gap Zeros plus write clock pulses to the device as in the previous gap cycle operation. During this gap cycle, firmware loads the data field sync byte in the FIFO buffer using adapter strobe ADSTB0+. Firmware also loads the data counter with a value equivalent to the number of bytes in the data field.

2.4.4.10 Write Data Field

When the gap counter reaches zero, the gap cycle is terminated, the data cycle is initiated, and the Serial Data Out Clock (SDOCLK+) is enabled. Note that at this time the gap counter is loaded from the device command register with the count for gap 3 plus gap 1 of the next sector.

Because the FIFO has been Zero-loaded through the serial data input, SDOCLK shifts data Zeros (DATAOS+) through the GWI multiplexer. These data Zeros, together with write clock pulses, are sent to the device (i.e., during a format write operation, the data fields are Zero-filled).

The data counter monitors the number of data field bytes sent to the device. When the last Zero bit of the last byte has been shifted out, the FIFO buffers' Output Register Full function (ORF000+) enables the CRC cycle counter which then proceeds to increment to CRCCY1+ when the data counters' Byte Complete function (BYTCMP-) comes true with the next NRZCLK+ pulse. CRCCY1+ terminates the data cycle and attendant SDOCLK+ clock pulses.

2.4.4.11 Write EDAC Field

CRC cycle 1 (CRCCY1+) and IDTFLD- select the output of the EDAC logic (EORG21+) for transfer through the GWI multiplexer and to the device. The CRC cycle counter is incremented by one as the bit counter indicates the transfer of each EDAC byte via its Byte Complete function (BYTCMP-). Transfer of the seventh EDAC byte advances the counter to CRCCY8+, signaling the end of the EDAC field.

2.4.4.12 Write Gap 3

When the CRC cycle counter has counted the last EDAC byte, CRCCY8+ causes the gap cycle and the gap counter enable flipflops to be set, thereby initiating another write gap cycle. During this write gap cycle, gap 3 of the present sector and gap 1 of the next sector are written.

2.4.4.13 Terminate Format Write Operation

Each sector of the track is written as described in the foregoing paragraphs until the second index pulse (INDEX2+) occurs, at which time a Nondata Service Request (NDTSRQ+) is sent to the controller. If the second index pulse occurs before the range expires, firmware immediately resets the adapter command register and the tag strobe flip-flop (i.e., tag gate out to the device) to terminate the format write operation. Normally the range expires just before the second index pulse, and gap Zeros are written to the end-of-track. Firmware then responds to the second index pulse by resetting the adapter command register and the tag strobe flip-flop to terminate the format write operation. This condition also exists if the specified range is less than normal; for example, one or two sectors can be written with the remainder of the track being filled with gap Zeros.

2.4.5 Search and Write Operation

This subsection describes adapter hardware used when the MPDC is performing a search and write operation. Firmware sets up the adapter, loads the sought-for sector ID in the FIFO buffer, strobes a read gate signal to the device enabling it to begin reading, and then waits for detection of an address mark. The device indicates detection of an address mark via bit zero (address mark found) of the Bus In interface lines (A1BIN0+). A firmware data service request is sent to the MPDC when an address mark is found as an alert to firmware that a sector ID field is about to be read.

As reading continues, the adapter detects the sector ID sync byte and begins to compare the sector ID being read with the sector ID in the FIFO buffer. If the sector ID being read is not the one being sought, firmware reloads the search argument and continues the search in the next sector. If the desired sector ID is found, a sync byte plus write data is strobed into the FIFO buffer and adapter hardware automatically switches both itself and the device from read mode to write mode.

During the search (read), the CRC logic develops an EDC code on the sector ID being read and compares it with the recorded EDC code. An unequal comparison indicates a read error in the sector ID field.

During write mode, gap field Zeros are supplied to the device by adapter hardware, while sync bytes and data bytes are extracted from the FIFO buffer. The FIFO buffer attempts to keep itself full by making hardware data service requests whenever its input register is empty. A 7-byte Error Detection and Correction (EDAC) code is developed on the data being written. This code is sent to the device upon termination of the data field.

The adapter's CRC cycle counter keeps track of the number of EDC and EDAC bytes, sets various device flip-flops to control reading or writing, and initiates firmware data service requests as required.

2.4.5.1 Search and Write Command Timing

A detailed timing diagram for adapter operation is provided in Appendix B and, together with Figures 2-13 through 2-15, is the basis for the following discussion of adapter logic.

2.4.5.2 Adapter Setup for Search and Write Operation

Execution of a search and write operation begins after firmware has set up the adapter logic. Firmware-controlled adapter strobes (ADSTBn) set up the adapter as follows:

1. ADSTB3+, 4+	Load data counter with a value of 4, which is the length of the sector ID field in bytes
2. ADSTB2+	Clears FIFO buffer and status flip-flops
3. ADSTB5+	Loads control tag code (code = 111) in tag register
4. ADSTB2+	Loads adapter command register, setting the following flip-flops: a. BUSYXX+ b. SEARCH+ c. WRITEX+
5. ADSTB2+	Sets Tag Strobe flip-flop TAGSTB+, which sends Tag Strobe A1TAGS+ to the device
6. ADSTB0+	Loads search argument into FIFO buffer

2.4.5.3 Search for Address Mark (Figure 2-13)

The adapter begins the search and write operation by conditioning the device to read, and then waits for the device to indicate that an address mark has been found. With search mode set, SEARCH+ sets the Read Gate flip-flop (RGTXXX+) as shown in Figure 2-13. Read gate causes the Address Mark search flip-flop (AMKSCH+) to be set and raises the read gate signal (A1OUT3+) to the device. The device begins reading the selected track and indicates detection of an address mark via the Bus In line A1BIN0+ (Address Mark Found). The address mark search flip-flop is reset when the mark is found.

2.4.5.4 Wait for Sync Byte (Figure 2-14)

After an address mark is found, reading continues through the remainder of gap 1, with serial read data (NRZDAT+) being clocked into the serial read data register by the clock

NRZCLK+. This register is examined for the occurrence of a sync byte (as indicated by the presence of the first One bit in the sync byte). When a sync byte is found, the sync byte comparator output (SYNC19+) goes true to set the Found One flip-flop (FONEXX+). FONEXX+ in turn sets the Data Cycle flip-flop (DATCYC+) and preloads the bit counter with a value of 8.

2.4.5.5 Read and Compare Sector ID Field (Figure 2-15)

Setting the data cycle flip-flop enables the FIFO buffer's Serial Data Out Clock (SDOCLK+) and the previously loaded sector ID begins shifting out of the buffer (DATAOS+) to be compared with the sector ID being read from the device. As long as the ID bits being compared are equal, the exclusive-OR function DATUEQ+ remains low. If any pair of compared bits is unequal, DATUEQ+ goes high, causing the Search Error flip-flop (SCHERR+) to be set. Although a search error is reported in adapter status, it does not indicate an adapter or device failure.

As the sector field is being read, the bit counter counts clock pulses from the device and generates a Byte Complete signal (BYTCMP-) for each byte read. Each time the bit counter reaches a count of three (BITCN3-), it decrements the data counter (DATC01+ through 12+) until its End-of-Field function (EOFXXX+) comes true when the counter reaches zero.

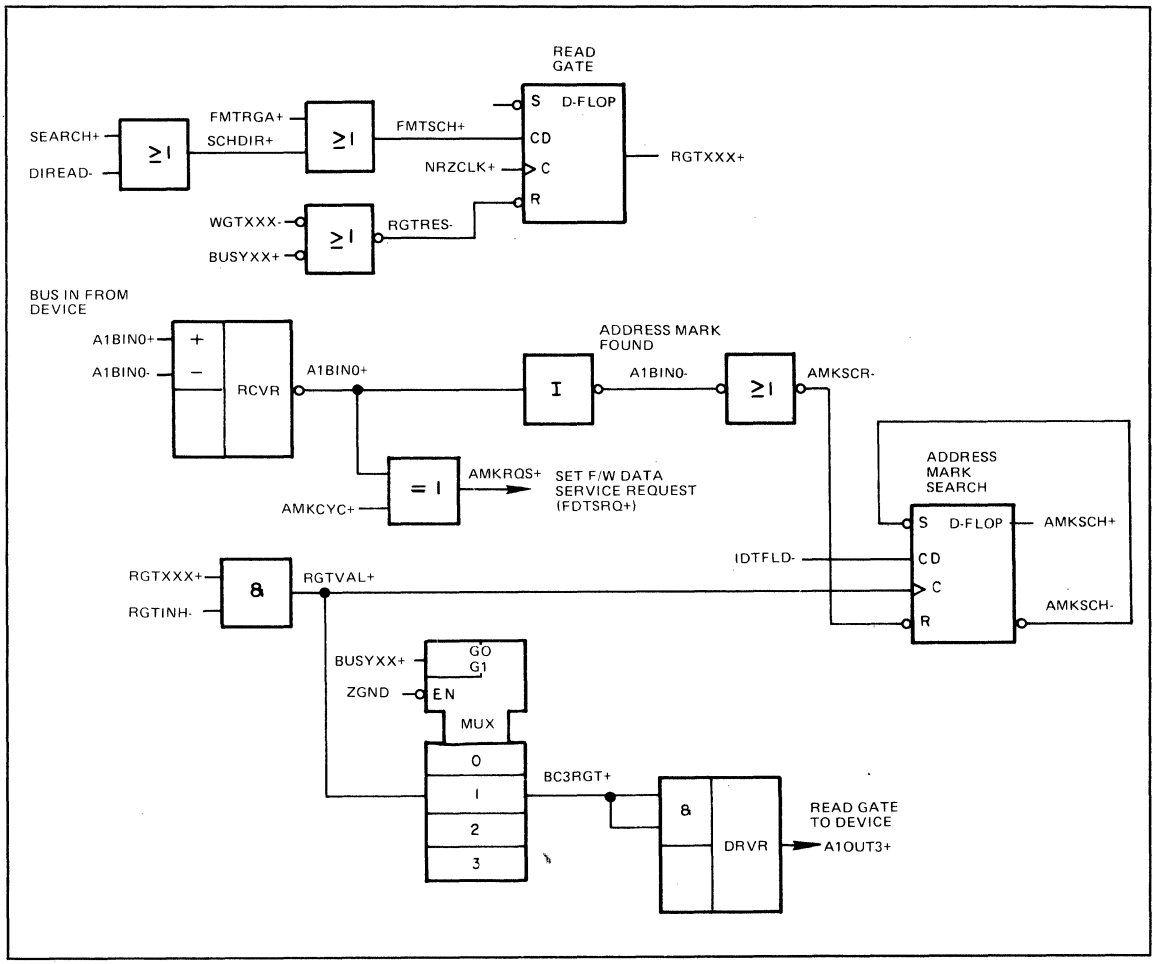


Figure 2-13 Read Gate/Address Mark Search

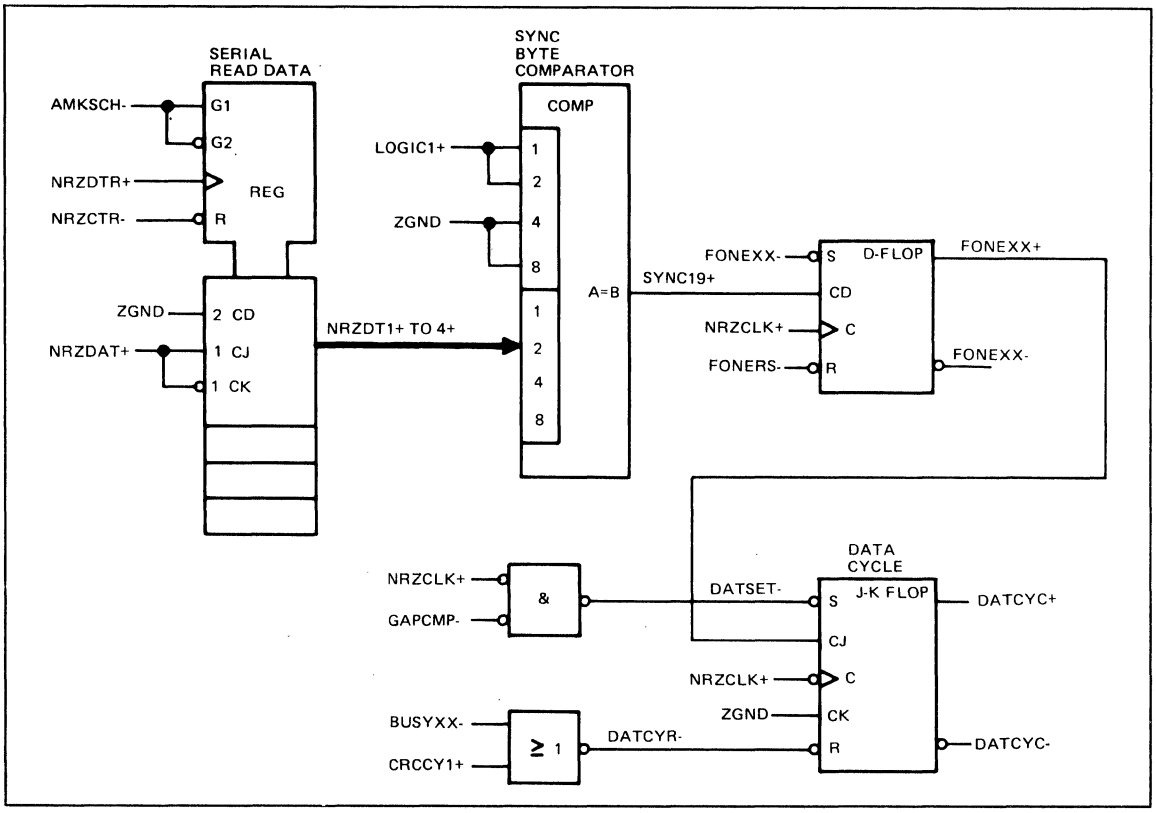


Figure 2-14 Sync Byte Detection

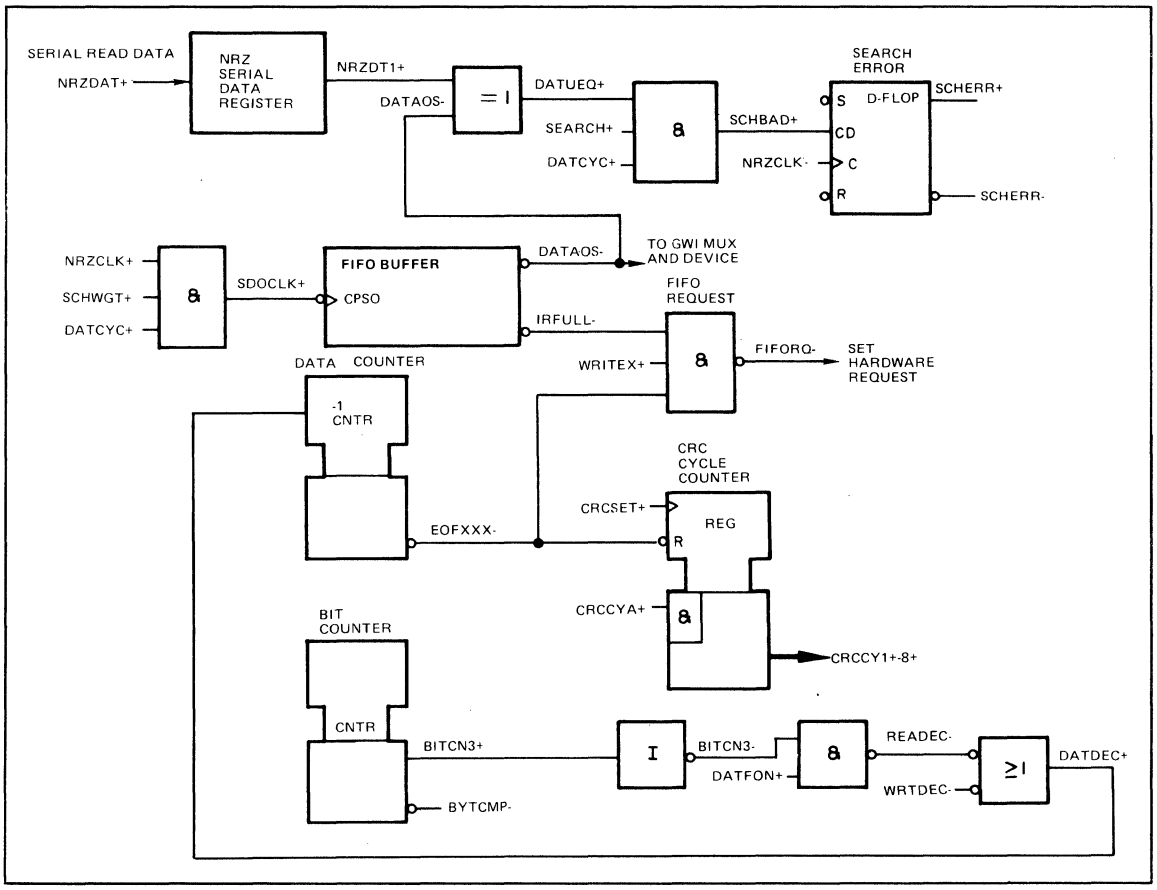


Figure 2-15 Compare Sector ID/FIFO Requests/CRC Cycle Counter

2.4.5.6 Read and Compare EDC Code

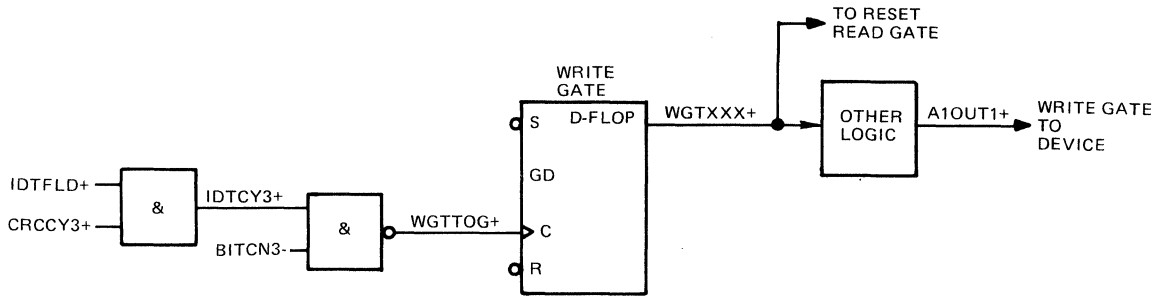
The data counter reaches zero and EOFXXX- comes true when the third bit of the last sector ID byte is counted by the bit counter. EOFXXX+ going high removes the forced reset from the CRC cycle counter, enabling it to increment to CRCCY1+ when the bit counter signals completion of the last sector ID byte (BYTCMP-). CRCCY1+ terminates the data cycle by resetting the Data Cycle flip-flop DATCYC+.

As the EDC code is read, the CRC cycle counter is incremented to CRCCY2+ and CRCCY3+ as the bit counter Byte Complete function (BYTCMP-) counts EDC code bytes 1 and 2, respectively. During this time, the CRC logic compares the recorded EDC code with the EDC code developed on the sector ID field just read. An unequal comparison indicates a read error (see subsection 2.4.7).

2.4.5.7 Write Gap (Correct ID Found, No Read Error)

CRC cycle 3 (CRCCY3+) indicates completion of the read portion of the search and write operation. When CRCCY3+ comes true, the Gap Cycle flip-flop GAPCYC+ is set and a firmware data service request is made to request the sync byte and the data field information. Firmware responds during the write gap operation by loading the requested information into the FIFO buffer (the FIFO buffer can be loaded while the gap is being written because gap Zeros are generated by adapter hardware, thereby leaving the buffer free to be loaded).

When the bit counter reaches a count of three (BITCN3-) at the beginning of CRC cycle 3, the Write Gate flip-flop (WGTXXX+) is set and the write gate signal is sent to the device via Bus Out line A1OUT1+, and read gate is reset. (See the following illustration.)



Set Write Gate, Reset Read Gate

If a Search Error (SCHERR+) occurred while reading the sector ID field, the IDTFLD+ flip-flop is reset at the beginning of CRC cycle 4 (CRCCY4+) and the write gate flip-flop will not be set.

The gap counter keeps track of the number of gap bytes and generates a Gap Complete signal (GAPCMP-) when the specified number of gap bytes has been written. GAPCMP- coming true results in the setting of the data cycle flip-flop and the resetting of the gap cycle flip-flop. Gap counter operation is as described for the format write operation.

2.4.5.8 Write Sync and Data Fields

The sync byte and data field information is extracted from the FIFO buffer and sent through the GWI multiplexer to the device as described in subsection 2.4.4.3. The FIFO buffer attempts to keep itself full during the write data field by sending frequent data service requests to the MPDC. As soon as a byte is loaded into the FIFO buffer, it drops from the input register into the FIFO stack (unless the FIFO is full). When this occurs, the Input Register Full function (IRFULL-) returns high and, if the end-of-field has not been reached, a hardware data service request is sent to the MPDC to request another byte of data.

Assuming the range to be correct, this operation continues until the data counter reaches zero (i.e., until End-of-Field function EOFXXX+ comes true) indicating that the last byte of the data field has been sent to the device. EOFXXX- thereafter prevents further data service requests by the FIFO buffer.

2.4.5.9 Write EDAC Field

Data being written is sent through the EDAC logic where a 7-byte Error Detection and Correction Code (EDAC) is developed. At the end of the data field, the CRC cycle counter is enabled and the EDAC code is selected by the GWI multiplexer for transfer to the device. The CRC cycle counter advances once for each EDAC byte, and reaches CRCCY8+ when the final EDAC byte has been transferred. CRCCY8+ sets the Gap Cycle flip-flop (GAPCYC+) for writing a 1-byte gap as a postamble, resets the Write Gap flip-flop (WGTXXX+), and initiates a firmware data service request to obtain the next adapter command.

2.4.6 Read Operations (Figures 2-16 and 2-17)

An overview of the functional flow of a format read operation is portrayed in Figures 2-16 and 2-17. Figure 2-16 shows the se quential steps of the operation, while Figure 2-17 shows data flow through, and control of, relevant adapter logic. Refer to the timing diagram in Appendix B and to the adapter reference manual for additional details.

In a format read (ID and data) operation, firmware sets up the adapter and initiates reading in the device. Upon detection of an index pulse, one complete formatted track is read. The adapter monitors Bus In line 0 from the device for an indication that an address mark has been found. The serial read data stream is monitored for sync bytes which define the beginning of the sector ID and the data fields. An EDC code is developed on the sector ID field and compared with the recorded EDC code for possible read errors in the ID field. An EDAC code is developed on the data field and compared with the recorded EDAC code for possible errors in the data field.

Software specifies the length of data fields (256 or 2304) and sector ID fields by placing the required field length in the data counter. The data counter is decremented once for each ID or data byte received, and reaches zero (end-of-field) when the last byte of the field has been detected. The CRC cycle counter keeps track of the number of EDC and EDAC bytes, controls the setting of error indicators, and initiates firmware requests at the end of these fields.

During a diagnostic mode format read operation, the adapter is set up by firmware as described above, except that Diagnostic Mode (DIAGMD+) is also enabled. In this operation, reading begins with the sector ID field following the first address mark detected.

2.4.7 CRC (EDC) Operation

An overview of the functional flow of EDC code development during write and read operations is shown in Figures 2-18, 2-19, and 2-20 and described in the following paragraphs. Refer to the timing diagram in Appendix B and to the adapter reference manual for additional details.

The CRC logic develops and checks a 2-byte Error Detection Code (EDC) on sector ID fields. During write operations, the CRC shift register is cleared while writing the address mark. When the sector ID field is reached, the EDC code is generated by half-adding the output of the CRC shift register (CRCOUT+) with the sector ID (includes sync byte) being written (CRCDAT+). The half-add result (CRCXOR+) is then placed in the shift register. After the EDC code has been developed, the cycle counter is set to CRCCY1+ to select the EDC code (CRDATA+) for transfer to the device via the GWI multiplexer.

During read operations, the CRC shift register is cleared while searching for the address mark (AMKCMD-). When the sector ID field is reached, the EDC code is generated by half-adding the CRC shift register output (CRCOUT+) with the sector ID being read (CRCDAT+). The Error flip-flop (IDTERR+) is enabled when CRCCY1+ comes true at the end of the sector ID field. As reading continues, the recorded EDC code is half-added with the EDC code just developed. The two codes should be identical. If any pair of bits is unequal, the exclusive-OR function (CRCXOR+) goes high and the error flip-flop is set on the next clock pulse. Errors are posted in adapter status (CRCERR+). A firmware request is generated when the cycle counter reaches CRCCY3+, and firmware responds by reading adapter status and checking for CRCERR+.

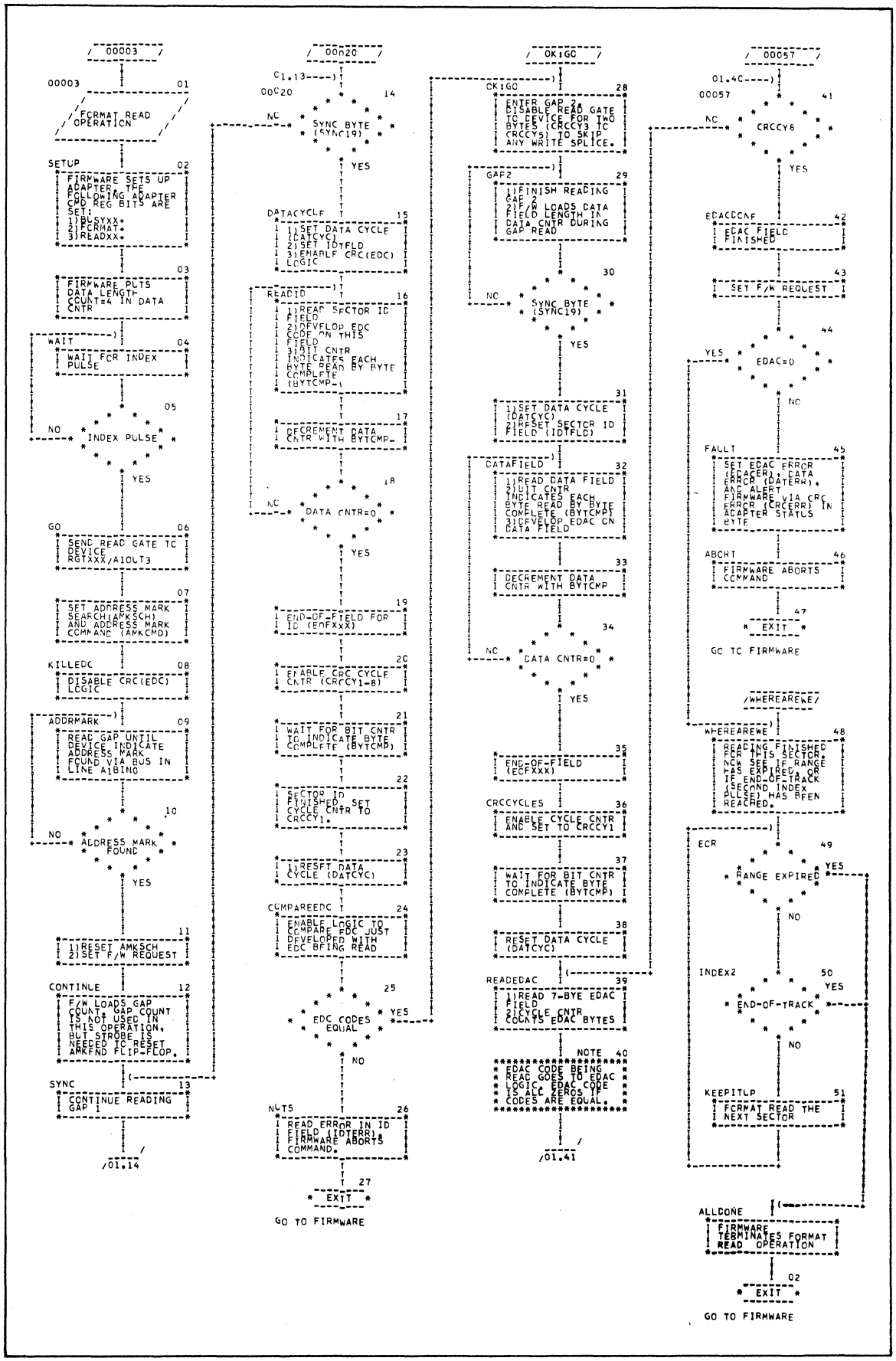


Figure 2-16 Format Read Flow Chart

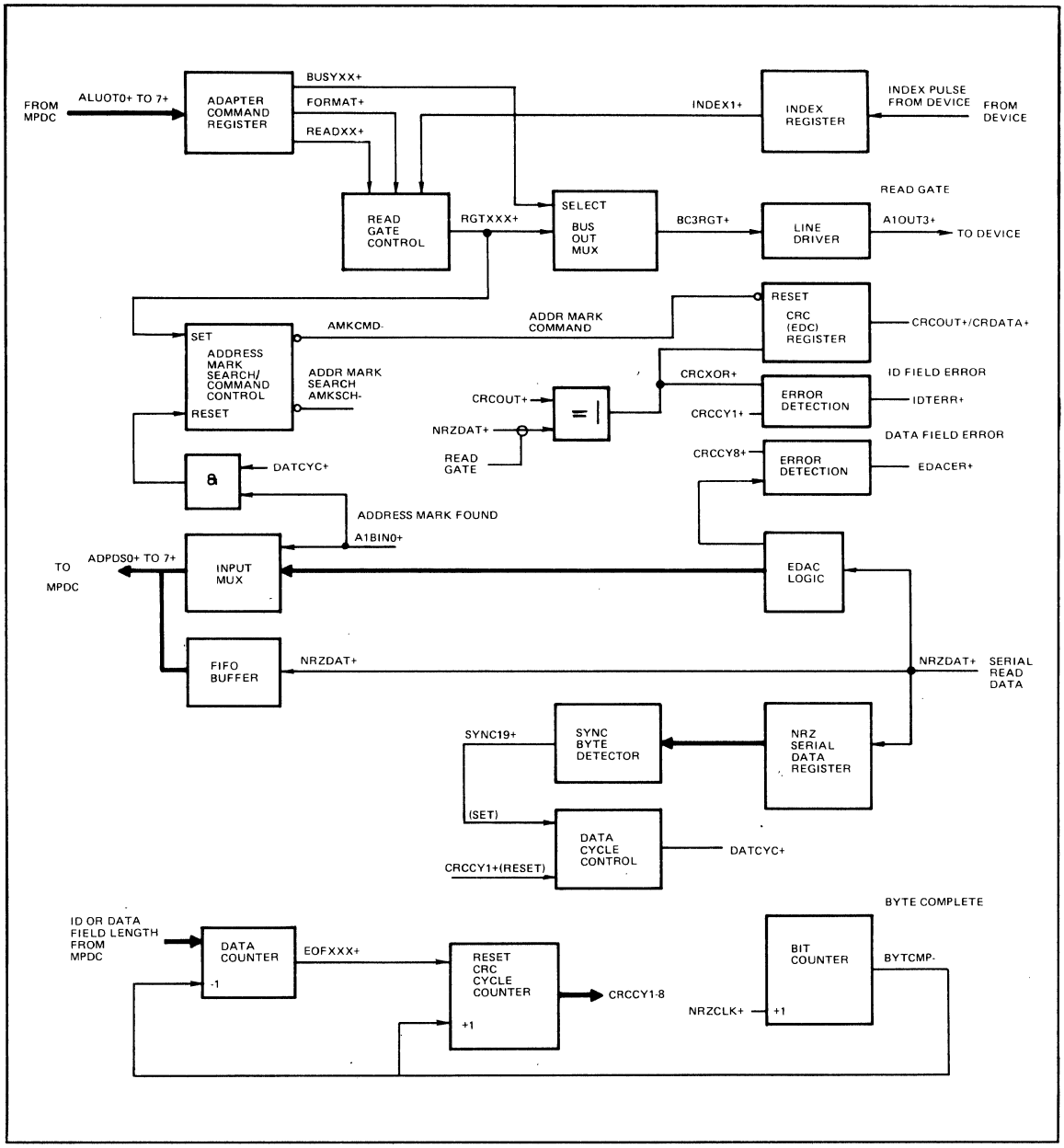


Figure 2-17 Format Read Logic

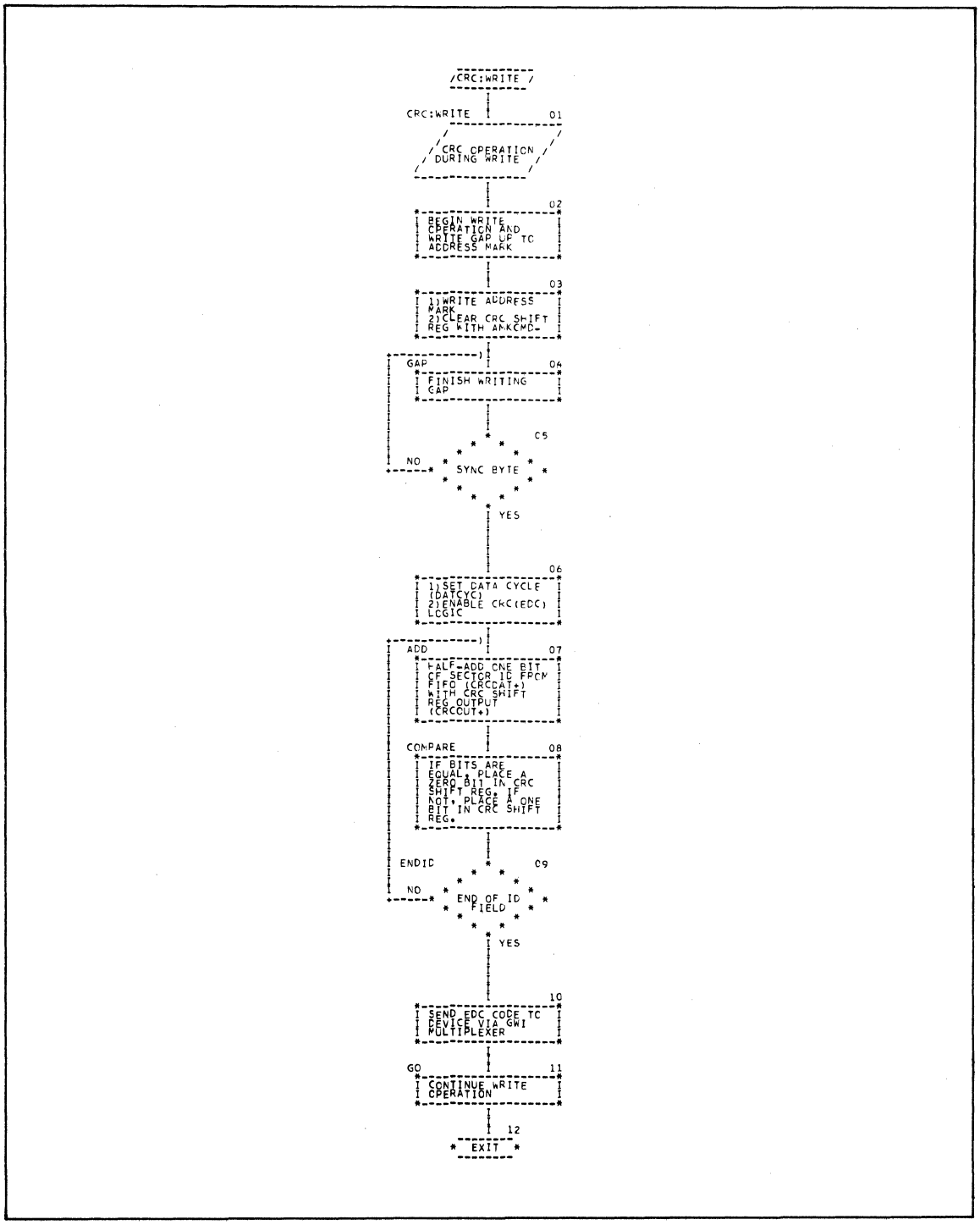


Figure 2-18 EDC During Write Flow Chart

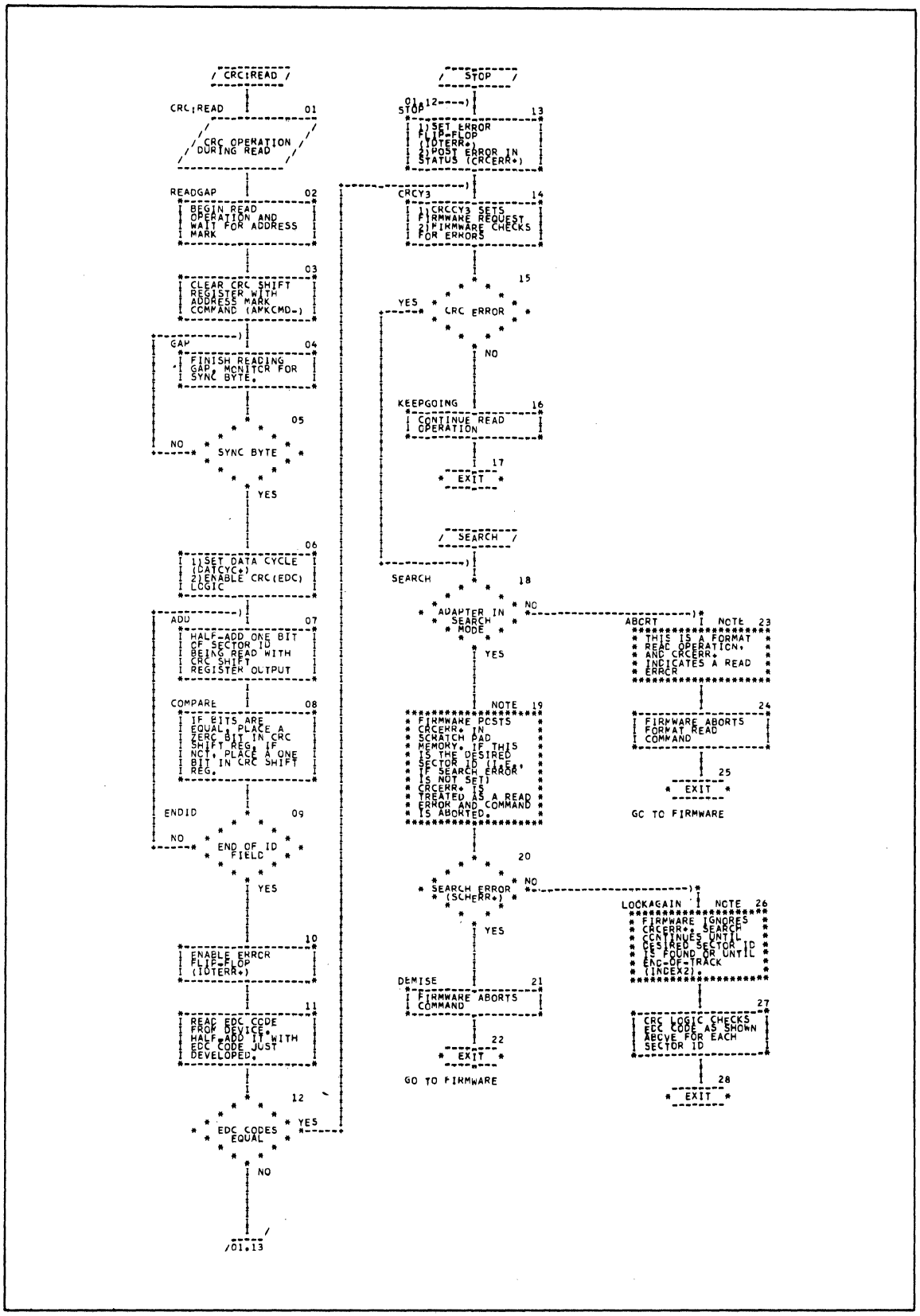


Figure 2-19 EDC During Read Flow Chart

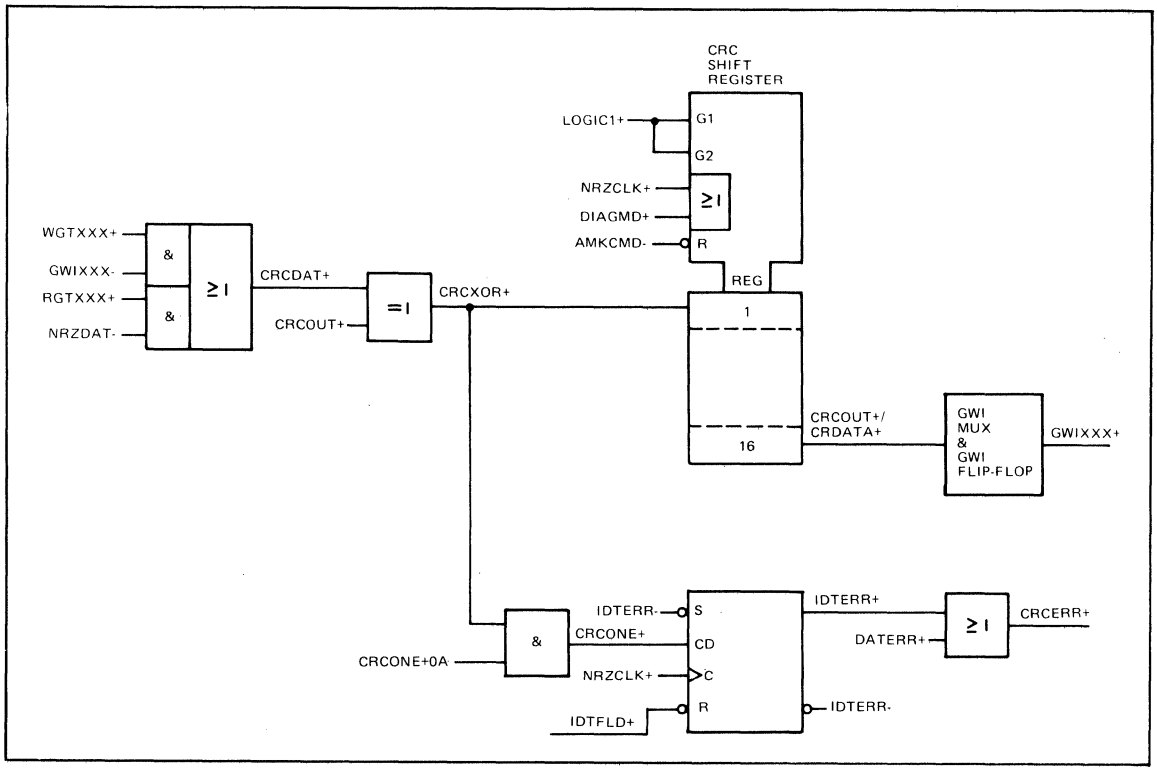
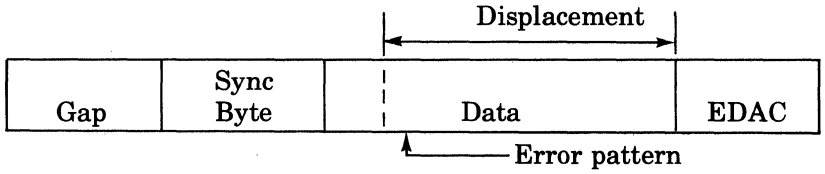


Figure 2-20 EDC Code Generation Logic

2.4.8 EDAC Operation

Data field integrity is assured by a 7-byte Error Detection and Correction (EDAC) code that can detect all errors, and can correct any single error burst of 11 bits or less. The EDAC code is hardware generated during a write operation and is appended to the data field on the media. During a read operation, an EDAC code is generated on both the data field and the EDAC field. The EDAC polynomial is such that the resultant EDAC code should be zero. If an error occurs, the EDAC code is analyzed by firmware to determine if the error is correctable. If it is, firmware uses the EDAC code to determine an error pattern and displacement of the error from the end of the data field as shown below.



During a write operation, the EDAC code is generated by connecting the four EDAC shift registers in series with feedback to make a single polynomial generator (Figure 2-21). Feedback is exclusive ORed into selected register positions with the input from the position to the left. Serial write data (GWIXXX+) is shifted through registers E3, E2, E1, and E0 to generate a 7-byte EDAC code. At the end of the data field, the write operation continues with the EDAC code being serially shifted out of the registers and sent to the device.

During a read operation, serial read data (NRZDAT+), including the recorded EDAC code, is shifted, with feedback, through the four shift registers connected in parallel (Figure 2-22). At the completion of the read operation, firmware checks each register to determine if the contents are zero. If all four registers contain zero, there is no error. If an error is detected, firmware checks to determine if it is correctable. A correctable error is defined as one in which register E0 contains a maximum error pattern of 11 bits, all four registers contain bits and can be shifted to match the pattern in register E0, and the displacement is within the data field.

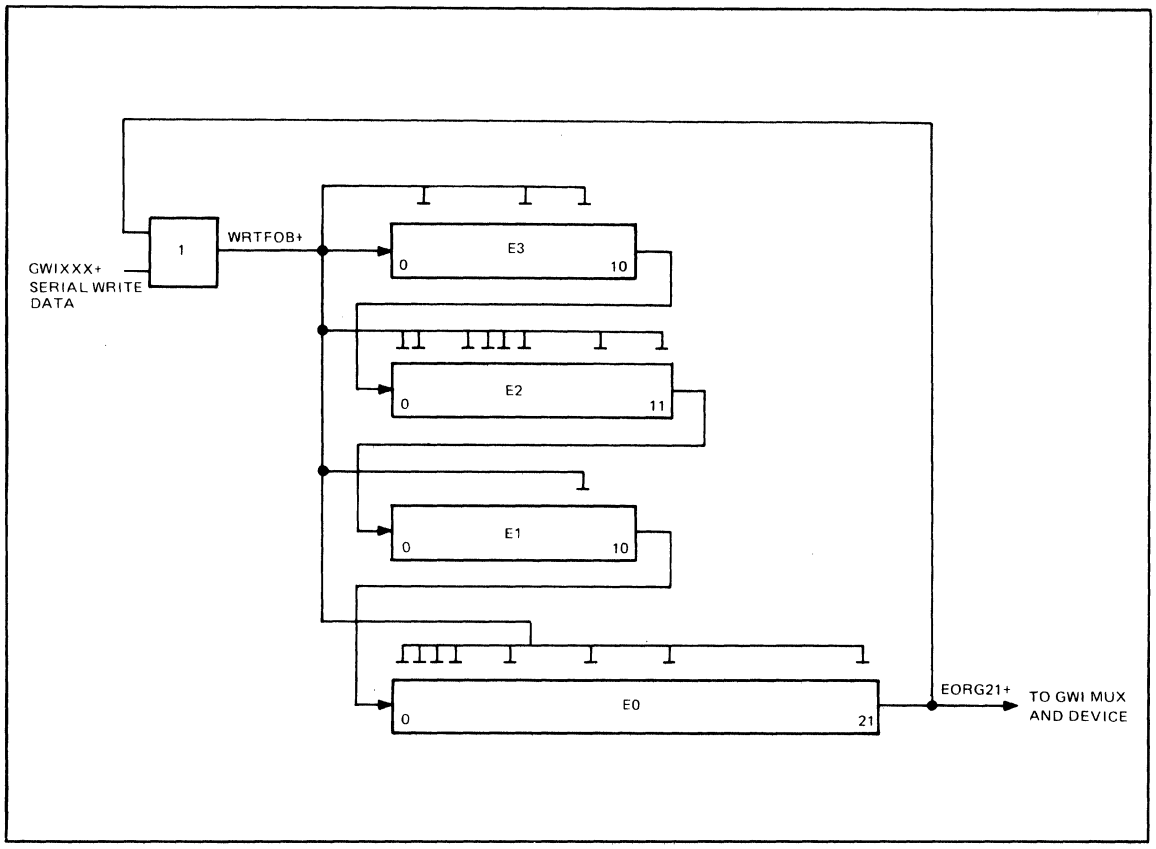


Figure 2-21 EDAC Generation During Write

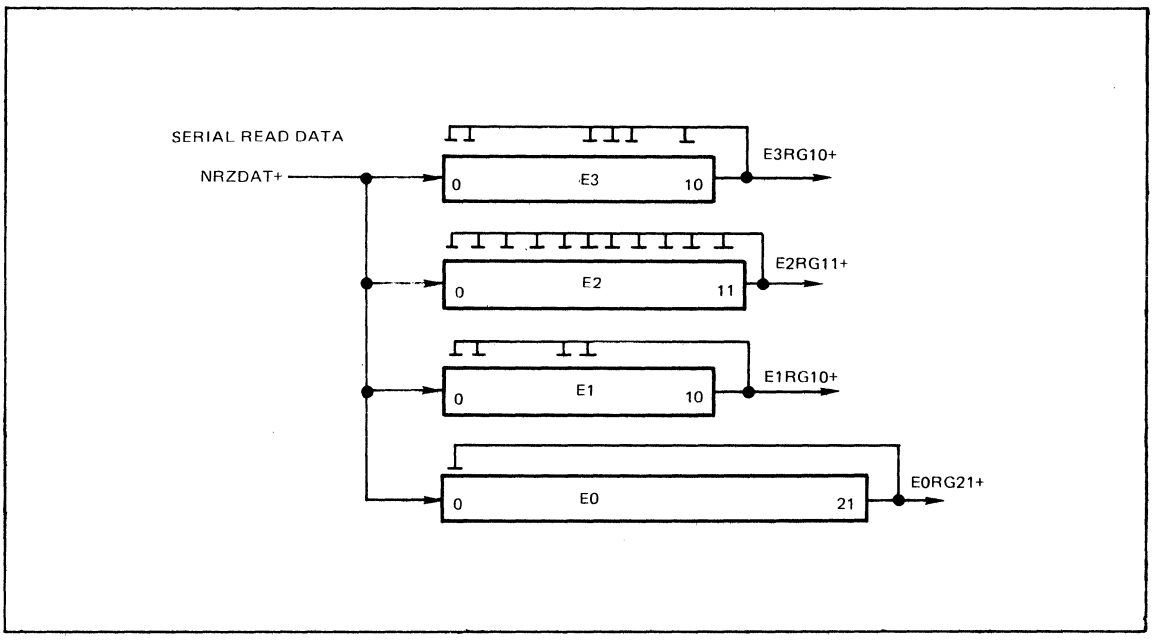


Figure 2-22 EDAC Generation During Read

2.4.9 Error Detection

Error conditions detected in adapter logic and reported in status via the input multiplexer are shown in Figure 2-23 and described below.

1. RWTERR+ Read/Write Clock Error. Indicates loss of one or more clock pulses while the adapter is busy.
2. OURERR+ Our Error. Indicates either a transfer rate error or a firmware slip error (firmware failure to condition adapter for the next field during a write operation).

   During a write operation, a transfer rate error occurs if the FIFO buffer output register becomes empty before the end-of-field is reached. During a read operation, a transfer rate error occurs if the FIFO buffer input register becomes full before end-of-field is reached.
3. CHKDIA+ Check Diagnostic. Indicates that a check diagnostic condition exists in the selected device.
4. CRCERR+ Indicates that a read error condition has been detected by the EDC code on the sector ID field (IDTERR+), or by the EDAC code on the data field (DATERR+).
5. SCHERR+ Search Error. Indicates, during a search operation, that the sector ID being read is not the same as the sought-for sector ID contained in the FIFO buffer.
6. SYNERR+ Sync Error. Indicates that the first data One bit has been detected in the sync field, but that it is not part of a sync byte.
7. INDEX2+ Index 2. Indicates occurrence of a second index pulse.

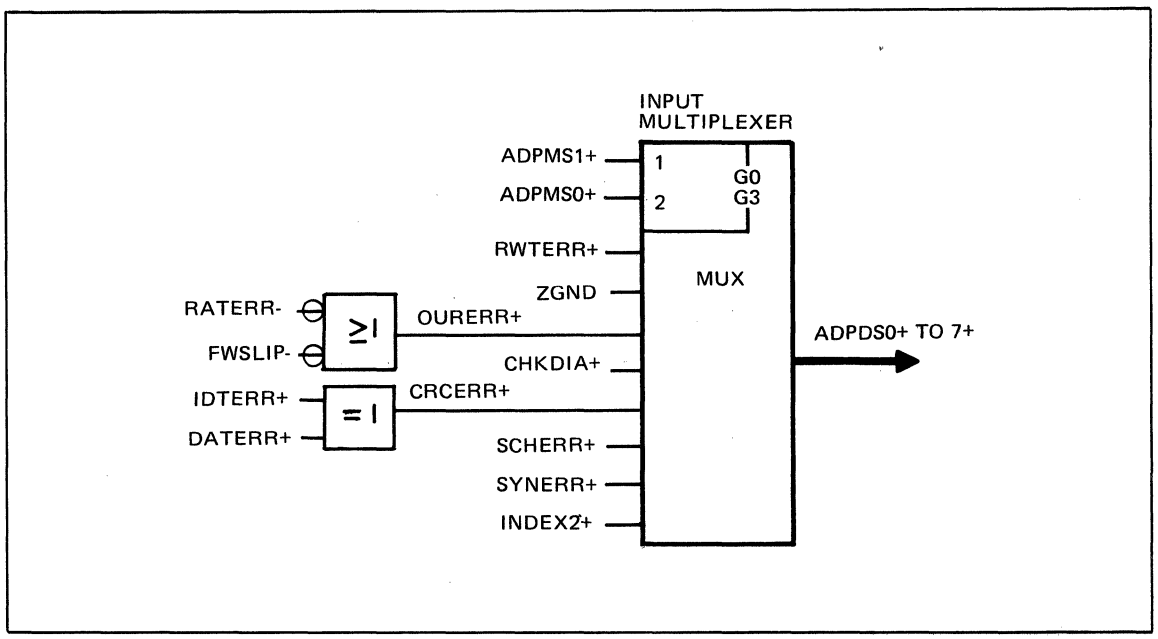


Figure 2-23 Error Conditions

2.4.10 Test Mode Operation (Figure 2-24)

The adapter test mode is used during firmware execution of Basic Logic Tests (BLTs) to verify the operation of adapter logic. Three test mode operations are performed:

- Test mode write
- Test mode read with no errors
- Test mode read with a correctable read error.

2.4.10.1 Test Mode Write (Format Set)

The following paragraphs, summarizing adapter logic operation during a test mode write operation, are based on Figure 2-24 and the detailed timing diagram contained in Appendix B.

Firmware sets up the adapter for a test mode write operation by clearing the FIFO buffer and status flip-flops. A sync byte and one data character (C5 hex) are then loaded into the FIFO buffer and allowed to fall through to the bottom of the FIFO register stack. A gap count of 29 decimal is then loaded into the gap counter and firmware loads the adapter command register with the following bits set:

- FORMAT+
- BUSYXX+
- WRITEX+

Once the data counter indicates end-of-field, firmware uses adapter strobe 5 to set the Test Mode flip-flop (TESTMD+).

As shown in Figure 2-24, the FIFO buffer consists of several FIFO chips configured to perform as a 32 x 8-bit register stack. Normally, the upper half of the FIFO loads the lower half to provide a continuous 32-byte fall-through stack. With test mode set, the parallel-load input of the lower half of the FIFO is disabled, thereby configuring the FIFO as two separately controllable 16-byte stacks.

Firmware, in test mode, forces a dummy index pulse which sets the adapter's write gate and gap cycle flip-flops, and causes the adapter to think that the beginning-of-track has occurred. Since FORMAT+ is set, the adapter logic behaves in a manner similar to a format write operation. Data is not sent to the device, however, because no device is selected during a test mode operation and the device, therefore, is not enabled (select hold low).

The lower FIFO output (DATAOS+) is selected for transfer through the GWI multiplexer. Because the FIFO Serial Data Out Clock (SDOCLK-) is not allowed during gap cycles, the GWI flip-flop is not toggled and GWIXXX+ (serial write data) remains a steady-state Zero. This information is sent through the EDAC logic, which does not change state because it is receiving zeros. When 16 bytes of gap Zeros have been counted by the adapter's gap counter, an address mark request and a firmware data service request are generated, as occurs during a format write operation. Following the 3 bytes of address mark, 10 more bytes of Zeros are sent through the GWI multiplexer and EDAC logic to complete the gap cycle.

The adapter now enters a data cycle during which the Serial Data Out Clock (SDOCLK-) and the Serial Data In Clock (SDICLK-) are enabled. The sync byte and data byte (C5 hex) are extracted from the lower FIFO and sent through the GWI multiplexer to the EDAC logic where the EDAC code is developed. The sync and data bytes are clocked into the upper half of the FIFO (NRZDAT+) by the serial data in clock. At the end of the data cycle, the sync and data bytes are in the upper half of the FIFO, the 7-byte EDAC code is in the EDAC logic, the serial data out clock is disabled, and the CRC cycle counter is enabled and set to CRCCY1+.

CRCCY1+ selects the EDAC logic (EORG21+) for transfer through the GWI multiplexer. The 7-byte EDAC code is sent through the read input control logic (NRZDAT+) and clocked into the upper half of the FIFO buffer by the serial data in clock. At the end of the test mode write operation, the sync byte, data byte (C5 hex) and 7 bytes of EDAC are located in the upper half of the FIFO buffer, from which they will be extracted and used during subsequent test mode read operation.

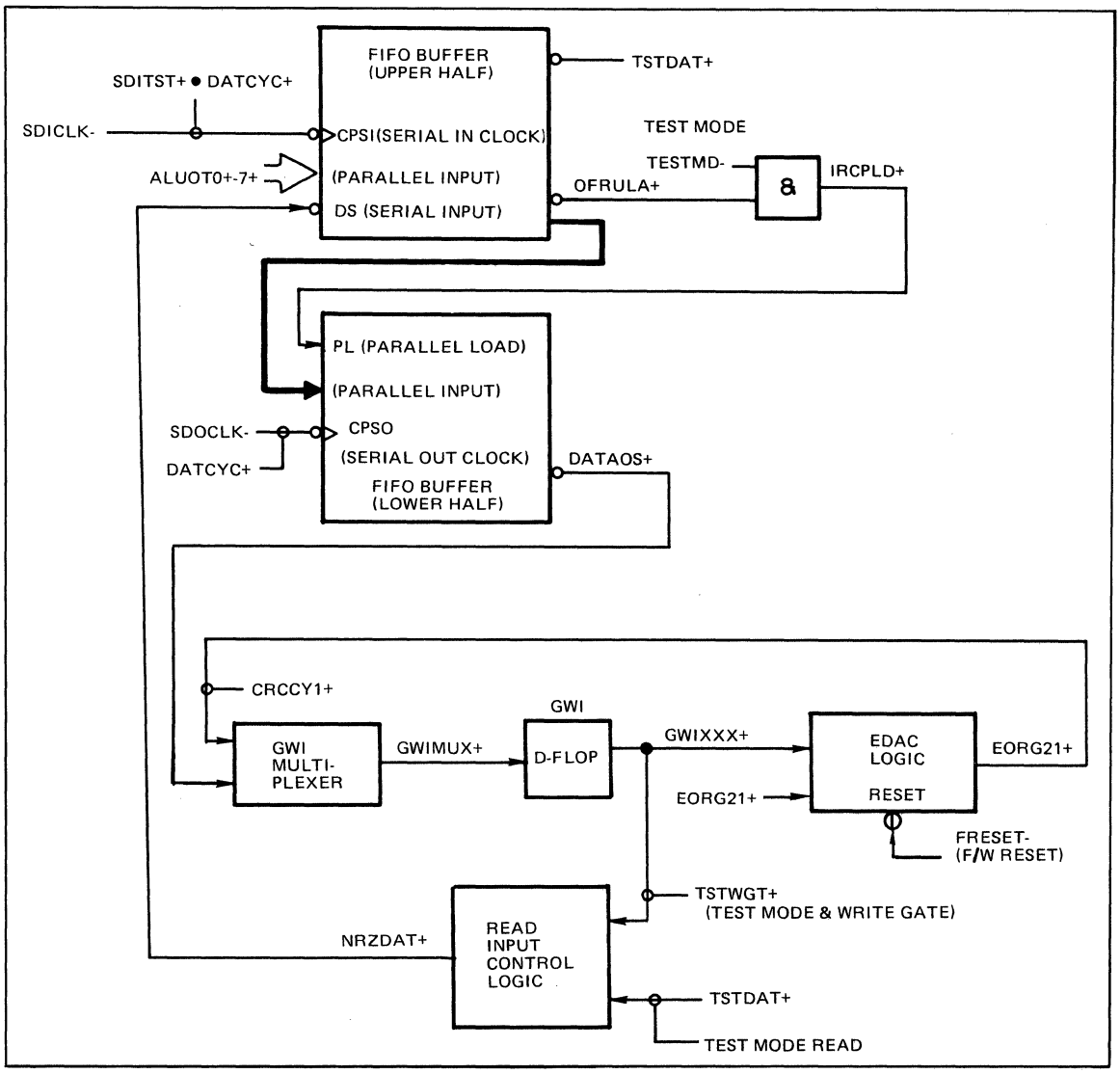


Figure 2-24 Test Mode Logic

2.4.10.2 Test Mode Read Operation (No Errors)

When the test mode read operation begins, the upper half of the FIFO buffer contains the sync, data, and EDAC bytes, and the FIFO remains configured as upper and lower halves (assuming that the preceding operation was a test mode write). The serial output (TSTDAT+) from the upper half of the FIFO is sent through the read input control logic (NRZDAT+) and clocked back into the upper half of the FIFO, with the sync and EDAC bytes being stripped off. The same information is also sent through the EDAC logic, and should result in an EDAC code of zero (indicating no read errors). The adapter test mode is reset at the end of the operation and TESTMD- going high restores the FIFO buffer to its normal 32 x 8-bit configuration. The data byte then falls through to the bottom of the FIFO stack, from which location it is sent to the MPDC.

2.4.10.3 Test Mode Read Operation (False Data)

In this operation, firmware loads a sync byte, a data byte with an error (hex 45), and seven EDAC bytes. The first EDAC byte also has one bit in error. Considered serially, the two bits in error are eleven bits apart, which is the limit for EDAC detection and correction. The test mode read operation proceeds as described above, except that the EDAC logic causes a read error to be set in the adapter status register. This operation provides assurance that the adapter can detect and correct errors.

Section 3

Theory of Operation-Cycle Flow

Firmware for the MPDC and storage module adapter (SMA) is resident within the MPDC read-only storage (ROS) memory. The ROS memory, located on a separate adapter board (BD2DT1), can house up to 4K locations (see subsection 3.1). When read from the ROS memory and decoded, a firmware command results in the specific action of various hardware elements.

A sequence of hardware operations can be obtained by performing designated serial execution of firmware commands. The groupings of firmware commands are referred to as firmware routines. The firmware routines link the Level 6 software and the MPDC subsystem hardware. Software commands are interpreted and executed by the firmware decoding of the command. The result of this decoding causes the MPDC to exit from the scanning process and select entry into the proper firmware routine. The sequencing of firmware commands and complete routines continues until all the routines required to complete the software command have been executed.

The MPDC also contains a 1K random access scratch pad memory (SPM). This memory has 256 locatons reserved for adapter and device-specific information, 256 locations for a common work area, and 512 locations for future use. The firmware can interrogate any location within the SPM for the purpose of interpreting or updating status, configuration, control, etc., information.

3.1 FIRMWARE COMMANDS

Firmware commands (words) are comprised of bit structures known as microinstructions and are relegated to a 4K by 16-bit deep storage area referred to as the ROS memory. The 16-bit firmware command is divided into four fields (see Figure 3-1): the op-code, the AOP register select, the BOP register select, and the micro-ops. The firmware commands are segmented into seven categories. Each is subdivided into various bit configurations (microinstructions) to perform a designated operation. The seven basic types of firmware commands are:

- Miscellaneous
- Megabus Logic
- Arithmetic Logic Unit (ALU)
- Constant
- Memory
- Test
- Branch

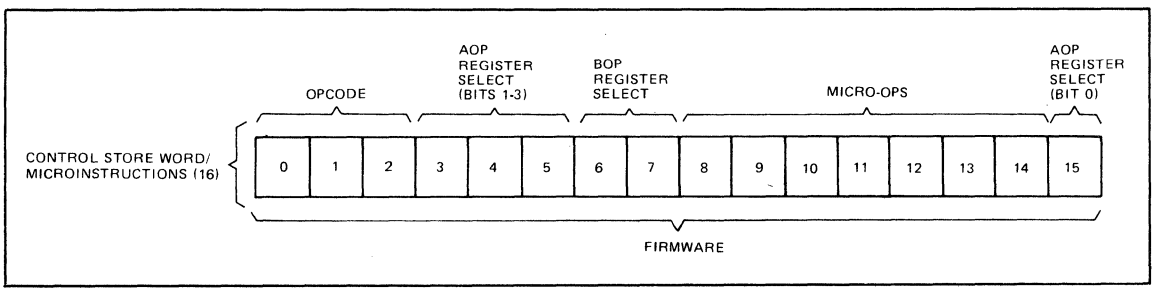


Figure 3-1 Microinstruction General Field Format

Each major category of firmware command is identified by a particular op-code (see Figure 3-2). The op-code is the binary decode of bits 0, 1, and 2 of the control store (firmware) word.

3.1.1 Miscellaneous Commmands

Miscellaneous commands, identified by an op-code of 000, are implemented primarily to perform clear and set operations involving registers and flip-flops in the MPDC and storage module adapter. This type of firmware instruction is typically comprised of a single microinstruction located in bits 3 through 15 of the firmware word.

In some instances it is advantageous to perform multiple set or reset operations during a single cycle to reduce firmware processing time. To accomplish this, some of the miscellaneous commands are composites of two or more microinstructions found in other firmware commands. Table A in Figure 3-2 lists the various miscellaneous commands indicating the specific microinstruction structure. Table A also shows the mnemonics, the operation performed, and the hexadecimal value for each type of miscellaneous command.

3.1.2 Megabus Logic Commands

Megabus logic commands have an op-code of 101 and are generally comprised of single microinstructions located in bits 3 through 15 of the firmware command. Megabus logic commands are similar to miscellaneous commands in that they perform set or reset operations; however, they are restricted to logic associated with the Megabus.

These commands also have logical control of the increment and decrement functionality of the output registers associated with Megabus transfers. Table B in Figure 3-2 lists the microinstruction configuration associated with the various types of Megabus logic commands. Also in this table are the mnemonic, hexadecimal equivalent of the microinstruction structure, and the operation to be performed for each Megabus logic command.

3.1.3 ALU Commands

ALU commands, which are specified by an op-code of 011, perform Designated logic or arithmetic operations on the contents of the A- and/or B-operand registers. These commands are comprised of microinstructions which define an A-operand (see Table C of Figure 3-2), microinstructions which define a B-operand and microinstructions located in bits 8 through 14 of the firmware command which specify what type of operation is to be accomplished by the ALU.

Mode, carry enable, and carry in are determined by the ALU-specific (bits 8 through 14) microinstructions, and the register selected as either the A- or B-operand input can be used as the storage area for the result.

The A- and B-operand multiplexer inputs and the microinstruction configuration for input selection are shown in Tables D and E of Figure 3-2.

3.1.4 Constant Commands

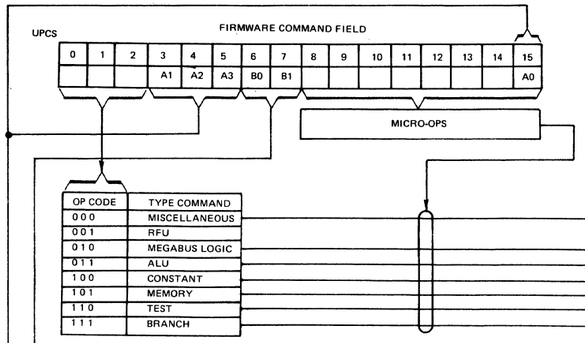
Constant commands, which have an op-code of 100, contain A-operand selection bits in microinstructions 3 through 5, and an 8-bit constant in microinstructions 6 through 15.

The constant command can be used in two ways: to load a constant directly from the firmware command into a register designated by the A-operand bits, or to perform a logical operation on another byte of information utilizing the ALU capabilities.

The configuration of the microinstructions for the A-operand constant fields of the firmware command is presented in Table F of Figure 3-2. Also listed in Table F are the operations to be performed and the mnemonics for the constant command format.

3.1.5 Memory Commands

The memory commands, which have an op-code of 101, are a combination of A-operand select microinstructions and scratch pad memory specific microinstructions or to control the



OP CODE	TYPE COMMAND
000	MISCELLANEOUS
001	RFU
010	MEGABUS LOGIC
011	ALU
100	CONSTANT
101	MEMORY
110	TEST
111	BRANCH

TABLE D AOP MULTIPLEXER INPUT SELECTION

A0	A1	A2	A3	SELECTED REGISTER (SRIA)	MNEMONIC	SELECTED REGISTER (SRIA)*	MNEMONIC
0	0	0	0	ACCUMULATOR	AACU	ACCUMULATOR	AACU
0	0	0	1	SCRATCH PAD MEMORY	ASPM	SCRATCH PAD MEMORY	ASPM
0	0	1	0	SCRATCH PAD ADDRESS (INDEXED)	ASPA	SCRATCH PAD ADDRESS (INDEXED)	ASPA
0	0	1	1	INDEX REGISTER	AIDX	SCRATCH PAD ADDRESS (INDEXED)	ASPAI
0	1	0	0	EDAC CORRECTION PATTERN 1	AADO	ADAPTER DATA REGISTER	AADO
0	1	0	1	EDAC CORRECTION PATTERN 2	AAD1	DEVICE DATA REGISTER	AAD1
0	1	1	0	ADAPTER STATUS	AAD2	ADAPTER COMMAND REGISTER	AAD2
0	1	1	1	DEVICE DATA	AAD3	ADAPTER DATA COUNTER LOWER	AAD3
1	0	0	0	BUS ADDRESS OUT	ABUS1	BUS REGISTER IN	ABUS1
1	0	0	1	BUS DATA OUT 1	ABUS2	BUS DATA IN 1	ABUS2
1	0	1	0	BUS DATA OUT 2	ABUS3	BUS DATA IN 2	ABUS3
1	0	1	1	BUS RANGE OUT	ABUS4	BUS ADDRESS IN	ABUS4
1	1	0	0	ADAPTER RFU	AAD4	ADAPTER DATA COUNTER UPPER	AAD4
1	1	0	1	ADAPTER RFU	AAD5	ADAPTER TAGS	AAD5
1	1	1	0	ADAPTER RFU	AAD6	RFU	AAD6
1	1	1	1	ADAPTER RFU	AAD7	ADAPTER DATA BYTE TAKEN	AAD7

SRIA = STORE RESULT IN SELECTED AOP REGISTER

TABLE E BOP MULTIPLEXER INPUT SELECTION

B0	B1	SELECTED DATA INPUT	MNEMONIC
0	0	ACCUMULATOR	BACU
0	1	SCRATCH PAD MEMORY	BSPM
1	0	BUS STATUS:	BBST
		BITS 0-3 (ZEROS)	
		BIT 4 BUS YELLOW IND.	
		BIT 5 BUS NAK	
		BIT 6 BUS PARITY ERROR	
		BIT 7 BUS RED IND.	
1	1	BOP CONSTANT	

TABLE A MISCELLANEOUS COMMANDS

OPERATION	BINARY VALUE	MNEMONIC	HEX CODE
NO OPERATION	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	NOP	0 0 0 0
RFU	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	–	0 0 0 1
SET BUS ACK	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0	SBA	0 0 0 2
SET QLT (BLT DONE)	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	QLT	0 0 0 4
CLEAR FLOPS AND REGISTERS	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0	CRF	0 0 1 0
RFU	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0	–	0 0 2 0
HALT	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0	HLT	0 0 4 0
RESET DIAGNOSTIC MODE	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0	R&D	0 0 8 0
RESET DEVICE ADAPTER	0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0	RDA	0 0 8 8
INITIALIZE	0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0	INI	0 0 9 8
SET DIAGNOSTIC MODE	0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0	STD	0 1 8 0
DISABLE HARDWARE DATA PATH	0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0	DHP	0 2 0 0
ENABLE READ PATH	0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0	ERP	0 6 0 0
ENABLE WRITE PATH	0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1	EWP	0 6 0 1
SET ERROR FLOPS	0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0	SEF	0 8 0 0
CLEAR COMMAND	0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0	CLR	1 0 0 0

TABLE B MEGABUS LOGIC COMMANDS

OPERATION	BINARY VALUE	MNEMONIC	HEX CODE
RESET INTERRUPT LATCH	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1	RIL	4 0 0 1
RESET REGISTER BUSY	0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0	RRB	4 0 0 2
SET REGISTER BUSY	0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	SRB	4 0 0 4
RESET CHANNEL READY	0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0	RCR	4 0 1 0
SET CHANNEL READY	0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0	SCR	4 0 1 8
DECREMENT RANGE COUNTER	0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0	DRC	4 0 4 0
RESET STATUS	0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0	RST	4 0 8 4
CLEAR BUS	0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0	CLB	4 0 8 6
INCREMENT ADDRESS COUNTER	0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0	IAC	4 1 0 0
CYCLE	0 1 0 A A A 0 0 0 0 1 0 0 0 0 A	CYC	N/A

AAAA = SELECT AOP REGISTER (SEE FIRMWARE COMMAND FIELD)

TABLE C ALU COMMANDS

OPERATION	BINARY VALUE	MNEMONIC	HEX CODE
AOP PLUS ONE	0 1 1 A A A B B C S 0 0 0 0 0 A	INC	N/A
AOP NEGATION	0 1 1 A A A B B C S 0 0 0 0 1 A	ANT	N/A
NOR A TO B	0 1 1 A A A B B C S 0 0 0 1 1 A	NOR	N/A
ZERO ALU	0 1 1 A A A B B C S 0 0 1 1 1 A	ZER	N/A
NAND A TO B	0 1 1 A A A B B C S 0 1 0 0 1 A	NND	N/A
BOP NEGATION	0 1 1 A A A B B C S 0 1 0 1 1 A	BNT	N/A
SUBTRACT B FROM A	0 1 1 A A A B B C S, 0 1 1 0 0 A	SUB	N/A
XOR A TO B	0 1 1 A A A B B C S 0 1 1 0 1 A	XOR	N/A
ADD A TO B	0 1 1 A A A B B C S 1 0 0 1 0 A	ADD	N/A
XNOR A TO B	0 1 1 A A A B B C S 1 0 0 1 1 A	XNR	N/A
TRANSFER BOP	0 1 1 A A A B B C S 1 0 1 0 1 A	XFB	N/A
AND A TO B	0 1 1 A A A B B C S 1 0 1 1 1 A	AND	N/A
LEFT SHIFT AOP	0 1 1 A A A B B C S 1 1 0 0 0 A	LSH	N/A
OR A TO B	0 1 1 A A A B B C S 1 1 1 0 1 A	ORR	N/A
AOP MINUS ONE	0 1 1 A A A B B C S 1 1 1 1 0 A	DEC	N/A
TRANSFER AOP	0 1 1 A A A B B C S 1 1 1 1 1 A	XFA	N/A

AAAA = SELECT AOP REGISTER (SEE FIRMWARE COMMAND FIELD)
BB = SELECT BOP REGISTER (SEE FIRMWARE COMMAND FIELD)
C = 1 = UTILIZE CARRY IN FROM PREVIOUS OPERATION (COTI)
S = 1 = STORE RESULT IN AOP (SRIA)

TABLE J BRANCH COMMANDS

OPERATION	BINARY VALUE	MNEMONIC	HEX CODE
GO TO	1 1 1 1 A A A A A A A A A A A A	GTO	FXXX
LOAD RETURN	1 1 1 0 A A A A A A A A A A A A	LRA	EXXX

A = BRANCH ADDRESS

TABLE F CONSTANT COMMANDS

OPERATION	BINARY VALUE	MNEMONIC	HEX CODE
LOAD CONSTANT TO AOP	1 0 0 A A A C C C C C 0 C 0 C C	LCN	N/A
AOP ANDED WITH CONSTANT	1 0 0 A A A C C C C C 0 C 1 C C	ACN	N/A
AOP ORED WITH CONSTANT	1 0 0 A A A C C C C C 1 C 0 C C	OCN	N/A

AAA = A1, A2, A3 TO SELECT AOP REGISTER
C = VALUE OF CONSTANT

TABLE G MEMORY COMMANDS

OPERATION	BINARY VALUE	MNEMONIC	HEX CODE
MEMORY WRITE	1 0 1 A A A 1 0 0 0 0 0 0 0 0 A	MWT	N/A
INCREMENT SPAC	1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0	IMA	A 1 0 0
DECREMENT SPAC	1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0	DMA	A 0 0 8
MEMORY WRITE AND INC.	1 0 1 A A A 1 1 0 0 0 0 0 0 0 A	WIA	N/A
MEMORY WRITE AND DEC.	1 0 1 A A A 1 0 0 0 0 0 1 0 0 A	WDA	N/A
SET SPM TEST MODE	1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0	SPT	A 0 8 0
RFU	1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0	–	A 0 4 0
LOAD REQUESTING CHANNEL	1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0	LRC	A 0 2 0
LOAD INDEX REG. WITH AOP	1 0 1 A A A 0 0 0 0 1 1 0 0 0 A	LIR	N/A
SET MODULE BAD PARITY	1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0	MBP	A 0 2 4
READ ONLY PARITY CHECK	1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0	RPC	A 0 0 2

AAAA = SELECT AOP REGISTER (SEE FIRMWARE COMMAND FIELD)
SPAC = SCRATCH PAD MEMORY ADDRESS COUNTER

TABLE H TEST COMMANDS

OPERATION	BINARY VALUE	MNEMONIC	HEX CODE
TEST FOR ZERO	1 1 0 A A A 0 0 0 1 T T T T T A	TFZ	N/A
TEST FOR ONE	1 1 0 A A A 0 0 1 0 T T T T T A	TFO	N/A
RETURN	1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0	RTN	C 2 0 0

AAAA = SELECT AOP REGISTER (SEE FIRMWARE COMMAND FIELD)
TTTTT = HEX CODE SELECTS TEST MUX INPUT

TABLE I TEST PARAMETERS

MNEMONIC	FUNCTION	HEX CODE	DESCRIPTION
TAHR	HDTSRQ+00	00	ADAPTER HARDWARE REQUEST
TBCA	SHRCOM+00	01	BUS CYCLE ACTIVE
TBSP	BSRSVP+30	02	BUS RESPONSE REQUIRED
TEQZ	ALUEQZ+00	03	ALU OUTPUT EQUALS 00
TEQF	ALUEQF+00	04	ALU OUTPUT EQUALS FF
TCOT	ALUCOT+00	05	ALU CARRY OUT
TREQ	CREREQ+00	06	CHANNEL REQUEST
TACK	ACKRSP+00	07	BUS ACK RESPONSE
TAX0	ALUAX0-00	08	AOP MULTIPLEXOR, BIT 0
TAX1	ALUAX1-00	09	AOP MULTIPLEXOR, BIT 1
TAX2	ALUAX2-00	0A	AOP MULTIPLEXOR, BIT 2
TAX3	ALUAX3-00	0B	AOP MULTIPLEXOR, BIT 3
TAX4	ALUAX4-00	0C	AOP MULTIPLEXOR, BIT 4
TAX5	ALUAX5-00	0D	AOP MULTIPLEXOR, BIT 5
TAX6	ALUAX6-00	0E	AOP MULTIPLEXOR, BIT 6
TAX7	ALUAX7-00	0F	AOP MULTIPLEXOR, BIT 7
TORZ	OFRGVL-00	10	OFFSET RANGE ZERO
TRGZ	EOR(XXX)+00	11	RANGE ZERO
TSBS	SBSOBS+00	12	SINGLE BYTE STORED
TSAW	SPAWRP+00	13	SPM ADDRESS WRAPAROUND
TADB	BUSY(XX)+00	14	ADAPTER BUSY
TNDR	NDSTRQ+00	15	NONDATA SERVICE REQUEST
TORH	OFRNGZ-00	16	OFFSET RANGE HISTORY
TDCN	MYDCNN+00	17	MY DATA CYCLE NOW
TBSY	BDRBSY+00	18	BUS DATA REGISTER BUSY
TUBR	UBRQ(XX)+00	19	UNSOLICITED BUS REQUEST
TINT	RESINT+00	1A	RESUME INTERRUPT
TNAK	NAKRSP+00	1B	NAK RESPONSE
TBYT	BBAD23+00	1C	BYTE MODE
TPTY	BSPYCK+00	1D	BUS PARITY CHECK
TNBR	NORQT3+00	1E	NO BUFFER REQUEST
TFDR	FDTSRQ+00	1F	FIRMWARE DATA SERVICE REQUEST

HEX CODE REPRESENTS TTTTT IN TEST COMMAND (TABLE H)

Figure 3-2 Firmware Word Decode

functionality of the scratch pad address counter. The memory commands also have the capability of manipulating the control flip-flops associated with the SPM. Table G of Figure 3-2 lists the various microinstruction configurations for the memory commands, their mnemonics, and the operations that they are designed to perform.

3.1.6 Test Commands

The test commands enable firmware to vary the sequential processing of firmware commands, depending upon the state of various hardware elements or control byte bit configurations. These commands, which have an op-code of 110, are used to determine if a designated function or bit is either set or reset, and to alter the routine sequencing according to the result of the test.

Test command fields (see Figure 3-2, Table H) are comprised of microinstructions which define an A-operand selection field, and microinstructions (bits 10 through 14 of the test command) which, when decoded into a hexadecimal code, determine the function to be tested as listed in Table I of Figure 3-2.

Return commands, which are not test commands, are classified within the test category. These commands also have an op-code of 110 and permit variations in the sequential execution of firmware commands. When the return command is processed, the microprogram control store address counter (UPAC) is set to the location that was previously stored in the selected subroutine return address register (SRAR).

3.1.7 Branch Commands

Branch operations have an op-code of 111 and are used to load either the UPAC or the SRAR. The microinstructions located in bits 4 through 15 represent the address to be loaded into the register designated by the state of the microinstruction in bit 3 (see Figure 3-2, Table J).

The load return address register is not actually a branch operation but is classified within this category. The load return address register sets up the contents of the SRAR with an address used by the firmware through utilization of a return command.

3.2 SCRATCH PAD MEMORY (SPM)

The MPDC is provided with a scratch pad memory containing 1024 locations each of which is 8 bits deep. The alterable contents of the SPM is implemented to allow for information storage or to store control bytes which direct the actions of each channel attached to the MPDC.

The 1K locations (see Figure 3-3) located in scratch pad memory are segmented into three areas:

- 256 Work Locations - Locations utilized to store information commonly pertinent to all available channels.
- 256 Channel Locations - Locations utilized to store information which is relevant to only an individual channel.
- 512 Unused Locations - Reserved for future use.

The 256 locations of SPM relegated to specific channel locations are further subdivided into 64-location quadrants. A single quadrant has 64 addressable locations that are channel-specific. Table 3-1 shows the topology of a quadrant indicating the address for each byte relative to the base location of the segment selected by the scratch pad memory index register. A description of the general usage and the terminology associated with each location of a quadrant is supplied in Table 3-2. Where a control byte has a bit-significant purpose, the bit structure and use are shown in Figures 3-4 through 3-10.

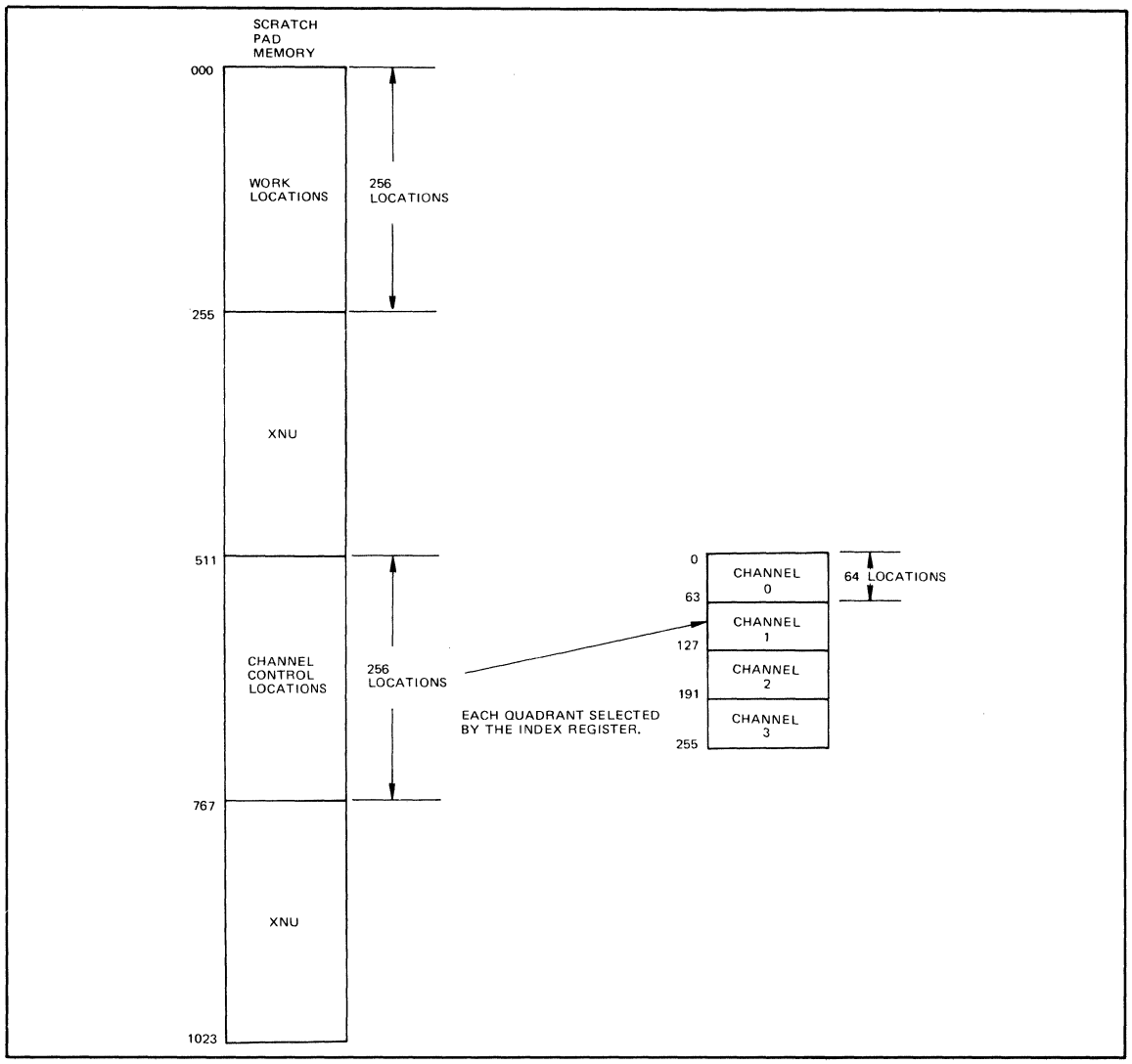


Figure 3-3 Scratch Pad Memory Segmentation

Table 3-1 Scratch Pad Memory Topology

HEX ADDRESS*	MNEMONIC	ASSOCIATED TERM
00	CWD1	Control Word, LSB
01	CWD2	Control Word, MSB
02	ILC1	Interrupt Level, LSB
03	ILC2	Interrupt Level, MSB
04	DATH	Data Length, MSB
05	DATL	Data Length, LSB
06	TSK1**	Task Word, LSB
07	TSK2**	Task Word, MSB
08	ADR1	Address, LSB
09	ADR2	Address, MSB
0A	RFU	-
0B	MOD1	Module Address
0C	RNG1	Range, LSB

Table 3-1 Scratch Pad Memory Topology (Cont.)

HEX ADDRESS*	MNEMONIC	ASSOCIATED TERM
0D	RNG2	Range, MSB
0E	OFR1	Offset Range, LSB
OF	OFR2	Offset Range, MSB
10	CNF1	Configuration Word A, LSB
11	CNF2	Configuration Word A, MSB
12	CNF3	Configuration Word B, LSB
13	CNF4	Configuration Word B, MSB
14	MAXSEC	Maximum Sector Number
15	MAXTRK	Maximum Track Number
16	MAXCY1	Maximum Cylinder Number, LSB
17	MAXCY2	Maximum Cylinder Number, MSB
18	STS1	Status Word One, LSB
19	STS2	Status Word One, MSB
1A	STS3	Status Word Two, LSB
1B	STS4	Status Word Two, MSB
1C - 1F	RFU	
20	RETCNT	Retry Count
21	RETTYP	Retry Type
22	RPSARG	RPS Argument
23	ERRBY	Error Byte
24	MON1	Channel Monitor
25	DMA1	DMA Control
26	DID1**	Device ID, LSB
27	DID2**	Device ID, MSB
28	CHN1	Channel Number, LSB
29	CHN2	Channel Number, MSB
2A	CPC1	CP Address, LSB
2B	CPC2	CP Address, MSB
2C	IDF1	Interrupt Vector, LSB
2D	IDF2	Interrupt Vector, MSB
2E	SUM1	Displacement Byte
2F	SUM2	Displacement Byte
30	SUM3	Displacement Byte
31	SUM4	Displacement Byte
32	BEG1	Beginning Byte, LSB
33	BEG2	Beginning Byte, MSB
34	END1	Last Byte, LSB
35	END2	Last Byte, MSB
36	MODOF1	Module Offset
37	ADDOF1	Address Offset, LSB

Table 3-1 Scratch Pad Memory Topology (Cont.)

HEX ADDRESS*	MNEMONIC	ASSOCIATED TERM
38	ADDOF2	Address Offset, MSB
39	ERRPT1	Error Pattern
3A	ERRTP2	Error Pattern
3B	ERRPT3	Error Pattern
3C	SFTCNT	Shift Count
3D	FWRV	Firmware Revision (Utilized by Software Only)
3E	DEVST**	Device Status
3F	PREST	Previous Status
FF***	LSTRW	Last Channel Read/Written

* Relative addresses to base location 00 of the Segment selected by the SPMIR

** Device-specific location

*** Absolute address from physical location 00 of the SPM

Table 3-2 SPM Segment Word Description

MNEMONIC	TERM	DESCRIPTION
CWD1 and CWD2	Control Words	These bit-specific words, which designate operations to be performed immediately (see Figure 3-4), are loaded by the previous output control word function code.
ILCI and ILC2	Interrupt Level	The interrupt level word consists of two bytes: bits 0-9 contain the CP's channel number and bits 10-15 represent a binary value from 0-63 indicating the priority of MPDC interrupt.
DATH and DATL	Data Length	These two bytes contain the hex value of the number of bytes in the data field. For complete data fields, this value represents either 256 or 2304 bytes.
TSK1	Task Work, LSB	This byte contains a count of the number of times the adapter status is read during tag code commands.
TSK2	Task Word, MSB	This byte contains information to be placed on the tag bus during an output tag code task. Bit 8=1 indicates to firmware to ignore ID field read errors.
ADR1 and ADR2	Address	This word is the main memory address sent across the Megabus to access information.

Table 3-2 SPM Segment Word Description (Cont.)

MNEMONIC	TERM	DESCRIPTION
MOD 1	Module Address	This byte is part of the main memory address and is used to select a 64K memory module.
RNG1 and RNG2	Range	The range is utilized to determine the number of bytes to be transferred across the Megabus. After the offset range is zero, the range is decremented for each byte until it is zero.
OFR1 and OFR2	Offset Range	The offset range is utilized to count the number of bytes not to be transferred (read only) across the Megabus. It is decremented for each byte read from the device until it is zero.
CNF1 to CNF4	Configuration Words	These words define the cylinder, track, and sector numbers (see Figures 3-5 and 3-6).
MAXSEC	Maximum Sector Number	This byte represents the highest sector number on a track that can be addressed.
MAXTRK	Maximum Track Number	This byte represents the highest track number on a cylinder that can be addressed.
MAXCY1 and MAXCY2	Maximum Cylinder Number	These two bytes represents the highest cylinder number that can be addressed.
STS1 to STS4	Status Word	These bytes store information concerning the device, device adapter firmware, and bus status (see Figures 3-7 and 3-8).
RETCNT	Retry Count	This byte represents a count of the number of retries allowed during a read operation with a read error.
RETTYP	Retry Type	A shifting bit pattern in this byte indicates the type of retry which is active. There are nine retry types, each of which is attempted three times. The nine retry types are:

ITEM NO.	ADAPTER HEX CODE	READ RETRY* TYPE
1.	00	Normal Retries.
2.	40	Retries with Clock Retarded.

Table 3-2 SPM Segment Word Description (Cont.)

MNEMONIC	TERM	DESCRIPTION

ITEM NO.	ADAPTER HEX CODE	READ RETRY* TYPE
3.	80	Retries with Clock Advanced.
4.	10	Retries with Offset In.
5.	50	Retries with Offset In and Clock Retarded.
6.	90	Retries with Offset In and Clock Advanced.
7.	20	Retries with Offset Out.
8.	60	Retries with Offset Out and Clock Retarded.
9.	A0	Retries with Offset Out and Clock Advanced.

*Listed in order of occurrence

MNEMONIC	TERM	DESCRIPTION
RPSARG	RPS Argument	This byte contains the sector address for initiating a search type data transfer.
ERRBY	Error Byte	This byte is used for temporary storage of the error pattern byte.
MONI	Channel Monitor	See Figure 3-9 for the bit-specific usage of this byte.
DMAI	DMA Control	See Figure 3-10 for the bit-specific usage of the data management control byte.
DID1 and DID2	Device ID Code	The MSB (DID2) is always hexadecimal 23 for the MPDC. The LSB (DID1) can be a hexadecimal 60, 61, 62, 63 or 6F. The complete storage module device ID codes appear as follows:

HEXADECIMAL CODE	DEVICE
2360	40 Megabyte Storage Module Device
2361	80 Megabyte Storage Module Device
2362	150 Megabyte Storage Module Device

Table 3-2 SPM Segment Word Description (Cont.)

MNEMONIC	TERM	DESCRIPTION
		HEXADECIMAL CODE / DEVICE 2363 — 300 Megabyte Storage Module Device 236F — No Storage Module Addressed or Reserved Device
CHN1 and CHN2	Channel Number	These bytes supply the channel number of the particular AMPDC.
CPC1 and CPC2	CP Address	The CP address is utilized by the MPC for loading of the address lines of the BDR whenever a request or a data transfer to the CP occurs.
IDF1 and IDF2	Interrupt sector	The interrupt vector word, consisting of two bytes, utilizes bits 0-9 for containing the MPDC channel number and bits 10-15 for containing a value of from 0-63, indicating the priority of the MPDC interrupt to the CP.
SUM1 to SUM4	Displacement Bytes	These four bytes contain the displacement from the base address of the read operation to the error byte location.
BEG1 and BEG2	Beginning Byte	These two bytes contain the main memory address of the first byte of the read operation.
END1 and END2	Last Byte	These two bytes contain the main memory address of the last byte of the read operation.
MODOF1	Module Offset	This byte contains the module offset.
ADDOF1	Address Offset	These bytes contain the address of the offset range.
ERRPT1	Error Pattern	These bytes contain the error pattern used in EDAC.
SETCNT	Shift Count	This byte is used by EDAC as a general purpose counter.
DEVST	Device Status	This byte represents the current device status (see Table 2-6).
PREST	Previous Status	This byte is used to store the previous device status of comparisons during status updates.
LSTRW	Last Channel Read/ Written	This byte contains the hex configuration of the channel number associated with the last read/ write operation performed.

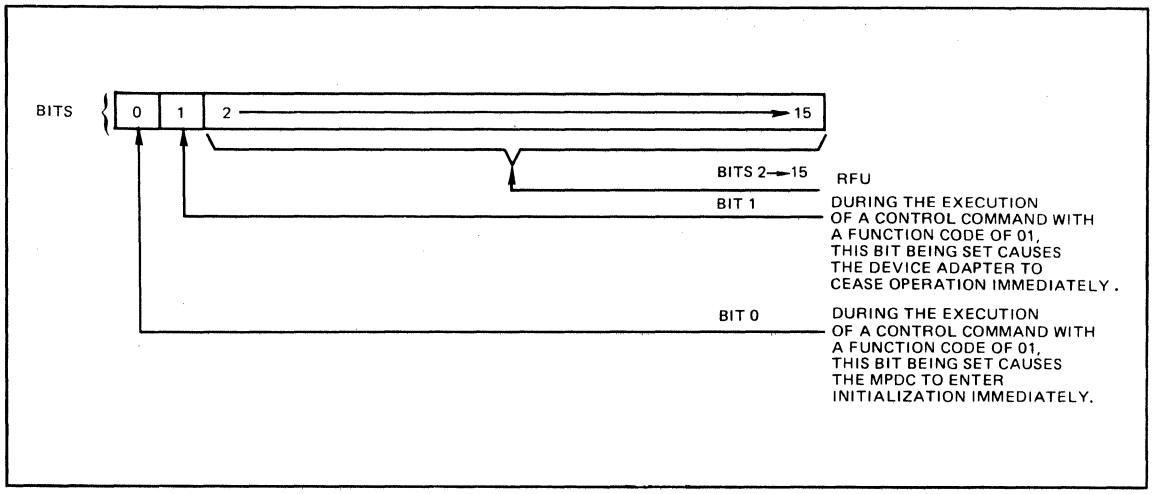


Figure 3-4 Control Word Bit Significance

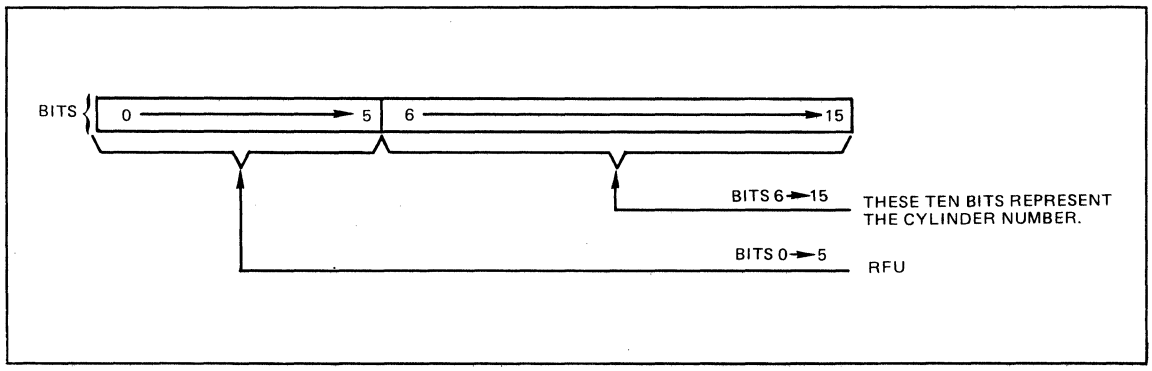


Figure 3-5 Configuration Word A Bit Significance

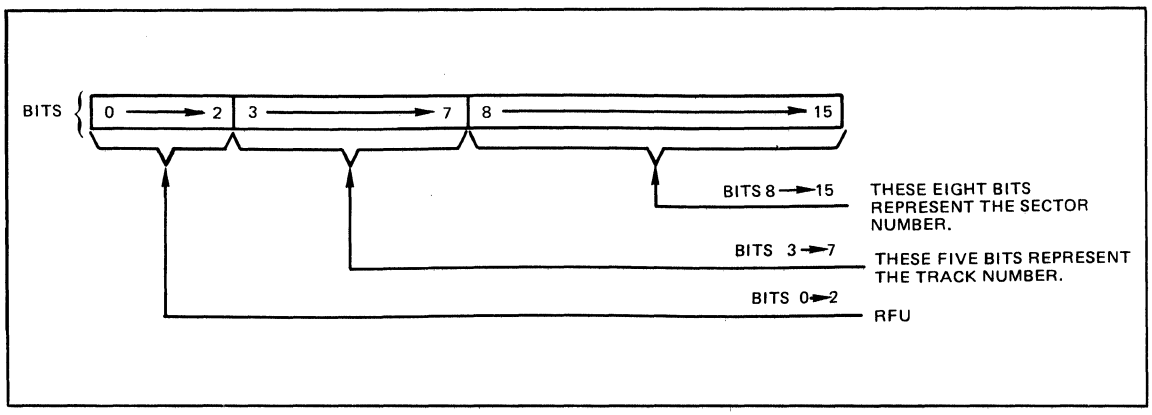


Figure 3-6 Configuration Word B Bit Significance

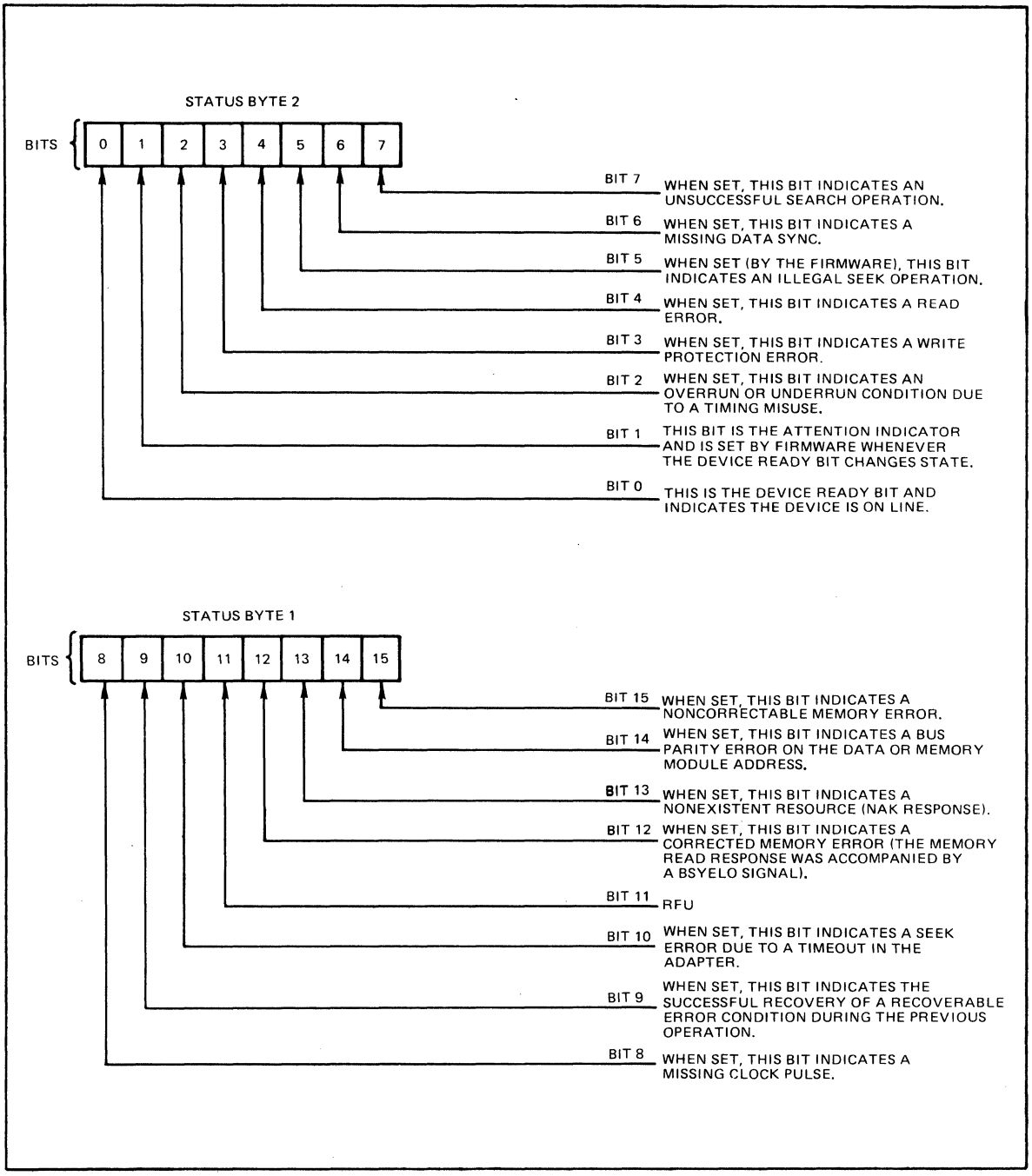


Figure 3-7 Status Word One Bit Significance

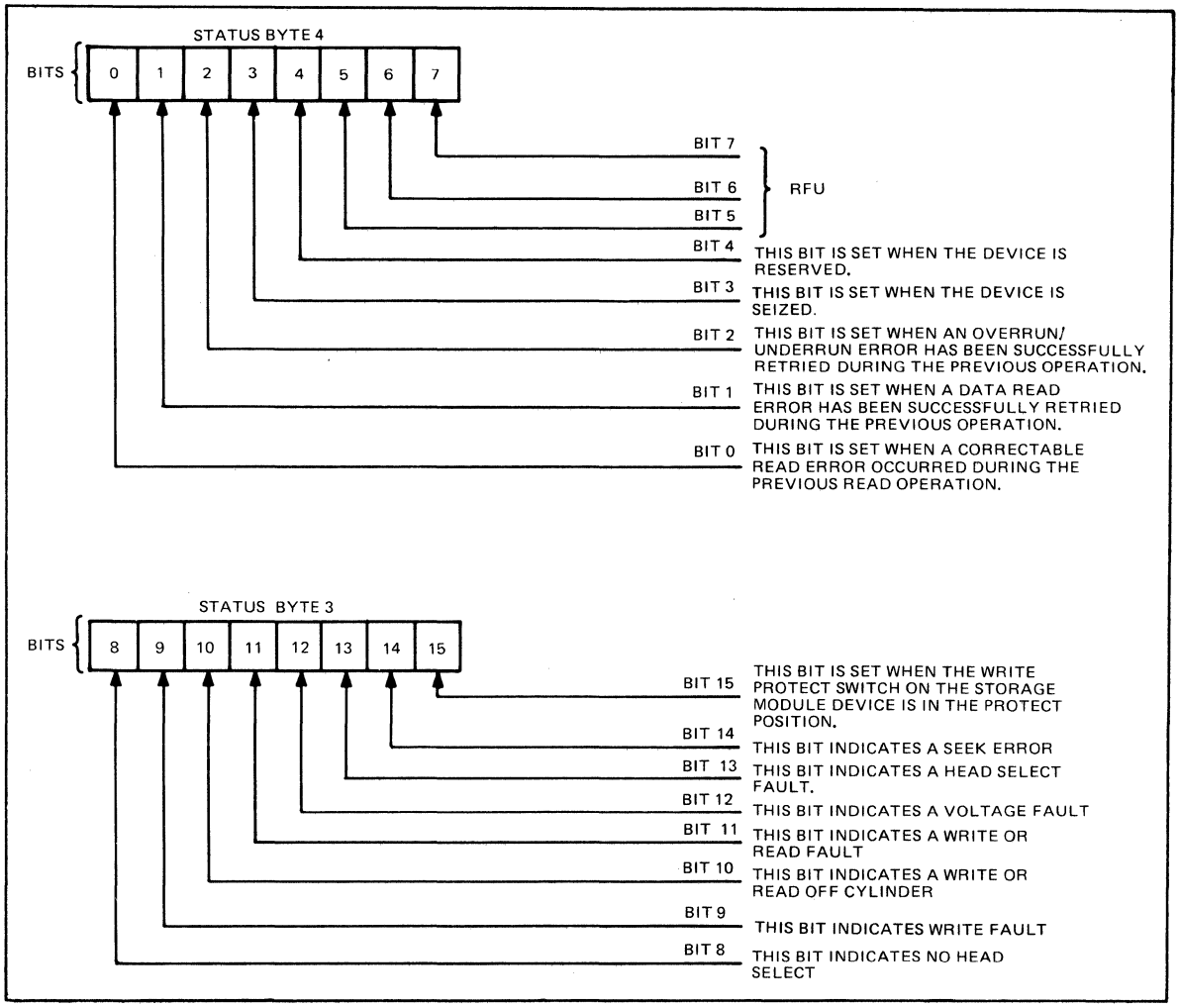


Figure 3-8 Status Word Two Bit Significance

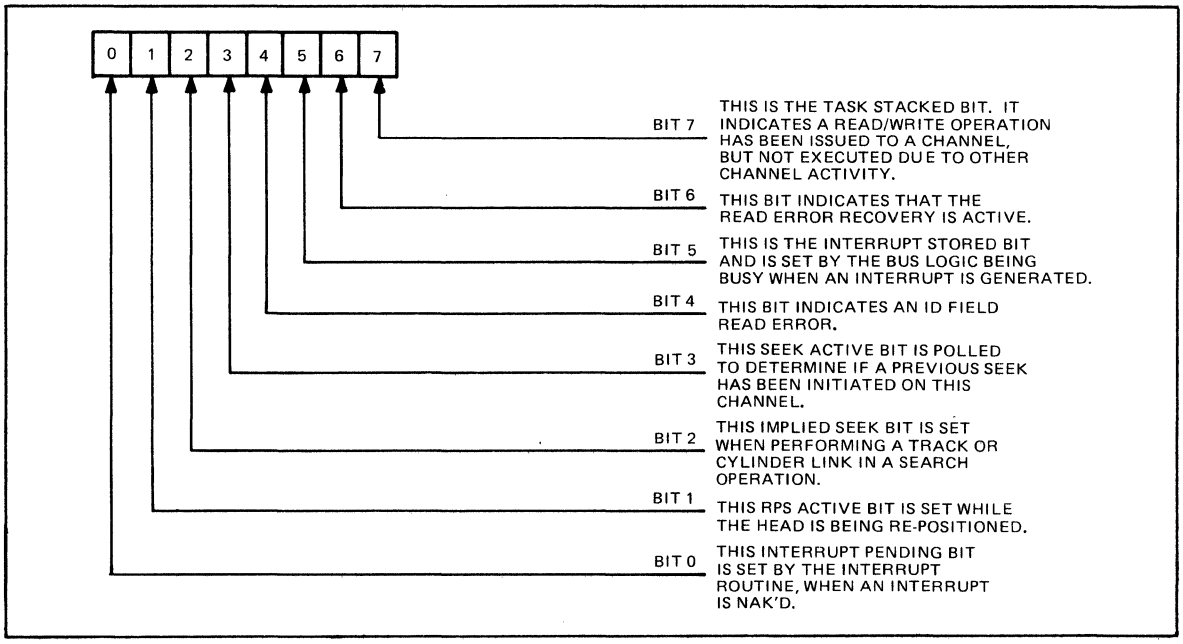


Figure 3-9 Channel Monitor Byte Bit Significance

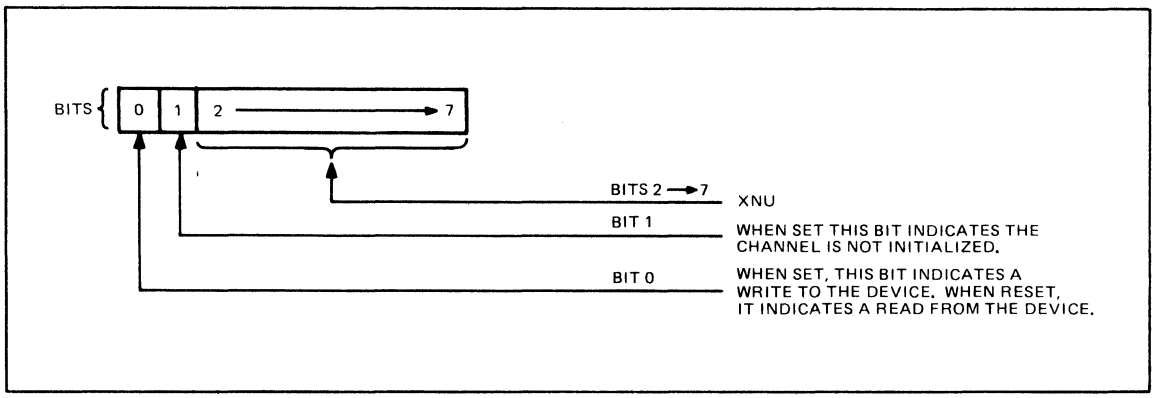


Figure 3-10 DMA Byte Bit Significance

3.3 FIRMWARE-RETRIEVABLE INFORMATION

Firmware is allowed access to certain SMA hardware generated information. This information is read from the SMA in byte form and is used by firmware to test adapter and device status (see Figures 3-11 through 3-14).

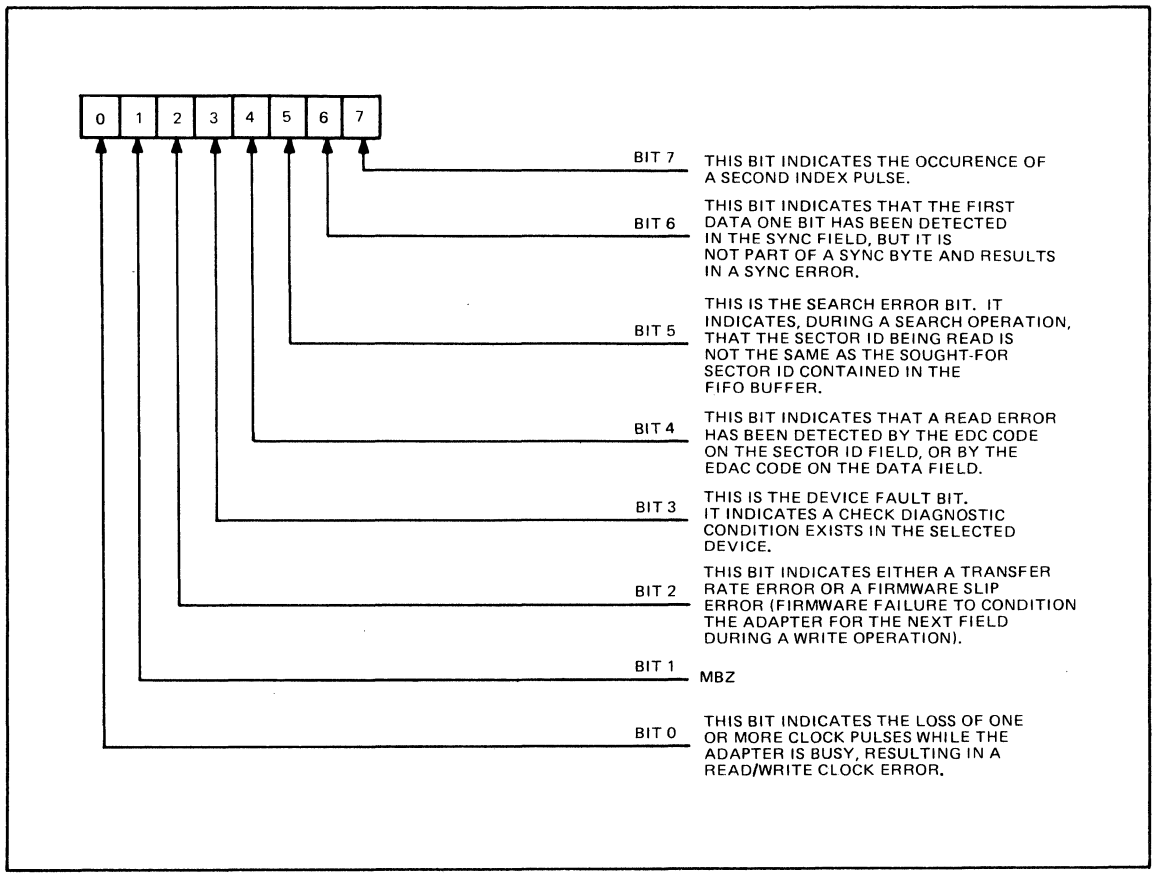


Figure 3-11 Adapter Status Byte Bit Significance

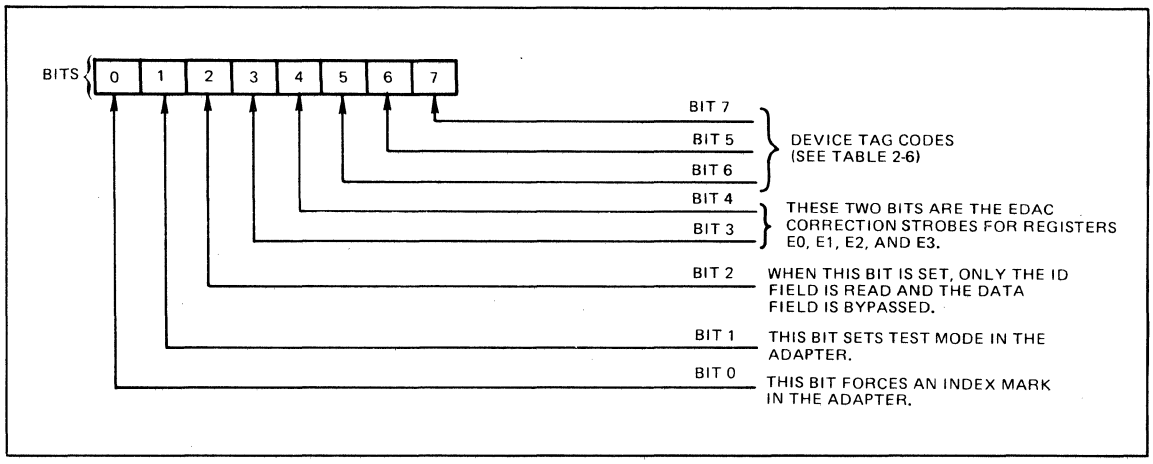


Figure 3-12 Adapter Tag Byte Bit Significance

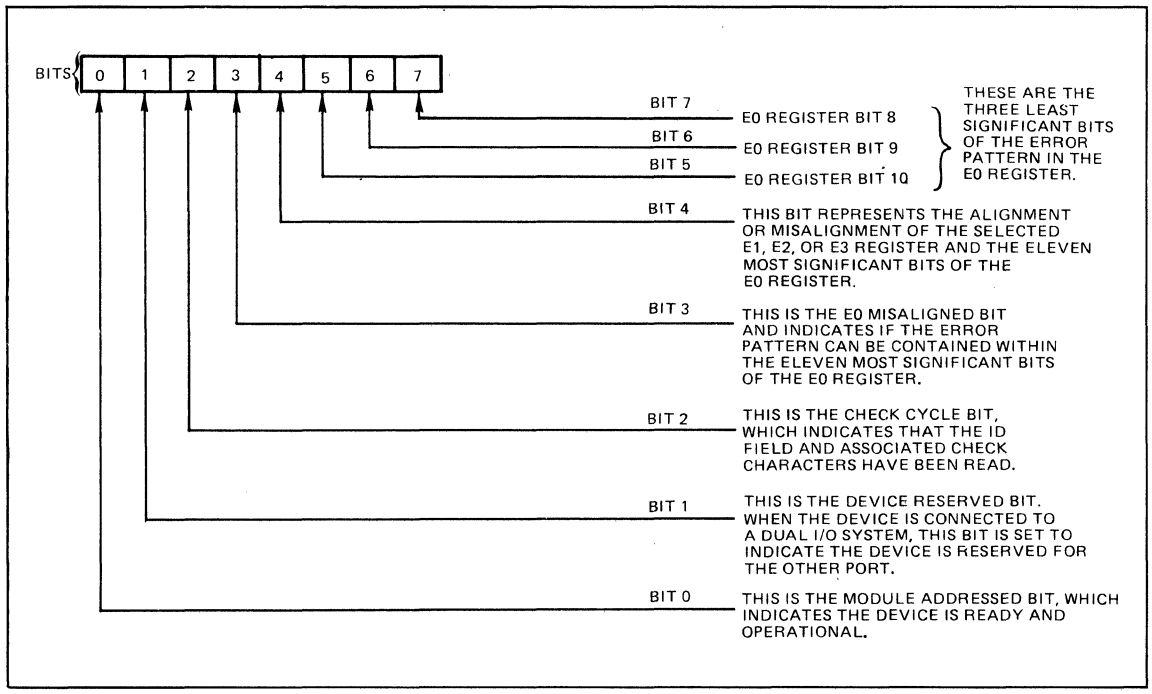


Figure 3-13 EDAC Pattern One Byte Bit Significance

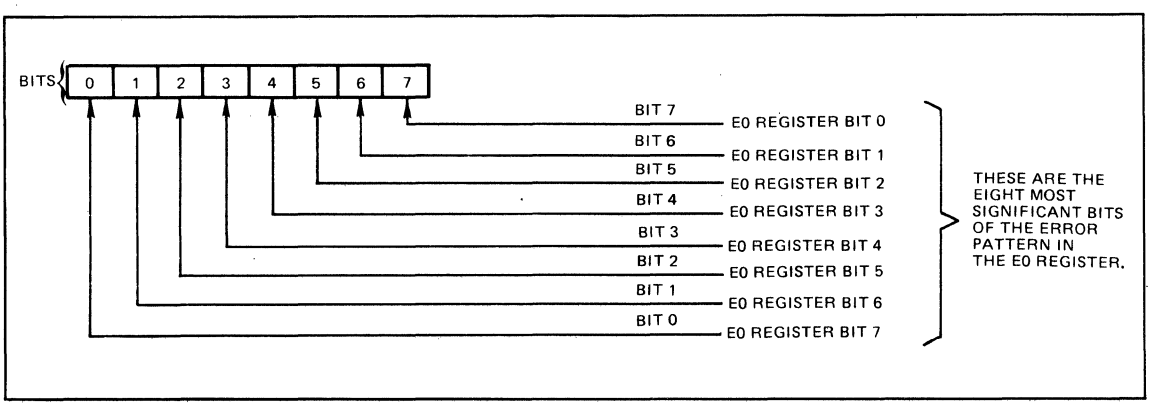


Figure 3-14 EDAC Pattern Two Byte Bit Significance

3.4 FIRMWARE FLOW

3.4.1 MPDC Overview Flow Chart

The MPDC-specific firmware is divided essentially into eight major functional areas or routines. Figure 3-15 is an overview flow chart depicting the process routines associated with the MPDC controlling capabilities, the interconnections between the routines, and the conditions for each available path.

Upon initialization of the MPDC, the firmware enters a routine for verification of the hardware referred to as the Basic Logic Test (BLT). The BLT either halts if an error is detected or enters the Clear Scratch Pad Memory routine.

The Clear Scratch Pad Memory routine, which resets all locations of the scratch pad memory (SPM) to Zeros, is utilized in one of two ways. The routine is entered during the processing of the BLT to clear the SPM and then returns to the BLT, or it is entered upon completion of the BLT to reset the SPM for the processing routines.

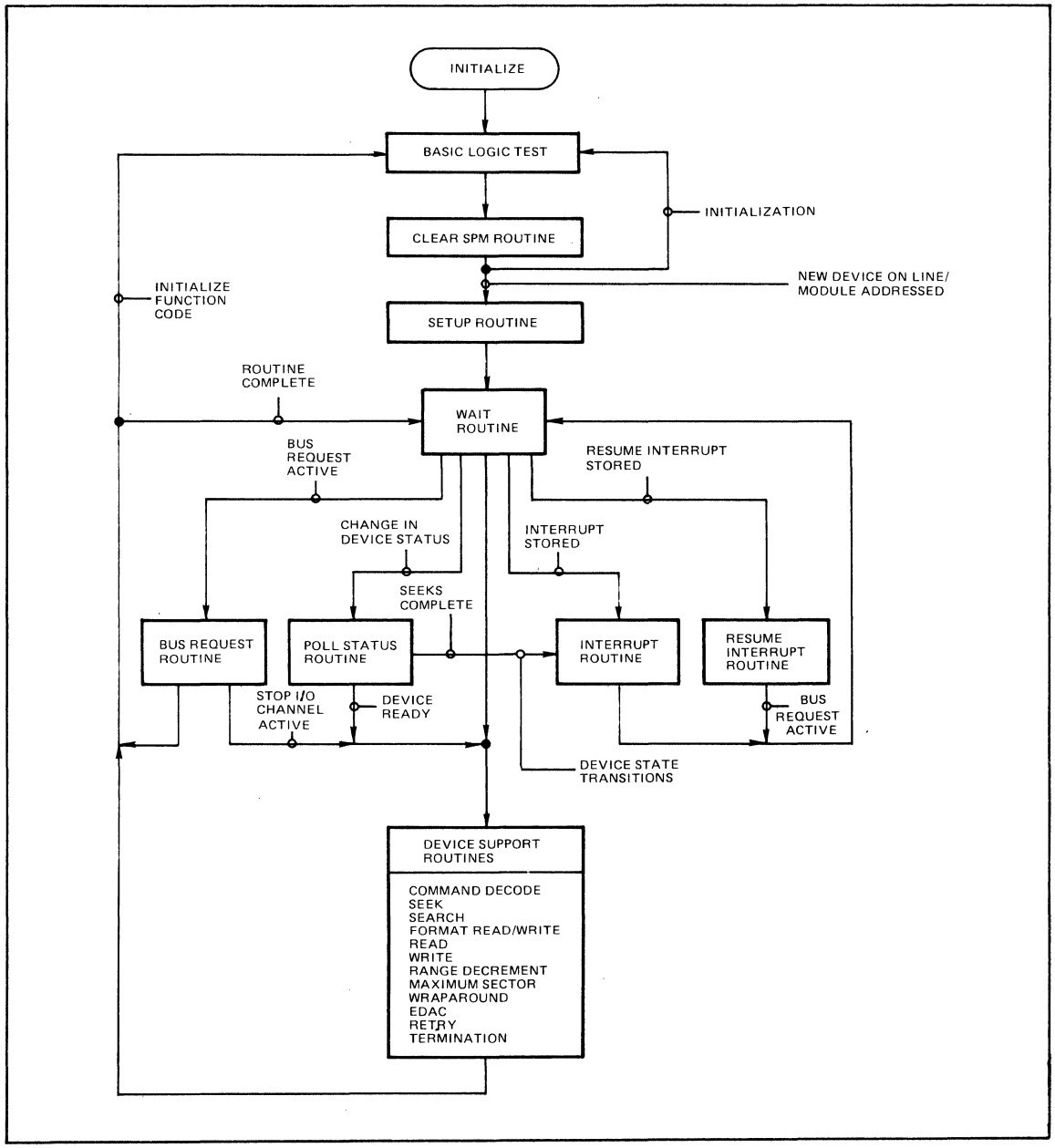


Figure 3-15 MPDC Overview Flow Chart

Upon completion of the Clear SPM routine, the firmware initiates the Setup routine. When the Setup routine is finished, the basic firmware used for verification and selection has been completed and will not be re-entered unless an initialize (software or hardware) is received.

At this point, processing of the Wait routine begins. Firmware decides which of three paths to take, depending on the type of channel/bus activity present. The Wait routine loops until one of these conditions occurs:

- A bus request is detected
- An adapter request is detected
- No requests are present and the adapter is not busy.

When one of these situations is present, the Wait routine exits to the corresponding area: Bus Request routine, the appropriate device support routine, or the Polling routine. The Bus Request and the device support routines re-enter the Wait routine when completed. The Polling routine enters either the Interrupt routine or the proper device support routine.

3.4.2 MPDC Firmware Routines

3.4.2.1 Basic Logic Test (BLT)

The BLT is a firmware routine which functionally verifies major logic components of the MPDC and adapter to ascertain operational capabilities.

The BLT is initiated by way of a Master Clear, which occurs as the result of a software command (function code = 01 hex), a maintenance panel reset, or the powering-up process. Upon entry, the microprogram address counter is set to zero and the LED located on the rear edge of the MPDC board is illuminated.

The routine then proceeds sequentially to ensure the functional capabilities of the following:

- ROS Scan
- Test and Skip operation
- Branch operation
- Status Flip-Flops
- ALU and Accumulator
- Bus Interface Register
- Return Address Register
- Bus Counters and Logic
- Index Register
- Scratch Pad Memory and Registers
- Bus Interface Shift Register
- Adapter First-In-First-Out (FIFO) Buffer
- Bus Drivers, Receivers, Address and Data Registers
- MPDC FIFO buffer and Adapter Read/Write Cycles
- Memory Yellow Counter
- Adapter Read, Write, and EDAC Logic.

Upon successful completion of the BLT (no halt incurred), a firmware command is issued which extinguishes the LED, and the firmware enters the Clear SPM routine.

3.4.2.2 Clear Scratch Pad Memory Routine

The Clear SPM routine is entered from the BLT under two circumstances. First, while in the initialization process, the BLT exits to the Clear SPM routine prior to initiating the SPM testing. Under this condition, when the Clear SPM routine finishes, it will return to the

BLT. Second, the Clear SPM routine is entered upon successful completion of the BLT. After completing the Clear SPM routine, an exit is made to the Setup routine. Regardless of the condition for entry, the Clear SPM routine resets all locations of the scratch pad memory.

3.4.2.3 Setup Routine

The Setup routine is entered from the Clear SPM routine and is used to set up the device and each of the four segments of the SPM with the device-specific information. The routine initializes the selected channel and reads the identification byte from the SMA. The scratch pad memory is then addressed, and the two bytes of ID and the firmware revision are written into the SPM. The routine then reads the device status from the SMA and loads it into the SPM at which point it branches to the recalibrate portion of the Seek routine to reset the device. The Setup routine is then re-entered, and the same procedure repeated with the next sequential channel until all channels are completed. When the last channel has been set up, this routine unconditionally exits to the Wait routine.

3.4.2.4 Wait Routine

The Wait routine utilizes the test multiplexer to determine if a request of any type is present. If there is a request active and it is from the Megabus, the firmware exits to the Bus Request routine. If the request exists as the result of the adapter, the firmware sets the index register with the number of the requesting channel. This enables the MPDC logic and SMA hardware associated with that channel. The Wait routine then exits to the device support routine that previously loaded the return address register (operation dependent).

When no requests of any type are active and the SMA is not busy, the Wait routine enters a polling process to test for any type of operation that has been stacked for any channel due to a busy condition. When any type of command (Seek, Read/Write) is stacked or a change in device status has occurred, the firmware enters the appropriate process routine and, if necessary, from there to a device support routine.

When no stacked operation is encountered on the indexed channel, the Wait routine sequentially examines each remaining channel to determine if an interrupt is stored. The stored interrupt can be the result of an error on the specific channel while the SMA is busy. When an interrupt is stored, the Wait routine exits to the Interrupt routine, or if no stored interrupts, the start of the Wait routine is re-entered to check for pending interrupts.

3.4.2.5 Bus Request

Upon detection of a Megabus transfer to the MPDC, the Wait routine loads the requesting channel number and branches to the Bus Request routine. Since the Interrupt and Resume Interrupt routines complete all Megabus transfers which they initiate, the Megabus transfer causing this request is unsolicited. This indicates that this bus cycle was initiated by the central processor or by another controller.

If no response is required, the pertinent information from the bus interface register, data and address segments, is stored in the scratch pad memory. This is accomplished by utilizing the function code augmented by the channel number in the index register as the SPM address. At this point the Bus Request routine decodes the function code to determine the operation to be performed and then branches to the Basic Logic Test, the Wait routine, or the appropriate device support routine via the Wait routine.

If a response to the Megabus transfer is required, the scratch pad memory is accessed by using the function code and index register as the address. The two bytes of data read from the scratch pad memory are loaded into the bus interface register, the proper cycle parameters are established, and the response cycle initiated. The response cycle is completed and the Megabus logic cleared prior to exiting to the Wait routine.

3.4.2.6 Poll Status Routine

The Poll Status routine is called when the polling process of the Wait routine detects a change in device status. The Poll Status routine analyzes the change to determine if any

seek operations have been completed or if the Ready state of the device has made a transition. The routine exits to the proper device support routine to either log and/or report the change. When the seeks have been successfully completed, the Poll Status routine exits to the Interrupt routine.

3.4.2.7 Interrupt Routine

The Interrupt routine is entered after detection of a device Ready transition, after the completion of a data transfer, or after the execution of an output control word.

This routine generates the interrupt vector and loads the bus interface register with the three bytes of CP address and the two bytes of interrupt vector. It also initiates the bus cycle and waits for a response before unconditionally exiting to the Wait routine.

3.4.2.8 Resume Interrupt Routine

The Resume Interrupt routine is entered from the polling process of the Wait routine when the resume interrupt function is set and a previous interrupt was NAKed. This routine unstacks all the interrupts which are pending as a result of being NAKed and reattempts them. To accomplish this, the firmware sequentially examines all channels, starting with channel 0. To determine if an interrupt is pending, the channel monitor byte is checked and, if warranted, an interrupt bus cycle is attempted. If the interrupt is acknowledged (ACK response) by the CP, the interrupt pending bit of the channel monitor byte is reset. In the case of no acknowledge (NAK response), the interrupt pending bit remains set, and the interrupt is attempted again on detection of the next resume interrupt test.

3.4.3 Device Support Firmware Flow Charts

The device support routines available to the storage module subsystem for the execution of device-specifc operations are:

- Command Decode
- Seek
- Search
- Format (read and write)
- Read
- Write
- Wraparound
- Range Decrement
- Maximum Sector
- EDAC
- Retry
- Termination

Each of the routines performs the functions designated and any in-process setup or verification required. The Termination routine completes device routine execution, determines error and status conditions, and exits to the MPDC Interrupt routine to report the status to software.

Appendix A

T&V Tests (MSUS1)

Through the use of the Test and Verification Program (MSUS1), a series of tests can be performed on the adapter and devices(s). As part of the tests, device specific commands are sent to the device(s). These commands require that specific cylinders, tracks and sectors be located, formatted, and written with test data. The disk areas are then read and the data compared with the data that was written. Any errors are recorded and issued in an error report.

For information pertaining to the sequence of steps required to run this program, refer to the program listing and the *Level 6 System Checkout and Operator's Guide* (Order No. AW94).

Appendix B
Adapter Timing Diagram

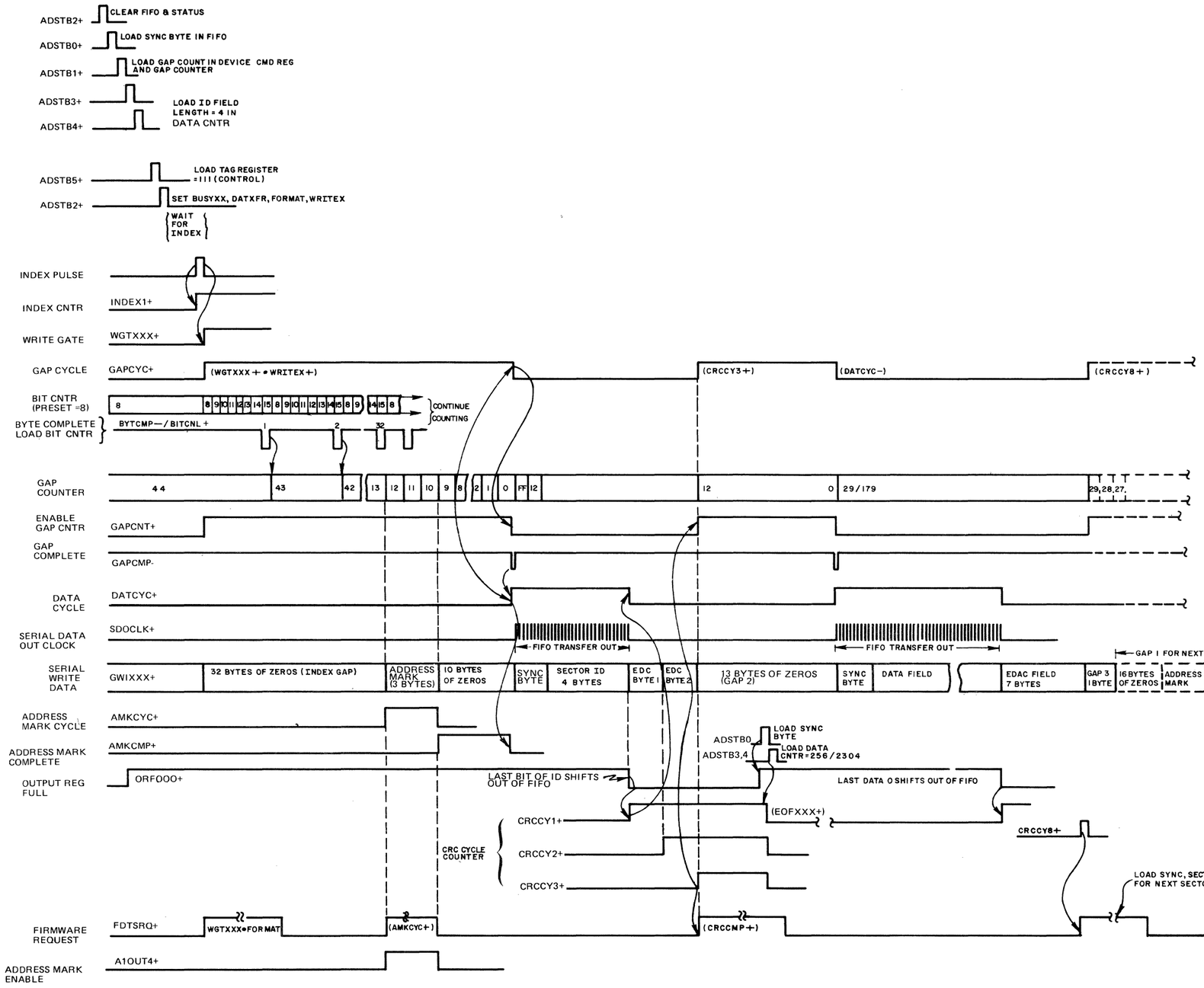


Figure B-1 Format Write Operation

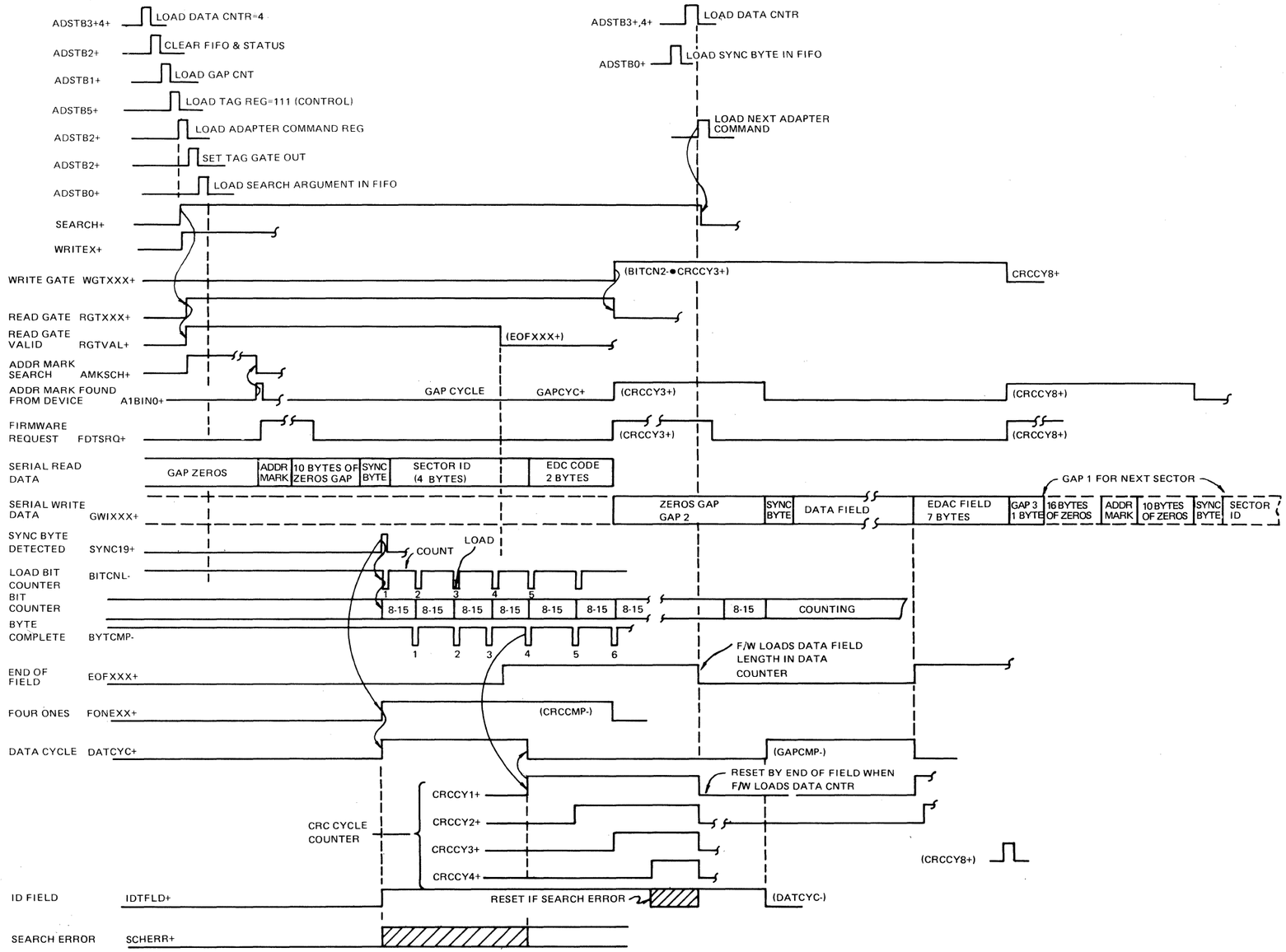


Figure B-2 Format Read Operation

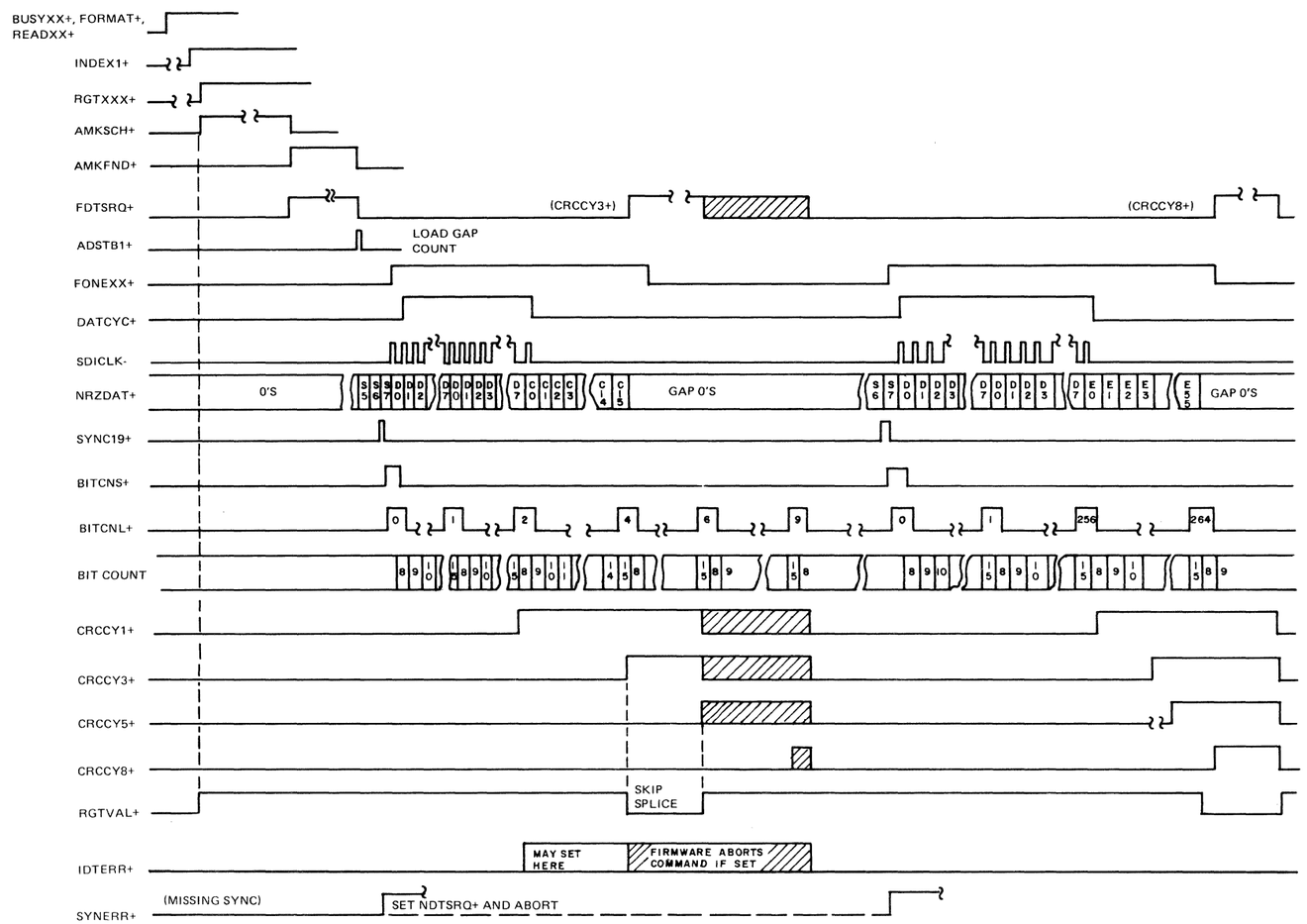


Figure B-3 Search and Write Operation

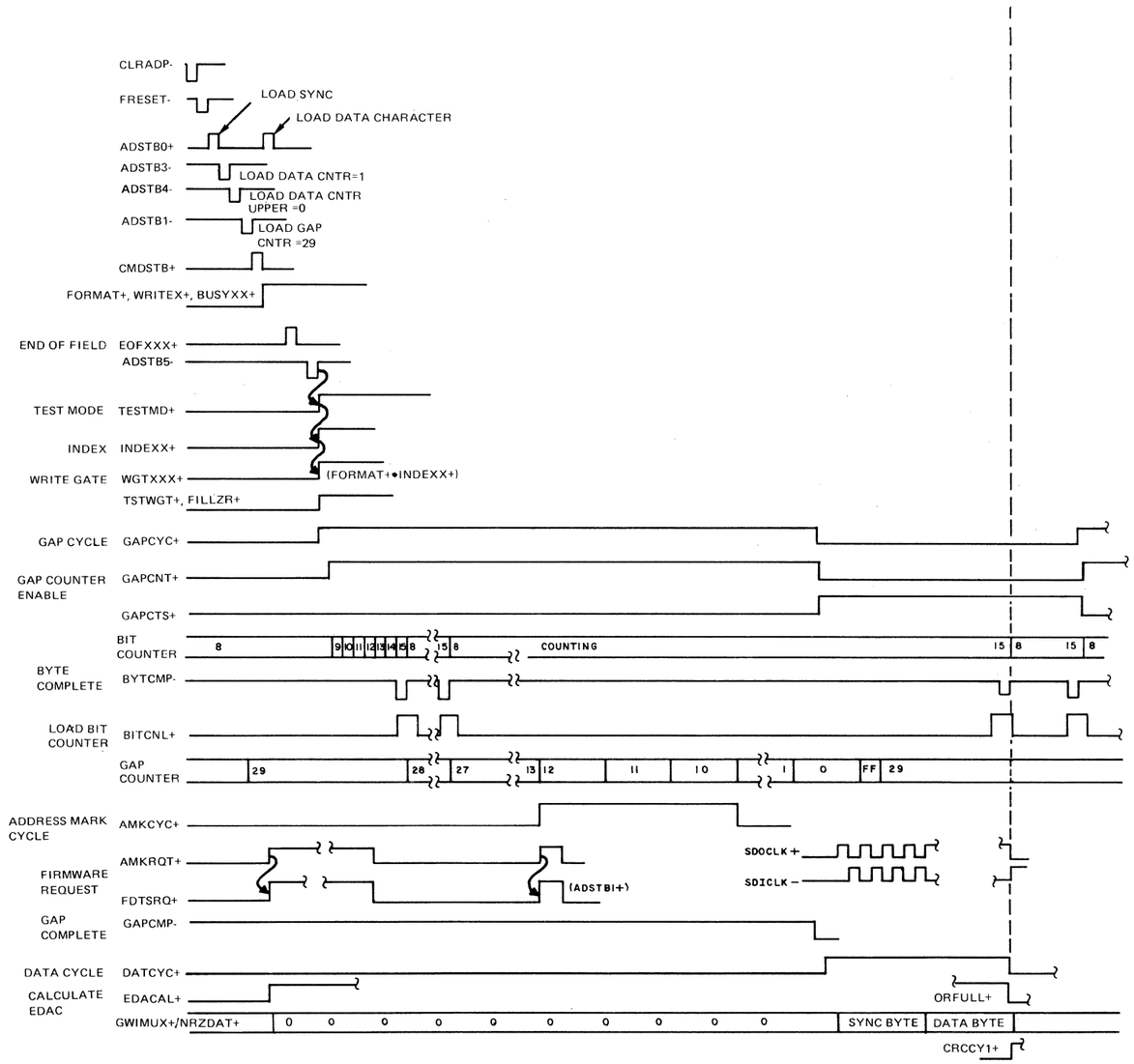


Figure B-4 Test Mode Write

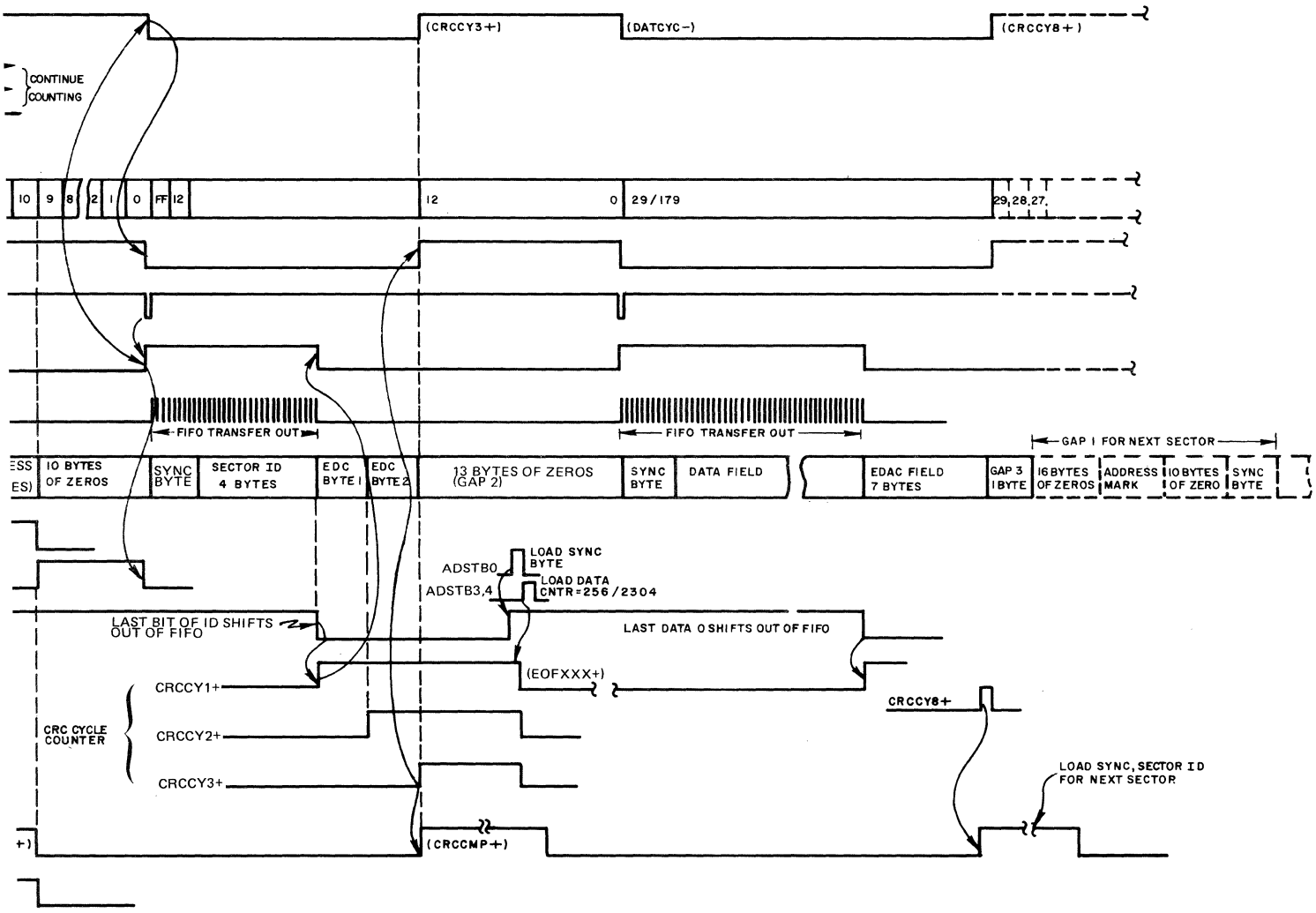


Figure B-1 Format Write Operation

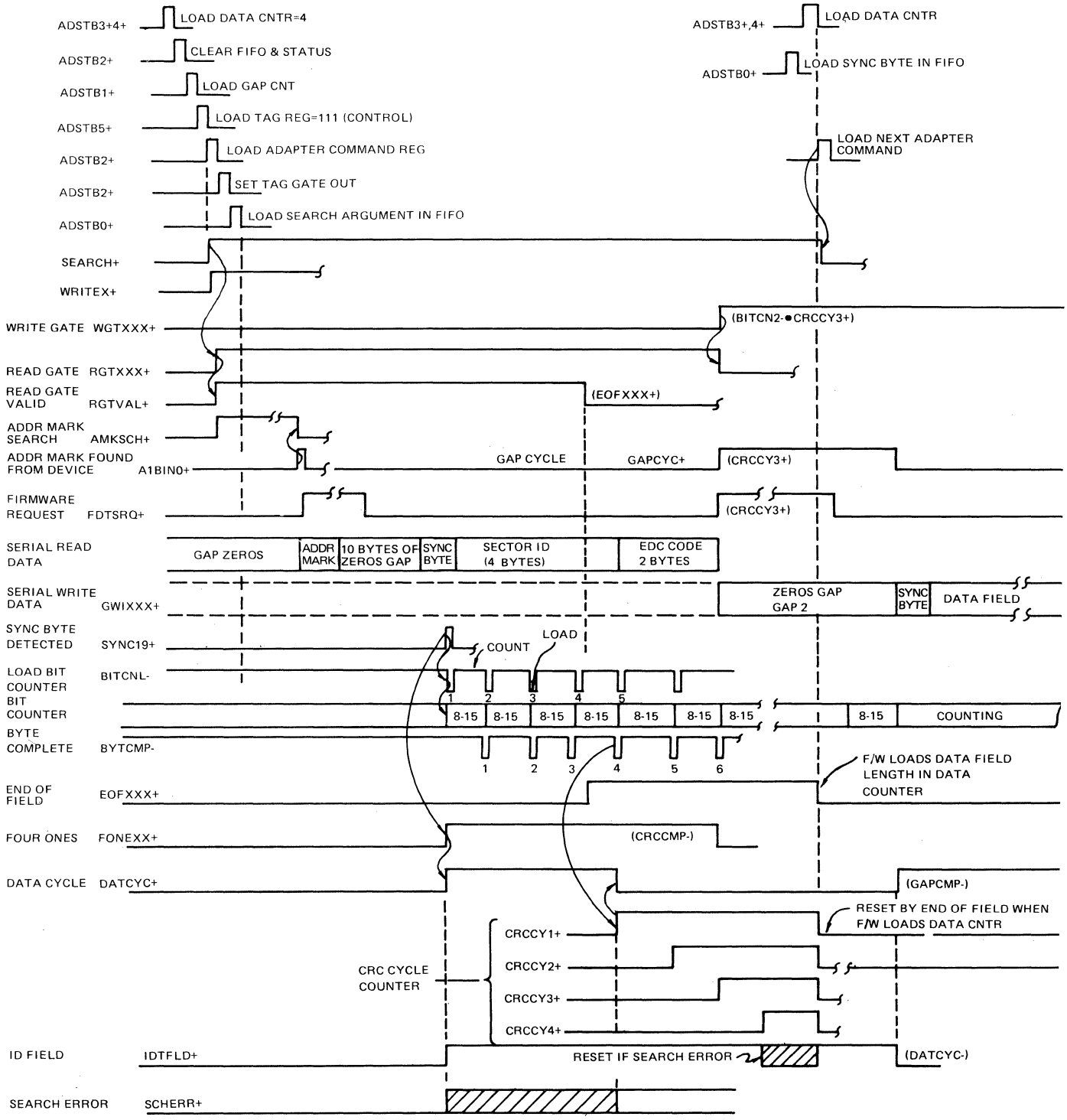


Figure B-2 Format Read Operation

Honeywell

Honeywell Information Systems
In the U.S.A.: 200 Smith Street, MS 486, Waltham, Massachusetts 02154
In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario M2J 1W5
In Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F.

21196, 3C878, Printed in U.S.A.

FN80, Rev. 1