28

TITLE: MASSBUS Specification

HISTORY: The Massbus interface concept was suggested by Gordon Bell in July, 1972. A committee was formed of: Tom Hastings, Steve Jenkins, Vic Ku, John Levy, and Pete McLean, who defined the interface through its first several design iterations. John Levy acted as secretary and published the following precursors to this document:

August 2, 1972, Standard Mass Storage Interface--Preliminary Specification

August 28, 1972, Mass Storage Interface Standard

October 16, 1972, Massbus Interface Standard

March 20, 1973, Massbus Interface Specification--Part 1

These four documents are sometimes informally referred to as "version 1", "version 2", "version 3", and "version 4" of the Massbus interface specification.

CONFIDENTIAL

REVIEW COPY

DATE	ECO #	AUTHOR	APPROVED	REV	SEC	PAGES	TITLE
							MASSBUS Secification DEC STD 159

SIZE	CODE	NUMBER	REV
A	DS	EL00159	A

digital EN-1047A-16-R275-(327)

DRA 118

PAGE REVISION CONTROL

PAGE NO.	PAGE REVISIONS												
1.0	A												
2.0	A												
3.0	A												
4.0	A												
5.0	A												
6.0	A												
7.0	A												
8.0	A												
9.0	A												
10.0	A												
11.0	A												
12.0	A												
13.0	A												
14.0	A												
15.0	A												
16.0	A												
17.0	A												
18.0	A												
ECO NO.													
DATE													
SEC. REV													
STD REV													

PAGE NO.	PAGE REVISIONS												
19.0	A												
20.0	A												
21.0	A												
22.0	A												
23.0	A												
24.0	A												
25.0	A												
26.0	A												
27.0	A												
28.0	A												
29.0	A												
30.0	A												
31.0	A												
32.0	A												
33.0	A												
34.0	A												
35.0	A												
35.5	A												
ECO NO.													
DATE													
SEC. REV													
STD REV													

PAGE REVISION CONTROL

PAGE NO.	PAGE REVISIONS	PAGE NO.	PAGE REVISIONS
36.0	A	54.0	A
37.0	A	55.0	A
38.0	A	56.0	A
39.0	A	57.0	A
40.0	A	58.0	A
41.0	A	59.0	A
42.0	A	60.0	A
43.0	A	61.0	A
44.0	A	62.0	A
45.0	A	63.0	A
46.0	A	64.0	A
47.0	A	64.5	A
48.0	A	65.0	A
49.0	A	66.0	A
50.0	A	67.0	A
51.0	A	68.0	A
52.0	A	69.0	A
53.0	A	70.0	A
ECO NO.		ECO NO.	
DATE		DATE	
SEC. REV		SEC. REV	
STD REV		STD REV	

PAGE REVISION CONTROL

PAGE NO.	PAGE REVISIONS	PAGE NO.	PAGE REVISIONS
71.0	A	89.0	A
72.0	A	90.0	A
73.0	A	91.0	A
74.0	A	92.0	A
75.0	A	93.0	A
76.0	A	94.0	A
77.0	A	95.0	A
78.0	A		
79.0	A		
80.0	A		
81.0	A		
82.0	A		
83.0	A		
84.0	A		
85.0	A		
86.0	A		
87.0	A		
88.0	A		
ECO NO.		ECO NO.	
DATE		DATE	
SEC. REV		SEC. REV	
STD REV		STD REV	

CONTENTS

1. INTRODUCTION

1.0 GENERAL

1.0.0 This document specifies a standard interface between controllers and mass-storage devices. It is a company standard applied to disks, drums, tapes, and other magnetic or cyclic storage media.

1.1 MOTIVATION

1.1.0 The following considerations motivated the generation of a standard interface.

1.1.1 The existing number of combinations of controllers and devices is too large. Past practice was to design and build a new controller for each peripheral device. Standardization allows construction of controllers which handle more than one drive in a series (e.g., RP04, RP05,...) and more than one type of drive (e.g., RK and RS disks). Controllers with such compatibility not only have a longer market life, but also give more flexibility to the customer.

1.1.2 A great deal of "re-inventing the wheel" takes place each time a new controller is designed. A standard interface specification provides a basis for design of new controllers when they are required. A written specification aids documentation of specific implementations, and when properly maintained, provides a forum for discussion of future evolution.

1.1.3 Prior peripheral interface designs were not adequate for the data rates anticipated in the next three to five years. The need to upgrade our designs provided an opportune moment for standardization.

1.2 GOALS OF THIS SPECIFICATION

1.2.1 This interface standard has been applied immediately to the RS03, RS04, and RP04 disks and the TM02 tape subsystem, and to controllers for these drives in the PDP-10 and PDP-11 product lines. Future peripherals, including magnetic tape, drums, disks, and possibly domain and semiconductor mass memories should also conform to this standard. Extension to other product lines is also anticipated.

1.2.2 Each peripheral device is expected to implement a subset of the functions called for in this interface standard. In particular, no device or controller to which the standard has been applied should perform in ways that conflict with this standard unless the exception is documented and approved.

1.2.3 It should be possible to design inexpensive controllers and sohisticated "universal" controllers, all of which conform to this standard.

1.3 GLOSSARY

1.3.0 The language of this specification is oriented to disk and drum devices; where applicable, extension of the concepts to magnetic tape and other devices is intended. The following terms are used:

1. DRIVE — The peripheral device which attaches to the Massbus and the device's associated digital electronics.

2. CONTROLLER — The unit to which data is transmitted from the drive. This unit may or may not be distinct from the central processing unit.

3. MASSBUS — The transmission medium connecting the drive with the controller; the MASSBUS interface standard. The name "MASSBUS" is a trademark of Digital Equipment Corporation. Consequently, only the following graphic forms are to be used:
MASSBUS
Massbus

4. THE DATA BUS — The part of the Massbus which transmits high-speed data using a synchronous clock signal.

5. THE CONTROL BUS — The part of the Massbus which transmits control and status information using an asynchronous "handshake".

6. MEMORY BUS — The wires which connect a controller with the central processor and/or main memory.

7. CRC — Cyclic redundancy check, extra words written on the storage medium to aid detection of errors in writing or reading data.

8. ECC — Error-correcting code, extra words written on the storage medium to aid detection and correction of errors.

9. WORD — Unless otherwise specified, a word is either sixteen or eighteen bits of data transmitted to or from the drive when reading or writing; 16 bits of control or status information.

10. FIELD — A contiguous sequence of bits written on a magnetic storage medium, without gaps. Also, on magnetic tape, a portion of a record.

11. STORAGE MEDIUM — The magnetic surface on which data bits are recorded.

12. MEDIUM — The storage medium.

13. BLOCK — A group of contiguous characters recorded on and read from a magnetic surface as a unit. A block may contain one or more complete records.

14. SECTOR — A portion of a disk storage medium, having a unique address. A sector is composed of one or more fields.

15. RECORD — A portion of a magnetic tape medium recorded between gaps.

16. HEADER — A field containing identifying information for the sector or block in which it occurs.

17. PARITY — Whenever used, parity is meant to be odd parity: the number of 1 bits is made odd by generating a 0 or a 1 on the parity line.

18. BIT NUMBER — Bits are numbered from the least significant end, starting with bit 0.

1.4 FLOWCHART NOTATION

1.4.1 In sections 3 and 4, a special flowchart notation is introduced to show both the sequence of events in controllers and drives and the timing restrictions which apply to these events (see, for example, 3.3.3, the flowchart for a read cycle on the control bus).

1.4.2 The flowchart is divided by two vertical lines which represent the physical and electrical separation of the controller and a drive. On the left are the events in the controller, on the right, the events in the drive. A line which crosses the double line represents one or more signals which are transmitted on the bus.

1.4.3 There are four kinds of enclosures in these flowcharts: boxes, diamonds, ovals, and "butterflies".

1.4.3.1 Boxes (rectangular enclosures) represent events which happen at one instant of time. For example, in 3.3.3, the second box from the top on the controller side contains "ASSERT DEM", indicating the event of asserting the signal named "DEM" on the control bus (at the controller end of the bus).

1.4.3.2 Diamonds are decision points, as in programming flowcharts. The purpose of diamonds is to direct the flow of events along one of two paths. The decision is always one which can be made on the basis of the state of the logic at that time. No time is taken to make the decision; diamonds never represent actions or events in the logic, only a test made mentally by the reader of the flowchart.

Diamonds contain a statement with a question mark. If the statement is true at that time, the flow line marked "Y" is taken; if false, the flow line marked "N" is taken.

Some diamonds have only one flow line exiting the diamond. If the answer to the statement in the diamond does not correspond to the Y or N on the one exiting flow line the flow stops here.

1.4.3.3 Ovals are terminals, as in programming flowcharts. Ovals are either starting points, having one flow line exiting, or connectors or termination points, having one flow line entering. Ovals which do not contain "END" are normally connectors to an additional flowchart.

1.4.3.4 "Butterflies" are small boxes with two compartments and an angled side which attaches to a flowline (it is drawn using a bisected "offpage connector" outline from a programming template). Each compartment contains a number or a "U". These represent timing restrictions. The bottom (or right-hand) number is a minimum, the top (or left-hand) one a maximum. Times are in nanoseconds. "U" means that the time restriction is unspecified. Empty compartments should not occur.

1.4.4 FLOW LINES

1.4.4.1 Flow lines direct the sequence and timing of events. Every flow line has a minimum and maximum time associated with it.

1.4.4.2 Flow lines which split into two lines are equivalent to two lines connecting the source box with the two following boxes. Both branches are to be actively followed simultaneously.

1.4.4.3 When two flow lines merge, the box entered by the merged line is activated by flow from either of the source flow lines. Conflicts due to multiple activation of events should not occur.

1.4.5 MINIMUM AND MAXIMUM TIMES

1.4.5.1 A flow line which has a "butterfly" touching it represents an elapsed time between the limits shown in the two compartments. This time may be under the control of the designer. The times shown are the permissible limits allowed by this specification.

1.4.5.2 A flow line which crosses between the controller and a drive represents an elapsed time of 0 nanoseconds minimum, 375 nanoseconds maximum. The actual time is not under control of the designer, and worst-case should always be assumed.

1.4.5.3 All other flow lines represent an elapsed time of 0 nanoseconds (0 nanoseconds minimum, 0 nanoseconds maximum).

2. SUMMARY

2.1 DIVISION OF FUNCTIONS BETWEEN CONTROLLER AND DRIVE

2.1.1 The controller performs the following functions:

2.1.1.1 Interfaces with the memory bus cables and signals.

2.1.1.2 Communicates with main memory in order to fetch and store data.

2.1.1.3 Buffers data in order to accommodate timing differences between the drive and the memory.

2.1.1.4 Communicates with the central processor in order to receive commands and send error and status information.

2.1.1.5 Implements commands which may require a sequence of functions in the drive (which might otherwise require a programmed sequence).

2.1.1.6 Interfaces with multiple drives.

2.1.2 The drive performs the following functions:

2.1.2.1 Records and plays back data.

2.1.2.2 Generates gaps and synchronization marks on the recording medium (and in general performs all functions which are highly medium-dependent).

2.1.2.3 Provides clock signals to synchronize data transmission between drive and controller.

2.1.2.4 Maintains error and status indicators and generates an attention signal when exceptional conditions occur.

2.1.2.5 Locates data by address (except magnetic tape).

2.1.2.6 Provides mechanisms for maintenance and diagnostic testing

2.1.2.7 Does error detection on the data and provides error correction patterns and positions.

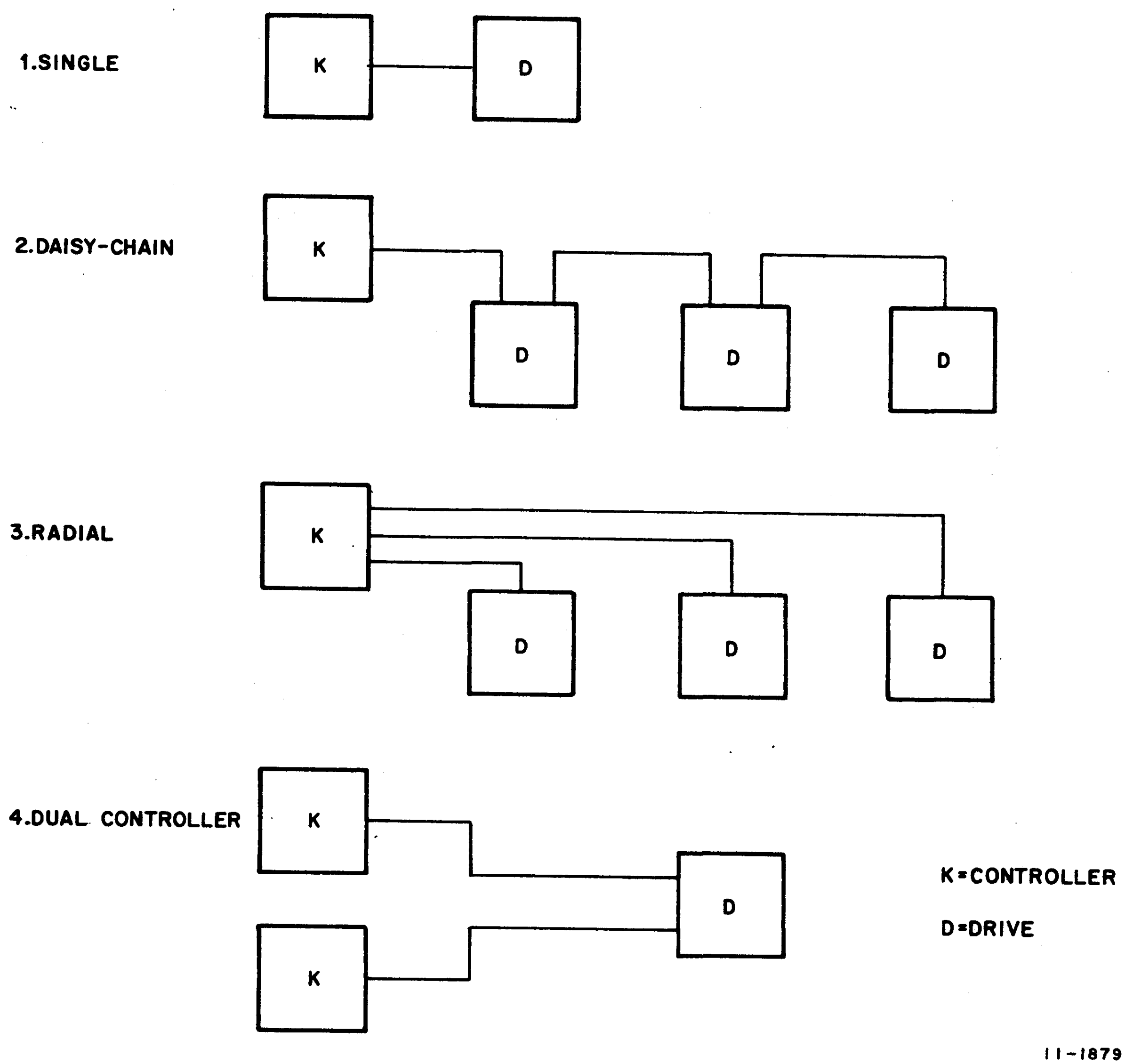
2.1.2.8 Does verification of header information.

2.2 CONFIGURATIONS

2.2.1 The four configurations shown in figure 2.2.1 are all expected to occur.

2.2.1.1	SINGLE	One controller and one drive; this could be a special case of daisy-chain or radial, below.
2.2.1.2	DAISY-CHAIN	One bus is threaded among the several drives.
2.2.1.3	RADIAL	A separate bus connects each drive with the controller.
2.2.1.4	DUAL CONTROLLER	Two controllers with separate paths to the drive share their control of the drive.

FIGURE 2.2.2


1.SINGLE
K
D
2.DAISY-CHAIN
K
D
D
D
3.RADIAL
K
D
D
D
4.DUAL CONTROLLER
K
K
D
K=CONTROLLER
D=DRIVE
11-1879

2.3 PHYSICAL CONSTRAINTS

2.3.1 The following goals are expected to be achieved by designs conforming to this specification.

2.3.1.1 Data rates of up to 36 MHz (bits).

2.3.1.2 Up to eight drives addressable per controller.

2.3.1.3 Cable (Massbus) lengths of up to 160 feet (allowing 10 feet per drive in the daisy-chain configuration).

2.4 GENERAL ABSTRACT

2.4.0 The following are the main features of the Massbus.

2.4.1 The Massbus is composed of two sections, containing a total of 55 signals. The data bus section, for high-speed data transmission, consists of a 19 bit parallel data path and six control lines. The control bus section, for control and status information transfer, contains a 17 bit path and 14 control lines.

2.4.2 Each drive contains up to thirty-two addressable registers (some drives may implement only a subset of these). Some of these registers, when written into, control the normal and maintenance operations of the drive. All registers may be read from, to obtain status information.

2.4.3 A sector format (for disk and drum) is specified containing header and data records. The length of the header record is fixed and specified (for those drives which implement it). The length of the data record is not specified, and considerable effort has been made to avoid implicit restrictions on its length.

3. THE CONTROL BUS

3.1 INTRODUCTION

3.1.1 The purpose of the control bus is to transmit control commands and information from controller to drive; to transmit status information from drive to controller; to notify the controller when an unusual (attention) condition exists in one or more drives; and to provide a master reset (all drives) signal from the controller.

3.2 CONTROL BUS SIGNALS

3.2.1 Control (C <0:15>); Bidirectional

3.2.1.1 These 16 bidirectional lines carry the control and status information.

3.2.2 Control Parity (CPA); Bidirectional

3.2.2.1 This bidirectional line carries a parity bit associated with the control lines (odd parity).

3.2.3 Drive Select (DS <0:2>); Controller To Drive

3.2.3.1 These three lines select the drive to be accessed.

3.2.3.2 When the Register Select lines RS<0:4> = 04(base 8), the Drive Select lines are ignored and all drives respond. This is for transmission of the Attention Summary psuedo register bits.

3.2.4 Register Select (RS <0:4>); Controller To Drive

3.2.4.1 These five lines select a register in the selected drive.

3.2.4.2 When the selected register RS <0:4> = 04(base 8), each drive gates out only one bit: its Attention Active (ATA) bit is driven onto one of the control lines in the bit position corresponding to the drive unit number.

3.2.5 Controller To Drive (CTOD); Controller To Drive

3.2.5.1 This line selects the direction of transfer. It is asserted when the transfer is from controller to drive.

3.2.6 Demand (DEM); Controller To Drive

3.2.6.1 This line is asserted by the controller to initiate a transfer "handshake".

3.2.7 Transfer (TRA); Drive To Controller

3.2.7.1 This line is asserted by the selected drive to complete a transfer "handshake".

3.2.7.2 When the selected register RS <0:4> = 04(base 8), this line is asserted by each drive, but the controller ignores the assertion. Instead, the controller completes the transfer after it "times out". This is necessary because each bit on the control lines is coming from a different drive.

3.2.8 Attention (ATTN); Drive To Controller

3.2.8.1 This line is asserted by a drive when it has an "Attention Active" condition. One or more drives may assert it at the same time.

3.2.9 Initialize (INIT); Controller To Drive

3.2.9.1 When this line is asserted by the controller, all drives perform a reset function.

3.2.9.2 The purpose of the Initialize line is to give an easy way to clear all drives at once; to allow initialization of the drives at system startup time; and to allow clearing of transient faults.

3.2.9.3 INIT should never be asserted while RUN is asserted.

3.2.10 Fail (FAIL); Controller to Drive

3.2.10.1 This line is asserted by the controller when power fails in the controller. It is negated when power in the controller is ok. The drive should ignore assertions of DEM and INIT while FAIL is asserted. See Section 11.5.1 for further details.

3.3 SEQUENCE AND TIMING OF CONTROL BUS TRANSFERS

3.3.1 Introduction

3.3.1.1 The flowcharts and timing diagrams below describe the normal operation of the control bus. In this section, "read" means a transfer from a drive register to the controller, and "write" means a transfer the other way.

3.3.1.2 Not all registers are implemented in all drives. When the register selected by the RS lines is unimplemented, the drive will still respond with the normal sequence shown in 3.3.4 and 3.3.8 below. After the sequence is complete, the drive will set the Illegal Register (ILR) error bit (see section 7). Zeros are transmitted on the C lines on a control bus read; on a write, the bits on the C lines are checked for parity, but are otherwise ignored.

3.3.1.3 Some registers are designated as read only. Performing a control bus write to one of these registers will not cause an error; the bits on the C lines are checked for parity, but are otherwise ignored.

3.3.1.4 While a drive is busy (DRY bit is reset), most registers cannot be modified by writing. The exceptions to this are the Attention Summary register, and possibly the Maintenance register (depending on the drive designer's specification). Performing a control bus write to an unmodifiable register while the drive is busy will cause the Register Modification Refused (RMR) error, after the normal sequence has been completed. The bits on the C lines are checked for parity, but are otherwise ignored.

3.3.1.5 The maximum cable delay time between controller and drive is 375 nanoseconds , assuming a 55 nanosecond driver and receiver delay and a 2 nanosecond per foot propagation delay over a 160 foot length. The minimum cable delay is 0. In the flowcharts below, wherever a flowline crosses between controller and drive, imagine a "butterfly" exists specifying max 375, min 0, over which the designer has no control.

3.3.1.6 The normal "handshake" sequences for control bus reads and writes are described in 3.3.2 and 3.3.6. Several unusual conditions can occur which will modify this sequence. One is addressing the Attention Summary register (see 3.3.5 and 3.3.9). Another is when the selected drive does not exist (see 3.8.1); this is the "Nonexistent Drive" error condition.

3.3.2 Control Bus Read Sequence

3.3.2.1 Refer to 3.3.4 for a timing diagram of this sequence. A flowchart of the sequence is shown in 3.3.3. This flowchart includes timing restrictions in a notation described in 1.4.

3.3.2.2 The normal read sequence is as follows.

1. The controller asserts the appropriate DS and RS lines and negates the CTOD line.

2. After waiting the deskew and set up time (min 225, max 325) and waiting, if necessary, for the TRA line to be negated, the controller asserts DEM.

3. After a cable delay, the selected drive receives the DEM assertion. 75 nanoseconds of set up time has been allowed so that the drive may use the DEM assertion edge as a strobe on the output of its DS lines comparator and its RS lines decoder. The controller should hold the DS and RS lines constant until the assertion of TRA is received.

4. Not more than 500 nanoseconds later (700 for dual controller drives), the drive has gated the contents of the selected register onto the C lines and has generated CPA. It asserts TRA (each C line and CPA must be held steady until step 7).

5. After a cable delay, the controller receives the TRA assertion. The controller may disable or change the DS, RS, and CTOD lines now (if the next control bus cycle is a read, the deskew of DS, RS, and CTOD for the next cycle may begin at this time).

6. After waiting for deskew time (min 150, max 250), the controller strobes or gates in the C lines and CPA. Parity checking may begin at this time.

   The controller may negate DEM at this time, or it may delay the negation of DEM in order to gate the C lines through without buffering.

7. After a cable delay, the selected drive receives the DEM negation. It disables the C lines and CPA and negates TRA.

8. After a cable delay, the controller receives the TRA negation. This completes the control bus read cycle (if the next control bus cycle is a write, the controller may begin deskew of the C lines at this time (or later). This cannot be begun sooner because the selected drive has been asserting the C lines until step 7. If the next cycle is a read, the controller may assert DEM now or when the RS, DS, and CTOD deskew time is finished, whichever occurs later).

FIGURE 3.3.3

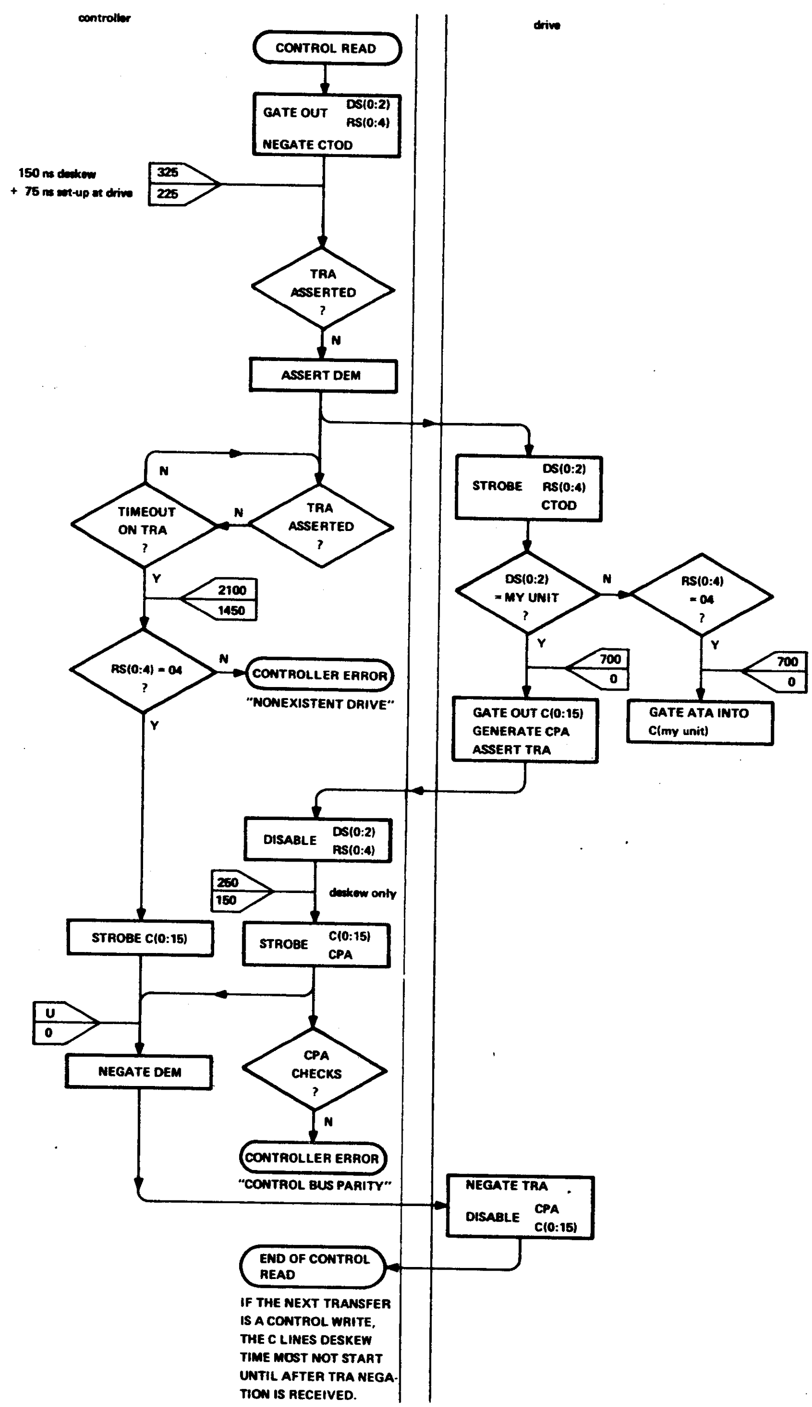
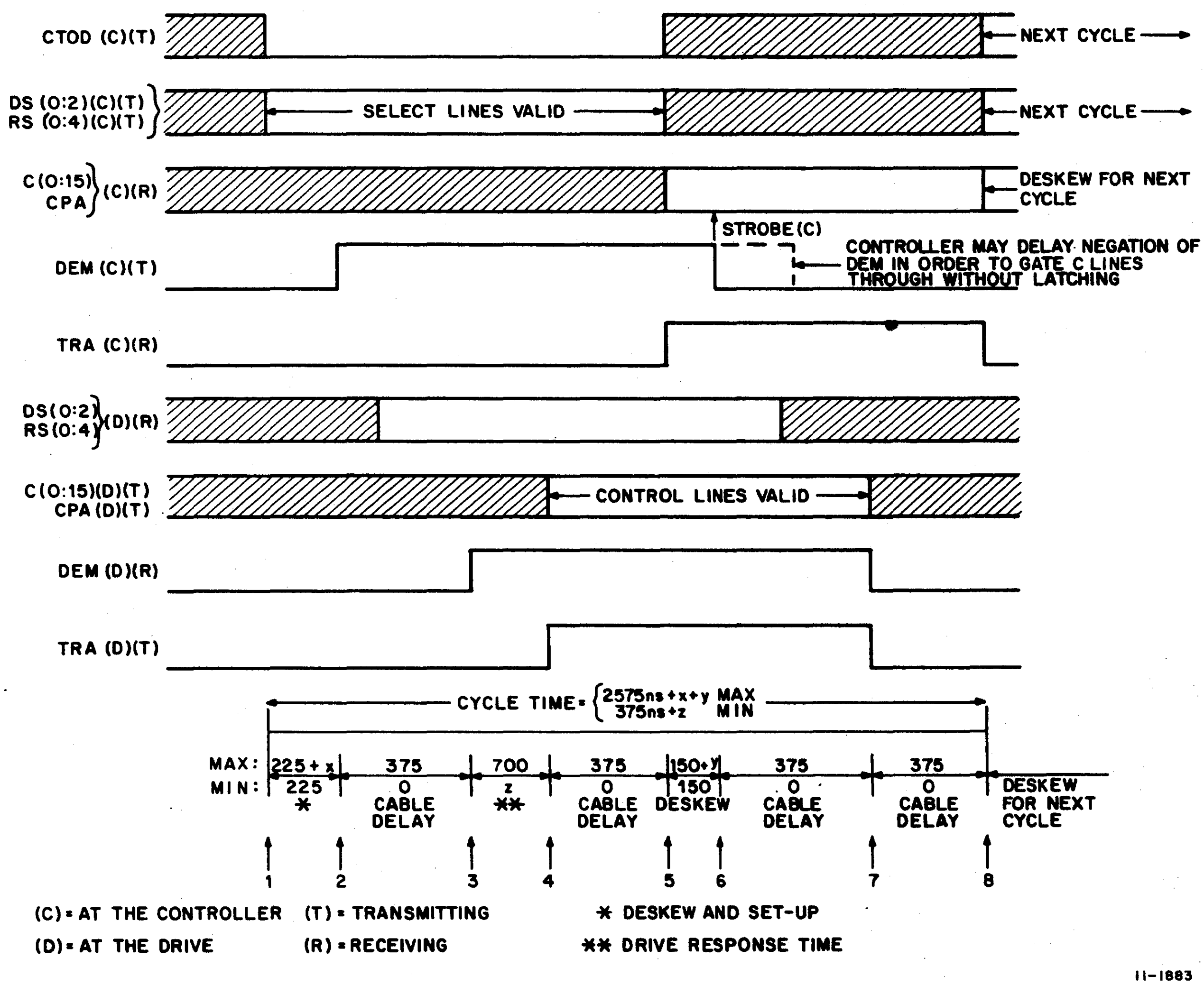

controller
drive
CONTROL READ
GATE OUT DS(0:2) RS(0:4)
NEGATE CTOD
150 ns deskew
+ 75 ns set-up at drive
325
225
TRA ASSERTED ?
N
ASSERT DEM
STROBE DS(0:2) RS(0:4) CTOD
N
TIMEOUT ON TRA ?
N
TRA ASSERTED ?
Y
2100
1450
DS(0:2) = MY UNIT ?
N
RS(0:4) = 04 ?
Y
700
0
Y
700
0
RS(0:4) = 04 ?
N
CONTROLLER ERROR
"NONEXISTENT DRIVE"
Y
GATE OUT C(0:15)
GENERATE CPA
ASSERT TRA
GATE ATA INTO C(my unit)
DISABLE DS(0:2) RS(0:4)
250
150
deskew only
STROBE C(0:15)
STROBE C(0:15) CPA
U
0
NEGATE DEM
CPA CHECKS ?
N
CONTROLLER ERROR
"CONTROL BUS PARITY"
NEGATE TRA
DISABLE CPA C(0:15)
END OF CONTROL READ
IF THE NEXT TRANSFER IS A CONTROL WRITE, THE C LINES DESKEW TIME MUST NOT START UNTIL AFTER TRA NEGATION IS RECEIVED.
11-1875

FIGURE 3.3.4


CTOD (C)(T)
NEXT CYCLE
DS (0:2)(C)(T)
RS (0:4)(C)(T)
SELECT LINES VALID
NEXT CYCLE
C(0:15)
CPA (C)(R)
DESKEW FOR NEXT CYCLE
STROBE(C)
DEM (C)(T)
CONTROLLER MAY DELAY NEGATION OF DEM IN ORDER TO GATE C LINES THROUGH WITHOUT LATCHING
TRA (C)(R)
DS(0:2)
RS(0:4) (D)(R)
C(0:15)(D)(T)
CPA (D)(T)
CONTROL LINES VALID
DEM (D)(R)
TRA (D)(T)
CYCLE TIME = 2575ns+x+y MAX, 375ns+z MIN
MAX: 225+x 375 700 375 150+y 375 375
MIN: 225 0 z 0 150 0 0
* CABLE DELAY ** CABLE DELAY DESKEW CABLE DELAY CABLE DELAY DESKEW FOR NEXT CYCLE
1 2 3 4 5 6 7 8
(C) = AT THE CONTROLLER (T) = TRANSMITTING * DESKEW AND SET-UP
(D) = AT THE DRIVE (R) = RECEIVING ** DRIVE RESPONSE TIME
11-1883

3.3.5 Reading The Attention Summary Register

3.3.5.1 The Attention Summary register is not a normal register. It is composed of one bit in each of up to eight drives. When this register (04) is read by the controller, each drive gates its ATA bit out onto one of the C lines (drive 00 onto C00, etc.).

Because all drives are responding at once, the normal handshake sequence is not valid. Instead of using the TRA assertion as a "ready to read" signal, the controller waits for the maximum delay time and then unconditionally strobes the C lines. The controller negates DEM after it has strobed the C lines. No drive may disable its C line output until it has negated TRA (typically after receiving the negation of DEM). When the TRA line becomes negated at the controller (after each drive has negated TRA), the controller knows that the cycle is complete.

3.3.5.2 Since no single drive is generating all of the C bits, it is impossible for any drive to generate a valid parity bit. Therefore, the controller must ignore the CPA line and must not check parity when reading the Attention Summary register.

3.3.5.3 The sequence for reading the Attention Summary register is as follows (refer to 3.3.5.4 for a timing diagram of this sequence).

1. The controller asserts code 04(base 8) on the RS lines and negates the CTOD line (the state of the DS lines does not matter).

2. After waiting the deskew and set up time (min 225, max 325 nanoseconds) and waiting, if necessary, for the TRA line to be negated, the controller asserts DEM.

3. After cable delay, each drive receives the DEM assertion. 75 nanoseconds of set up time has been allowed so that the drive may use the DEM assertion edge as a strobe on the output of its RS lines decoder. The controller should hold the RS lines constant until the assertion of TRA is received.

4. Not more than 250 nanoseconds after receiving the DEM assertion, each drive gates out its ATA bit onto a C line, and asserts TRA (the C line must be held steady until step 6).

5. After cable delay, the controller receives the TRA assertion from the nearest (or fastest) drive. However, since the response of other drives must also be waited

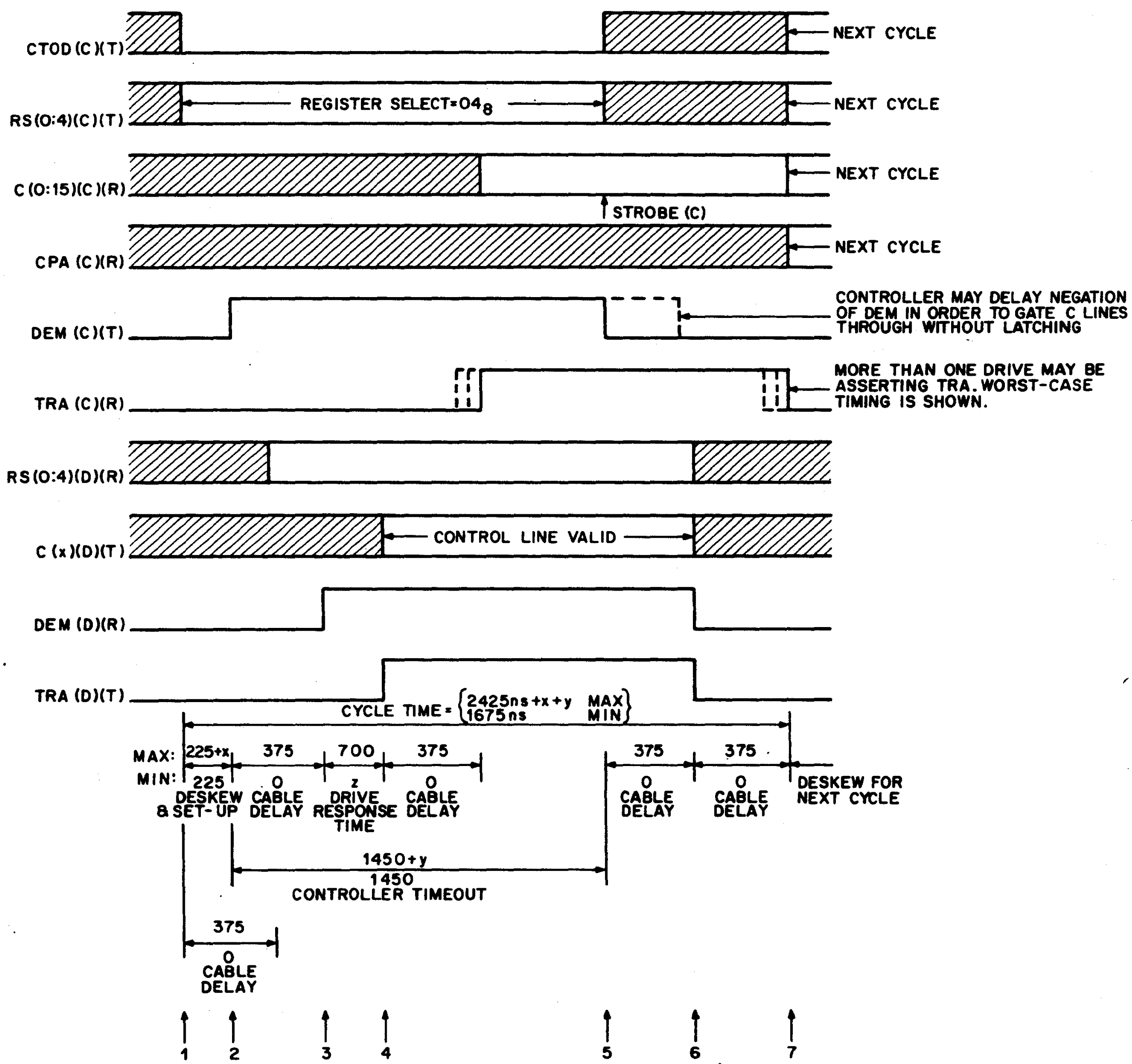
for, the controller does not use the TRA assertion to start deskew. Instead, it waits at least 1450 nanoseconds from the assertion of DEM (step 2). At this time, the controller strobes the C lines and negates DEM (but does not check parity). The controller may disable or change the DS, RS, and CTOD lines now (if the next cycle is a read, the deskew of DS, RS, and CTOD for the next cycle may begin at this time).

6. After cable delay, each drive receives the DEM negation. It disables the C line it was driving and negates TRA.

7. After cable delay, the controller finally receives the negation of TRA after all drives have negated it. This completes the Attention Summary register read cycle.

   If the next control bus cycle is a write, the controller may begin deskew of the C lines at this time (or later). This cannot be begun sooner because some drive may have been asserting a C line until step 6. If the next cycle is a read, the controller may assert DEM now or when the RS, DS, and CTOD deskew time is finished, whichever occurs later.

FIGURE 3.3.5.4



(C) = AT THE CONTROLLER (T) = TRANSMITTING

(D) = AT THE DRIVE (R) = RECEIVING

11-1884

3.3.6 Control Bus Write Sequence

3.3.6.1 Refer to 3.3.8 for a timing diagram of this sequence. A flowchart of the sequence is shown in 3.3.7. This flowchart includes timing restrictions in a notation described in 1.4.

3.3.6.2 The normal write sequence is as follows.

1. The controller asserts the appropriate DS and RS lines, asserts the CTOD line, and gates out a word on the C lines (the C lines and CPA must be held steady until step 5).

2. After waiting the deskew and set up time (min 225, max 325 nanoseconds) and waiting, if necessary, for the TRA line to be negated, the controller asserts DEM.

3. After a cable delay, the selected drive receives the DEM assertion. 75 nanoseconds set up time has been allowed so that the drive may use the DEM assertion edge as a strobe on the output of its DS lines comparator and its RS lines decoder. The controller should hold the DS and RS lines constant until the assertion of TRA is received.

4. Not more than 250 nanoseconds later, the drive has strobed or gated in the C lines and CPA. It asserts TRA. Parity checking may begin at this time.

5. After a cable delay, the controller receives the TRA assertion. It negates DEM. The controller may disable or change the DS, RS, CTOD, and C lines now, and begin the deskew for the next cycle.

6. After a cable delay, the selected drive receives the DEM negation. It negates TRA.

7. After a cable delay, the controller receives the TRA negation. This completes the control bus write cycle.

   For the next cycle, the controller may assert DEM now or when the RS, DS, and CTOD deskew time is finished, whichever occurs later.

FIGURE 3.3.7

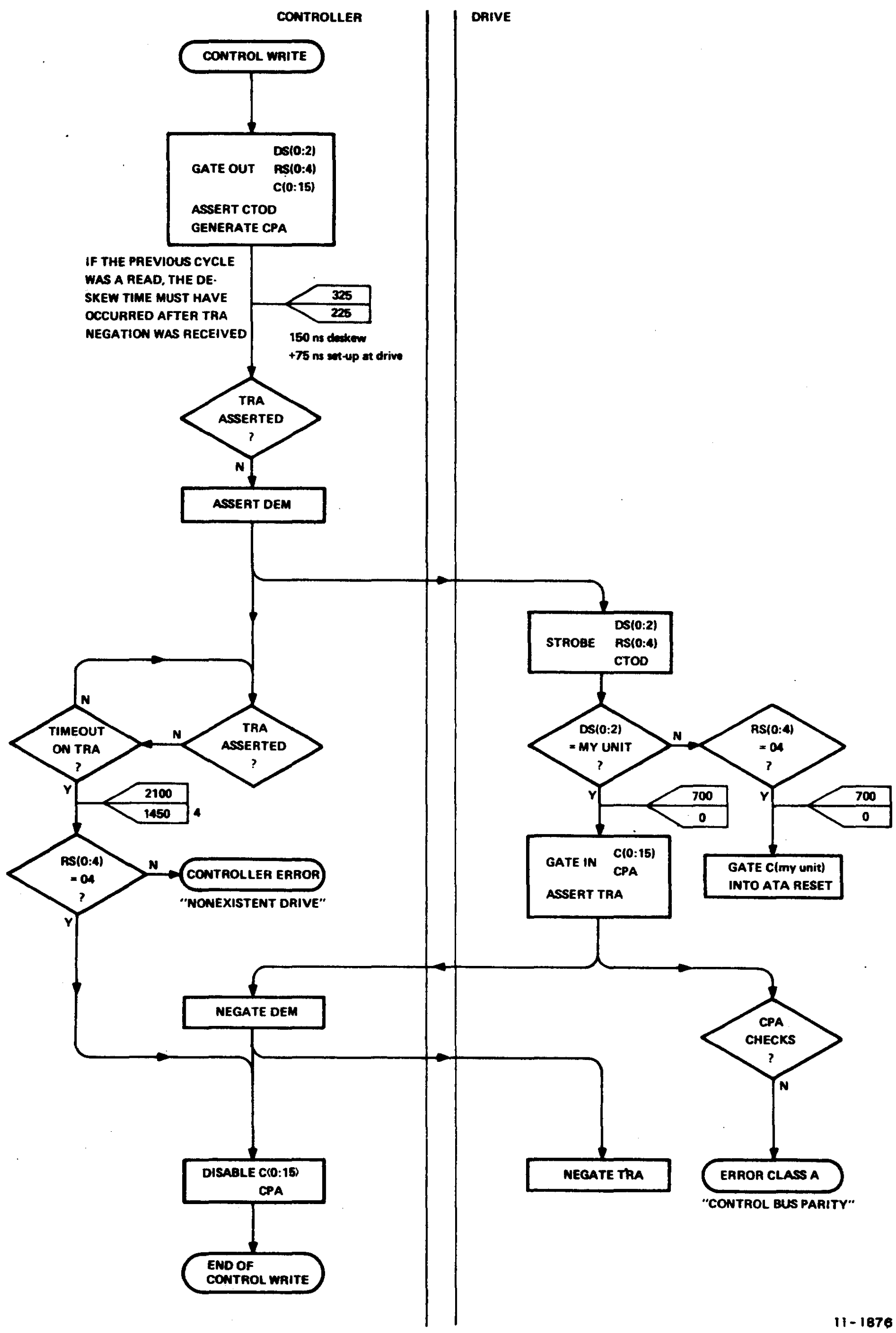
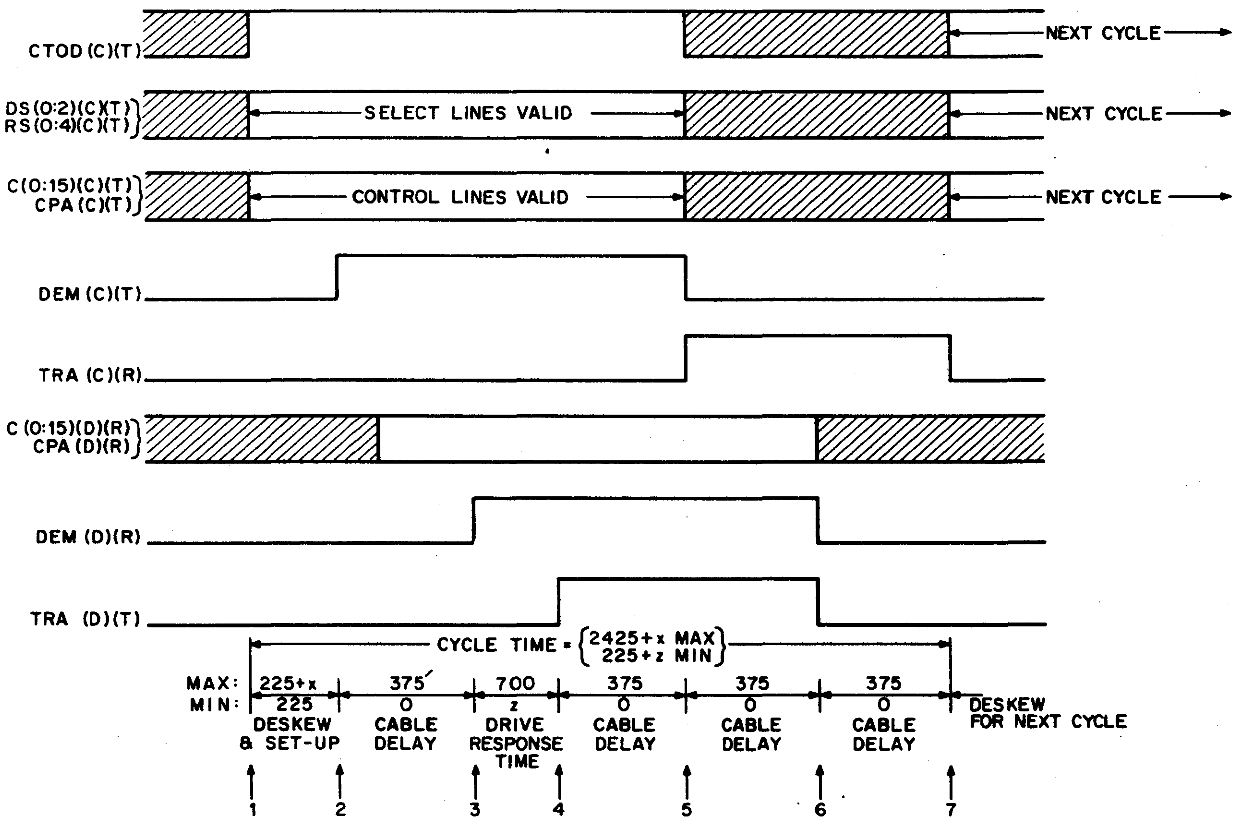

CONTROLLER
DRIVE
CONTROL WRITE
GATE OUT DS(0:2) RS(0:4) C(0:15)
ASSERT CTOD
GENERATE CPA
IF THE PREVIOUS CYCLE WAS A READ, THE DE-SKEW TIME MUST HAVE OCCURRED AFTER TRA NEGATION WAS RECEIVED
325
225
150 ns deskew
+75 ns set-up at drive
TRA ASSERTED ?
N
ASSERT DEM
STROBE DS(0:2) RS(0:4) CTOD
TIMEOUT ON TRA ?
N
TRA ASSERTED ?
N
Y
2100
1450
4
RS(0:4) = 04 ?
N
CONTROLLER ERROR
"NONEXISTENT DRIVE"
Y
DS(0:2) = MY UNIT ?
N
RS(0:4) = 04 ?
Y
700
0
Y
700
0
GATE IN C(0:15) CPA
ASSERT TRA
GATE C(my unit) INTO ATA RESET
NEGATE DEM
CPA CHECKS ?
N
DISABLE C(0:15) CPA
NEGATE TRA
ERROR CLASS A
"CONTROL BUS PARITY"
END OF CONTROL WRITE
11-1876

FIGURE 3.3.8



(C) = AT THE CONTROLLER (T) = TRANSMITTING

(D) = AT THE DRIVE (R) = RECEIVING

11-1885

3.3.9 Writing The Attention Summary Register

3.3.9.1 Writing into the Attention Summary register has the special purpose of resetting selected ATA bits in the drives. This register (04) is composed of one bit in each drive; when it is written into, each drive gates in one of the C lines (C00 into drive 00, etc.). If the line is asserted, that drive will reset its ATA bit (if the line is negated, the drive does not change the state of ATA).

Because all drives are responding at once, the normal handshake sequence is not valid. Instead of using the TRA assertion as a "data received" signal, the controller holds the C lines valid for the maximum delay time (at least 1450 nanoseconds) and assumes that they are received by that time.

3.3.9.2 Since all C bits are coming from the controller in this case, the controller generates a valid parity bit on the CPA line. The drives will check the parity as in a normal write sequence.

3.3.9.3 The sequence for writing the Attention Summary register is as follows (refer to 3.3.9.4 for a timing diagram of this sequence).

1. The controller asserts code 04(base 8) on the RS lines, asserts the CTOD line, and gates out the C lines and CPA (the state of the DS lines does not matter). The C lines and CPA must be held steady until step 5.

2. After waiting the deskew and set up time (min 225, max 325 nanoseconds) and waiting, if necessary, for the TRA line to be negated, the controller asserts DEM.

3. After cable delay, each drive receives the DEM assertion. 75 nanoseconds of set up time has been allowed so that the drive may use the DEM assertion edge as a strobe on the output of its RS lines decoder The controller should hold the RS lines constant until it has strobed the C lines and negated DEM.

4. Not more than 250 nanoseconds after receiving the DEM assertion, each drive gates in the C lines and CPA, and asserts TRA. The assertion of the appropriate one of the C lines is used to reset ATA in each drive. Parity checking may be done at this time.

5. After cable delay, the controller receives the TRA assertion from the nearest (or fastest) drive. However, since other drives must also have time to respond, the controller does not use the TRA assertion to terminate the

cycle. Instead, it waits at least 1450 nanoseconds from the assertion of DEM (step 2). At this time the controller negates DEM.

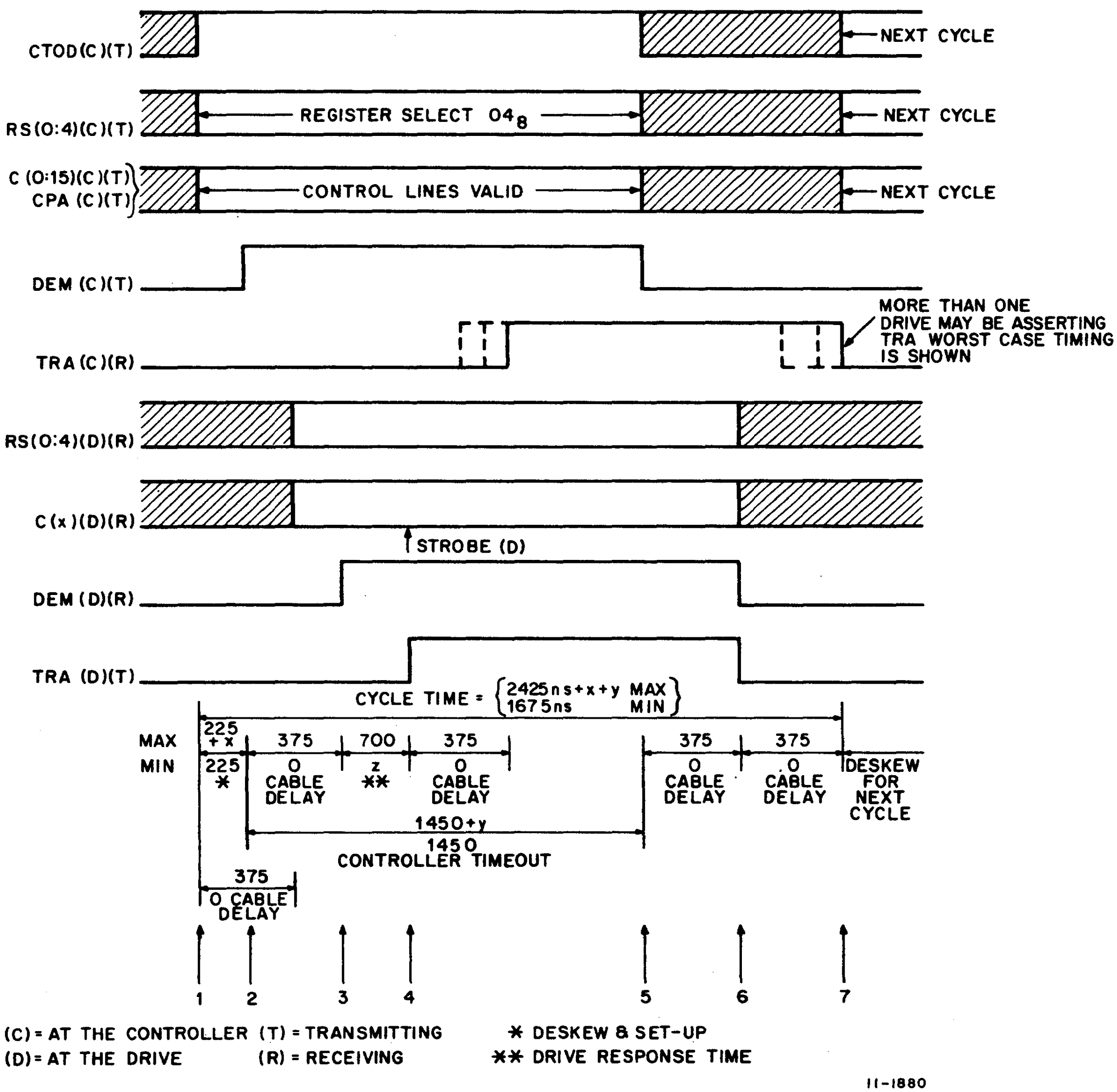
The controller may disable or change the DS, RS, CTOD, and C lines now, and begin the deskew for the next cycle.

6. After a cable delay, each drive receives the DEM negation, and negates TRA.

7. After all drives have negated TRA and after cable delay, the controller finally receives the TRA negation. This completes the Attention Summary register write cycle.

   For the next cycle, the controller may assert DEM now or when the RS, DS, and CTOD deskew time is finished, whichever occurs later).

FIGURE 3.3.9.4


CTOD(C)(T)
NEXT CYCLE
RS(0:4)(C)(T)
REGISTER SELECT 04$_8$
NEXT CYCLE
C (0:15)(C)(T)
CPA (C)(T)
CONTROL LINES VALID
NEXT CYCLE
DEM (C)(T)
TRA (C)(R)
MORE THAN ONE DRIVE MAY BE ASSERTING TRA WORST CASE TIMING IS SHOWN
RS(0:4)(D)(R)
C(x)(D)(R)
STROBE (D)
DEM (D)(R)
TRA (D)(T)
CYCLE TIME = {2425ns+x+y MAX, 1675ns MIN}
MAX
MIN
225 +x
225 *
375
0 CABLE DELAY
700
z **
375
0 CABLE DELAY
375
0 CABLE DELAY
375
0 CABLE DELAY
DESKEW FOR NEXT CYCLE
1450+y
1450
CONTROLLER TIMEOUT
375
0 CABLE DELAY
1
2
3
4
5
6
7
(C) = AT THE CONTROLLER (T) = TRANSMITTING
(D) = AT THE DRIVE (R) = RECEIVING
* DESKEW & SET-UP
** DRIVE RESPONSE TIME
11-1880

3.4 COMMAND INITIATION

3.4.1 General

3.4.1.1 To initiate a command in a drive, the controller (or the CPU, via the controller) writes a word into the Control register (00) of the drive. This word will contain a command code in bits 1 through 5 and a GO bit (always set, when starting a command) in position 0. The drive begins executing the command (if it is a valid one) as soon as the DEM line is negated on the control bus write which loaded the control register.

Command codes fall into two classes: non data transfer commands (such as Drive Clear, and Seek) and data transfer commands (such as Read Data). Bits 0 through 5 (including the GO bit) are 01 through 47(base 8) for non data transfer command codes; 51(base 8) through 77(base 8) are for data transfer codes (see section 6 for a description of commands and their codes).

3.4.2 Non Data Transfer Commands

3.4.2.1 Non data transfer commands have effect only on the state of the drive. The controller merely writes the command word into the drive as it would for any register. At the completion of the command execution, the drive typically raises an attention condition in order to signal its completion (see section 10 for a description of the attention condition).

3.4.2.2 If the non data transfer command code written into the drive is not recognized by the drive as a valid command, the drive will normally signal an error by raising an attention condition.

3.4.3 Data Transfer Commands

3.4.3.1 When any data transfer command is written into a drive, the controller expects data transfer on the data bus to begin soon thereafter. Normally, the controller will set a "controller busy" status bit related to the data bus as soon as the data transfer command code is written into a drive. Data transfer then follows, as described in section 4. See also section 10 for notes on programming related to data transfer.

3.5 THE ATTENTION LINE (ATTN)

3.5.1 The Attention line is the means by which drives which are not doing data transfers signal their need for service. The normal response of the controller is to cause an interrupt in the CPU when the Attention line is asserted. The CPU can then inspect the Attention Summary register and proceed from there.

3.5.2 Whenever the ATA bit in a drive is set, that drive is also asserting the Attention line. Up to eight drives may be asserting the Attention line at one time.

3.5.3 Drives do not set the ATA bit (and therefore do not assert the Attention line) while they are executing commands of any sort (the drive ready (DRY) status bit is set whenever ATA gets set). No internally generated changes of status should normally occur in a drive after it has set its ATA bit.

3.5.4 See section 8.1 for treatment of the attention condition when operating with dual controller drives.

3.6 THE INITIALIZE LINE (INIT)

3.6.1 The initialize line is used to perform a "system reset" of all drives attached to a controller. This line may be asserted by the controller at any time; however, the controller may not assert INIT and RUN simultaneously (see sec. 4.2.5.3).

3.6.2 When the INIT line is asserted by the controller, it will have a minimum pulse duration of 400 nanoseconds.

3.6.3 When a drive receives the assertion of INIT, it immediately aborts the execution of any ongoing command and then performs all functions described for the drive clear command (see 6.2.1.7).

3.6.4 In dual controller drives, a drive which is switched to controller B does not respond to the assertion of INIT on controller A, and vice versa.

3.7 CONTROL BUS PARITY CHECKING

3.7.1 Both the controller and the drive normally generate and check parity on control bus transfers. The CPA line is asserted or not to make an odd modulo 2 sum of the sixteen C lines and the CPA line.

3.7.2 When the controller reads the Attention Summary register (04), correct parity cannot be generated because not all C bits are being generated in one drive. Therefore, the controller must ignore the result of parity checking in this case.

3.7.3 All drives and controllers will have a means of disabling parity checking while otherwise operating normally. The purpose is to permit smaller, limited feature controllers or drives which do not contain parity circuitry to operate with standard drives or controllers.

Note: Not all current drives contain this feature.

3.7.4 When a control bus parity error is detected during a write into a drive register, the register should be loaded, and then PAR (or the appropriate error indicator) ERR, and ATA are set. In the case of a write into the Control register (00), the register should be loaded, but detection of a control bus parity error should inhibit the setting of the GO bit, and no command execution should occur.

3.8 OTHER CONTROL BUS TIMING CONSTRAINTS

3.8.1 If the controller selects a drive (DS<0:2>) which is not present, there will be no response to the DEM assertion. The controller should wait a minimum of 1450 nanoseconds after asserting DEM before declaring the error. It is recommended that this time not exceed 1800 nanoseconds.

These times are the same as shown above for the timeout in reading and writing the Attention Summary register. The same timer circuit may be used for both. The error will not occur when reading or writing the attention summary register.

3.8.2 The DEM line is always negated during the first deskew portion of the control bus timing cycle. Therefore, it is always negated for at least 225 nanoseconds before being asserted.

3.8.3 Because all of the timing of the control bus is based on a maximum electrical length of 160 feet, it is not possible to extend the Massbus beyond this length, such as by using bus repeaters.

3.8.4 Controllers and drives must insure that, when they are generating them, the signals on each of the C lines and on CPA are fixed and stable (either asserted or negated) from the time they are gated out until the DEM line is negated. This may require that certain status lines be buffered or latched when their states are being gated onto the C lines.

4. THE DATA BUS

4.1 INTRODUCTION

4.1.1 The purpose of the data bus is to transmit blocks of data at high speed between the controller and drives and to control the initiation and termination of block transmissions. Data transmission is "synchronous".

4.2 DATA BUS SIGNALS

4.2.1 Data (D<0:17>); Bidirectional

4.2.1.1 These lines carry 18 bits of data between controller and drive.

4.2.2 Data Parity (DPA); Bidirectional

4.2.2.1 This line carries a parity bit associated with the Data lines (odd parity).

4.2.3 Sync Clock (SCLK); Drive To Controller

4.2.3.1 This line carries a clock signal, generated by the drive, which controls the gating and strobing of data on the Data lines.

4.2.4 Write Clock (WCLK); Controller To Drive

4.2.4.1 This line returns the Sync Clock (SCLK) signal to the drive during transmission of data from controller to drive. It tells the drive when to strobe the data lines during a write.

4.2.5 Run (RUN); Controller To Drive.

4.2.5.1 This line controls the start-up and continuation of data transfer commands.

4.2.5.2 The RUN line is first asserted by the controller to start the execution of a data transfer command which has been placed in the Control register of a drive. Thereafter, the drive inspects the RUN line at the trailing edge (negation) of each EBL pulse; if the RUN line is still asserted, the operation normally continues.

4.2.5.3 RUN should not be asserted while INIT is asserted.

4.2.6 End Of Block (EBL); Drive To Controller

4.2.6.1 This line is pulsed by the drive at the end of each block (sector or record) of data transmitted. At the trailing edge (negation) of the pulse, the drive inspects the RUN line. If RUN is asserted the drive continues to read or write. If RUN is negated, the drive terminates the operation and disconnects from the data bus.

4.2.6.2 The EBL line is always pulsed at least once by the drive which is commanded to read or write. This is necessary, even in case of error, so that the controller can tell when the drive has disconnected from the data bus.

4.2.6.3 The duration of an EBL pulse is at least 1.5 microseconds. This is long enough to allow the controller to respond to the leading edge by negating the RUN line, and have the negation safely recognized when the trailing edge occurs.

4.2.6.4 After a drive has disconnected from the data bus (at the negation of an EBL pulse), no more state changes shall occur due to the data transfer operation just terminated.

4.2.7 Exception (EXC); Bidirectional

4.2.7.1 This line is asserted by a drive performing a data transfer to indicate that an error condition has been detected.

4.2.7.2 Once asserted by a drive, the EXC line remains asserted until the trailing edge of the last EBL pulse.

4.2.7.3 The time from the assertion of EXC to the trailing edge of the next EBL pulse must not be less than 1.5 microseconds. This is to allow the controller time to respond to the EXC assertion safely before the drive next inspects the RUN line.

4.2.7.4 The EXC line may be asserted by the controller in order to abort a data transfer command, but the controller may not assert EXC and RUN simultaneously.

4.2.8 Occupied (OCC); Drive To Controller

4.2.8.1 OCC is asserted by the drive as soon as a valid data transfer command is recognized and accepted. It is negated at the trailing edge of the last EBL pulse.

NOTE:

> OCC assertion is not conditional on receiving RUN assertion.

4.3 SEQUENCE AND TIMING OF DATA BUS TRANSFERS

4.3.1 Introduction

4.3.1.1 The data bus is used for transmission of data from and to the drive recording medium. Timing of transfers is controlled by a clock which is generated by the drive.

4.3.1.2 Transfers are oriented towards blocks of data which are transmitted as a group (e.g., sectors on a disk, records on mag tape). The drive will normally send and receive data only as whole blocks. If the number of data words desired by the CPU is not an integral times the number of words per block, it is up to the controller to stop the transfer to memory on reads or to provide filler words on writes.

4.3.1.3 The data bus is shared among all drives. Only one drive may be attached to it at a time. The controller should prevent a data transfer command from being loaded into a drive while OCC is asserted.

4.3.1.4 A drive attaches itself to the data bus and asserts OCC when a data transfer command is loaded into its Control register. After transferring one or more blocks of data (unless a class B error occurs), the drive disconnects from the data bus and negates OCC. Disconnect always occurs at the trailing edge (negation) of an EBL pulse.

4.3.1.5 For detailed description of error conditions and their effects on data bus signals, see section 7.

4.3.2 Data Bus Read Sequence

4.3.2.1 This section describes a typical data bus read sequence with no errors. 4.3.2.3 is a timing diagram of a read of a single sector with four words. 4.3.2.2 is a flowchart, with timing restrictions, of the read sequence. The following sequence occurs on a data bus read (refer to 4.3.2.3).

1. A read command is loaded into the Control register of the drive. If the command is valid, the drive enables its data bus receivers and drivers and asserts OCC.

2. Not more than 100 microseconds after step 1, the controller asserts RUN.

3. After a cable delay, the drive receives the RUN assertion. Disk drives now begin searching for the desired sector. Tape drives begin tape motion.

4. When the drive has read the first data word and gated it onto the D lines, it generates DPA and asserts SCLK.

5. After a cable delay, the controller receives the SCLK assertion.

6. The drive negates SCLK no less than T nanoseconds after asserting it, where T is either 225 nanoseconds or 30 percent of the nominal burst data period of the drive, whichever is greater. The Data lines should be maintained valid for no less than one half of the SCLK interval after SCLK is negated.

7. After a cable delay, the controller receives the SCLK negation. The controller strobes the D lines and DPA, and checks the parity.

8. If there is more data to be read in this block, then not less than T nanoseconds after step 6, the drive gates out the next data word onto the D lines, generates DPA, and asserts SCLK. Steps 5, 6, and 7 then follow.

9. After the negation of SCLK (step 6) on the last word of data in the block, the drive asserts EBL.

10. After a cable delay, the controller receives the EBL assertion. At this time, the controller must decide whether or not to have the drive read the next block of data without disconnecting from the data bus (the controller may already have negated the RUN line).

11. If the controller decides not to read the next block, it negates the RUN line not later than 500 nanoseconds after step 10.

12. After a cable delay, the drive receives the RUN negation (the RUN line may already have been negated).

13. Not less than 1500 nanoseconds after step 9, the drive negates EBL. At this time the drive strobes the RUN line. If RUN has been negated, the drive disconnects from the data bus (the DRY bit should be set and OCC negated at this time).

14. After a cable delay, the controller receives the EBL negation (the controller may now generate an end-of-transfer interrupt, and start another data transfer).

FIGURE 4.3.2.2

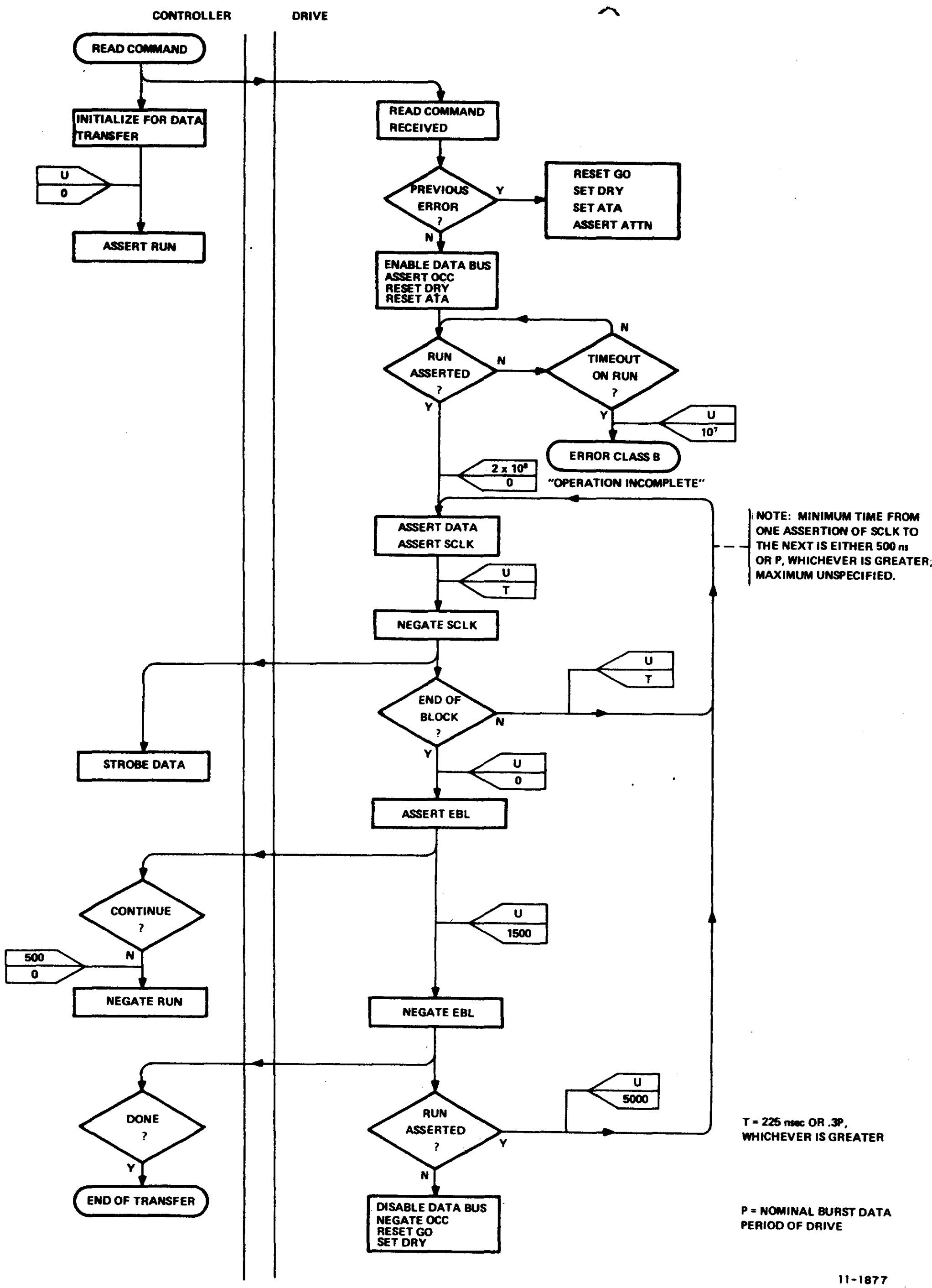
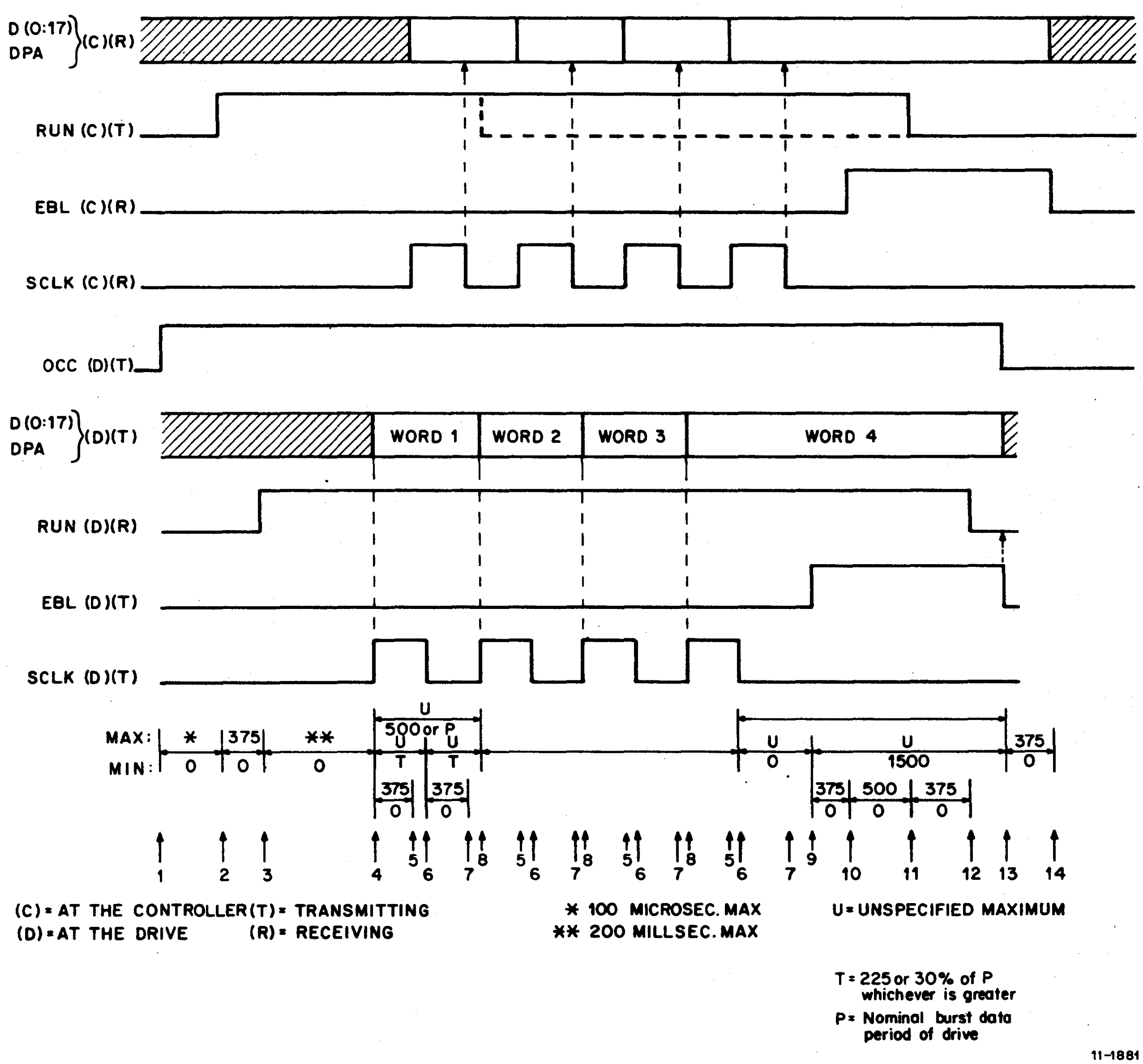

CONTROLLER
DRIVE
READ COMMAND
INITIALIZE FOR DATA TRANSFER
U
0
ASSERT RUN
READ COMMAND RECEIVED
PREVIOUS ERROR ?
Y
N
RESET GO
SET DRY
SET ATA
ASSERT ATTN
ENABLE DATA BUS
ASSERT OCC
RESET DRY
RESET ATA
RUN ASSERTED ?
TIMEOUT ON RUN ?
U
10⁷
ERROR CLASS B
"OPERATION INCOMPLETE"
2 x 10⁸
ASSERT DATA
ASSERT SCLK
NOTE: MINIMUM TIME FROM ONE ASSERTION OF SCLK TO THE NEXT IS EITHER 500 ns OR P, WHICHEVER IS GREATER; MAXIMUM UNSPECIFIED.
T
NEGATE SCLK
END OF BLOCK ?
STROBE DATA
ASSERT EBL
CONTINUE ?
500
NEGATE RUN
1500
NEGATE EBL
DONE ?
END OF TRANSFER
5000
T = 225 nsec OR .3P, WHICHEVER IS GREATER
DISABLE DATA BUS
NEGATE OCC
RESET GO
SET DRY
P = NOMINAL BURST DATA PERIOD OF DRIVE
11-1877

FIGURE 4.3.2.3



D (0:17) DPA (C)(R)
RUN (C)(T)
EBL (C)(R)
SCLK (C)(R)
OCC (D)(T)
D (0:17) DPA (D)(T)
WORD 1
WORD 2
WORD 3
WORD 4
RUN (D)(R)
EBL (D)(T)
SCLK (D)(T)
MAX:
MIN:
500 or P
U
T
375
0
1500
*
**
1 2 3 4 5 6 7 8 9 10 11 12 13 14
(C) = AT THE CONTROLLER (T) = TRANSMITTING
(D) = AT THE DRIVE (R) = RECEIVING
* 100 MICROSEC. MAX
** 200 MILLSEC. MAX
U = UNSPECIFIED MAXIMUM
T = 225 or 30% of P whichever is greater
P = Nominal burst data period of drive
11-1881

4.3.3 Data Bus Write Sequence

4.3.3.1 This section describes a typical data bus write sequence with no errors. 4.3.3.2 is a flowchart of the write sequence, showing timing restrictions. See 4.3.3.3 for a timing diagram of a write of a single sector with four words.

The following sequence occurs on a data bus write.

1. A write command is loaded into the Control register of the drive. If the command is valid, the drive enables its data bus receivers and drivers and asserts OCC.

2. Not more than 10 milliseconds after step 1, the controller has gated the first word onto the D lines, and has generated DPA. It asserts RUN.

3. After a cable delay, the drive receives the RUN assertion. Disk drives now begin searching for the desired sector. Tape drives begin tape motion.

4. When the drive is ready to accept the first word, it asserts SCLK.

5. After a cable delay, the controller receives the SCLK assertion. The controller asserts WCLK.

6. After a cable delay, the drive receives the WCLK assertion. The drive now strobes the D lines and DPA. It checks parity (the drive may now begin to write the word on the medium).

7. The drive negates SCLK no less than T nanoseconds after asserting it, where T is either 225 nanoseconds or 30 percent of the nominal burst data period of the drive, whichever is greater.

8. After a cable delay, the controller receives the SCLK negation. The controller negates WCLK; it then gates out the next word on the D lines, and generates DPA.

9. After a cable delay, the drive receives the WCLK negation.

10. If more words are to be written, then not less than T nanoseconds after negating SCLK, the drive asserts it again (the interval between the first and second SCLK pulses shown in 4.3.3.3 is longer than the later intervals, since this is probably typical behavior for disk drives. The restrictions on the intervals are not different). Steps 5, 6, 7, 8, and 9 then follow.

11. After the negation of SCLK (step 7) for the last word in the block, the drive asserts EBL.

12. After a cable delay, the controller receives the EBL assertion. At this time, the controller must decide whether or not to have the drive write the next block of data without disconnecting from the data bus (the controller may already have negated the RUN line).

13. If the controller decides not to write the next block, it negates the RUN line not later than 500 nanoseconds after step 12.

14. After a cable delay, the drive receives the RUN negation (the RUN line may already have been negated).

15. Not less than 1500 nanoseconds after step 11, the drive negates EBL. At this time the drive strobes the RUN line. If RUN has been negated, the drive disconnects from the data bus (the DRY bit should be set and OCC negated at this time).

16. After a cable delay, the controller receives the EBL negation (the controller may now generate an end-of-transfer interrupt, and start another data transfer).

FIGURE 4.3.3.2

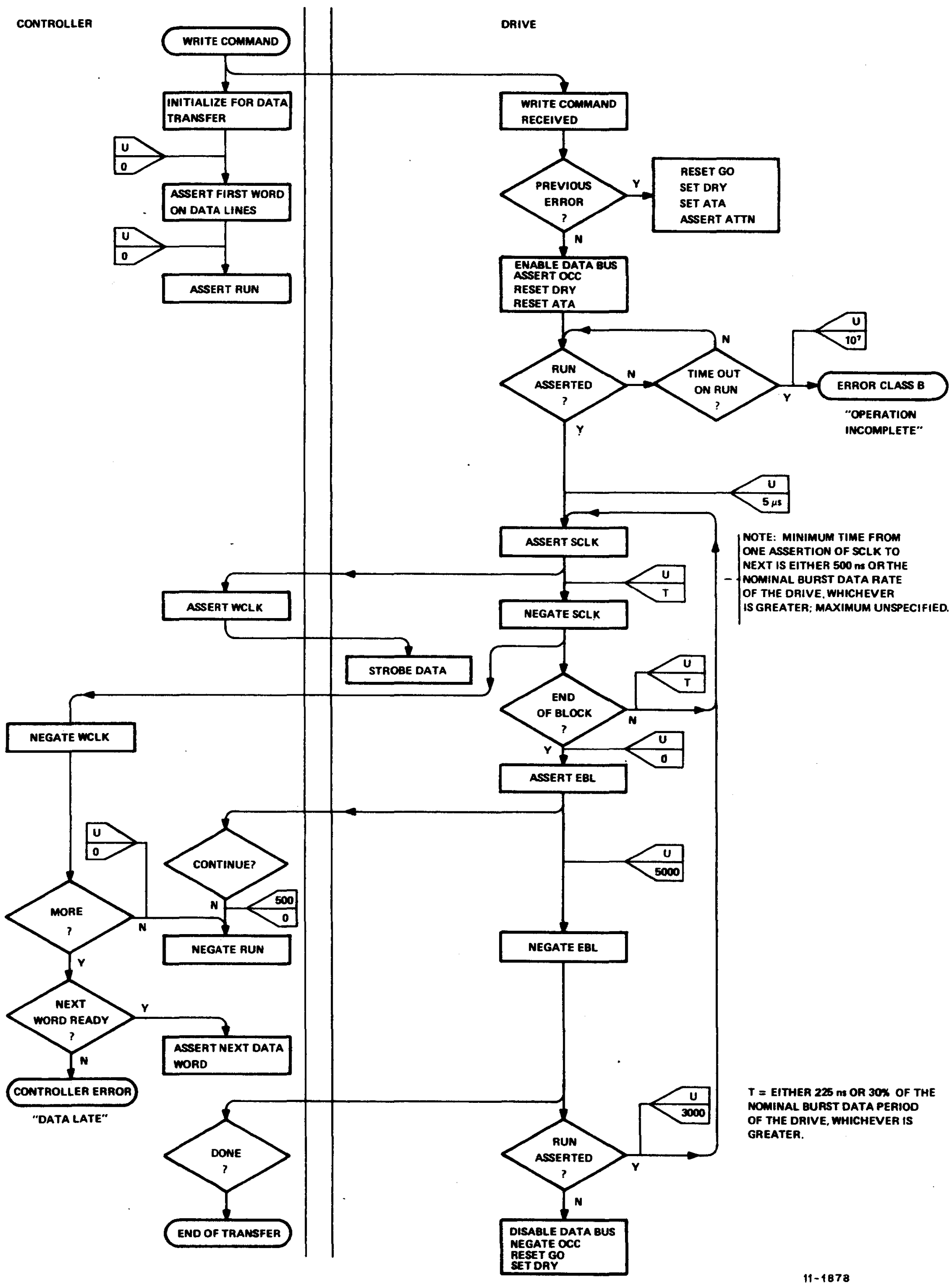
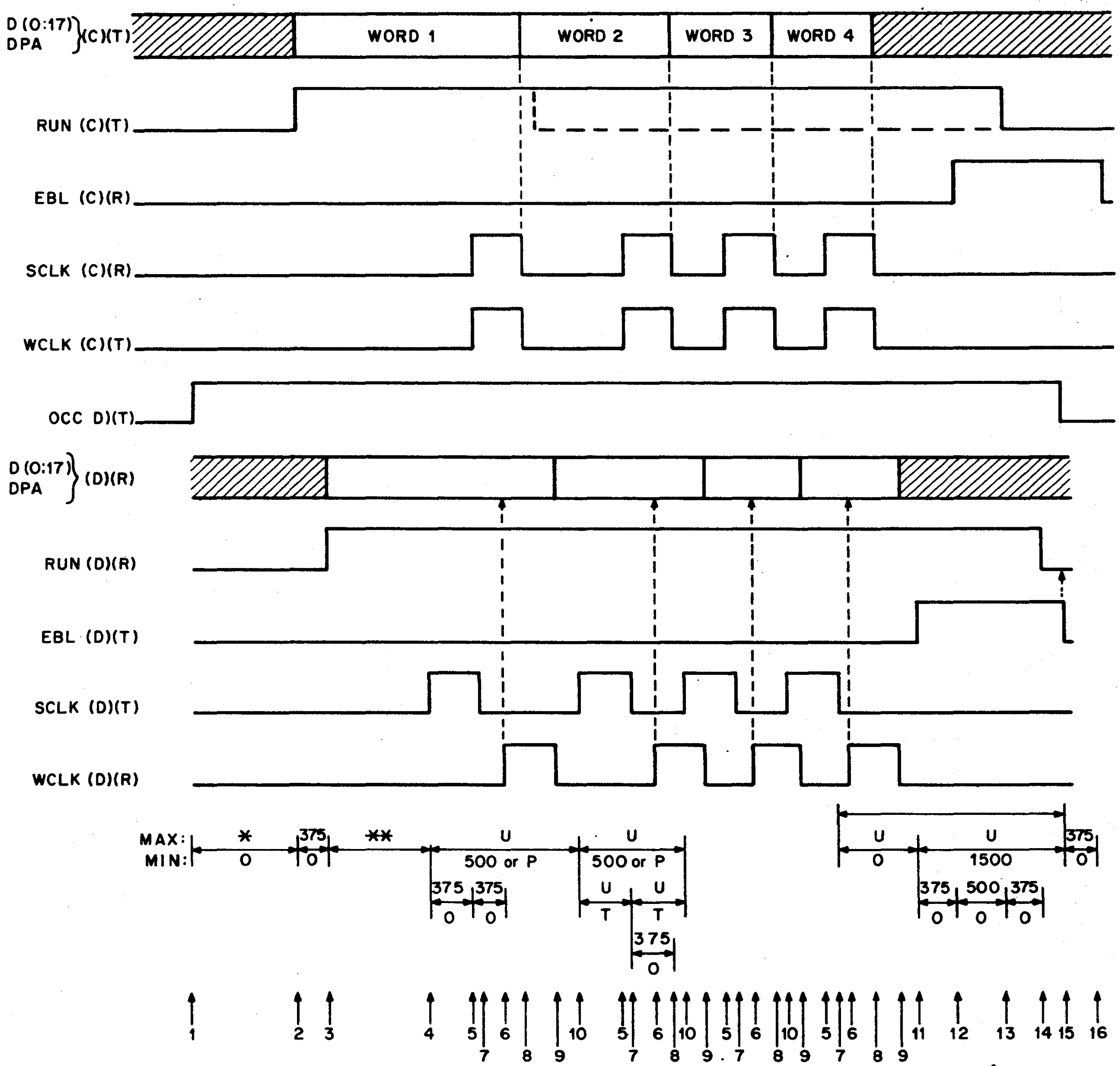

CONTROLLER
DRIVE
WRITE COMMAND
INITIALIZE FOR DATA TRANSFER
U
0
ASSERT FIRST WORD ON DATA LINES
U
0
ASSERT RUN
WRITE COMMAND RECEIVED
PREVIOUS ERROR ?
Y
RESET GO
SET DRY
SET ATA
ASSERT ATTN
N
ENABLE DATA BUS
ASSERT OCC
RESET DRY
RESET ATA
U
10⁷
N
RUN ASSERTED ?
N
TIME OUT ON RUN ?
Y
ERROR CLASS B
"OPERATION INCOMPLETE"
Y
U
5 μs
ASSERT SCLK
NOTE: MINIMUM TIME FROM ONE ASSERTION OF SCLK TO NEXT IS EITHER 500 ns OR THE NOMINAL BURST DATA RATE OF THE DRIVE, WHICHEVER IS GREATER; MAXIMUM UNSPECIFIED.
ASSERT WCLK
U
T
NEGATE SCLK
STROBE DATA
U
T
END OF BLOCK ?
N
NEGATE WCLK
Y
U
0
ASSERT EBL
U
0
CONTINUE?
U
5000
MORE ?
N
500
0
N
NEGATE RUN
NEGATE EBL
Y
NEXT WORD READY ?
Y
ASSERT NEXT DATA WORD
N
CONTROLLER ERROR
"DATA LATE"
T = EITHER 225 ns OR 30% OF THE NOMINAL BURST DATA PERIOD OF THE DRIVE, WHICHEVER IS GREATER.
U
3000
DONE ?
RUN ASSERTED ?
Y
N
END OF TRANSFER
DISABLE DATA BUS
NEGATE OCC
RESET GO
SET DRY
11-1878

FIGURE 4.3.3.3



(C) = AT THE CONTROLLER (T) = TRANSMITTING * 100 MICROSEC MAX.

(D) = AT THE DRIVE (R) = RECEIVING ** 200 MILLISEC. MAX.

T = 225 or 30% of P, whichever is greater.

P = Nominal burst data period of drive.

11-1882

4.4 DATA BUS PARITY CHECKING

4.4.1 Both the controller and the drive normally generate and check parity on data bus transfers. The DPA line is asserted or not to make an odd modulo 2 sum of the eighteen D lines and the DPA line.

4.4.2 All drives and controllers will have a means of disabling parity checking while otherwise operating normally. The purpose is to permit smaller limited-feature controllers or drives which do not contain parity circuitry to operate with standard drives or controllers.

Note: Not all current drives contain this feature.

4.5 ERROR SIGNALING

4.5.1 The EXC line is used to signal an error condition by the drive which is doing a data transfer while the transfer is going on. See section 7 for a complete description of how various error conditions are handled.

4.5.2 The drive which is doing a data transfer never asserts the ATTN line (of the control bus) while the data transfer is going on. If an error occurs, the drive does assert ATTN after disconnecting from the data bus (after the DRY bit is set and OCC negated).

4.5.3 When the controller asserts EXC, it must always negate RUN; under normal conditions, the controller negates RUN whenever the EXC line is asserted. See section 7 for descriptions of cases where this may not be true, and the possible consequences.

4.6 DATA TRANSFER COMMAND TERMINATION

4.6.1 Data transfer commands are normally terminated at the trailing edge (negation) of the EBL pulse (when the RUN line has been negated). In abnormal or error situations, the drive still must generate an EBL pulse, in order to define to the controller when the drive has disconnected from the data bus.

4.6.2 When a drive negates EBL, it disconnects from the data bus if any one of the following is true:

1. The RUN line is negated.

2. The EXC line has been asserted by the controller

3. The drive has asserted EXC due to a Class B error.

The controller cannot distinguish 3 above from the case of a Class A error (in which the drive will continue if RUN remains asserted). Therefore, case 3 may result in a "NO SCLK" timeout after 250 milliseconds (see section 7).

4.6.3 In mag tape, a Frame Count register substitutes for sector detection on disks. The mag tape drive generates an EBL pulse when the frame count becomes zero.

4.6.4 During data transfers, control bus writes are not permitted into any registers except the Attention Summary register and the Maintenance register.

4.7 ADDITIONAL DATA BUS TIMING RESTRICTIONS

4.7.1 Because the timing of the data bus is based on a maximum electrical length of 160 feet, it is not possible to extend the Massbus beyond this length, such as by using bus repeaters.

4.7.2 There is a point in time for each sector (on disk) beyond which an attempt to read or write that sector will fail (on this revolution). Let us call this the "Sector Decision Point". The drive should assure that the trailing edge of the EBL pulse occurs not later than 5 microseconds before the "Sector Decision Point" of the following sector. The purpose is to allow the controller sufficient time, after a transfer terminates on a drive, to write a new desired sector address and a new command into that drive, and not lose a revolution when the desired sector is the next one.

4.7.3 When the drive asserts EXC, it must assure that (whenever EBL is next asserted) EXC and EBL are asserted together for at least 1500 nanoseconds. This may sometimes require extending the normal duration of the EBL pulse.

4.7.4 When a severe (Class B) error occurs in a drive during a data transfer, the drive designer must take steps to assure that the SCLK, EXC, and EBL lines do not get "stuck" in the asserted state. Also, the SCLK pulse should not be allowed to "glitch" (have a very short duration) in any error case.

4.7.5 After the controller has asserted RUN, an error should be declared if no SCLK pulse is received for 250 milliseconds or more (unless OCC is asserted). An error should also be declared if, after RUN is negated, no EBL pulse has occurred for 250 milliseconds or more (unless OCC is asserted). The assertion of OCC should inhibit both timeout errors.

4.7.6 After the drive has received a valid data transfer command, the drive should declare an error (Class B, which causes EXC and EBL pulses to occur) if RUN assertion has not been received for 10 milliseconds or more.

4.7.7 The drive will not assert EXC (without a simultaneous EBL pulse) until after RUN assertion (or EXC assertion) has been received.

4.7.8 A drive should assert SCLK no sooner than 2 microseconds after receiving the DEM assertion on the writing of a valid command into the Control register.

4.7.9 The controller does not assert WCLK on read and write-check operations.

4.8 RECOMMENDED PULSE DURATIONS

4.8.1 The EBL pulse should be initiated as early as possible. Its duration, if possible, should be at least 2 to 5 microseconds, as long as this is consistent with 4.7.2. More than the 5 microsecond margin specified in 4.7.2 should be provided if possible.

4.8.2 The controller should assert RUN as early as possible. Under normal circumstances, RUN should remain asserted until the controller recognizes the end of a transmission (by word count or block count).

4.8.3 The drive should be prepared to abort a data transfer in an orderly way during the time after an EBL pulse negation and before the next sector decision point (see sec. 4.7.2). Some controllers may routinely use the EXC line to stop transfers during this time.

5. DRIVE REGISTERS

5.1 INTRODUCTION

5.1.1 There are a maximum of 32 registers addressable in each drive; a drive will normally implement a subset of these. When a register which is defined in this section is implemented, it must be assigned to the Massbus location specified in its definition.

5.2 DESCRIPTION OF DRIVE REGISTERS

5.2.1 Massbus Location 00

5.2.1.1 This location contains the Control register. This register receives the command code, and displays Port Status, Format, and GO bits.

5.2.1.2 All Massbus devices must implement this register.

5.2.2 Massbus Location 01

5.2.2.1 This location contains the Status register. This Read-Only register displays all non-error status bits plus the composite error bit.

5.2.2.2 All Massbus devices must implement this register.

5.2.3 Massbus Location 02

5.2.3.1 This location contains the Error 1 register

5.2.3.2 All Massbus devices must implement this register.

5.2.4 Massbus Location 03

5.2.4.1 This location contains the Maintenance register. This register is to be defined by the drive designer, for use in diagnostic and maintenance functions.

5.2.4.2 All Massbus devices must implement this register.

5.2.5 Massbus Location 04

5.2.5.1 This location contains the Attention Summary register. This is a special pseudo-register which displays the "Attention Active" status of all drives, one bit per drive. Each of these ATA bits may be cleared by writing a "1" into the appropriate position in this register.

5.2.5.2 All Massbus devices must implement this register.

5.2.6 Massbus Location 05

5.2.6.1 All 16 bits of this register are to be implemented as Read/Write bits to assure that diagnostic routines have at least one register in which all 16 bits of the control bus are received and latched.

5.2.6.2 This location contains the Frame Count register of Massbus tape drives.

5.2.6.3 This location contains the Desired Sector/Track Address register of Massbus drum and disk devices.

5.2.7 Massbus Location 06

5.2.7.1 This location contains the Drive-Type register. This Read-Only register contains information identifying the drive model and characteristics.

Engineering note 3.5 shows sample settings of all bits in this register for the drives which have been assigned Drive Type Numbers.

5.2.7.2 All Massbus devices must implement this register.

5.2.8 Massbus Location 07

5.2.8.1 This location contains the Look-Ahead register of Massbus disk drives. This Read-Only register displays information on the rotational position of a disk.

5.2.8.2 This location contains the Tape Character Check register of Massbus tape drives.

5.2.9 Massbus Location 12(base 8)

5.2.9.1 This location contains the Desired Cylinder Address register of Massbus moving head disk drives. This register receives the address of a cylinder to which a Seek is to be done.

5.2.10 Massbus Location 13(base 8)

5.2.10.1 This location contains the Current Cylinder Address register of Massbus moving head disk drives. This Read-Only register displays the address of the cylinder at which a moving-head disk is positioned.

5.2.11 Massbus Location 14(base 8)

5.2.11.1 This location contains the Serial Number Register of certain Massbus devices. This register must be implemented on drives with a removable medium (i.e., magtape, cartridge disk, etc.). On other drives, this register may be implemented if desired by the drive designer.

The Serial Number register is not required on drives with a non-removable medium because the drive serial number can be recorded on the medium.

5.2.11.2 This register is intended to allow identification of the drive unit and to distinguish it from other drives, possibly of the same type, attached to a single controller. It is also intended as an aid to field service trouble reporting, by giving software the means to consistently identify the particular drive which has failed or has given error indications.

5.2.11.3 The serial number is displayed as four decimal digits, which are the last four digits of the drive serial number as normally defined and stamped on the cabinet.

5.3 DETAILS OF MANDATORY DRIVE REGISTERS

5.3.1 Control Register (00); Read/Write

5.3.1.1 Bit 0 (GO); Read/Write

A command (bits 1-5 of this register) which is to be executed is always transmitted with this bit set. When the assertion of the "GO" bit is received the drive resets the DRY status bit. When reading from this register, this bit always corresponds to the opposite of the DRY status bit.

5.3.1.2 Bits 1-5 (F1-F5), Function Code; Read/Write

These bits are the command which is being or has been executed by the drive. See section 6.

5.3.1.3 Bits 6-10

These bits are reserved for use by the controller.

5.3.1.4 Bit 11 (DVA) Drive Available; Read-Only

This bit is intended for use in dual-controller configurations. The bit normally appears set. It appears reset when the drive is switched to the other controller.

5.3.1.5 Bit 12-13 (spare); Read-Only

These two bits are spares, but may only be used as read- only status bits. This is to allow controllers to also use these bit positions for writable local control bits.

5.3.1.6 Bits 14-15

These bits are reserved for use by the controller.

5.3.2 Status Register (01); Read-Only

5.3.2.1 Bit 0 (spare)

5.3.2.2 Bit 1 (BOT) Beginning Of Tape; Read-Only

Set while magtape is positioned over the Beginning-Of-Tape marker.

5.3.2.3 Bit 2 (EOF) End-Of-File; Read-Only

Set when a file mark record has been read on magtape. Reset at the beginning of each read and write command.

5.3.2.4 Bits 3-5 (spare)

5.3.2.5 Bit 6 (VV) Volume Valid; Read-Only

This bit is reset whenever the validity of the volume mounted in a drive with removable media is in question. In the RP04, for example, VV is reset whenever MOL changes state.

5.3.2.6 Bit 7 (DRY) Drive Ready; Read-Only

This bit is set when the drive is ready to accept a command. Upon completion of a data transfer, "DRY" should never be asserted before the negation of the last "EBL" pulse. For other operations, the drive designer should explicitly specify when "DRY" is asserted. It is reset by the drive when it begins to execute a command; however, there is no minimum time for it to remain negated.

5.3.2.7 Bit 8 (DPR) Drive Present; Read-Only

In drives with dual controller ports, this bit is set to indicate that the drive is switched to this controller. It is reset while the drive is in the neutral state or is switched to the other controller. In single-controller drives, this bit is always set.

5.3.2.8 Bit 9 (spare)

5.3.2.9 Bit 10 (LBT) Last Block Transferred; Read-Only

This bit is set by a disk drive at the end of the transfer of the last block (i.e., set at the assertion of end of block for the last block). It is reset whenever a new value is written into the Desired Sector/Track Address register. The purpose is to provide an indication of "end of drive" before an error occurs if early detection is wanted for spiral reads or writes.

NOTE:

If a read or write attempts to continue into the next block, the AOE error occurs. LBT will remain asserted.

5.3.2.10 Bit 11 (WRL) Write Locked; Read-Only

This bit is set when the drive is in the write locked state and will therefore not accept write commands. The state may be caused by a panel switch, a write protect ring (magtape), or by the matching of the desired track field with a setting of write-protect track switches. This bit is reset only by changing the state of the switch.

5.3.2.11 Bit 12 (MOL) Medium On Line; Read-Only

This bit is set when all applicable conditions in the following list are satisfied.

A. Removable medium is mounted

B. Door closed

C. Moving heads loaded

D. LOAD/RUN switch is in the RUN position

E. Spindle speed is ok.

F. Unit number plug is inserted

The bit is reset when any applicable condition becomes false. An attention condition is raised for every change of state of this bit.

5.3.2.12 Bit 13 (PIP) Positioning Operation in Progress; Read-Only

This bit is set during the execution of some non-data-transfer commands (for example: Search, Seek) it is always reset at the termination of the command.

5.3.2.13 Bit 14 (ERR) Composite Error; Read-Only

This bit is set whenever any bit in any error register is set. It provides a summary indication of the existence of some error condition. While this bit is set, the only command which may be executed is the Drive Clear command. This bit may be reset by writing 0's into the error registers, by executing a Drive Clear command, or by asserting the INIT line.

5.3.2.14 Bit 15 (ATA) Attention Active; Read-Only

This bit is set at the occurrence of an attention condition (see sec. 9.4). It may be reset by executing a Drive Clear command, by asserting the INIT line, or by writing a 1 into the appropriate position of the Attention Summary register (see section 9.4.4).

This bit also appears in the Attention Summary register, in the bit position corresponding to the drive unit number.

Whenever this bit is set, this drive is asserting the ATTN line.

5.3.3 Error 1 Register (02); Read/Write

5.3.3.1 Bit 0 (ILF) Illegal Function; Read/Write

Set when a function code is received in the Control register (bits 1-5) with the GO bit (bit 0) set, and the code does not correspond to an implemented command on this drive.

5.3.3.2 Bit 1 (ILR) Illegal Register; Read/Write

Set when a control bus read or write is attempted from or to a non-existent register (writing into a "read- only" register does not cause this error).

5.3.3.3 Bit 2 (RMR) Register Modification Refused; Read/Write

Set when a control bus write is attempted into any existing drive register (except Attention Summary or Maintenance) while an operation is in progress on this drive.

5.3.3.4 Bit 3 (PAR) Parity; Read/Write

Set when a parity error is detected on the data bus during a write operation, or when a parity error is detected on the control bus while writing into a register. This bit applies only to information being transmitted from controller to drive.

NOTE:

Drive designers may, at their option, implement an additional bit (DPE) to signal a parity error on the data bus section. See sec. 7.6.4.

5.3.3.5 Bit 4 (FER) Format Error; Read/Write

Some drives have a medium with more than one format. This bit is set when the Format Selector bit does not match the format identification on the medium (see desired Cylinder Addresss register, 5.2.9) during a read or write operation.

5.3.3.6 Bits 5-8 (spare)

5.3.3.7 Bit 9 (AOE) Address Overflow Error; Read/Write

Set during a read or write operation when the last addressable sector has been read or written and the controller attempts to continue reading or writing. The error is detected when the Desired Sector/Track Address register (and the Desired Cylinder Address register, if present) is incremented by the drive, causing overflow from the highest order address. The invalid address error (IAE, see bit 10) is not set in this case.

5.3.3.8 Bit 10 (IAE) Invalid Address Error; Read/Write

Set when the drive receives a command (with GO bit set) which requires the use of the Desired Sector/Track Address register, and the contents of that register do not correspond to any existing sector on this drive.

Also set when the drive receives a command (with GO bit set) which requires the use of the Desired Cylinder Address register, and the contents of that register do not correspond to any existing cylinder on this drive.

5.3.3.9 Bit 11 (WLE) Write Lock Error; Read/Write

Set when a write command (with GO bit set) is received while the write locked (WRL, see 5.3.2.10) status bit is set.

5.3.3.10 Bit 12 (DTE) Drive Timing Error; Read/Write

Set when an internal timing failure is detected by the drive during a read or write operation.

5.3.3.11 Bit 13 (OPI) Operation Incomplete; Read/Write

Set when an operation fails to complete within the expected time. Examples: a Search command has not completed after the disk index mark has been detected twice; a Seek operation has not completed after the maximum allowable time; a read or write command has been received (with GO bit set), but the RUN line has not been asserted after 10 milliseconds (or more).

5.3.3.12 Bit 14 (UNS) Drive Unsafe; Read/Write

Set when a condition has occurred in the drive which makes it impossible to operate normally. Examples: power line voltage below limit; temperature limit exceeded.

5.3.3.13 Bit 15 (DCK) Data Check Error; Read/Write

Set when the data checking circuitry of the drive (CRC or ECC) has detected an error in a block of data while reading.

5.3.4 Maintenance Register (03); Read/Write

5.3.4.1 Bit 0 (DMD) Diagnostic Mode; Read/Write

This bit controls certain diagnostic and maintenance functions, to be defined by the drive designer (example: in moving-head disk drives, a write command will not be executed while the heads are offset from cylinder centerline, except when this bit is set).

5.3.4.2 Bits 1-15

These bits are to be defined by the drive designer.

5.3.5 Attention Summary Register (04); Read/Write-1-To-Clear

5.3.5.1 This register consists of one to eight status bits, each corresponding to the ATA status bit of one drive. Bit 0 is the ATA bit of drive 0, bit 1 is the ATA bit of drive 1, and so on to bit 7. Bits 8-15 are not used.

5.3.5.2 The timing of transfers on the control bus to and from this register is special. See sections 3.3.5 and 3.3.9 for a description.

5.3.5.3 When this register is selected, all drives respond. When reading the register, each drive presents its ATA bit in the appropriate bit position. When writing, each drive receives a bit from the control bus, and if the bit is set, the drive resets its ATA bit. For each bit position and drive, the table below applies when writing into this register.

Bit Written	ATA before	ATA after
0	0	0
0	1	1
1	0	0
1	1	0

This scheme allows the program to reset the ATA bits which were already seen and acted upon, without accidentally resetting other ATA bits which may have become set in the meantime.

5.3.6 Drive Type Register (06); Read-Only

5.3.6.1 Bits 0-8 (DT0-DT8); Drive Type Number; Read-Only.

These bits contain a code number which identify the drive model and major variations. The codes are assigned by the Massbus committee and are to be wired into the drive at manufacture. See engineering note 3.5 for a list of Drive Type numbers which are currently defined.

5.3.6.2 Bits 9-10; (spare)

5.3.6.3 Bit 11 (DRQ) Drive Request Required; Read-Only.

Set to indicate a drive with dual controller ports, which therefore must be requested before use and released after use (see 8.1)

5.3.6.4 Bit 12 (spare)

5.3.6.5 Bit 13 (MOH) Moving-Head; Read-Only

Set to indicate that this is a moving-head disk drive, which therefore has a Desired Cylinder Address register and other moving-head control and status bits.

5.3.6.6 Bit 14 (TAP) Tape Drive; Read-Only

Set to indicate that this is a tape drive.

5.3.6.7 Bit 15 (NSA) Not Sector-Addressed; Read-Only

Set to indicate that this drive does not have a desired sector track address register.

5.3.6.8 Engineering note 3.5 shows sample settings of all bits in this register for the drives which have been assigned Drive Type numbers.

5.3.7 Serial Number Register (17(8)); Read-Only

5.3.7.1 Bits 0-3 (SN01, SN02, SN04, SN08); Read-Only

Lowest order digit of the serial number, encoded as an 8-4-2-1 decade.

5.3.7.2 Bits 4-7 (SN11, SN12, SN14, SN18); Read-Only

Second lowest decade.

5.3.7.3 Bits 8-11 (SN21, SN22, SN24, SN28); Read-Only

Third decade.

5.3.7.4 Bits 12-15 (SN31, SN32, SN34, SN38); Read-Only

High order (4th) decade.

6. COMMANDS

6.1 COMMAND CODES

6.1.1 Command codes are divided into two types. The first type are commands which do not cause data transmission (and therefore do not use the data bus). The second type are those which do cause data transmission over the data bus.

Command codes are listed below by octal value. The two- digit octal code represents the low-order 6 bits (bits 0-5) of the Control register. This includes the GO bit (bit 0), which is always set when a command is initiated. Therefore, all of the codes below are odd.

The two types of commands are distinguished by the code group. Codes 01-47 are non-data-transfer commands. Codes 51-77 are data transfer commands.

6.1.2 Non-data-transfer commands

Command Code (octal)	Moving-Head disk	Drum And Fixed-head Disk	Magnetic Tape
01	No Operation	No Operation	No Operation
03	Unload	Spare	Rewind, Offline
05	Seek	Spare	Spare
07	Recalibrate	Spare	Rewind
11	Drive Clear	Drive Clear	Drive Clear
13	Release	Spare	Spare
15	Offset	Spare	Spare
17	Return To Centerline	Spare	Spare
21	Readin Preset	Readin Preset	Readin Preset
23	Pack Acknowledge	Spare	Spare
25	Spare	Spare	Erase
27	Spare	Spare	Write File Mark
31	Search	Search	Space Forward
33	Spare	Spare	Backspace
35	Spare	Spare	Spare
37	Spare	Spare	Spare
41	Spare	Spare	Spare
43	Spare	Spare	Spare
45	Spare	Spare	Spare
47	Spare	Spare	Spare

6.1.3 Data transfer commands

Command Code (octal)	Moving-Head Disk	Drum And Fixed-head Disk	Magnetic Tape
51	Write Check Data	Write Check Data	Write Check Forward
53	Write Check Header and Data	Spare	Spare
55	Spare	Spare	Spare
57	Spare	Spare	Write Check Reverse
61	Write Data	Write Data	Write Forward
63	Write Header And data	Spare	Spare
65	Spare	Spare	Spare
67	Spare	Spare	Spare
71	Read Data	Read Data	Read Forward
73	Read Header And data	Spare	Spare
75	Spare	Spare	Spare
77	Spare	Spare	Read Reverse

6.1.4 Codes 51-57 are reserved for Write Check commands. Codes 61-67 are reserved for Write commands. Codes 71-77 are reserved for Read commands.

6.1.5 The Write Check command codes must correspond to the Read command codes one-for-one, because the action of the drive is the same for Write Check and for Read.

6.2 COMMAND DESCRIPTIONS

6.2.1 Non-data-transfer commands

Unless otherwise noted, each of these commands causes the drive to become busy (GO=1, DRY=0) (see 5.3.2.6) for a finite time, and causes an Attention condition at the termination of the command.

6.2.1.1 No Operation (01)

This command takes no time to execute. The drive does not become busy. The GO bit is reset immediately, and no Attention condition is generated.

6.2.1.2 Unload (03)

For moving-head disk, this command causes the heads to retract and the drive to be taken off line (MOL bit reset; see 5.3.2.11). However, an Attention condition is not raised unless the operation fails to complete. This is because it is assumed that the program which issues an unload command does not need to be advised of the "going offline" change of status caused by that command. ATA should be asserted by the drive when it becomes ready after coming back online after an Unload command.

The drive electronics are not powered down by this operation.

6.2.1.3 Rewind, Offline (03)

The selected magtape begins rewinding and goes offline. GO is reset and DRY, ATA, and SSC become asserted. Operator intervention is required to bring the drive back online.

6.2.1.4 Seek (05)

For moving-head disk, this command initiates a seek to a new cylinder. The Desired Cylinder Address register contents control the destination of the seek.

Bits 0-7 of the Offset register are reset, and the heads are positioned to the centerline of the new cylinder.

6.2.1.5 Recalibrate (07)

For moving-head disk; causes the heads to be re-indexed and then positioned to cylinder 000. The Current Cylinder Address register (bits 0-9) and the Offset register (bits 0-7) are cleared.

6.2.1.6 Rewind (07)

For magtape; causes the tape to be wound back to the beginning-of-tape mark and then stopped.

6.2.1.7 Drive Clear (11(base 8))

This command is the only command which will be accepted by a drive while the ERR bit is set (see 5.3.2.13). The command causes all of the following to be reset:

All error registers
The ATA and ERR bits in the Status register
The ATA bit for this drive, in the Attention Summary register
The Desired Sector/Track Address register

Bits 0-9 of the Desired Cylinder Address register
Bits 0-7 of the Offset register
The ECC Position register
The ECC Pattern register
Some bits of the Maintenance register, as specified by the drive designer.

It does not reset any of the following:

The Control register (except the GO bit is reset as usual at the completion of this command)
The Drive Type register
The Look-Ahead register
The Current Cylinder Address register
The Serial Number register
The HCI, ECI bits in the Offset register
The FMT22 bit in the Desired Cylinder Address register
The DMD bit in the Maintenance register
The Status register (except ATA and ERR bits).

No Attention condition is generated by this command.

6.2.1.8 Release (13(base 8))

This command is meaningful only on Dual Controller drives (see 8.1).

When executed, this command causes the drive to switch to the neutral state (if no request from the other controller was pending) or to the other controller (if a request was pending).

No Attention condition is generated by this command, on the controller which initiated the command.

6.2.1.9 Offset (15(base 8))

For moving-head disks; causes the drive to displace the heads a small amount (25-1575 microinches for RP04) from cylinder centerline. This is used in recovery of data which may have been recorded off the centerline.

The Offset amount and direction are controlled by bits 0-7 of the Offset register.

Performing an Offset command with an Offset amount of 0 specified in the Offset register does not necessarily guarantee that the heads are returned to cylinder centerline. (see 6.2.1.10)

6.2.1.10 Return to Centerline (17(base 8))

For moving-head disks; causes the drive to reposition the heads to the centerline of the cylinder currently being addressed (see 6.2.1.9). Bits 0-7 of the Offset register are reset.

6.2.1.11 Read-In Preset (21(base 8))

Causes the drive to be placed in a known state. Refer to Drive documentation for details of specific drive actions (which registers are cleared, bits which are set, etc.).

6.2.1.12 Pack Acknowledge (23(base 8))

For moving head disks; sets the VV bit for the command controller. This command must be issued before any data transfer or positioning commands can be given if MOL changes state.

6.2.1.13 Erase (25(base 8))

For magtape; the drive writes forward on the tape, erasing, until the End-Of-Tape mark is encountered.

6.2.1.14 Write File Mark (27(base 8))

For magtape; causes a file mark record to be written.

6.2.1.15 Search (31(base 8))

For disk drives, the current sector field of the Look-Ahead register is compared to the desired sector field of the Desired Sector/Track Address register. When they are equal, the command execution is complete.

The purpose of this command is to provide automatic angular position sensing in the drive, with an Attention signal being used as a signal that a desired position has been reached. The desired sector field would normally be set by software to one or more sectors ahead of the sector in which a data transfer is to take place.

6.2.1.16 Space Forward (31(base 8))

For magtape; the tape is moved forward over one record.

6.2.1.17 Backspace (33(base 8))

For magtape; the tape is moved backward over one record.

6.2.2 Data transfer commands

These commands, when issued to a drive, cause the drive to attach to the data bus. The timing restrictions for transfers are described in section 4.

Interrupts to the CPU are generated by the controller, as desired by the controller designer. Attention occurs only on error detected by the drive (see section 7).

6.2.2.1 Write Check...(51(base 8)-57(base 8))

Each of the commands in this group causes the drive to perform exactly the same action as the corresponding Read command. The difference is that the controller, instead of transmitting the data to memory, reads data from memory and compares the disk data and memory data, word for word.

6.2.2.2 Write data (61(base 8))

For disks; this command causes transmission of data from memory to the data field of the disk sector.

If the disk has a header field, the validity of the header is checked by the drive before data transmission begins for each sector.

If the disk is a moving-head disk, a seek to the cylinder indicated in the Desired Cylinder Address register will be performed automatically if the heads are not already positioned there (the "implied seek" feature).

6.2.2.3 Write forward (61(base 8))

For magtape; this command causes transmission of data from memory to a tape record.

Tape records are always written in the forward direction, but can be Read in either direction.

6.2.2.4 Write header and data (63(base 8))

For disks with header fields; this is the "format" command, which creates formatted header records. The drive expects to receive 4 words of format information, to be written into the header field, plus the usual number of data words, which will be written with the data field, for each sector. The first two words of the header are to contain bits which match the contents of the Desired Cylinder Address register and the Desired Sector/Track Address register in that order (see section 10).

The header field is not checked before writing. The "implied seek" will occur if required.

6.2.2.5 Read data (71(base 8))

For disk drives; this command causes transmission of data from the data field of the sector to memory.

If the disk has a header field, the validity of the header is verified before data transmission begins for each sector.

The "implied seek" will occur if required.

6.2.2.6 Read forward (71(base 8))

For magtape; this command causes transmission of data from a tape record to memory.

6.2.2.7 Read header and data (73(base 8))

For disks with header fields; this is the only command that allows the header field to be seen by software. The header, followed by the data field, is transmitted to memory.

The validity of the header field is checked before the data field is transmitted. However, the header and the data field are always transmitted.

The "implied seek" will occur if required.

6.2.2.8 Read reverse (77(base 8))

For magtape; this command causes transmission of data from a tape record to memory, but the direction of motion of the tape is reversed from normal.

Because the order of words transmitted from the drive is reversed from the order in which they were written, they will be written into memory in the reverse order unless the controller also reverses its sequence of memory access.

7. ERROR HANDLING

7.1 INTRODUCTION

7.1.1 Massbus Drive errors are grouped into two categories, "Class A" and "Class B", according to when they are to be handled.

7.1.2 A "Class A" error is defined to be a Drive error which can be handled at the completion of a non data transfer command or, in the case of a data transfer command, at a convenient block boundary. Controllers typically terminate data transfers at the end of the current block (but additional blocks may be transferred).

7.1.3 A "Class B" error is defined to be a Drive error which must be handled immediately; the occurence of a Class B error causes the drive to terminate command execution as soon as possible.

7.1.4 The responsibility for defining the set of events which constitutes a given error shall lie with the drive designer. Errors which are Class A do not normally disrupt the successful transmission of data. The drive designer should insure that, when defining a Class B error, he structures that definition so that the termination of the command execution occurs at an appropriate point in that command execution.

7.1.5 The responsibility for the classification of specific drive errors as "Class A" or "Class B" shall ultimately lie with the drive designer.

7.2 USE OF ATTENTION (ATTN)

7.2.1 Massbus drive errors are signalled to the controller with the ATTN line.

7.2.2 If an error occurs while the drive is ready (DRY=1), the drive both asserts ATTN and sets ATA immediately.

7.2.3 If an error occurs while a command execution is in progress (DRY=0), the drive asserts ATTN and sets ATA when the drive becomes ready (when DRY is set) at the termination of the command execution.

7.3 USE OF EXCEPTION (EXC)

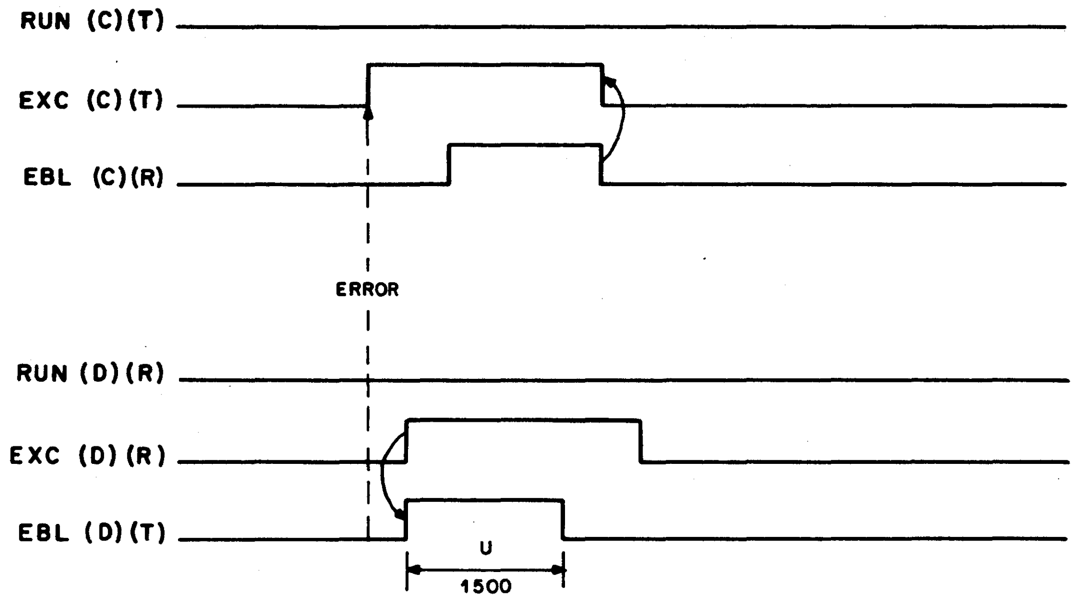
7.3.1 The EXC line is asserted by Massbus drives when they detect an error during a data transfer operation. See sections 7.4.3 and 7.5.3 for a detailed description of when EXC is asserted by a drive.

7.3.2 EXC may be asserted by a Massbus controller (if it has negated RUN) in order to terminate a data transfer operation. The drive doing the data transfer responds by asserting EBL and terminating the data transfer.

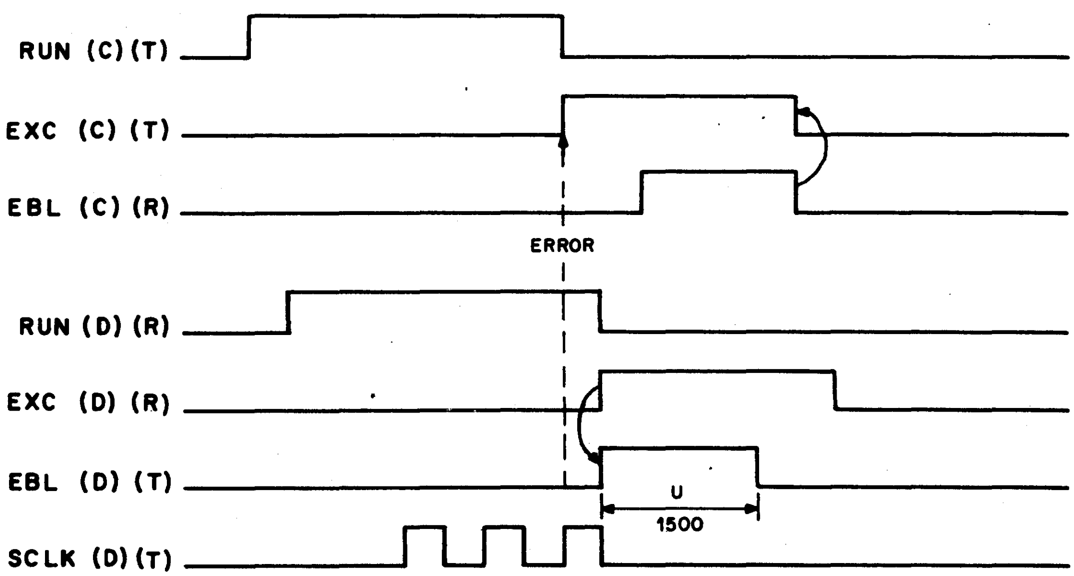
Note 1: There are cases in which a drive may not recognize a controller's assertion of EXC. Thus, it is not always possible for a controller to cause an immediate termination of a data transfer by asserting EXC.

Note 2: No error is set and the ATA bit is not set when a data transfer is aborted by the controller (by asserting EXC).

FIGURE 7.3.2.1



A) CONTROLLER ERROR,BEFORE RUN ASSERTION



B) CONTROLLER ERROR,AFTER RUN ASSERTION

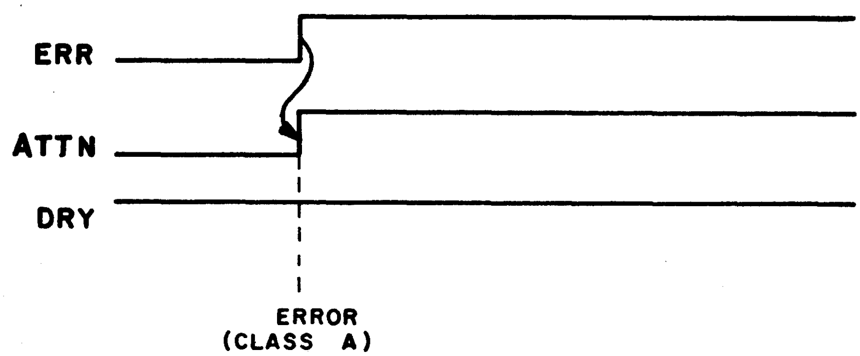
(C)= AT THE CONTROLLER (T)= TRANSMITTING U= UNSPECIFIED
(D)= AT THE DRIVE (R)= RECEIVING

11-2743

7.4 CLASS A ERROR HANDLING PROTOCOL

7.4.1 Class A errors which occur while the drive is ready cause ATTN to be asserted and ATA to be set upon detection of the error (see figure 7.4.1.1).

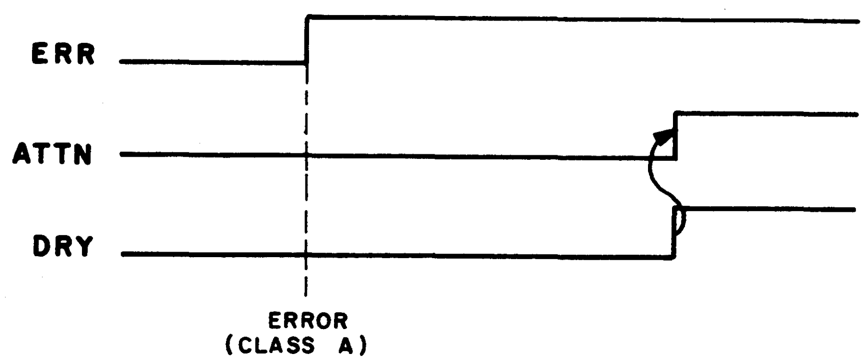
FIGURE 7.4.1.1



DRIVE ERROR WHILE DRIVE IS READY

7.4.2 Class A errors which occur while the drive is performing a non data transfer operation are signalled at the completion of the command execution, when DRY becomes asserted. After DRY has been asserted, the drive asserts ATTN and sets ATA (see figure 7.4.2.1).

FIGURE 7.4.2.1



DRIVE ERROR WHILE DRIVE IS PERFORMING
NON-DATA TRANSFER OPERATION

11-2745

7.4.3 Class A errors which occur while the drive is performing a data transfer operation are handled according to the following sequence: (see fig. 7.4.3.1 for a flowchart; see fig. 7.4.3.2 for timing diagrams)

1. The selected drive asserts EXC. If the error occurs before RUN has been asserted, the drive waits until it receives the RUN assertion. It then asserts EXC.

   If the error occurs during the EBL pulse, it may be handled in one of two ways:

   A) EXC is immediately asserted. The drive designer must insure that the next negation of EBL occurs no less than 1500 nanoseconds after the assertion of EXC (so that both EXC and EBL are asserted together for at least 1500 nanoseconds).

   B) The error is not signalled until the end of the data transfer. When DRY becomes asserted, the error is handled as described in sec. 7.4.1.

      Note: If the error affects the validity of the data which was just transferred, alternative B is not permitted. See sec. 4.2.6.4.

2. The controller, on receiving the assertion of EXC, decides whether or not to continue the data transfer. If the controller decides to stop, it negates RUN. If it decides to continue, it leaves RUN asserted.

3. The drive continues the data transfer until the next EBL pulse occurs. At the trailing edge of EBL, the drive inspects the RUN line. If RUN is asserted, the drive begins transferring the next block of data (leaving EXC asserted). If RUN is negated, the drive terminates the operation.

4. When DRY becomes asserted (at the trailing edge of the final EBL pulse), the drive negates EXC, asserts ATTN, and sets ATA. The cause of the error should be clearly indicated by the state of the various error bits in the drive.

FIGURE 7.4.3.1

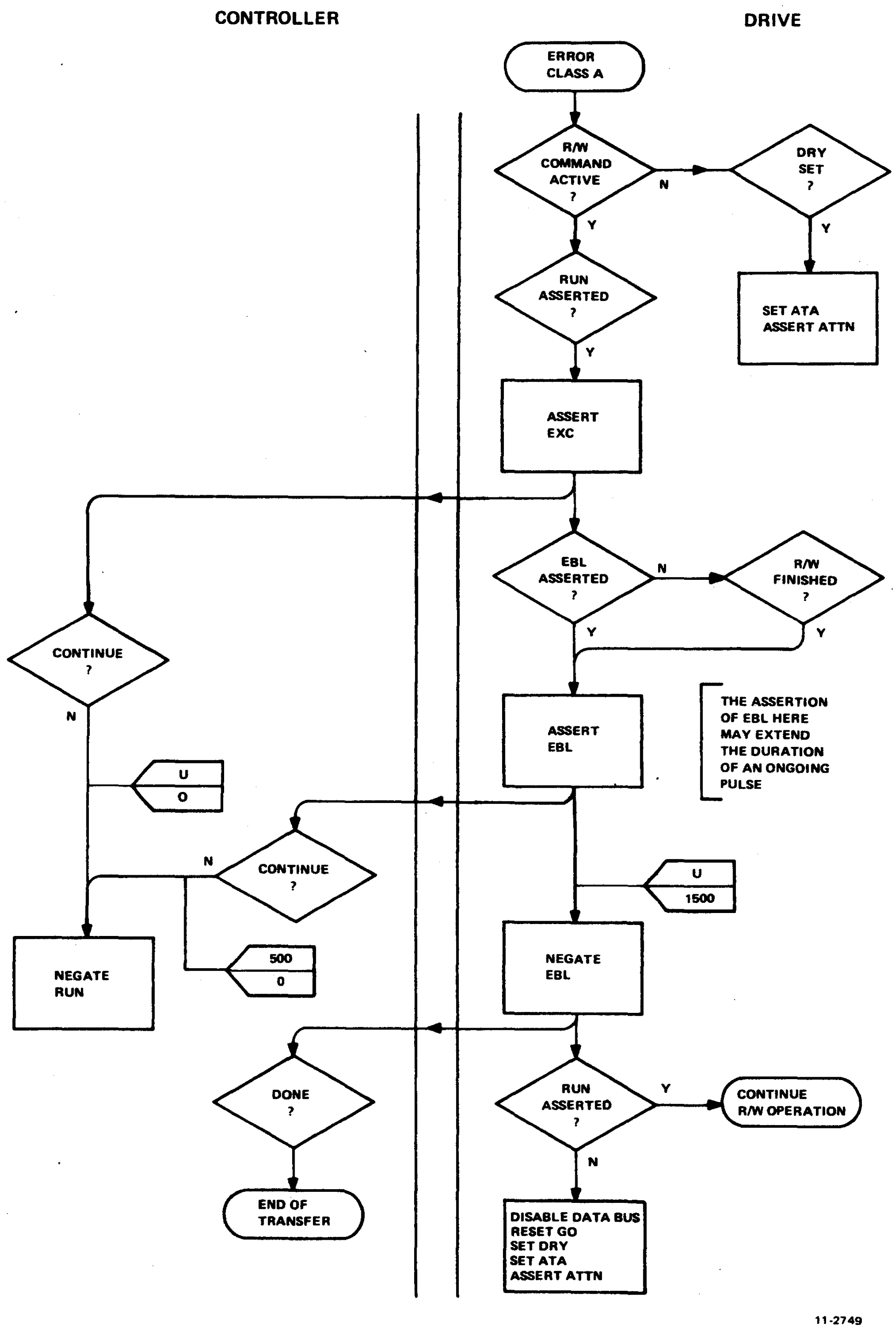
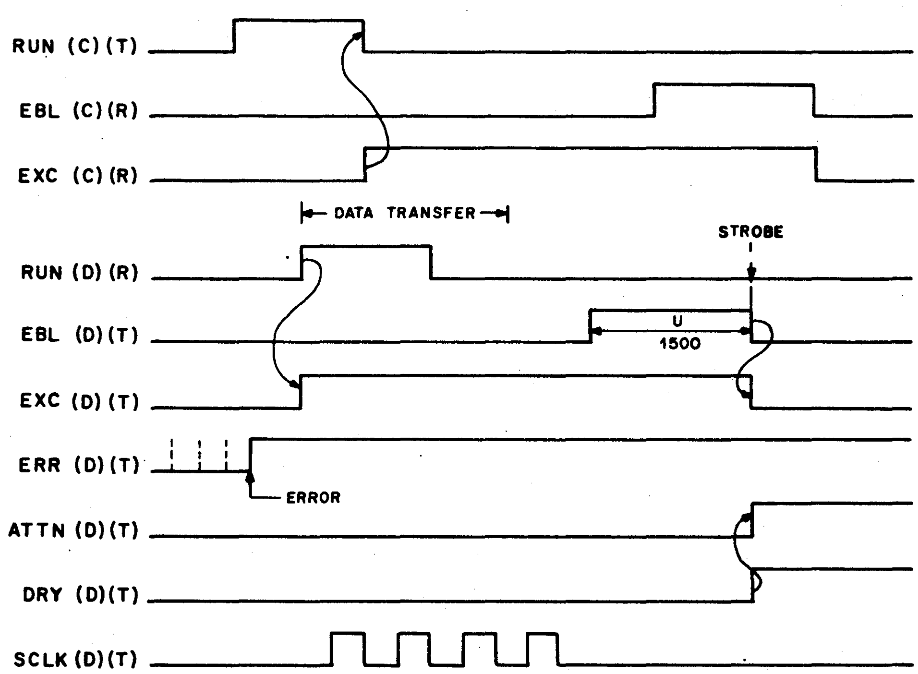
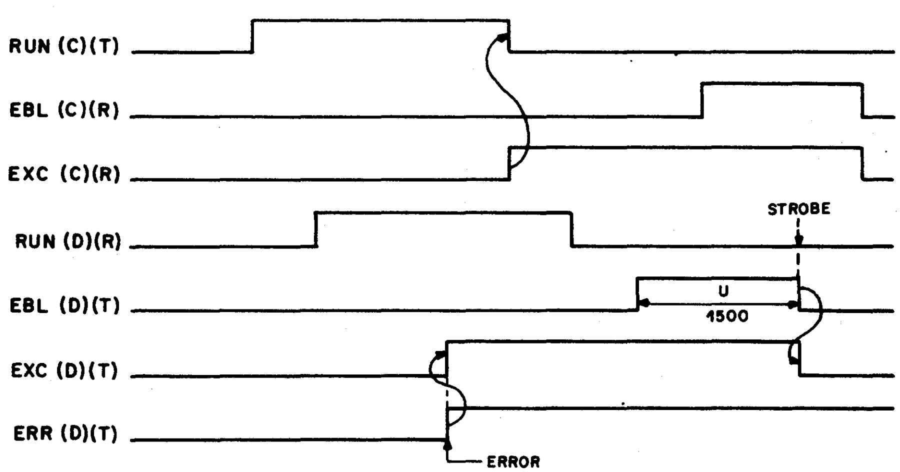

CONTROLLER
DRIVE
ERROR CLASS A
R/W COMMAND ACTIVE ?
N
Y
DRY SET ?
Y
SET ATA ASSERT ATTN
RUN ASSERTED ?
Y
ASSERT EXC
EBL ASSERTED ?
N
R/W FINISHED ?
Y
Y
CONTINUE ?
N
ASSERT EBL
THE ASSERTION OF EBL HERE MAY EXTEND THE DURATION OF AN ONGOING PULSE
U
0
CONTINUE ?
N
U
1500
500
0
NEGATE RUN
NEGATE EBL
DONE ?
RUN ASSERTED ?
Y
CONTINUE R/W OPERATION
N
END OF TRANSFER
DISABLE DATA BUS
RESET GO
SET DRY
SET ATA
ASSERT ATTN
11-2749

FIGURE 7.4.3.2



A) CLASS A ERROR-BEFORE RUN ASSERTION



B) CLASS A ERROR-AFTER RUN ASSERTION,BEFORE EBL ASSERTION

(C)= AT THE CONTROLLER (T)= TRANSMITTING U= UNSPECIFIED
(D)= AT THE DRIVE (R)= RECEIVING

11-2744

7.5 CLASS B ERROR HANDLING PROTOCOL

7.5.1 Class B errors which occur while the drive is ready (DRY=1) cause ATTN to be asserted and ATA to be set upon recognition of the error (see fig. 7.4.1.1).

7.5.2 Class B errors which occur during a non data transfer operation should cause the command execution to terminate as soon as possible. When DRY becomes asserted, ATA is set and ATTN is asserted (see figure 7.4.2.1).

7.5.3 Class B errors which occur during a data transfer operation should be handled according to the following sequence (see fig. 7.5.3.1 for a flowchart; see fig. 7.5.3.2 for timing diagrams):

1. Command execution should immediately terminate (it is, however, the choice of the drive designer to define exactly when an error "occurs").

2. The drive asserts EXC and EBL.

   Note: If the Class B error occurs while SCLK is asserted, the drive designer must insure that the error does not cause SCLK to "glitch" (to have an arbitrarily short duration). This may be avoided by delaying the assertion of EXC and EBL until SCLK is negated or by holding SCLK asserted until EBL is negated.

3. After insuring that EXC and EBL have been asserted together for at least 1500 nanoseconds, the drive negates them, disconnects from the Data Bus, and becomes ready (DRY=1) if possible.

FIGURE 7.5.3.1

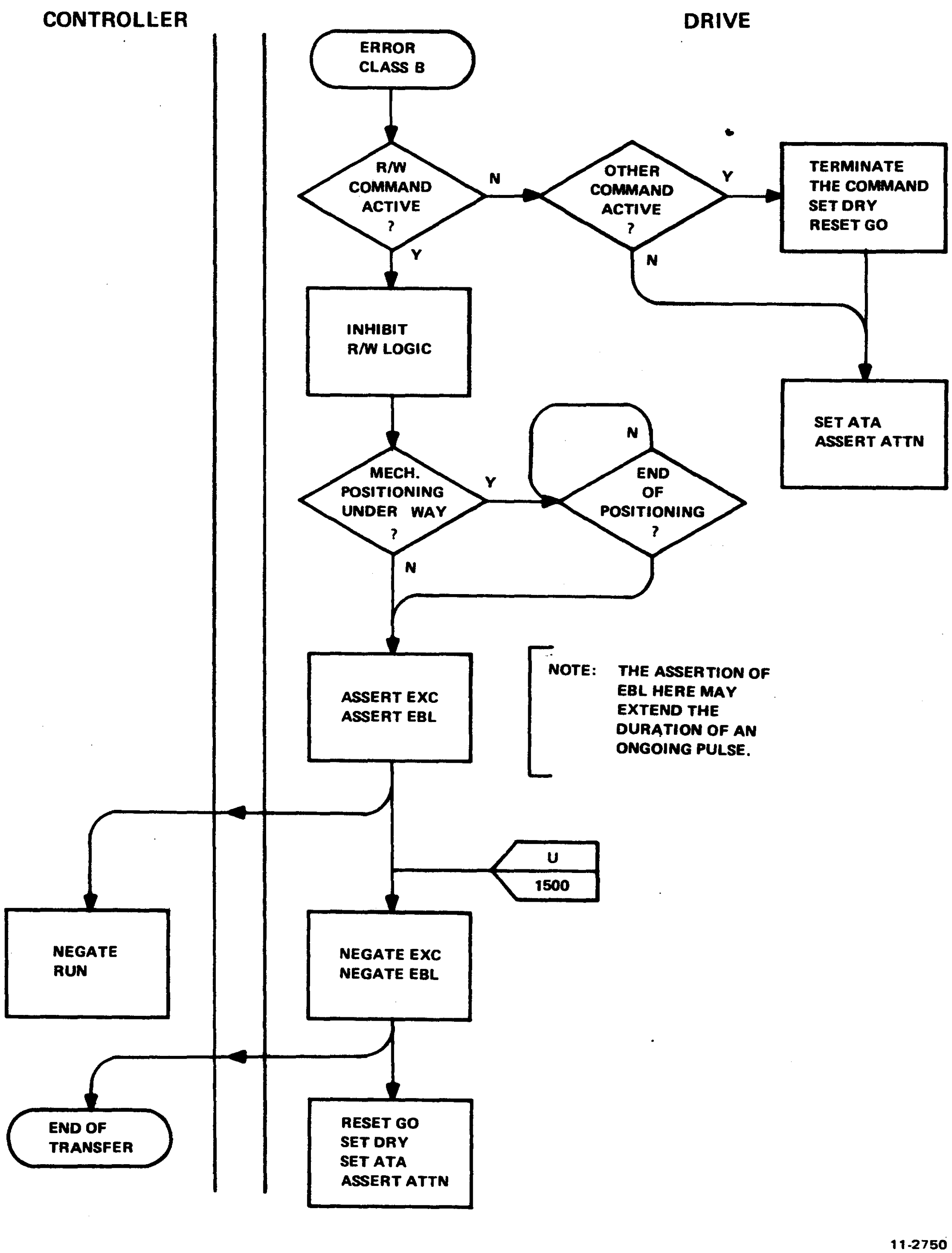
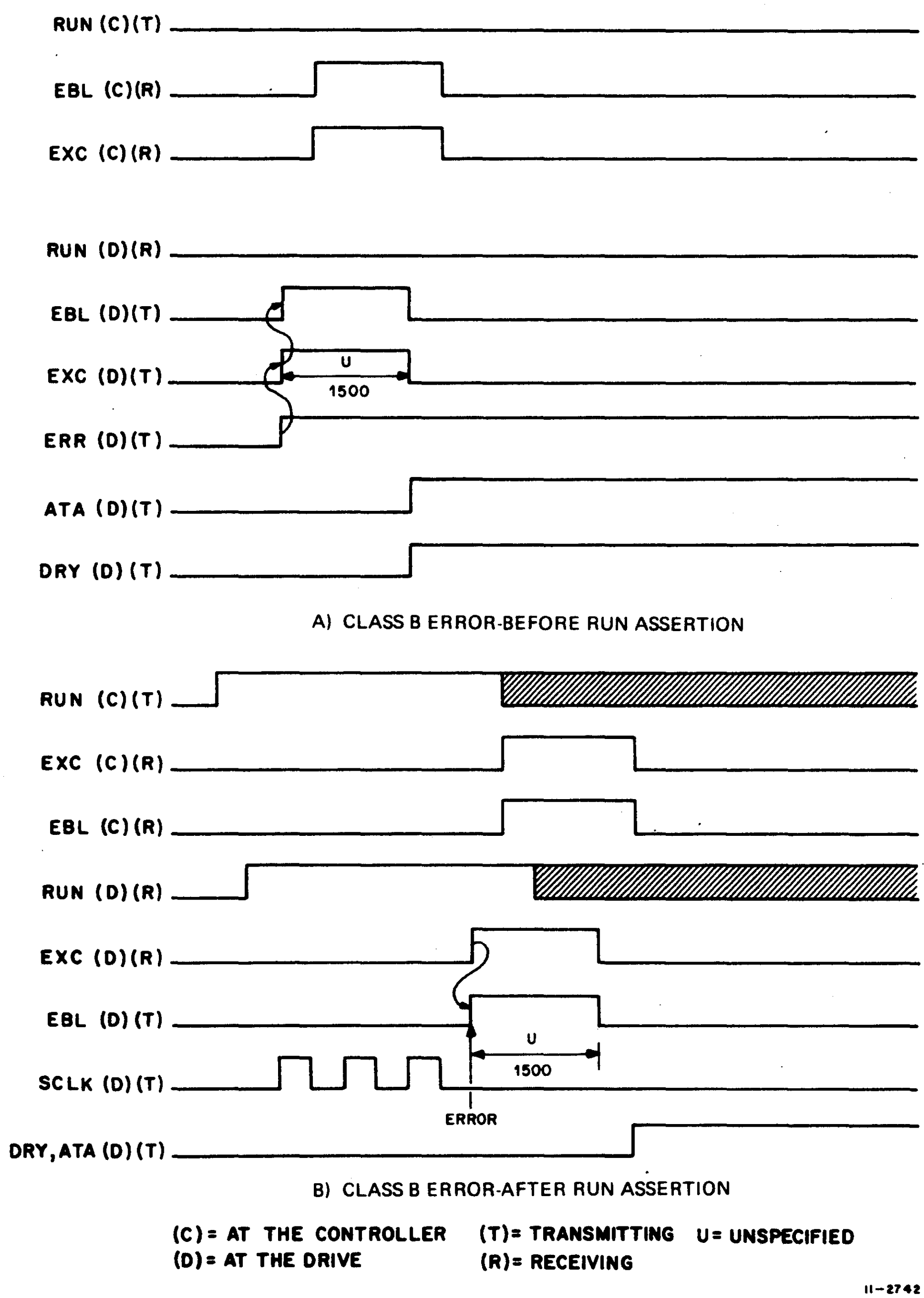

CONTROLLER
DRIVE
ERROR CLASS B
R/W COMMAND ACTIVE ?
N
Y
OTHER COMMAND ACTIVE ?
Y
N
TERMINATE THE COMMAND SET DRY RESET GO
INHIBIT R/W LOGIC
SET ATA ASSERT ATTN
N
MECH. POSITIONING UNDER WAY ?
Y
END OF POSITIONING ?
N
ASSERT EXC ASSERT EBL
NOTE: THE ASSERTION OF EBL HERE MAY EXTEND THE DURATION OF AN ONGOING PULSE.
U
1500
NEGATE RUN
NEGATE EXC NEGATE EBL
END OF TRANSFER
RESET GO SET DRY SET ATA ASSERT ATTN
11-2750

FIGURE 7.5.3.2


RUN (C)(T)
EBL (C)(R)
EXC (C)(R)
RUN (D)(R)
EBL (D)(T)
EXC (D)(T)
U
1500
ERR (D)(T)
ATA (D)(T)
DRY (D)(T)
A) CLASS B ERROR-BEFORE RUN ASSERTION
RUN (C)(T)
EXC (C)(R)
EBL (C)(R)
RUN (D)(R)
EXC (D)(R)
EBL (D)(T)
U
1500
SCLK (D)(T)
ERROR
DRY, ATA (D)(T)
B) CLASS B ERROR-AFTER RUN ASSERTION
(C) = AT THE CONTROLLER
(D) = AT THE DRIVE
(T) = TRANSMITTING
(R) = RECEIVING
U = UNSPECIFIED
11-2742

7.6 EXAMPLES OF CLASS A ERRORS

7.6.1 The examples given in this section describe how existing Massbus drives handle these Class A errors. Each of these examples deals with an error which is flagged in Massbus register 02 (Error 1 register; See sec. 5.3.3). Because all Massbus drives contain this register, each of these examples represents an error condition which every Massbus drive must handle. While each drive designer must define each error condition within the context of the drive he is designing (see Sec. 7.1.4), these examples may be used as a guide in arriving at that definition.

7.6.2 Illegal Register (ILR)

This error occurs when the device control logic decodes a nonexistent register address from the Register Select lines (RS<0:4>). The error bit (Error 1 register, bit <01>) is cleared by a Drive Clear, Massbus INIT assertion, or by writing 0's into the Error 1 register. Attempting to write into a read only register will not cause the ILR error to occur. The bits received will be ignored and no other errors will be flagged.

This error is handled in the sequence described in Sec. 7.4.

7.6.3 Register Modification Refused (RMR)

This error occurs when a write is attempted into any register (except the Attention Summary or Maintenance Registers) during an operation. It is cleared by Drive Clear, Massbus INIT assertion, or by writing 0's into the register.

This error is normally handled as described in sec. 7.4; however, the RS04 Drive, when performing a Search command, aborts the command execution upon detecting an RMR error. This happens because the ATTN line is asserted at the successful completion of a Search command; if RMR did not abort the execution, the controller would not recognize the error.

7.6.4 Massbus Parity Error (PAR)

This error occurs when a parity error is detected while writing into a register on the Control bus or sending data to the drive during a write operation.

When a Control Bus parity error is detected during a write into a drive register, the register should be loaded, and then PAR (or the appropriate error indicator), ERR, and ATA are set. In the special case of a write into the Control Register

(00), the register is loaded, but detection of a Control Bus parity error should inhibit the setting of the GO bit, and no command execution should occur.

Except as noted above, this error is handled as described in sec. 7.4.

7.7 EXAMPLES OF CLASS B ERRORS

7.7.1 The examples given in this section describe how existing Massbus drives handle these Class B errors. Each of these examples deals with an error which is flagged in Massbus register 02 (Error 1 register). Because all Massbus drives contain this register, each of these examples represents an error condition which every Massbus drive must handle. Although each drive designer must define each error condition within the context of the drive he is designing (see Sec. 7.1.4), these examples may be used as a guide in arriving at that definition.

7.7.2 Illegal Function Error (ILF)

This error occurs when the GO bit is set while the function code in the Control register does not correspond to an implemented command on the selected drive.

This error is handled as described in Sec.7.5.

The ILF bit in the Error 1 register (Bit 0) will be reset when a Drive Clear command (function code 11) or an INIT pulse is received.

7.7.3 Format Error (FER)

This error occurs when the prerecorded (during pack formatting) "flag" bit on the header is not equal to the corresponding flag bit in the "offset" register. This generally implies that the wrong medium has been mounted on a Massbus device.

This error is handled as described in Sec. 7.5.2.

7.7.4 Invalid Address Error (IAE)

This error occurs when the address in the "Desired Cylinder Address" and "Desired Sector/Track Address" registers is invalid. The Illegal Address Error will occur only if the GO bit is set and the function is one which makes use of the "Desired Block Address" register. It does not occur, for example, on either the "Seek" (05) or "Return to Zero" (17) functions.

8. OPTIONS

8.1 DUAL CONTROLLER

8.1.1 Two independent Massbus controllers may share a single Massbus drive if that drive has implemented a dual controller option.

8.1.2 A drive which implements a dual controller option will have three states (with respect to its two Massbus Controllers):

A. Connected to controller (A)

B. Connected to controller (B)

C. Neutral (not connected to either controller)

While in the neutral state, the drive is not connected to either controller, but its registers can be read by either controller.

When the drive is connected to controller B, only controller B can operate on the drive's registers (except for the Attention Summary pseudo-register).

Similarly, when the drive is connected to controller A, only controller A can operate on the drive's registers (except for the Attention Summary pseudo-register).

8.1.3 A dual controller drive implements a manual switch with three positions: "A", "B", and "A/B". The "A" and "B" positions override the programmable switching by locking the drive to one controller. The "A/B" position allows the drive to be connected to either A or B under program control or to rest in the Neutral state. When a drive is connected to controller A under program control (with the manual switch set to "A/B"), it is said to be seized on A. When a drive is connected to controller A by setting the manual switch to "A", the drive is said to be "switched" to A.

Note: The manual switch takes effect only while the drive is being powered up. It is ignored at all other times.

8.1.4 While a drive is in the neutral state, an INIT pulse will cause ATA to be reset on only the controller from which the INIT came (there is really a separate flipflop for each controller). A drive which is connected to a controller will ignore any INIT pulses from the other controller.

8.1.5 While the switch is set to A/B, a change of state of the MOL bit will cause both ATA bits to be set (except on an unload command). However, a drive whose switch is in the A position will not set the ATA bit of controller B when MOL changes state, and vice versa.

8.1.6 A drive which is seized on one controller will return to neutral (not seized on either controller) if after one second no read or write commands have been issued and there is no request pending from the other controller.

9. HARDWARE DESIGN NOTES

9.1 INTRODUCTION

9.1.1 This section is primarily a collection of Massbus folklore. It is intended to inform drive designers of some of the hardware issues which have been noted and resolved up to this time.

9.2 NOTES ON MASSBUS PROTOCOL

9.2.1 When performing a data transfer command, it should be noted that RUN may already be asserted when the GO bit becomes set.

9.2.2 The OCC line in the Data Bus section (see section 4.2.8) is to be asserted by the drive during the entire data transfer. OCC is asserted as soon as a valid data transfer command is recognized and accepted. It is negated at the trailing edge of the last EBL pulse. The assertion of OCC does not depend on receiving RUN assertion.

9.2.3 The RH11 Massbus controller decrements the Unibus Address register (RHBA) on READ/WRITE CHECK REVERSE (57,77) commands. These commands cause words to referenced "backwards" in memory while tape motion is "backwards".

9.2.4 The RS04 fixed head disk drive aborts a Search command execution immediately upon the detection of an RMR (Register Modification Refused) error condition. It immediately sets DRY, RMR, ERR, and ATA. See section 7.6.3 for additional information.

9.2.5 When a Read or Write is done from or to the last addressable block on the medium (for example, block 07777(base 8) on a drive with 4096 blocks), the Last Block Transferred (LBT) bit is set at the end of the transfer. This allows the controller or CPU to implement "spiral" reads and writes by incrementing the drive number for the next block, when the LBT condition is detected. The LBT bit is reset whenever a new Read or Write command is started. The LBT condition does not cause an Attention condition. Therefore, the LBT bit must be actively sampled in order to implement spiral reads, writes, etc.

9.2.6 If a command code other than Drive Clear is written into the Control Register while any error bit is set, only the following will happen:

1. The command code is loaded.

2. The GO bit is reset.

3. The ATA bit is set (if not already set).

9.2.7 In general, a channel (such as on DECsystem-10) may signal "complete" to the controller some time after the EBL pulse. The controller has kept the RUN line asserted, so that the drive is going to continue the transfer. The controller then negates the RUN line and asserts EXC in order to abort the continuation. The drive responds with another EBL pulse. This can normally be completed before the first data word of the next sector is transferred. In transferring the last addressable sector, however, the AOE bit may be set before the channel has told the controller to stop. The error is spurious, because the channel in fact knows that the transfer is complete.

9.3 NOTES ON MASSBUS TIMING

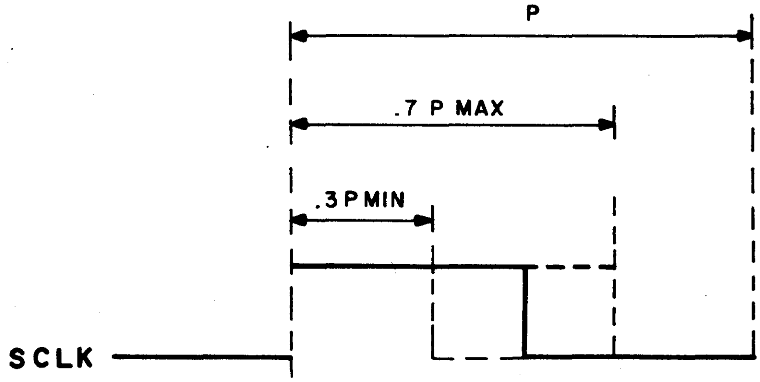
9.3.1 Drive designers should insure that no serious problem occurs if the DEM line is negated prematurely. Current drives currently ignore DEM except as a handshake initiator until they assert TRA. In no case do they place data on or strobe data from the Control bus lines (C<00:15>) unless DEM is asserted.

9.3.2 The safest way to insure the validity of Control Bus transfers is to start the deskew for reads and writes only after receiving the TRA negation from the previous transfer.

9.3.3 This specification specifically does not require SCLK pulses to be evenly distributed, but this was intended to allow elongation of the interval rather than contraction of it, relative to the nominal data rate. It seems that good design practice requires that a drive never issue SCLK pulses closer together than the nominal (burst) data rate of that drive.

This suggests that the minimum duration of SCLK assertion and negation should be expressed in terms of the nominal burst data period of the drive. Let P be the nominal burst data period. Then the minimum SCLK assertion duration should be ".30 p or 225 nanoseconds, whichever is larger". This guarantees that the duty cycle of the SCLK pulse is not worse than 70% to 30% in either direction. See sec. 4.3 for further details; figure 9.3.3.1 shows a typical SCLK assertion.

FIGURE 9.3.3.1



P=500 NANOSEC OR NOMINAL BURST DATA PERIOD,
WHICHEVER IS GREATER

11-2746

9.3.4 Section 3.6.2 of this specification requires the INIT pulse, when it occurs, to be at least 400 nanoseconds in duration. This duration was chosen to be long enough to assure that circuits in all drives have time to respond to it. It was then proposed that the minimum width be extended to 1500 nanoseconds, in order to allow time for the reset function to be completed in all drives before the controller tried to initiate a command. The extended time was intended only as a restriction on the controller. We now agree that it is sufficient to restrict the controller, so that it does not attempt to initiate a command until at least 1500 nanoseconds after the INIT signal has been asserted. The INIT pulse minimum width thus remains at 400 nanoseconds.

9.3.5 The timeout in the controller should occur in two phases:

1. After RUN is asserted, there is a timeout if no SCLK pulse is received within 250 milliseconds.

2. After as many SCLK pulses are received as are desired by the controller, and RUN is negated, there is a timeout if EBL is not eventually asserted and negated.

Note: Both phases of the timeout should be inhibited while OCC is asserted.

9.3.6 The RP04 will have a nominal response time of 600 nanoseconds, with a worst case time of 700 nanoseconds (DEM assertion to TRA assertion).

9.3.7 The drive designer should explicitly specify how long it takes his particular drive to execute a Drive Clear command.

9.4 NOTES ON THE USE OF ATA (ATTENTION ACTIVE)

9.4.1 A drive is normally expected to set ATA whenever it comes on line or goes off line (the MOL bit changes state). In powering up the logic of a drive, we assume that it is making the transition from off line to on line, even if "power off" is the only offline state. Therefore, ATA should be set on power up, unless there is some other condition which keeps the drive off line. On power down, ATA should also be set and held as long as possible, both because of the online to offline transition (if the drive was on line) and because of the Unsafe (UNS) error condition becoming set.

9.4.2 If the controller asserts EXC during a data transfer and there is no error in the drive, the drive aborts the transfer, but does not create an error due to receiving EXC assertion, and does not set ATA. See sec. 7.3.2.

9.4.3 While ERR is reset and DRY is set, all valid function codes are acceptable. If a valid function code (with the GO bit set) is loaded into the drive Control register while ATA is set, the drive should reset ATA and proceed with the command execution.

9.4.4 In the case of a persistent error in a drive which cannot be cleared by a Drive Clear command or even by INIT, it is highly desirable to be able to clear ATA, so that this drive does not cause perpetual interrupts, and thus block other drives on this Massbus. Thus, it is necessary to allow resetting of the ATA bit (by writing a 1 into the Attention Summary register) even while ERR is set.

9.4.5 Since all drives are affected by a write to the Attention Summary register, and since it is desirable to be able to affect the ATA bits of non busy drives at any time, a busy drive (one which has its DRY bit negated) must be prepared to allow such a write to occur without causing an RMR error.

It is apparent that ATA is never set in a busy drive. Therefore, the busy drive does not have to inspect the bit being written because it cannot affect ATA. It should also be noted that a drive which is performing a data transfer can ignore a parity error which occurs on a write to the Attention Summary register, again because such a write cannot affect the ATA bit (which must be in a negated state).

Note: It is acceptable for a drive not to assert TRA in response to a write to the Attention Summary register.

10. PROGRAMMING NOTES

10.1 INTRODUCTION

10.1.1 This section is a collection of notes and examples which should prove useful to those persons who are concerned with the programming and general software implications of the Massbus.

10.2 GENERAL IMPLICATIONS OF THE MASSBUS FOR SOFTWARE

10.2.1 The Massbus Interface standard has several effects on the design of devices which comply with it. A device normally must contain all the necessary logic and registers to carry out various functions without the supervision of the Massbus controller. For a disk these various functions include Seeks, Searches, disk address incrementing for spiral data transfers, and automatic error recovery operations. Because the devices are "smarter", there are certain implications for how the software should be written.

1. Operations once started in a drive can not be cleanly aborted.

2. The typical controller clear or initialize command issues a Massbus INIT. A Massbus INIT causes all devices on the Massbus to be initialized. The initialize performs the same actions as a Drive Clear command in all drives. Therefore, this should only be done at system start up time or in the most severe error cases (e.g., the subsystem is hung up).

3. Because the Massbus is to be used with a variety of mass storage devices, the function codes have been placed in groups. Therefore, their values and structures are different from older devices.

4. Because the spiral I/O logic (incrementing the desired disk address) is in the drives, automatic overflow from one device to another is normally not possible at the hardware level.

5. System throughput can be optimized to a very fine degree. This is due to the ability to interrogate and initiate positioning functions in all devices even while the controller and one of its devices is busy with a data transfer.

6. Since the currently selected unit in the controller can be changed at any time, the software must take care not to lose the unit number of the device performing a data

transfer. This is of particular importance when writing interrupt handling routines. The Unit Select bits should normally be saved and restored.

7. In the general case, more than one type of Massbus device may be on the Massbus. For example, one Massbus controller may be controlling a mixture of RS04's (fixed head disks), RP04's (moving head disks), and TU16's (magtape).

8. Massbus devices have a much higher data rate than similar, older devices. It is important for system designers to consider the possible interference and interaction between I/O and CPU in memory access.

9. Most moving head disks will have a header record which precedes the data record in each sector. The length of the header will be four words. The first word will contain the cylinder address, in the same format as the Desired Cylinder Address register. The second word will contain the block address, in the same format as the Desired Block Address register. The third and fourth words may contain (a) a repetition of the first and second word, (b) CRC words, or (c) other unrelated information.

   The drive will read and examine the first two words of the header during Read Header, Read Data and Write Data operations. The drive compares the first two words of the header against the cylinder and block addresses and gives the Header Compare Error (HCE) if they do not agree. The Read Header command will cause all four words to be transmitted to the controller. The Write Header command will always expect four words to be sent from the controller. The drive does not check the header words being written.

   For more information on the use of the header record, please refer to the documentation of the specific drive being programmed.

10.3 PROGRAMMING NOTES ON MASSBUS TIMING

10.3.1 It takes at least one (1) microsecond to clear a slave or to initialize all slaves. This could cause a problem if the program assumes that the drive is ready to accept a command immediately after the Drive Clear command is accepted.

10.3.2 If data overrun occurs in the RH10, it may assert EXC. This will occur not later than four microseconds after the EBL pulse trailing edge (at the controller), and not before the EBL pulse leading edge (at the controller).

The RH10 depends on having at least five microseconds between EBL pulse trailing edge and the next sector decision point (see sect. 4.7.2) The RH10 may also assert EXC during a read operation, at any time. It is most likely to occur during the gap time but not after the first SCLK pulse.

10.3.3 The following conditions are known to cause a controller timeout (due to no OCC assertion). In the RH11, this timeout causes the MXF error.

1. A parity error occurs, detected by the drive, while writing a data transfer command into the drive Control Register.

2. An illegal data transfer command is written into the drive Control register.

3. A data transfer command is written into the drive Control register while ERR is asserted in the drive.

4. When a data transfer command is issued, the drive is either offline, busy, or connected to another controller.

10.4 PROGRAMMING NOTES ON MASSBUS COMMANDS

10.4.1 The TU16 Error register (02) is now a read only register. Since all error bits can be set by a suitable sequence of commands, the ability to write the error register is unnecessary for maintenance purposes. INIT and Drive Clear commands will still clear the Error Register.

10.4.2 In Massbus devices, DRY is negated whenever the drive is offline, in Standby, or is otherwise unable to accept a command. Thus, issuing a command while offline is treated the same as issuing a command while busy; the RMR error is set, but no other action is taken until DRY is asserted. If, when DRY is asserted, an error is present (ERR=1), ATA will be set. This treatment seems to be consistent and adequate.

10.4.3 If offsetting is used to recover data from cylinder 0 the RP04, it is possible that a subsequent "bootstrap load" sequence will read with the heads still offset from centerline. This can occur if a recovery routine left the heads in an offset position, and then the sequence of: INIT pulse, Readin Preset command, and Read Data command is issued. Since the heads are returned to track centerline only by the Return to Centerline command and by a seek to a different cylinder, the offset could remain. It is therefore recommended that error recovery routines which use offsetting always execute a Return to Centerline command at the end of the recovery sequence.

10.4.4 The Readin Preset command causes the Tape Control register of the TU16 to set for: slave 0, 556 bpi, odd parity; a rewind operation is initiated. This is for use by the RH10 controller. Whenever DMD (Diagnostic Mode) is set, the MOL bit is also set.

10.4.5 While Drive clear is being executed, DRY is negated, with all that that implies.

10.4.6 Loading a "1" into the Ignore Frame Count (IFC) bit of the TU16 will cause an entire record to be read from tape regardless of the contents of the frame counter and will suppress setting of FCE on read operations. Ignore Frame Count is meant to be used for readin mode only, since setting it does not inhibit the check for FCL=1 before a Read Fwd.

10.4.7 Note that the Write Lock (WRL) status bit is asserted in current magtapes during rewinds. Also, the RS03 will assert WRL whenever a track is selected (Desired Block Address) which has its Write Inhibit switch turned on.

10.4.8 The Illegal Address Error (IAE) will occur only if the GO bit is set and the function is one which makes use of the Desired Block Address. It does not occur, for example, on either a Return to Centerline or a Seek function.

10.4.9 Software should not regard a disk transfer as complete until all error checking and correction has been done. Power fail should cause an unfinished read to be repeated.

10.4.10 For moving head disks, note that there is no way to stop a Seek which is in progress without risking losing track of the arm position. Therefore, if ever a Seek is aborted (by INIT, Drive Clear, or an error), the Recalibrate command should be issued. This will insure that the drive reestablishes proper arm indexing. When the Return to Centerline command is performed, the drive also resets the Desired Cylinder Address register.

10.4.11 It should be noted that the RMR error will not occur on writes to registers which are located solely in the Massbus controller.

10.5 NOTES ON THE ATTENTION CONDITION

10.5.1 The Attention Active (ATA) bit in the Status register is the indicator of the Attention condition for the drive. When it is set, the ATTN line is asserted by the drive, and the ATA bit also appears set in the appropriate position of the Attention Summary pseudo-register.

The Attention condition can be caused by any one of the following events:

1. An error occurs. The specific error condition is indicated in an Error register, and the composite error (ERR) bit is set.

2. A non data transfer operation is completed. The Positioning In Progress (PIP) bit is reset and the Drive Ready (DRY) bit is set by by the drive as the operation is completed. Some examples of non data transfer operations are the Search, Seek, and Rewind commands.

3. The recording medium comes on line or goes off line. The Medium On Line (MOL) bit is set or reset by the drive at these events.

4. In dual controller drives, an Attention condition is raised when the drive switches to this controller if the drive was not available when it was requested. This event corresponds to the Drive Present (DPR) being set by the drive as it switches from the other controller to this one. If the drive was in the neutral position when the drive was requested, it would have switched immediately. No Attention condition is raised in that case.

10.5.2 The Attention Active (ATA) bit of a drive can be reset by the controller by doing any one of the following:

1. Write a Drive Clear function into the drive Control register (function word = 000011(8))

2. Write a new operation code (with the GO bit set) into the Control register. As the new command is accepted, the drive resets its ATA bit.

3. Write a word into the Attention Summary register with a 1 in the bit position corresponding to this drive. Note that this is permitted even while ERR is set (see sec. 9.4.4).

   Note 1: Although the ATA bit can be reset by writing a "1" into the Attention Summary pseudo-register, it cannot be set by doing so.

   Note 2: The PDP-11 "BIS" and similar instructions should not be used to clear ATA bits. The Attention Summary register should be be written only with entire words.

4. Assert the Initialize (INIT) line. This also causes all drives to reset and go to an initial power up state. This action takes precedence over any other operation going on in the drive.

10.5.4 The normal sequence for program handling of ATA caused interrupts should always be the following:

1. In the interrupt routine, read the Attention Summary register.

2. Write the Attention Summary register, using either the word read from it in (1) or a word with fewer "1" bits (reset only the ATA bits which are going to be serviced on this interrupt).

3. Now read the Status register of the drive or drives which are to be serviced (this may be done outside of the interrupt routine).

It is necessary to reset the ATA bit before reading the Status register to insure that no subsequent error or attention conditions are lost.

In a simple configuration, it is permissable to leave the ATA bits set (and interrupts disabled) until the next transfer is ready to begin.

In all cases, the program should verify that the drive is ready before issuing a data transfer command.

11. ELECTRICAL SPECIFICATION

11.1 INTRODUCTION

11.1.1 The electrical specification is intended to define the electrical parameters of the bus, and to assign values to these parameters.

11.2 COMPONENTS

11.2.1 Line drivers

11.2.1.1 The line drivers are single input, differential-output devices. Their input characteristics are those of TTL integrated circuits. Their output characteristics are:

A. Sink Side; V(OL)=0.6 volts @ 60 mA.

B. Source Side; V(OL)=1.5 volts @ -60 mA.

C. Propagation Delay time: Tpd = 30 nanoseconds, max.

11.2.2 Line Receivers

11.2.2.1 The line receivers are differential-input, single-output devices, with input sensitivity of 25 millivolts and maximum Propagation delay time of 25 nanoseconds.

They have open-collector outputs with a maximum sink capability of 16 milliamps at V(OL) = 0.4 Volts, and maximum high state sink (leakage) current of 250 microamps. Each output has a 3.3 K-ohm resistor pullup to +Vcc.

11.2.3 Cable

The standard cable for use inside enclosures is a flat 40 conductor cable with shield and drain wire. Its maximum length is 120 feet. Its characteristics are as follows.

A. Common Mode Characteristic Impedance (Wire to Shield) 75 +/- 8 ohms.

B. Differential Mode Characteristic Impedance (Wire to Wire) 130 +/- 10 ohms.

C. Losses- The rise time degradation is approximately given by

dV(L)/dT = dV(0)/dT * EXP (L * -5.77 x 10E-3)
Where dV(L) = rate of voltage change at a point L feet from the driver end.

L = distance along cable.

11.2.3.2 The standard cable for outside enclosures is a 60 twisted pair, round bundle. Its characteristic impedance is 130 ohms.

11.2.3.3 Ground continuity must be maintained throughout the cable system. The flat cable and twisted pair cable drain wires must be connected to each other and to logic ground.

11.2.3.4 The assumed maximum propagation delay for a Massbus signal is 375 nanoseconds (in one direction). This is calculated from 160 feet maximum length times 2 nanoseconds per foot plus 55 nanoseconds for driver and receiver delays. The assumed worst-case skew between signals on the Massbus is 150 nanoseconds. This is calculated from a 375 nanoseconds delay times a 33% variation in delay plus a 25 nanosecond safety factor.

11.3 STANDARD TRANSCEIVER MODULES

11.3.1 Purpose

11.3.1.1 Standard modules exist to buffer the Massbus cabling from local logic circuitry. The M5904 is the Controller Transceiver, M5903 is the Drive Transceiver, and H870 is the Terminator. Use of these modules assures that the cable system will be virtually free of stubs.

11.3.2 Use

11.3.2.1 Each Massbus interface is accomplished with three of the appropriate modules. Connections to the modules are made with flat, 40 conductor cable. Cabling runs either directly to another set of modules or to a junction with twisted pair cable.

11.3.2.2 The reliable operation of the Massbus Transceiver Modules can only be assured if the transmission system is well designed. Impedance must be uniform throughout the system. No stubs longer than six inches can be used, and no multiple stubs are acceptable.

11.3.2.3 Proper operation of the Massbus system requires a low resistance connection between all logic circuitry grounds. Remote peripherals must be connected to the controller, preferably with a heavy metal braid. Frame connections may not be adequate, so direct connection to power supply or logic backpanel ground should be made. The standard flat cable includes a drain wire on conductor VV to provide a well-defined common-mode impedance. The standard twisted-pair cable includes shields and drain wires to provide well-defined common mode impedance. These drain wires can be used for the electrical connection by themselves if these is no danger of additional currents being present between the cable terminal points. Each system must provide proper grounding between logic supplies independent of the Massbus cable.

11.3.2.4 The M5904 Controller Transceiver Module requires the following voltages and worst case currents:

+5 V(DC) @ 3.3 Amps
-15 V(DC) @ 0.165 Amps

The M5903 Drive Transceiver requires:

+5 V(DC) @ 1.7 Amps
-15 V(DC) @ 0.165 Amps

The currents given are per module, so a three module bank of M5904's draws 9.9 Amps @ +5V and 0.5 Amps @ -15V. The three M5903's draw 5 Amps @ +5V and 0.5 Amps @ -15V.

11.3.2.4 A maximum of 0.6 volts of common mode noise can be tolerated by Massbus Transceiver modules.

11.4 APPROVED MASSBUS HARDWARE

11.4.1 The following is a listing of hardware considered to form the core of the Massbus interface system. Any items not on this list may not be used without complete justification to a project Design Review Committee.

ITEM	DEC Purchase Spec. No.	Name
Line Driver	19-11341	DEC 75113
Line Receiver	19-10268	DEC 75107B
	19-10275	DEC 75108B
Flat Cable	17-00034	
Twisted Pair Cable	17-00033	

11.4.2 It is impossible to create a bus extender, bus repeater, or bus switch for the Massbus as it is currently specified. This is primarily due to electrical limitations, propagation delay, skew, etc.

11.5 MASSBUS SIGNALS

11.5.1 The FAIL signal is defined as asserted (high) when power fails in the controller. It is negated (low) when power in the controller is ok. The drive should use this signal to inhibit the DEM and INIT signals whenever FAIL is asserted.

The controller must provide a circuit which sinks up to 20 ma with a voltage rise of less than 0.5 V when power is ok, and makes a single transition to high impedance at least 1.5 microseconds before the controller line drivers fail.

The drive will pull up the FAIL line to +5 V through a diode to prevent a failed drive from pulling the line down.

If possible, the controller should insure that any FAIL pulse will be at least 1 microsecond in duration.

On power-up, the controller should keep the FAIL circuit at high impedance until at least 1.5 microseconds after the line drivers are powered up and stable.

11.5.2 The other Massbus signals are defined in sections 3 and 4 of this specification. For a signal table, see fig. 11.5.3 and 11.5.4.

FIGURE 11.5.3

M5903 SIGNAL TABLE

SIGNAL NAMES A	SIGNAL NAMES B	SIGNAL NAMES C	DRIVER INPUT PIN	RCVR OUTPUT PIN	DRIVER ENABLE PIN	RCVR ENABLE PIN	HEADER PIN –	HEADER PIN +
D00	D06	D12	AE1	AD1	AB1	AJ1	A	B
D01	D07	D13	AC1	AA1	↓	↓	D	C
D02	D08	D14	AM2	AK1	↓	↓	E	F
D03	D09	D15	AL2	AF1	↓	↓	J	H
D04	D10	D16	AP2	AM1	↓	↓	K	L
D05	D11	D17	AN2	AL1	↓	↓	N	M
C00	C06	DPA	AU1	AS1	AU2	AR1	P	R
C01	C06	C12	AT2	AN1	BS2	AP1	T	S
C02	C08	C13	BB1	BA1	↓	↓	U	V
C03	C09	C14	AV2	AV1	↓	↓	X	W
C04	C10	C15	BJ2	BE1	↓	↓	Y	Z
C05	C11	CPA	BD1	BC1	BF2	↓	BB	AA
SCLK	EXC	OCC	BP2	BL1	BM1	BK1	CC	DD
RS3	RS0	DS0		BF1		BH1	FF	EE
RS4	RS1	DS1		BU1		↓	KK	LL
CTOD	RS2	DS2		BV2		↓	MM	NN
WCLK	INIT	DEM		BR1		BS1	RR	PP
RUN	SPARE	SPARE		BP1		BN1	TT	SS
ATTN	EBL	TRA	BL2		BJ1		JJ	HH
–	–	FAIL		BT2			UU	
+3V	+3V	+3V		BR2				
+5V	+5V	+5V	AA2	BA2	AV1			
			AE2*					
–15V	–15V	–15V	AB2					
GROUND	GROUND	GROUND	AC2/ AT1/ BC2/ BT1/				VV	

*FOR –15V OPERATION' THESE PINS MUST BE CONNECTED ON THE BACKPANEL.
FOR –20V OPERATION, AE2 IS *NOT* TO BE USED.

11-2747

FIGURE 11.5.4

M5904 SIGNAL TABLE

SIGNAL NAMES A	SIGNAL NAMES B	SIGNAL NAMES C	DRIVER INPUT PIN	RCVR OUTPUT PIN	DRIVER ENABLE PIN	RCVR ENABLE PIN	HEADER PIN –	HEADER PIN +
D00	D06	D12	AE1	AD1	AB1	AL1	A	B
D01	D07	D13	AC1	AA1	↓	↓	D	C
D02	D08	D14	AK1	AJ1	↓	↓	E	F
D03	D09	D15	AF1	AH1	↓	↓	J	H
D04	D10	D16	AN1	AM1	↓	↓	K	L
D05	D11	D17	AL2	AM2	↓	↓	N	M
C00	C06	DPA	BD2	AU2	AU1	AS1	P	R
C01	C07	C12	AP1	AR1	AT2	BB1	T	S
C02	C08	C13	BD1	BC1	↓	↓	U	V
C03	C09	C14	AV2	BA1	↓	↓	X	W
C04	C10	C15	BJ1	BH1	↓	↓	Y	Z
C05	C11	CPA	BE1	BF1	↓	↓	BB	AA
SCLK	EXC	OCC	BR2	BP2	BN1	BN2	CC	DD
RS3	RS0	DS0	BM1		BM2		FF	EE
RS4	RS1	DS1	BU2		↓		KK	LL
CTOD	RS2	DS2	BP1		↓		MM	NN
WCLK	INIT	DEM	BV1		BU1		RR	PP
RUN	SPARE	SPARE	BS1		BR1		TT	SS
ATTN	EBL	TRA		BK1		BL1	JJ	HH
–	–	FAIL	BV2				UU	
+3V	+3V	+3V		BK2				
+5V	+5V	+5V	AA2/ BA2/ AV1				–	+
–15V	–15V	–15V	AB2					
GROUND	GROUND	GROUND	AC2/ AT1/ BC2/ BT1/ AN2					

These are relative polarity for a logic 1 at module input pin.

11-2748