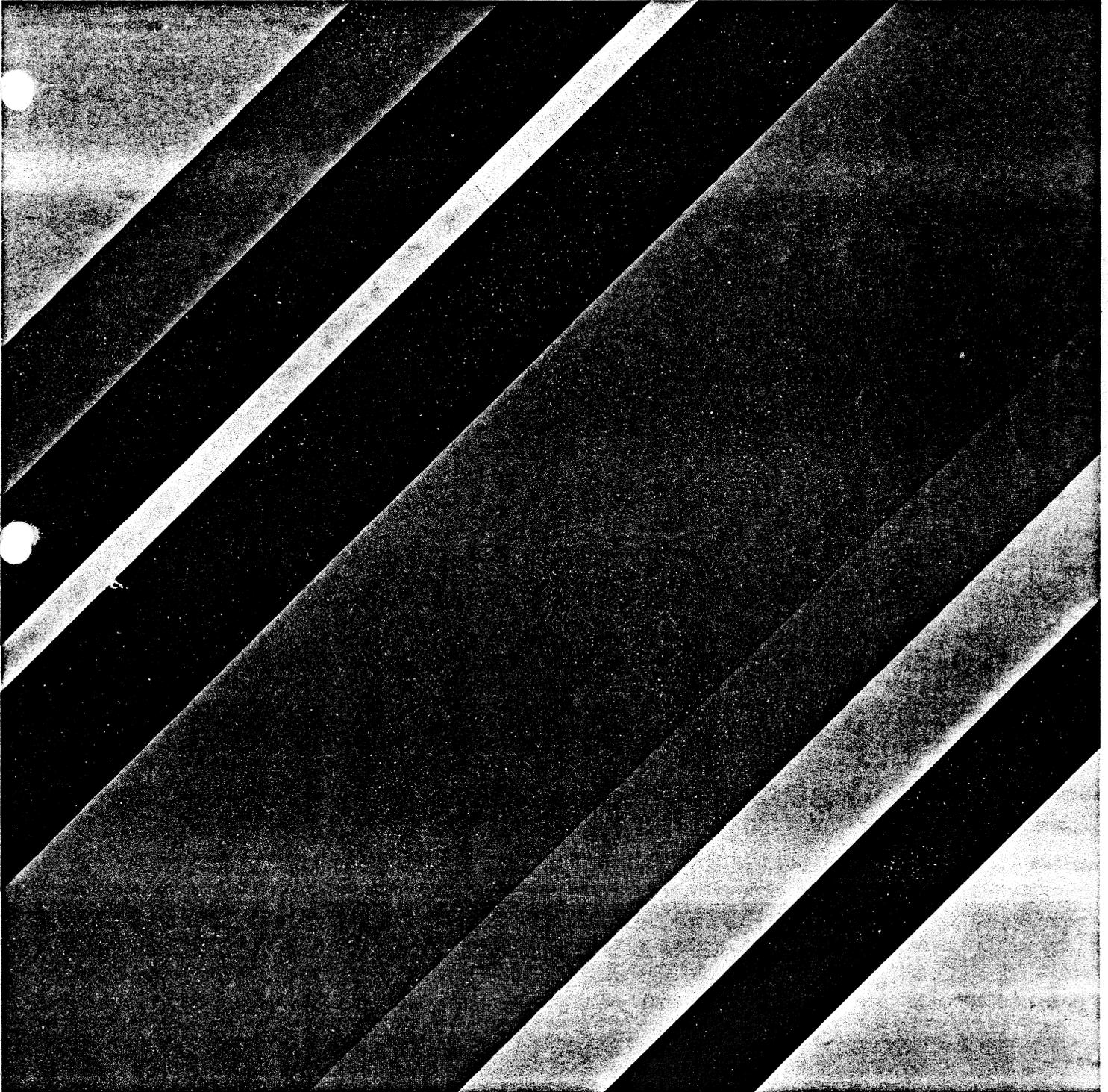


PRODUCT DESCRIPTION

D120/D140 MIDIDISK CONTROLLER

OEM SALES



MDC

MIDIDISK DRIVE CONTROLLER

Product description for the D120 / D140 disk drive controller

DOCUMENT REFERENCE No.: BEM 1174 / AO DECEMBER 1977

Cii Honeywell Bull
OEM SALES
6, AV. DES USINES
90001 BELFORT/FRANCE

MDC CONTROLLER - PRODUCT DESCRIPTION

NOTES

MDC CONTROLLER - PRODUCT DESCRIPTION

TABLE OF CONTENTS

1	INTRODUCTION	3
2	OPERATIONAL MODES	
	2.1 Buffered mode	3
	2.2 Real time mode - autonomous halt	3
	2.3 Real time mode - external halt	3
3	INTERFACE	
	3.1 Interface loading	4
	3.2 Interface logic	4
	3.3 Interface connector	5
	3.4 Interface lines	6
	3.4.1 Buffered mode interface lines..	6
	3.4.2 Real time mode interface lines..	7
	3.5 Signal lines	8
	3.5.1 Control lines.....	10
	3.5.2 I/O Bus.....	11
4	I/O BUS STATES	
	4.1 Status mode	12
	4.2 Command mode	12
	4.3 Read Data mode	12
	4.4 Write Data mode	12
5	COMMAND INSTRUCTIONS	12
6	READ OPERATION	17
7	WRITE OPERATION	18
8	STATUS READ OPERATION	
	8.1 Status Register	19
	8.2 Status Bits	21
	8.3 Flag line	23
9	DATA TRANSFER	24
10	SECTOR FORMAT	26
11	BAD SECTOR PROCESSING.....	26
12	READ/WRITE OPERATION FLOW CHARTS	27
13	SECTOR UPDATING OPERATION FLOW CHART.....	27

MDC CONTROLLER - PRODUCT DESCRIPTION

LIST OF ILLUSTRATIONS

Figure	1 Interface line loading.....	4
	2 Interface connector.....	5
	3 Interface signals and pin nos - buffered mode.....	6
	4 Interface signals and pin nos - real time mode.....	7
	5 Buffered controller interface circuits.....	8
	6 Real time controller interface circuits.....	9
	7 Read operation - buffered mode.....	17
	8 Read operation - real time mode.....	17
	9 Data buffer.....	18
	10 Write operation - buffered mode.....	18
	11 Write operation - real time mode.....	19
	12 Status transfer timing.....	20
	13 Data transfer timing - buffered mode.....	24
	14 Data transfer timing - real time mode.....	25
	15 Sector format.....	26
	16 Read/write operation flow chart - buffered mode.....	28
	17 Read/write operation flow chart - real time mode.....	29
	18 Sector updating flow chart - buffered mode.....	30

LIST OF TABLES

Table	1 I/O Bus states.....	11
	2 Subsystem command set.....	13
	3 Status Register bits.....	20
	4 BSF status.....	21

1 INTRODUCTION

Any user-system based on an 8 bit external bus - for example a micro-processor based usersystem - can be interfaced with the MDC controller, designed to simplify the D120 / D140 Disk Drive system connection.

Adaptable to specific user systems the controller can operate as a slow buffered device, or as a rapid transfer device working in real time with autonomous or external halt at the end of processing.

Interfacing is simple - only 15 lines for data and control signals are required.

2 OPERATIONAL MODES

Three different modes of operation have been designed into the controller, allowing a universal approach to different system architectures, and varied applications of the disk drives. These operational modes are discussed in detail in the following paragraphs. Throughout the remainder of the text, differences are noted where applicable.

2.1 BUFFERED MODE

For systems working with several peripheral devices, where a disk subsystem is accorded a low priority, this mode allows a slow, asynchronous data transfer into a memory buffer. Capacity of the buffer is $4 \times 256 \times 9$ bits (including a data integrity parity bit). This allows storage of up to 4 consecutive sectors, each sector containing a maximum of 256 data bytes (1 byte = 8 bits)

Another application of the buffered mode is updating. In this case the buffer is addressed and partial updating can be effected without transferring a block of complete sectors from the disk to the processor.

2.2 REAL TIME MODE - AUTONOMOUS HALT

As the name suggests, data is transferred at a rate of 916 Kbytes/second in real time in blocks of up to four sectors. After the processor initiates a read/write operation the controller executes the required operation and automatically halts when the specified number of sectors has been processed.

2.3 REAL TIME MODE - EXTERNAL HALT

This operational mode is similar to the real time mode - autonomous halt, with a transfer rate of 916 Kbytes/second. However, the difference is in that the number of sectors to be processed is not specified. After the processor initializes a read/write command, the controller starts execution of the command and continues the required operation until a halt instruction is received from the processor.

As a result, it is possible to read a track several times, or, write a complete track.

MDC CONTROLLER - PRODUCT DESCRIPTION

3 INTERFACE

The user system controller interface comprises 15 lines. These lines are described in detail in § 3.4 and shown schematically in figures 3 and 4.

3.1 INTERFACE LOADING

Each of the 15 lines is terminated in 240 ohms to the + 5V power supply and 660 ohms to ground. A similar termination at the system end interface is recommended. Lines asserted by the Controller are driven by an open collector driver, while lines asserted by the system interface are received by a standard TTL gate device.

3.2 INTERFACE LOGIC

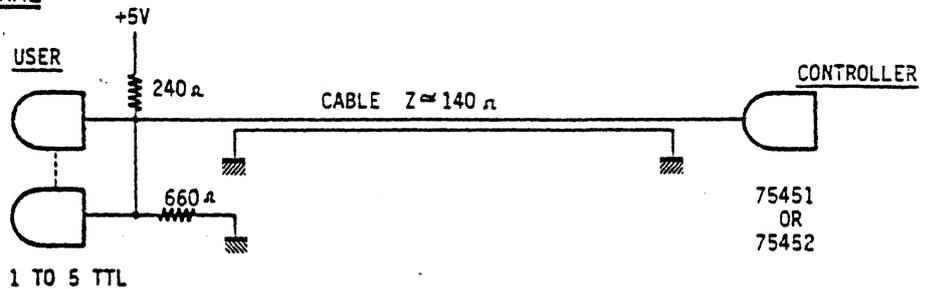
The logic used at the interface is negative TTL. Logic levels are defined hereunder :

- HIGH : from 2.8 v to 5.25 volts, which corresponds to logic zero, or not asserted.
- LOW : from 0 v to 0.4 volts, which corresponds to logic one, or ground asserted, or, asserted.

NOTE

An asterisk following the signal name indicates that the signal is significant at low level.

UNIDIRECTIONAL



BIDIRECTIONAL

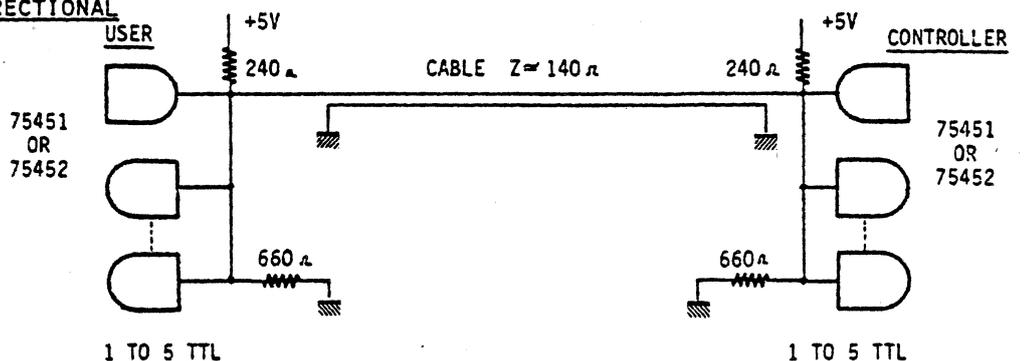
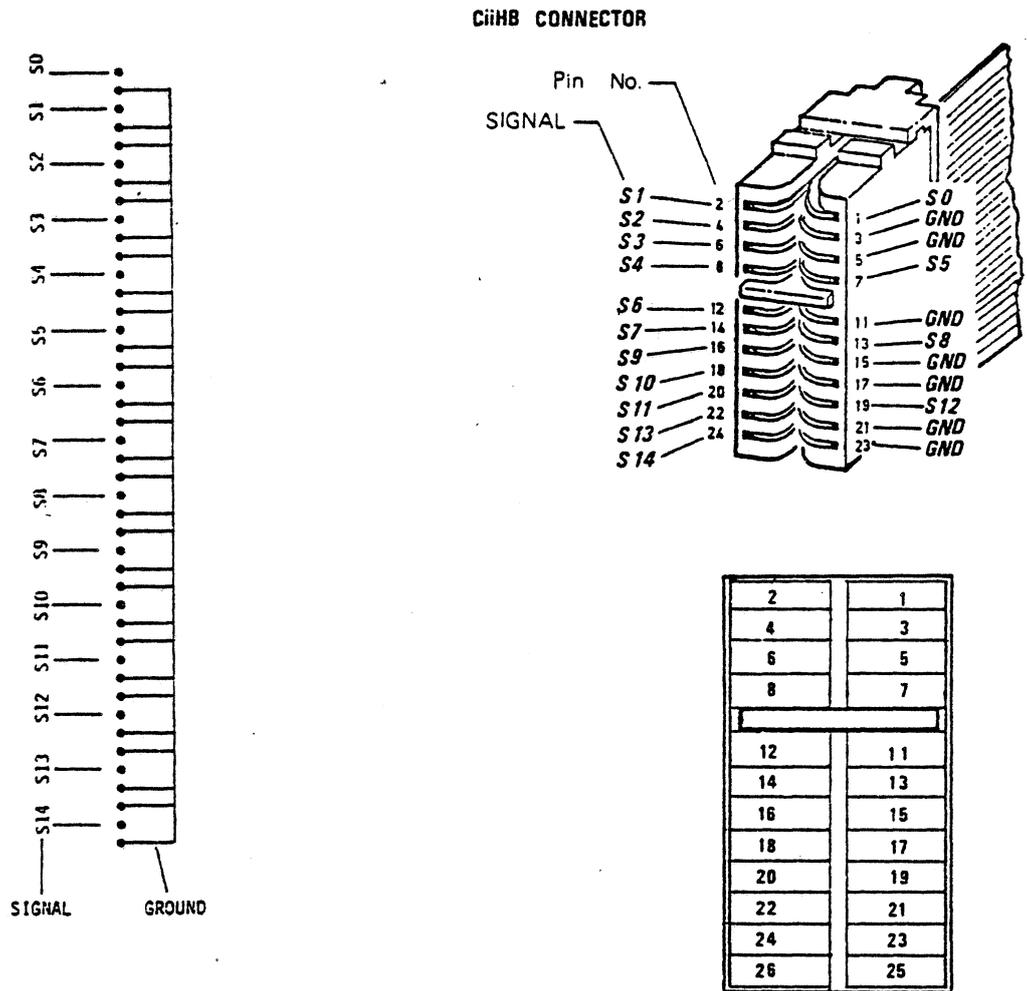


Figure 1. Interface Line loading

3.3 INTERFACE CONNECTOR

Physical connection between the Controller and the user system is ensured by a 43 conductor flat cable. The standard CiiHB cable connector is shown in figure 2. Other connectors, type HE9 may also be accommodated.

Figures 3 and 4 contain the identification of each interface signal and the number of its connector pin.



VIEW ON FLAT CABLE
ENTERING THE Cii HB CONNECTOR

HE9 TYPE CONNECTOR

NOTE Signal/pin allocation is the same as the CiiHB connector. Pins 25 and 26 are not connected.

Figure 2. Interface connector

MDC CONTROLLER - PRODUCT DESCRIPTION

3.4 INTERFACE LINES

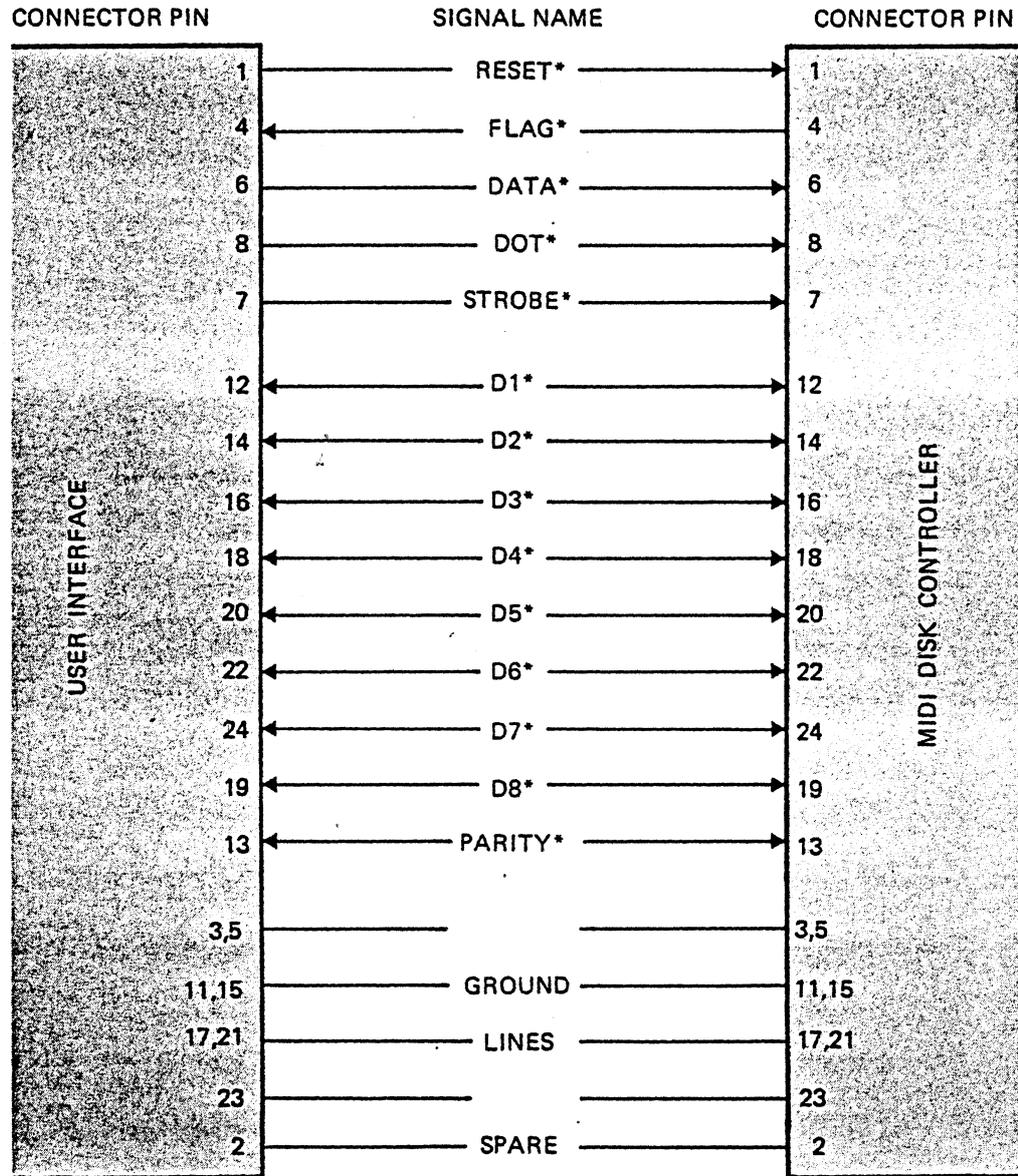


Figure 3. Interface signals and pin Nos. - Buffered Mode

MDC CONTROLLER - PRODUCT DESCRIPTION

3.4.2 Real Time Mode interface lines

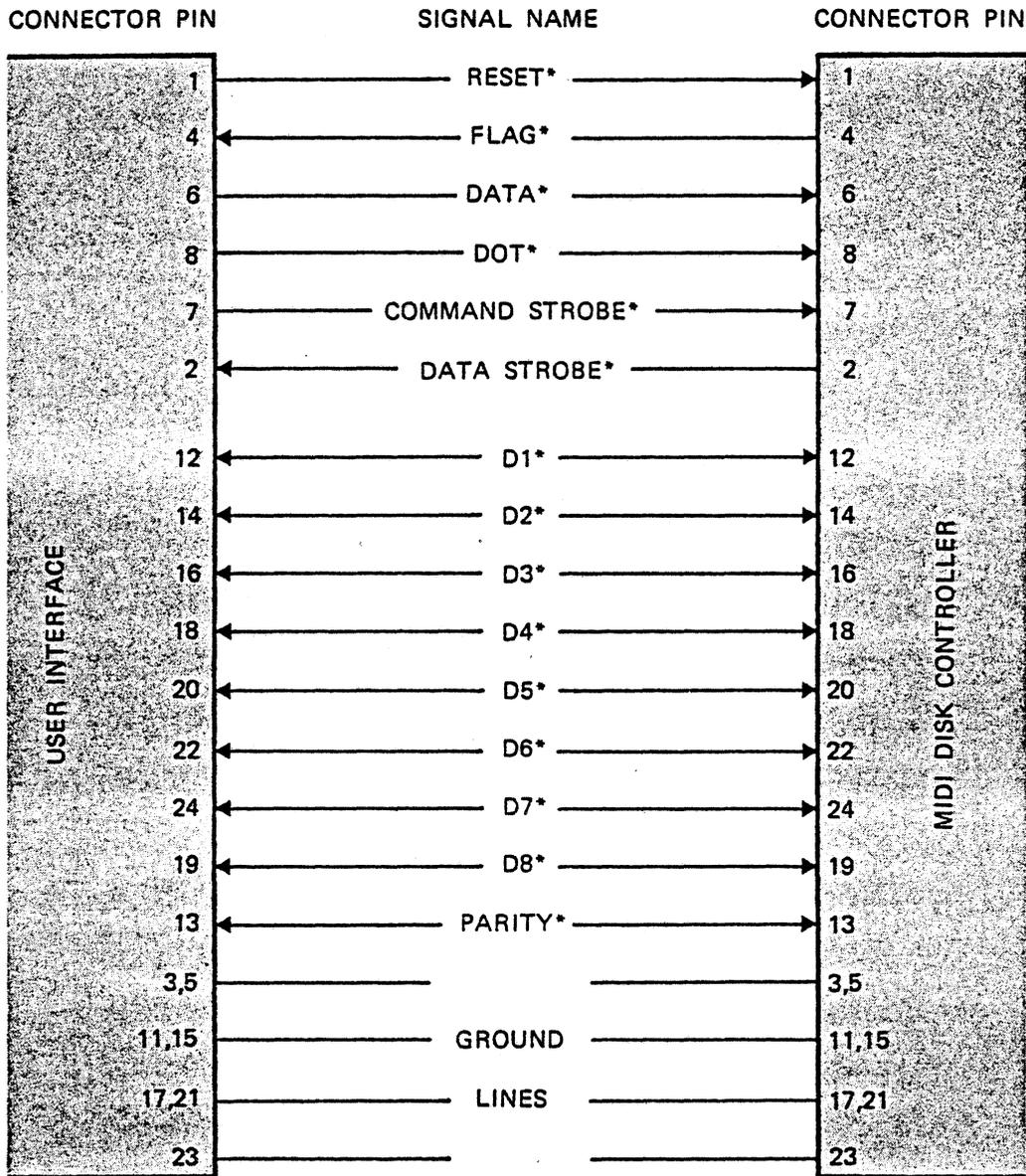


Figure 4. Interface signals and pin. Nos. - Real Time Mode

MDC CONTROLLER - PRODUCT DESCRIPTION

3.5 SIGNAL LINES

Interface lines fall into two categories - the I/O Bus (8 lines) and 5 control lines (or 6 control lines with real time operation) - refer to the following paragraphs for a full detailed description. Figure 5 below, shows the interconnection between the interface lines and the controller buffer and registers. When the controller is operating in real time mode, no buffer is required. The controller interface circuits for this mode are shown in figure 6.

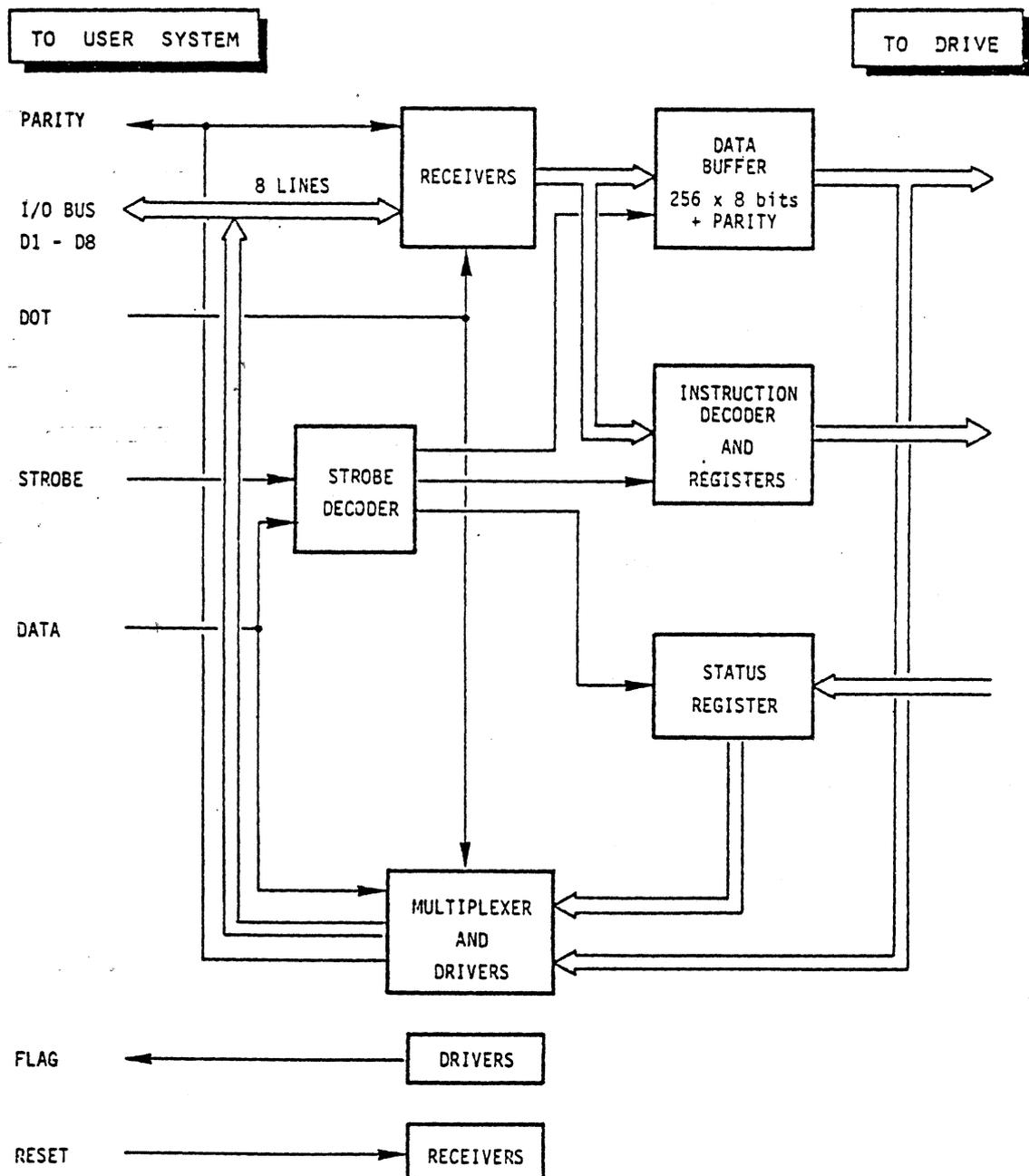


Figure 5. Buffered controller interface circuits

MDC CONTROLLER - PRODUCT DESCRIPTION

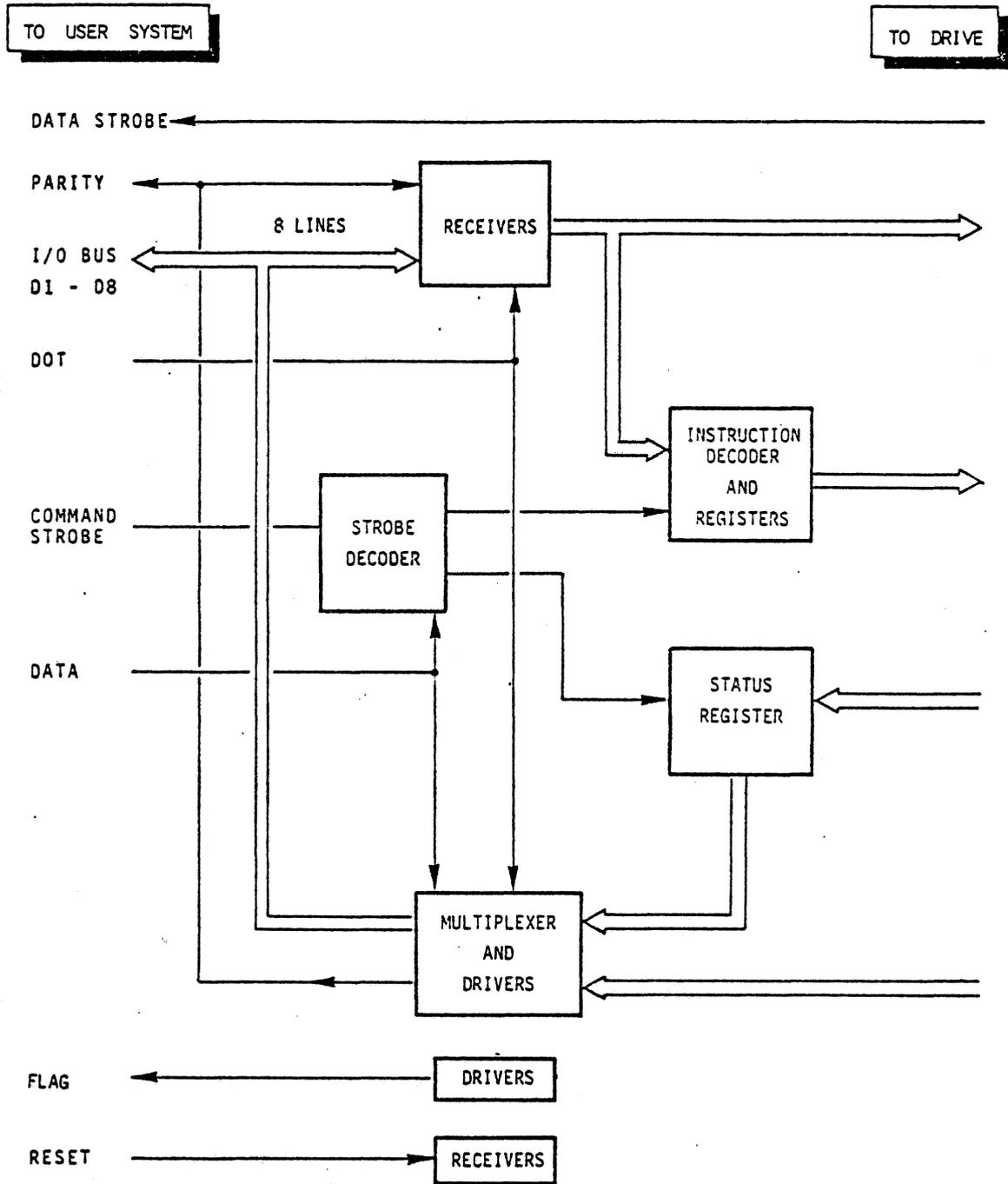


Figure 6. Real Time controller interface circuits

MDC CONTROLLER - PRODUCT DESCRIPTION

3.5.1 Control Lines

3.5.1.1 RESET*

The RESET line is a computer interface originated, ground asserted control. It is asserted by the computer for approximately 300 ms upon power up and may be asserted by the interface to unconditionally reset the MDC.

3.5.1.2 FLAG*

The FLAG is an MDC originated, ground asserted control. When asserted, this signal informs the computer interface that the MDC requires service.

3.5.1.3 DATA*

The DATA line is a computer originated ground asserted signal. When it is asserted, the I/O lines (D1 to D8) contain a data byte. When it is not asserted the I/O lines will contain either a command or the controller status byte.

3.5.1.4 DOT*

The DOT line is a computer interface generated, ground asserted signal. When asserted, it indicates that the direction of data transfer is to the MDC. When not asserted, the direction of data transfer is from the MDC.

3.5.1.5 STROBE* or COMMAND STROBE*

The STROBE line is a computer interface generated, ground asserted signal. When asserted, it indicates that information is being transferred. The type of information transferred depends on the operational mode : in buffered mode, the information is Data, Commands and Status ; in real time mode, only Commands or Status are transferred. This line should be asserted for a minimum of 0.4 and a maximum of 2.0 microseconds. The computer interface should not change the state of any other lines during the interval beginning 0.2 microseconds before assertion of STROBE and ending 0.2 microseconds after the release of STROBE. The release of STROBE is interpreted by the MDC as the completion of a transfer.

MDC CONTROLLER - PRODUCT DESCRIPTION

3.5.1.6 DATA STROBE*

The DATA STROBE is an MDC originated ground asserted signal. When asserted in real time mode, it signifies that data is to be transferred to or from the MDC.

3.5.1.7 PARITY*

The PARITY is a bi-directional ground asserted line which carries the parity bit of the I/O Bus lines. This line is asserted if the number of logic ones on the I/O Bus is even.

3.5.2 I/O Bus : D1 - D8

The I/O Bus comprises 8 lines, designated as D1 to D8, with D8 as the least significant bit. The Bus is bi-directional and ground asserted.

Four different types of information may be transferred over I/O Bus, namely, read, write, command and status modes. Each mode is accompanied by its own strobe and is determined by the states of the control lines, DATA and DOT.

4 I/O BUS STATES

Lines DATA, DOT determine the command state of the I/O Bus. The I/O Bus has 4 states :

- 1 - READ DATA TRANSFER
- 2 - WRITE DATA TRANSFER
- 3 - COMMAND CONTROL
- 4 - READ STATUS TRANSFER

The truth table given below shows the logic conditions of the DATA and DOT lines and the state selected. Signal STROBE or COMMAND STROBE must also be present.

Table 1. I/O Bus Table

DATA STATE	DOT STATE	I/O BUS STATE
0	0	READ STATUS
0	1	CONTROL COMMAND
1	0	READ DATA
1	1	WRITE DATA

MDC CONTROLLER - PRODUCT DESCRIPTION

4.1 STATUS MODE

This mode requires DATA and DOT to be unasserted and causes the MDC to place status information on the I/O BUS. A strobe asserted in this mode is interpreted as a Read Status Strobe (RSS).

4.2 COMMAND MODE

This mode requires DOT asserted and DATA unasserted and causes the MDC to interpret the D lines as a command. D lines are examined only while STROBE is asserted. A strobe in this mode is interpreted as an Output Command Strobe (OCS).

4.3 READ DATA MODE

This mode requires DATA asserted and DOT unasserted and causes the MDC to place the currently held data byte on the I/O Bus. This data is valid in buffered mode during a data service request until the trailing edge of Read Data Strobe (RDS) is sensed by the MDC, and in real time mode during a data Strobe.

4.4 WRITE DATA MODE

This mode requires both DATA and DOT to be asserted and causes the MDC to interpret the D lines as a data word to be written (D lines are examined only while Write Data Strobe in buffered mode or Data Strobe in real time mode is asserted).

5 COMMAND INSTRUCTIONS

Table 2 lists all the commands that the controller will execute. To command the controller, the user system transfers the appropriate command instruction on the I/O Bus, asserts DOT and then strobes in with the STROBE or COMMAND STROBE line.

To initiate a read or write operation the user system must transfer a command sequence of three consecutive bytes.

Byte 1 gives the least seven significant bits of the track address where the operation is to start. Byte 2 identifies the two most significant bits of the track address and the head select. Byte 3 gives the six bit sector address and whether the data is to be read or written.

MDC CONTROLLER - PRODUCT DESCRIPTION

In each read or write operation, three consecutive commands instructions are necessary.

Command bytes 1 and 2 are reserved for track and head addressing. Command byte 3 contains either a WRITE or READ Command, together with the sector address where the operation is to be performed.

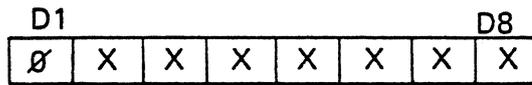
When the controller enters the BUSY state, the read/write commands - i.e., command with bit D1 = zero - are non-operational until the current operation is terminated. All other control commands (command with bit D1 = One) are normally executed.

Table 2. Subsystem command set

I/O BUS LINES									COMMAND
D1	D2	D3	D4	D5	D6	D7	D8	D8	
∅	T2	T3	T4	T5	T6	T7	T8		TRACK SELECT - Selects track TT and prepares to enter the read or write mode
∅	∅	∅	N0	N1	H1	T∅	T1		TRACK SELECT - Selects track TT, the number of sectors NN to be chained, and head H and prepares to enter the read or write mode subsequent to the next command.
∅	∅	S0	S1	S2	S3	S4	S5		READ SECTOR SS - Moves on the previously selected track and head, enters read mode, and reads data starting at sector SS.
∅	1	S0	S1	S2	S3	S4	S5		WRITE SECTOR SS - Moves on previously selected track and head, enters write mode and writes received data starting at sector SS.
1	∅	∅	1	∅	∅	∅	∅		CLEAR DATA SERVICE - Resets the DATA SERVICE and BUSY status bits.
1	∅	∅	∅	∅	∅	∅	∅		RESET - The controller and disk drive are unconditionally reset.
1	∅	1	∅	∅	∅	∅	∅		TERMINATE - this indicates the end of an operation in read or write mode.
1	1	∅	∅	B6	B7	B8	B9		BUFFER SELECT - Selects word BB in the controller buffer (lowest order bits).
1	1	1	∅	B2	B3	B4	B5		BUFFER SELECT - Selects word BB in the controller buffer.
1	1	1	1	∅	∅	B0	B1		BUFFER SELECT - Selects word BB in the controller buffer (highest order bits).

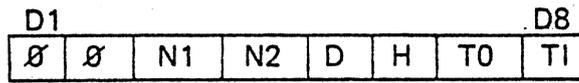
MDC CONTROLLER - PRODUCT DESCRIPTION

5.1 COMMANDE BYTE No. 1



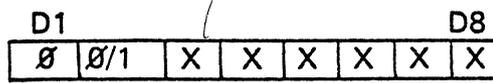
bits 2 to 8 may be 1 or 0, and signify the seven low order bits of the track address.
bit D1 is always 0

5.2 COMMAND BYTE No. 2



bits 7,8 may be 1 or 0. Used with bits 2-8 of command byte No. 1 allow addressing up to 511 tracks.
head select : 0 = upper head
1 = lower head
disk identification
gives the number of sectors to be processed -i.e., up to a maximum of four. These bits are not significant in real-time mode with external halt.
always set to zero.

5.3 COMMAND BYTE No. 3

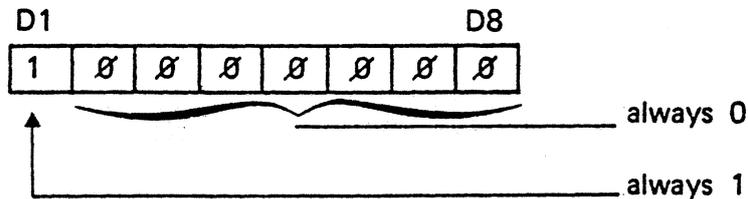


bits D3-D8 give the starting sector number where a read or write operation is to be performed. Bit D3 is the least significant bit.
00 : indicates a read operation
01 : indicates a write operation

5.4 SUBSYSTEM INITIALIZATION

This command is required for unconditionally resetting the MDC. The memory buffer is not scratched, but the buffer address pointer is reset.

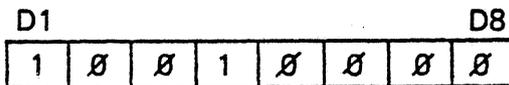
Format :



5.5 CLEAR DATA SERVICE

This command is required, during an up-dating operation effected in buffered mode, to unconditionally abort a read operation in the MDC. On the completion of this, commands DATA SERVICE and BUSY bits are reset. See also updating flowchart, figure 18.

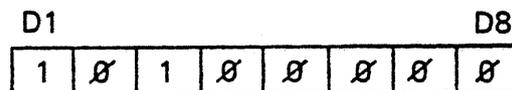
Format :



5.6 DATA TRANSFER COMPLETE, or TERMINATE

This command is sent in the buffered mode of operation to inform the MDC that no more data transfer cycles are required by the user-system. It is also sent in real time mode - external halt during the last sector being read or written in order to stop the data transfer.

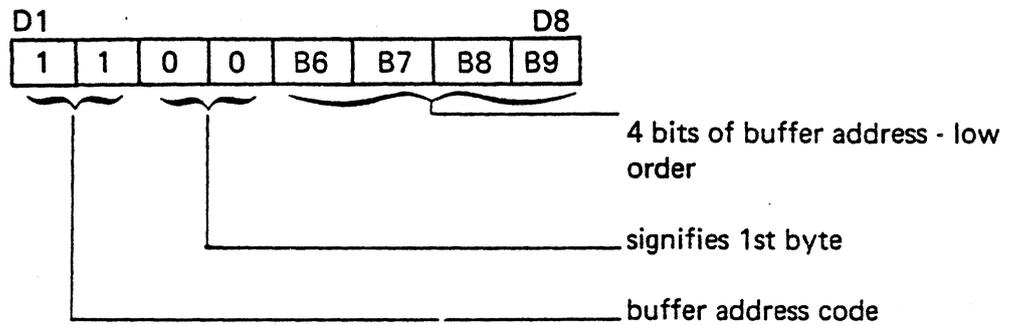
Format :



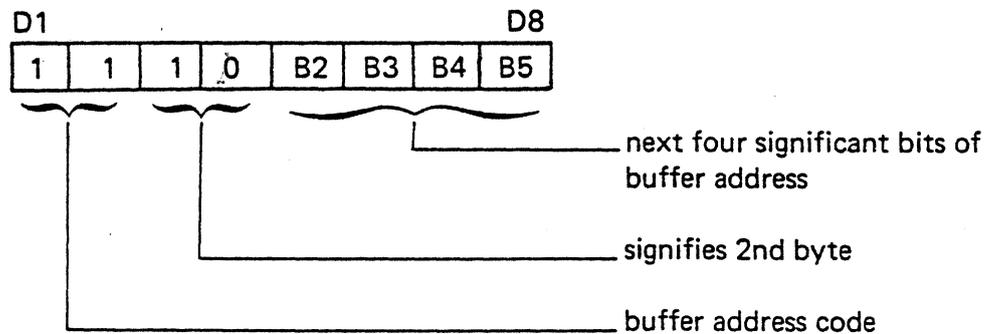
5.7 BUFFER MEMORY ADDRESS

The three commands described below are used in buffered mode only during an updating operation in order to set the buffer memory address. For further details, refer to the updating flowchart, figure 18.

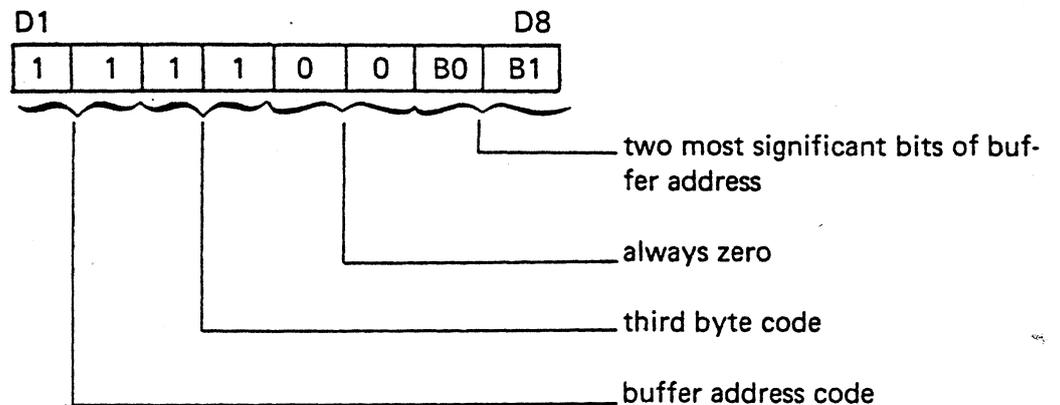
Byte No 1



Byte No 2



Byte No 3



6 READ OPERATION

For a read operation, the user-system performs the 3 byte command sequence described in § 5. If the controller is operating in buffered mode, it waits for the end of the operation, indicated by assertion of the FLAG line, and then initiates asynchronously, X bytes transfer from the controller buffer (with X value of 1 to N x 256, N being the number of sectors to be chained). See figure 7.

If the controller is operational in real time mode, the buffer is bypassed and the data is transferred directly from the disk into the user-system.

BUFFERED MODE READ SEQUENCE

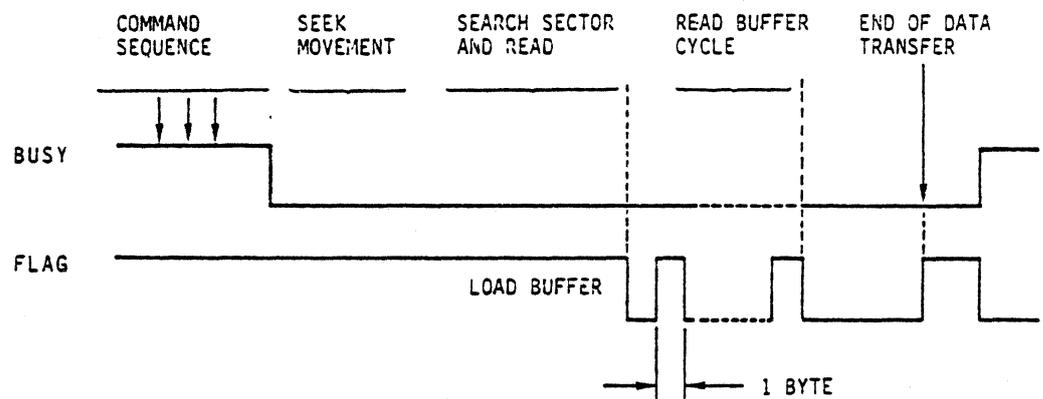


Figure 7. Read operation - buffered mode

REAL TIME MODE - READ SEQUENCE

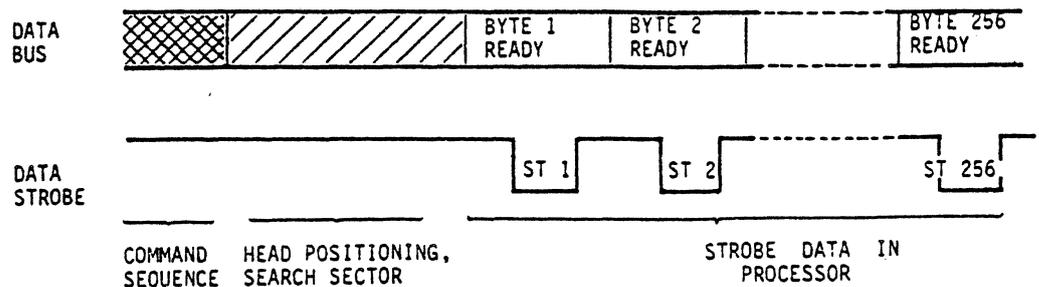


Figure 8. Read operation - real time mode

7 WRITE OPERATION

For a write operation, the user system first performs the 3 byte command sequence, as in the read operation and next initiates (asynchronously, if the MDC is operating in buffered mode) an X bytes transfer into the Data Buffer, (with X = value of 1 to N x 256 ; N being the number of sectors to be chained) and then indicates the end of transfer with a TERMINATE command. If less than N x 256 bytes are transferred into the Data Buffer the remaining data field will be zero filled (see figures 9 and 10).

After the FLAG line assertion indicating the end of operation, the user system must read the Status Register to check that no error occurred in the disk sub-system during the operation. Refer to section 8 for Status Register description. If the MDC is operating in real time mode, then the data is transferred synchronously with DATA STROBE to the MDC. See figure 11.

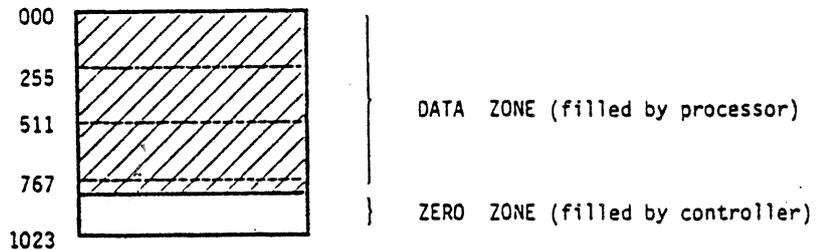


Figure 9. Data buffer

BUFFERED MODE WRITE SEQUENCE

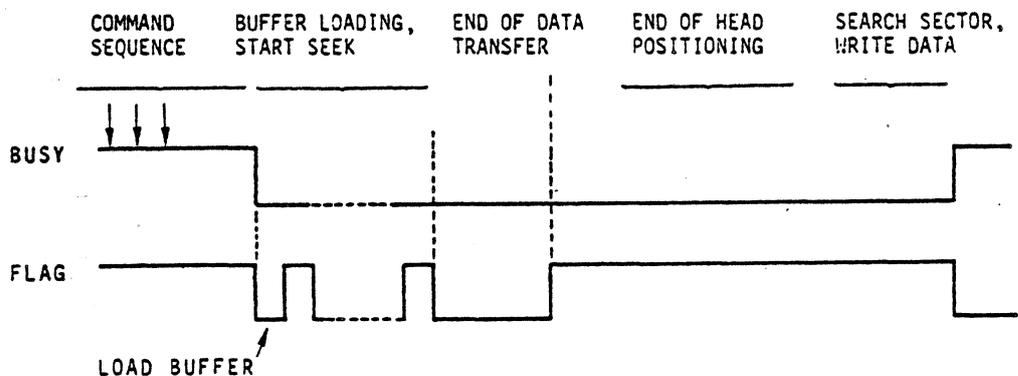


Figure 10. Write operation - Buffered mode

MDC CONTROLLER - PRODUCT DESCRIPTION

REAL TIME MODE WRITE SEQUENCE

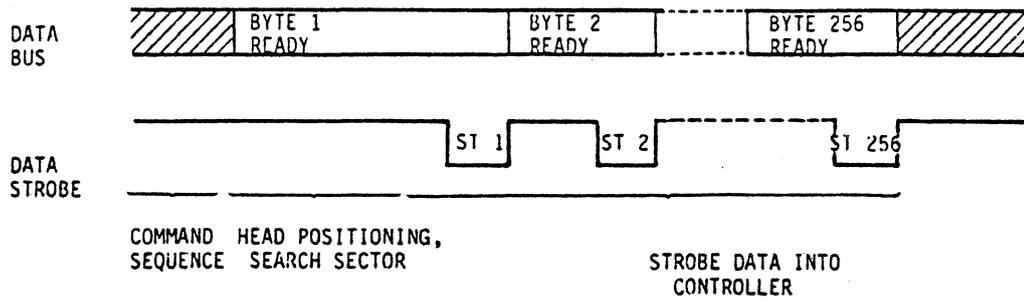


Figure 11. Write operation - real time mode

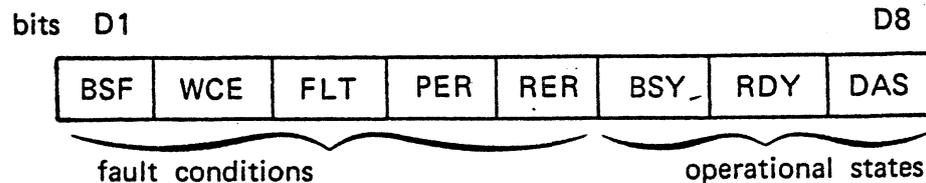
8 STATUS READ OPERATION

The status of the subsystem can be read (sampled) at any time by the external control unit by monitoring the DATA and DOT lines. However, in order to properly reset the status lines that are transient, the computer interface will usually pulse the strobe line after or during the sampling operation, generating an RSS. (= DATA, DOT unasserted, followed by strobe).

If the Write Count Error, the Fault Status, the Parity Error Status or the Read Error Status lines are set, they will be reset on the trailing edge of the Read Status Strobe pulse ; if one of the error bits is set after the leading edge of a RSS strobe, it will only be reset by the next RSS strobe in order to allow proper sampling operation. Figure 12 shows the recommended timing of a status transfer.

8.1 STATUS REGISTER

The status register is used to store information coming from the disk subsystem. This information comprises three functional bits giving the operational status of the disk subsystem and five incident bits indicating fault conditions.



The eight bits are described in detail in § 8.2

MDC CONTROLLER - PRODUCT DESCRIPTION

Table 3. Status Register bits

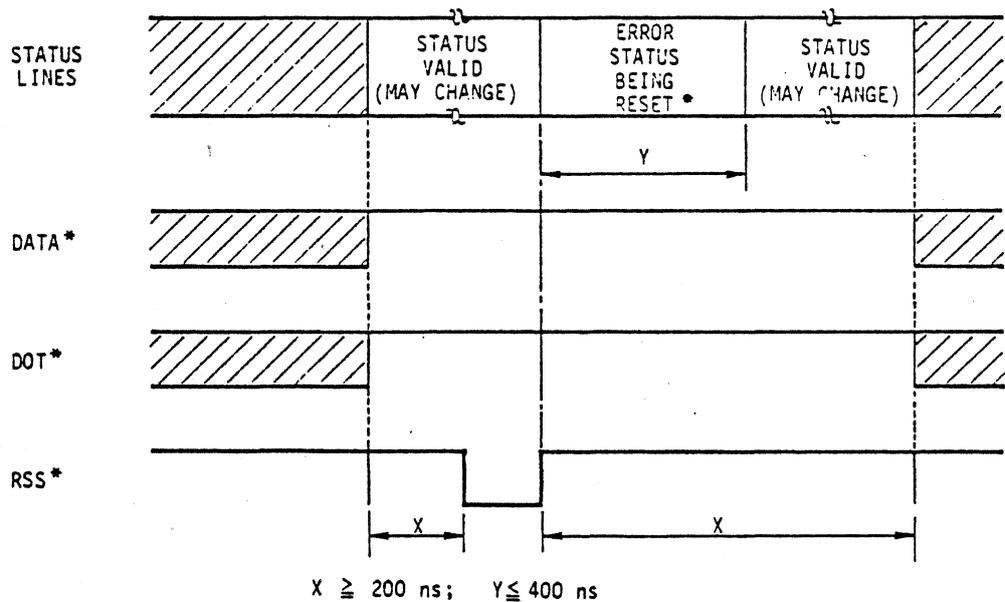
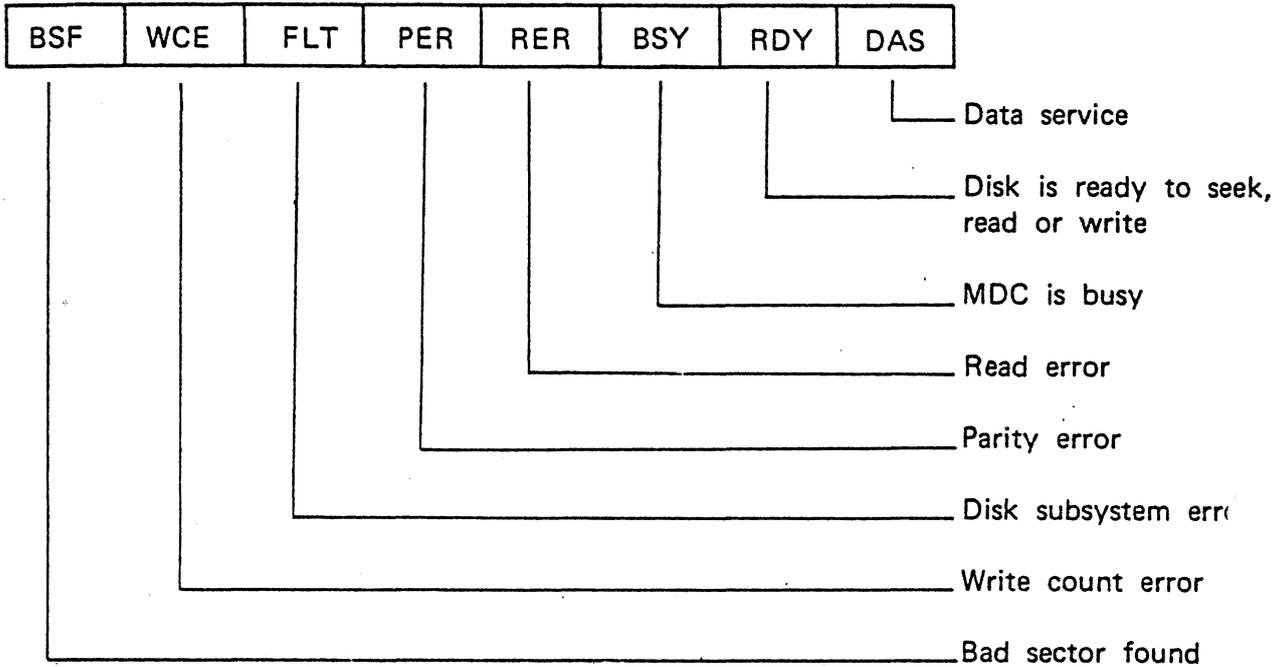


Figure Status transfer timing.

(*) BITS D2, D3, D4 and D5 of the status byte. - and bit D1, See § 8.2.1.

Figure 12. Status transfer timing

MDC CONTROLLER - PRODUCT DESCRIPTION

8.2 STATUS BITS

8.2.1 BSF - Bad Sector Found

Bad Sector status, bit D1, indicates that at least one sector which has been processed by the MDC is found defective. As a result, the operation is aborted and the flag line is asserted in the case where the MDC does not process a bad sector saving operation. Refer to table 4.

Table 4. BSF status

OPERATIONAL MODE	MDC HANDLES SAVING OPERATION	FLAG LINE ASSERTED	ABORT OPERATION
BUFFERED	YES Bad sector status may be permanently set to logic zero	NO	NO
REAL TIME INTERNAL HALT	YES Bad sector status may be permanently set to logic zero	NO	NO
REAL TIME EXTERNAL HALT	NO	YES	YES

NOTE

The BSF status is cleared by an RSS, by dispatching a reset command, or by asserting the RESET LINE.

8.2.2 WCE - WRITE COUNT ERROR

Write count status is set in buffered mode if more than $N \times 256$ bytes (N being the number of sectors to be chained) are issued to the buffer and is indicated by the assertion of D2. This status is cleared by a Read Status Strobe pulse, by sending a Reset command or by asserting the reset line. This bit is always set to logic zero in the real time operational mode.

MDC CONTROLLER - PRODUCT DESCRIPTION

8.2.3 FLT - Fault

Fault status is indicated by the assertion of D3 and is set under the following conditions :

- A failure of the disk to complete a search operation within approximately 500 milliseconds.
- A failure of the MDC to detect a sector pulse within 500 microseconds.
- A failure of the MDC to complete a read/write operation on the addressed sector within the period between two index pulses.
- A failure of the MDC to detect a proper device clock signal.
- A failure in the disk drive.

Fault status is cleared by RSS, by issuing a reset command or by asserting the RESET line.

8.2.4 PER - Parity Error

Parity error status is indicated by the assertion of D4 and is set in the case of :

- A failure to match the parity bit of the addressed sector written on the track recognition area of this sector and the parity bit of the track sector calculated in the MDC. This error may occur during a read/write operation.
- A failure of the MDC to match the parity bit of a data byte written in the buffer and the parity bit of this data byte when read out of the buffer. This error may occur during a data transfer operation from the MDC buffer to computer when in read mode and during a write operation into the disk when in write mode.
- A failure of the MDC to match the parity line and the parity of the 8 bidirectional lines during an output command strobe, a write data strobe in buffered mode or a data strobe in real time mode write operations.

Parity error status is cleared by RSS, by sending a Reset command, or by asserting the Reset line.

MDC CONTROLLER - PRODUCT DESCRIPTION

8.2.5 RER - Read Error

Read error status is indicated by the assertion of D5 and is set :

- When the data syncfield is not detected correctly
- When the MDC had read and transferred a block of data and then finds that the Cyclic Redundancy Check characters at the end of the block do not agree with what it has calculated.

Read error status is cleared by a read status strobe pulse, by sending a Reset command, or by asserting the Reset line.

8.2.6 BSY - Busy

Busy status is indicated by the assertion of D6 whenever the Midi Disk Controller is in the Read or Write mode.

8.2.7 RDY - Ready

Ready status is indicated by the assertion of D7 whenever a disk is in the drive, the door is closed, the disk is spinning at the specified rate and the heads have been positioned over a track. The Midi Disk Controller will delay a read/write operation until the Ready line is true.

8.2.8 DAS - Data Service

The Data Service request is indicated by the assertion of D8 in buffered mode and occurs whenever data should be transferred. This line is cleared by the positive-going edge of either a Read Data Strobe pulse or a Write Data Strobe pulse depending on whether the system is reading or writing, and by a Clear Data Service Command. This bit is always set to logic zero with the controller in real time mode of operation.

8.3 FLAG LINE

The flag line is true when one of the following conditions exist :

1. The Data Service status bit in buffered mode is true, indicating that data is ready to be transferred in either a read or write operation. Appropriate transfer of data (as signaled by a Write Data Strobe pulse or Read Data Strobe pulse) resets the Data Service bit.
2. An error condition exists, caused by the rise of Bad Sector Found (if any), Write Count Error, Fault, Parity Error or Read Error. The assertion of the Flag line will allow the computer to fall through its data transfer.
3. The busy line drops after a Terminate command is issued (while the system is in either the buffered read or write mode), and raises the computer/controller Flag line. The raising of this line signals to the external control unit that the MDC has completed writing the data in write mode or sensed an end of data buffer reading operation in Read mode.

MDC CONTROLLER - PRODUCT DESCRIPTION

The Flag line is reset by generating a Read Status Strobe, issuing a Re-set command or pulsing the Computer/Controller Reset line.

9 DATA TRANSFER

9.1 BUFFERED MODE

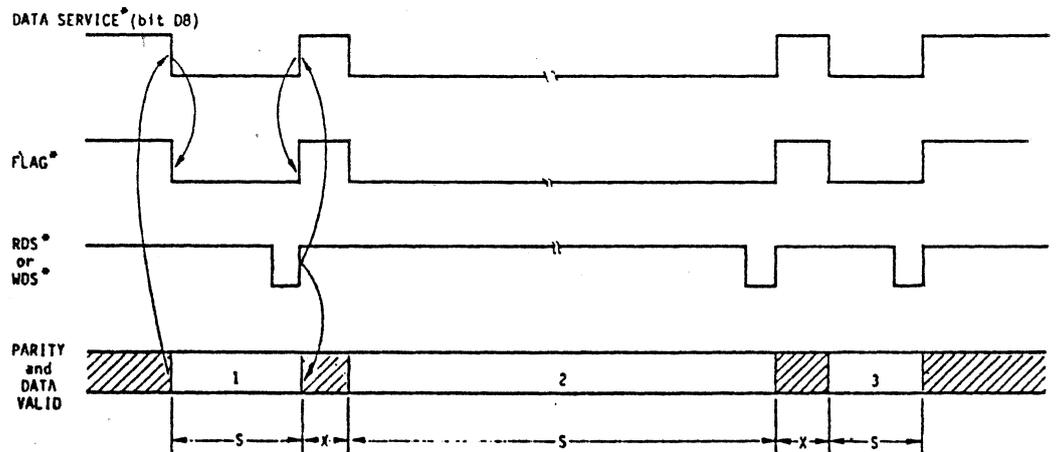
The controller is said to be buffered because it includes a data buffer with a capacity of four sectors.

When the system is in the Read mode, data is transferred from the disk to the buffer and then to the external control unit at any rate established by the external control unit ; the buffer might be read more than once by the CPU before a terminate command is issued.

When the system is in Write mode, the external control unit transfers data to the buffer at any rate it desires. When the MDC receives a terminate command (the buffer is full), it automatically transfers the data into the disk.

To end a write operation, the Terminate command is issued. When the subsystem receives this command, it immediately lowers the FLAG line and, if less than a full sector was transferred, fills out the sector with zero bytes and writes the CRC characters.

Finally, it causes the FLAG line to go high to signal the completion of the write operation.



NOTE : $X = \text{MDC processing time} : 0.9 \mu\text{s} < X < 1.9 \mu\text{s}.$

$S = \text{service time with no timing restrictions. External control may leave the flag unserviced indefinitely.}$

Figure 13 . Data transfer timing - Buffered mode

MDC CONTROLLER - PRODUCT DESCRIPTION

Figure 13 illustrates the data transfer requirements. One byte is transferred at a time, where the external control unit is not required to respond within a specific time interval after the Flag goes true. Thus the external control unit can treat the system as a truly asynchronous device. However, if the Read Data Strobe/Write Data Strobe frequency is fast enough, the buffer can be written or read with a maximum transfer rate of 916 Kbytes/second.

9.2 REAL TIME MODES

Data transfer is synchronous with the disk and each data block consists of 256 bytes. Figure 14 gives the detailed timing for each byte exchange during read or write operations.

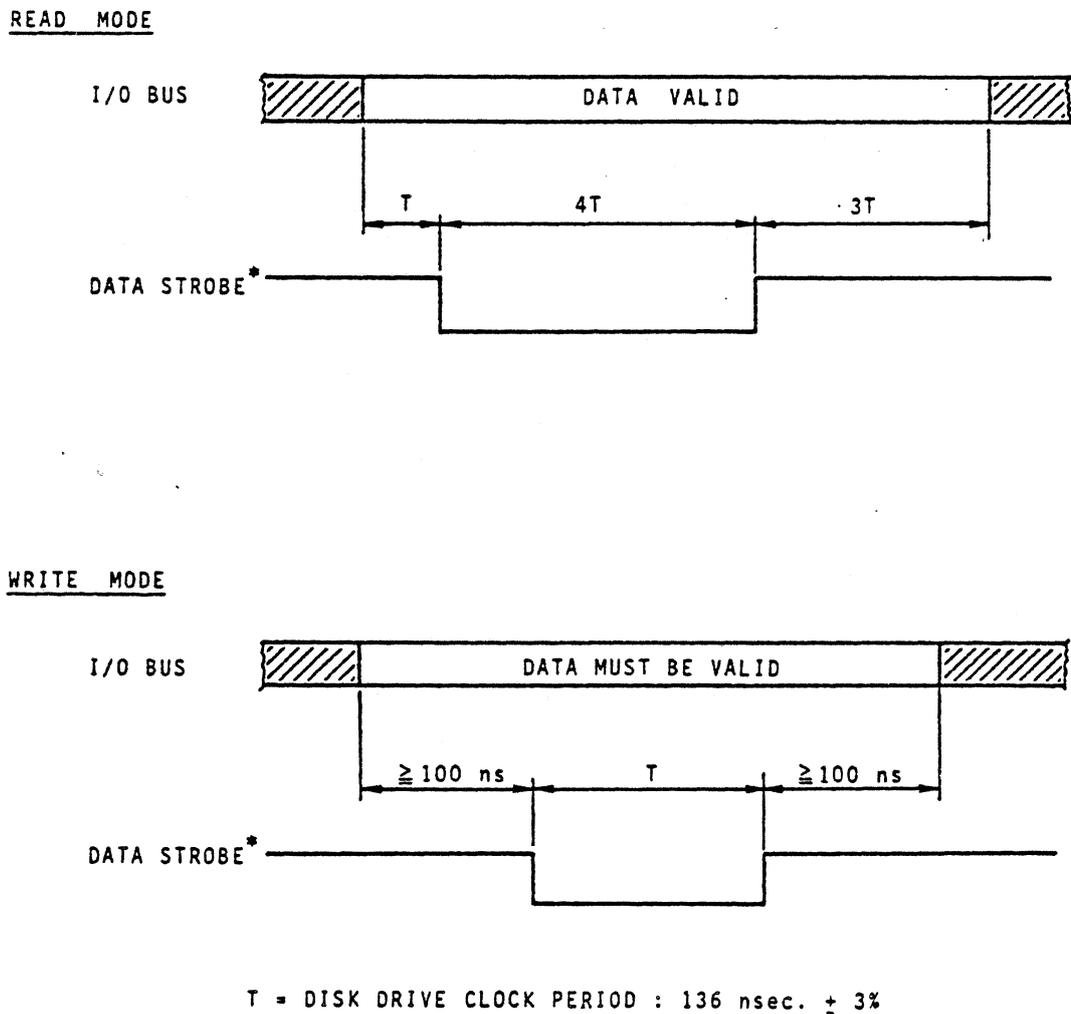


Figure 14. Data transfer Timing - Real Time Mode

MDC CONTROLLER - PRODUCT DESCRIPTION

10 SECTOR FORMAT

There are 392 tracks on each disk face with each track containing 48 sectors in which data is formatted in blocks of 256 bytes.

Data integrity is ensured by a longitudinal polynomial ($x^{16} + x^{15} + x^2 + 1$) used for cyclic redundancy checking (at the system interface level and in the buffer memory, data is checked by means of a lateral parity bit)

Figure 15 illustrates the format for each sector.

GAP 10 bytes 0's	DATA SYNC. 1 byte FD16	DATA 256 bytes -	CRC 2 bytes -	POSTAMBLE 1 byte FF16	GAP - -
------------------------	------------------------------	------------------------	---------------------	-----------------------------	---------------

Figure 15. Sector format

11 BAD SECTOR PROCESSING

A disk cartridge is not defect-free ; consequently, the data blocks must not be recorded in the defective, or bad, sectors. Bad sectors are indicated at the device interface level by an appropriate signal.

The system will be responsible for bad sector saving and management of the retrieval operation. The MDC allows several possibilities of management, outlined below.

11.1 BAD SECTOR MANAGEMENT - BUFFERED MODE

Bad sector management is processed completely by the MDC (2 spare sectors are provided at the end of each track), and the user system ignores the disk defects.

The Bad Sector Found status bit (BSF) may however be set by the MDC and used for logging purposes by the system.

11.2 BAD SECTOR MANAGEMENT - REAL TIME MODES

In these modes of operation, bad sector management may be either processed by the MDC or taken over by the user system.

- MDC Management

Each defective sector is saved in the immediate following sector. In this case, the MDC must wait for an INDEX pulse prior to beginning a read/write operation, giving a read/write latency time of 8.33 msec (half disk revolution).

The Bad Sector Found (BSF) status bit may, however be set by the MDC and used for system logging.

- System Management

In this case, when the MDC detects a defective sector it immediately aborts the current read/write operation. It then asserts the Flag line to allow the system to stop the data transfer and sets the BSF status bit. The system then initiates its own save operation. The total number of sectors available is 50 per track (with a maximum number of two being defective).

12 READ/WRITE OPERATION FLOWCHARTS

Figure 16 illustrates a typical operation sequence in buffered mode. Note, there are no timing restrictions and that it is not necessary to wait for a valid CRC error status at the end of a read transfer. Figure 17 illustrates a typical operational sequence in real time modes. If an error occurs, the FLAG line remains asserted until the corresponding error status is reset.

13 SECTOR UPDATING OPERATION FLOWCHART

To update the data content of N sectors (with $N \leq 4$) the user system issues a read command sequence (three commands), waits for the FLAG line to go high and resets it with the CLEAR DATA SERVICE command.

Next, the user system dispatches a write command sequence, waits for the FLAG line and then sends the three command sequence for buffer addressing. This selects the starting buffer word and initiates "X" number of transfers to the buffer (X words updated, with X as value, from 1 to 256). Finally, the computer sends the Terminate command, waits for the assertion of the FLAG line and checks if the operation was done correctly. A flowchart illustrating an updating operation is shown in figure 18.

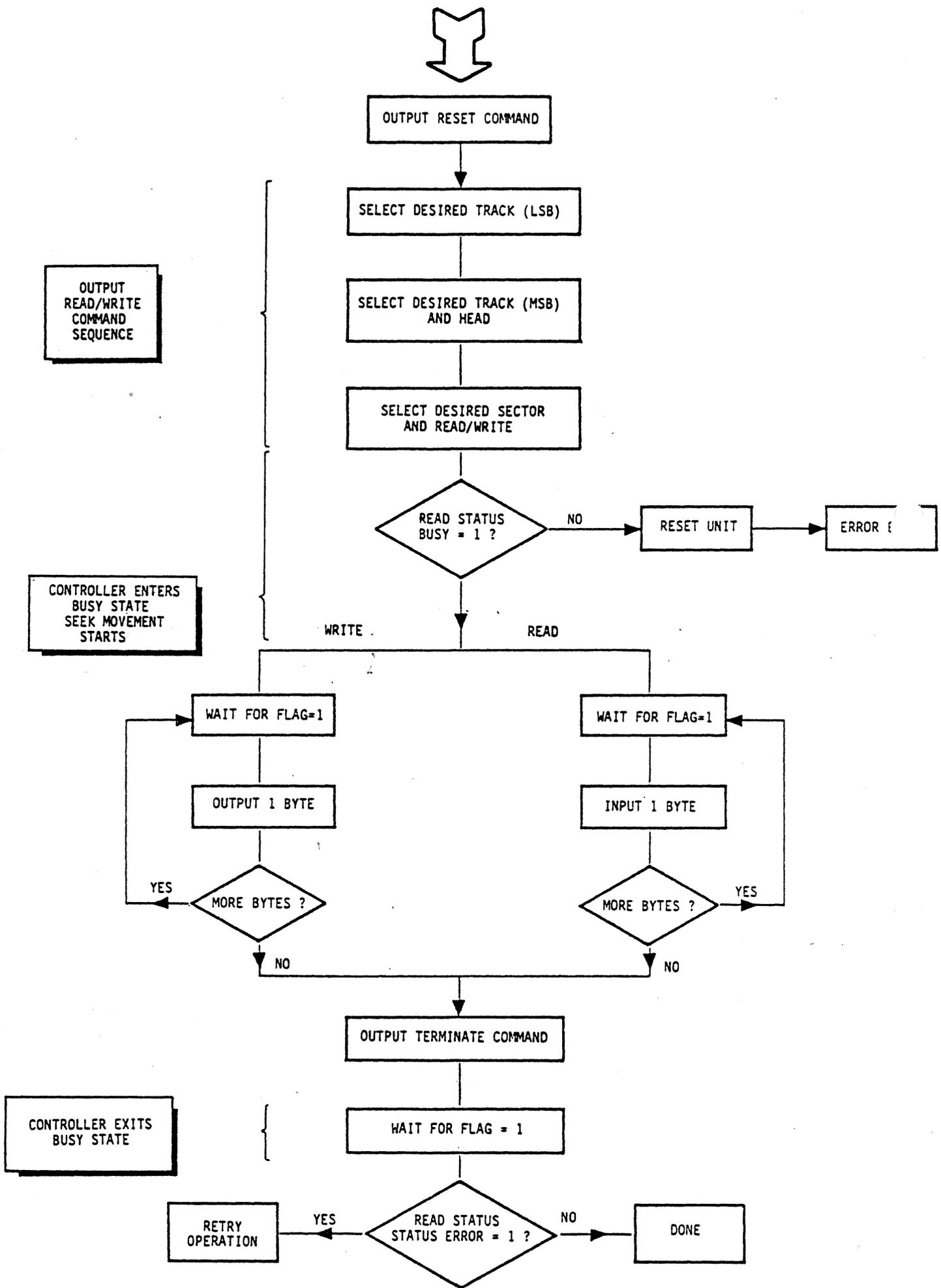


Figure 16. Read/write operation flowchart - buffered mode

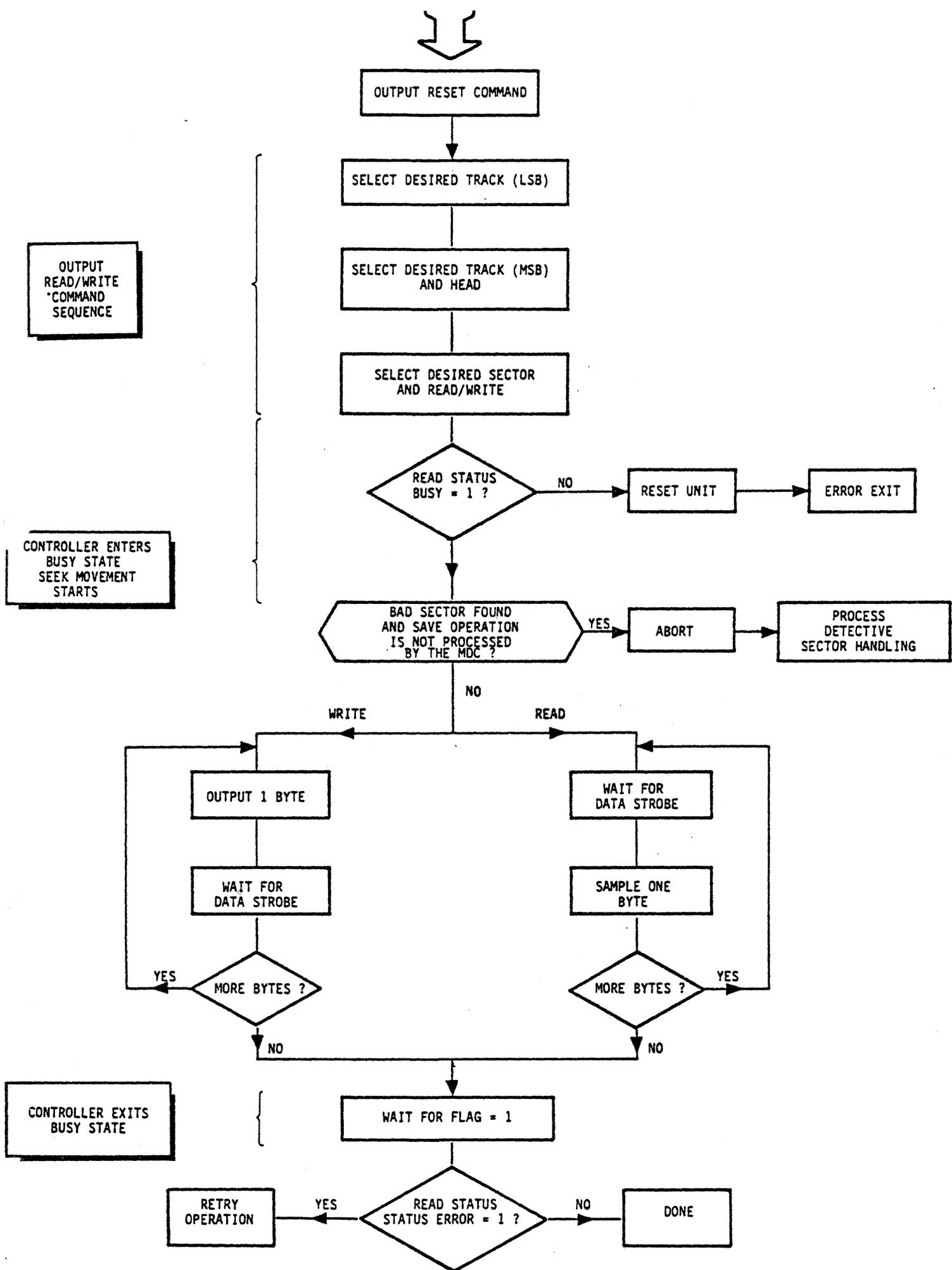


Figure 17. Read/write operation flowchart - real time mode

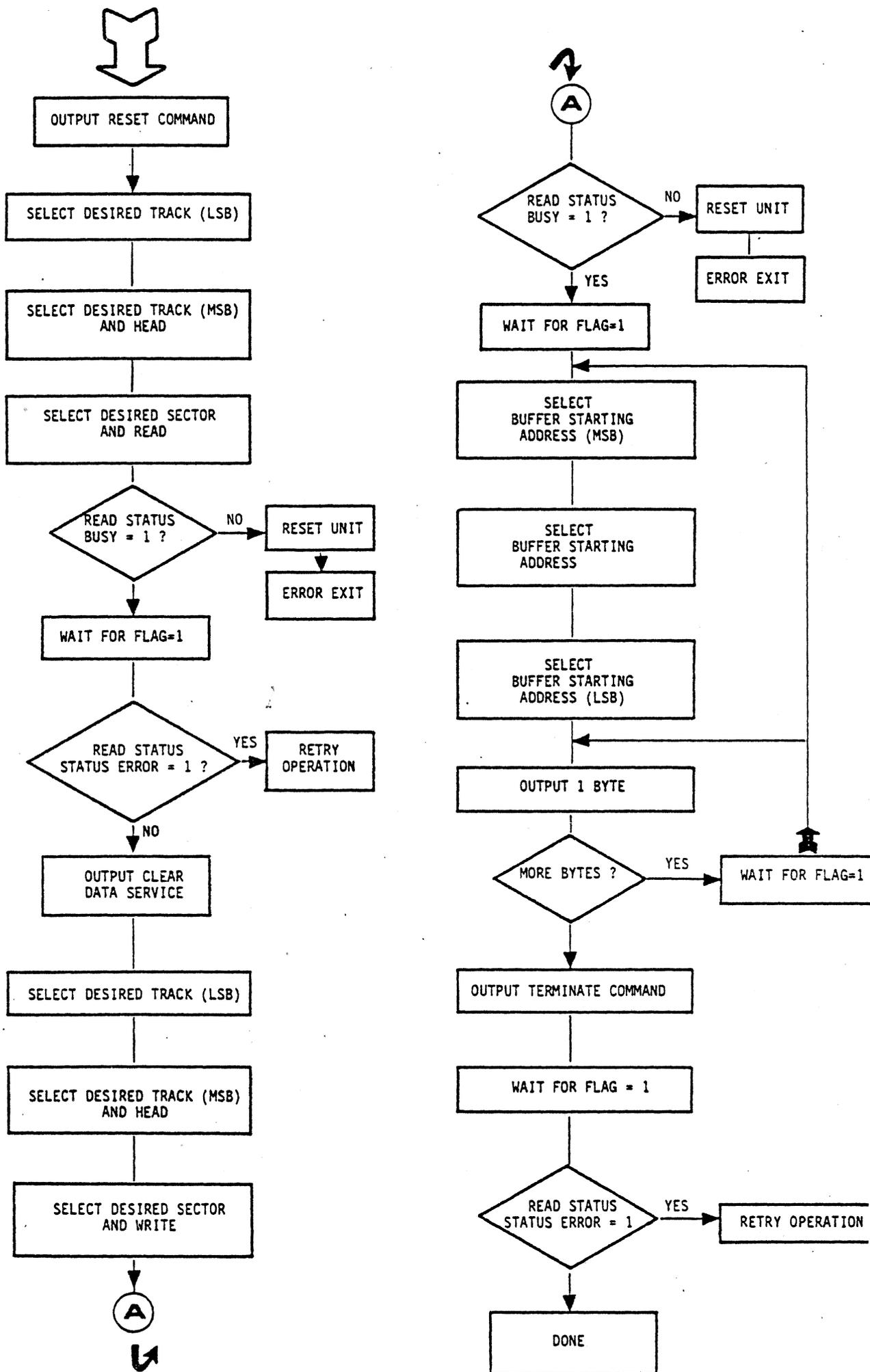


Figure 18. Sector updating flowchart - buffered mode