RL8-A omnibus controller technical manual

EK-RL8A-TM-001

digital equipment corporation • maynard, massachusetts

1st Edition, April 1978

Printed in U.S.A.

This document was set on DIGITAL's DECset-8000 computerized typesetting system.

CONTENTS

CONTENTS (Cont)

FIGURES

TABLES

CHAPTER 1
GENERAL DESCRIPTION

1.1 PURPOSE AND SCOPE

This manual describes the physical, electrical, and programming characteristics of the RL8-A Omnibus Controller. It is intended as an aid for maintenance personnel in troubleshooting and servicing the RL8-A/RL01 Disk Subsystem. Prerequisites for this manual include a basic understanding of the PDP-8 processors and peripherals as well as a background in disk principles and magnetic recording techniques.

1.2 INTRODUCTION

The RL8-A/RL01 Disk Subsystem is a random access, mass storage system that stores up to 5.24 million bytes or 3.5 million 12-bit words (per disk) in fixed length blocks on preformatted disk cartridges. The RL8-A controller provides the interface and the control functions between all Omnibus PDP-8s and the RL01 disk drives.

The RL8-A (M8433) interface and control logic is contained on a single multilayer hex width module. It is inserted into the Omnibus backplane and connected to the first drive via a BC80J-20 interface cable. Connections between the controller and the drives are made using the conventional daisy chain method with a 7012122 cable.

The RL8-A controller can control up to four RL01 disk drives (Figure 1-1). It can communicate with one or more drives, but can only transfer data to or from one drive at a time. In addition, provisions are available which allow a second controller and four additional drives. Data transfers to and from memory are accomplished via the single cycle data break (DMA) facility. In a two controller configuration, activities (i.e., seeks, read header, etc.) can be done through one controller while the other controller is performing data transfers.

CAUTION

Simultaneous data transfers (DMAs) through both controllers should not be attempted due to data rate limitations.

NOTE

The RL8-A controller is not program compatible with any other PDP-8 disk controller.

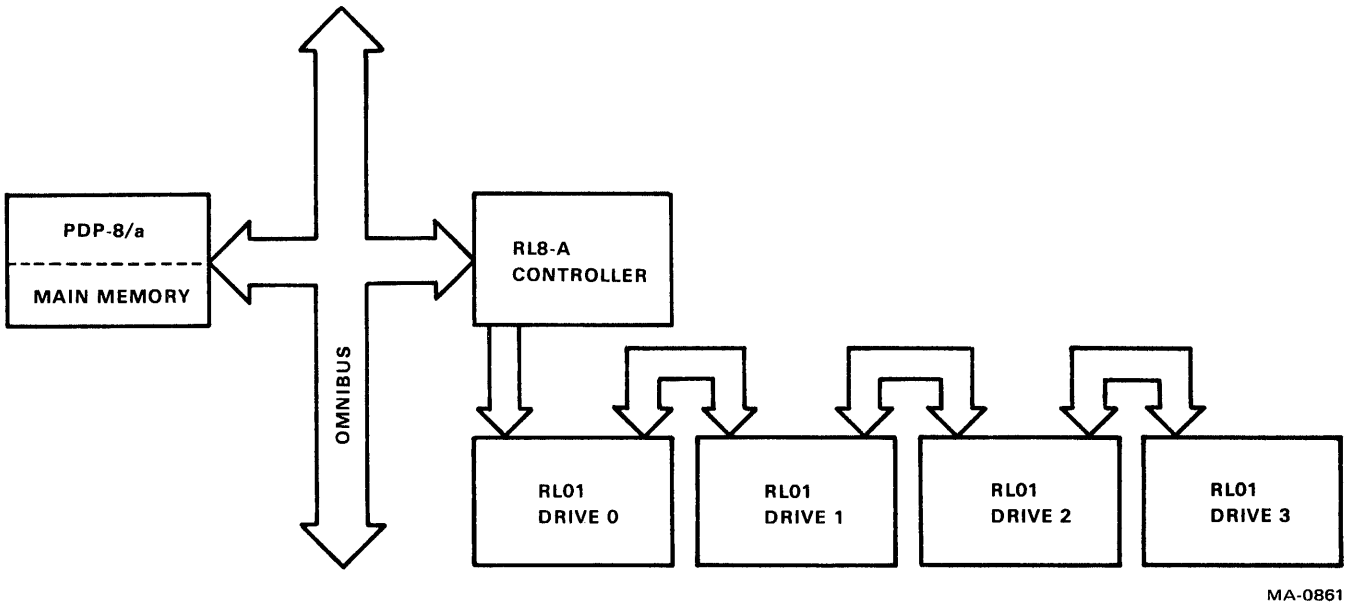


Figure 1-1 RL8-A/RL01 Block Diagram

1.3 RL01 DISK DRIVE

The RL01 disk drive is a random access, moving head, dual surface single-platter drive that is used for system applications requiring on-line direct memory access. Each drive has two movable read/write heads using RL01K disk cartridges as the storage medium.

The RL01 disk drive employs the "servo-in-data" concept to provide high field performance and maintainability. This concept permits the derivation of head positioning and track counting information from pulses prerecorded within the data record. In effect, each read/write head seeking to a desired track becomes its own servo transducer and therefore an alignment cartridge is not required for head alignments.

The disk drive consists essentially of a spindle motor, head positioner, drive electronics, power supply, cabinet and front panel assembly. The storage medium for the drive is a top-loading, single-disk cartridge with a total data storage capacity of approximately 5.2 million bytes.

1.3.1 Data Format

The RL01K disk cartridge has two recording surfaces capable of storing up to approximately 5.2 million bytes or 3.5 million 12-bit words of data on 512 tracks (Figure 1-2). Servo head positioning information and header information are prerecorded on the disk at the factory and cannot be modified or rewritten in the field. The drive logic prevents accidental over-writes of this data to ensure pack formatting integrity. A bad sector file is also formatted on the disk at cylinder 377_8 (a maximum of 128_8 bad sectors are allowed per pack). This track should not be used for read/write operations.

Every track on the RL01K disk cartridge is subdivided into 40 equal-length sectors which are labeled from 0_8 to 47_8. The sectors are further subdivided into data fields of either 170 (12-bit) or 256 (8-bit) words depending on the mode and the data cyclic redundancy check (CRC). Data modified by a write operation includes only the data preamble (PRZ). Preambles and CRC are generated and checked by the control hardware.

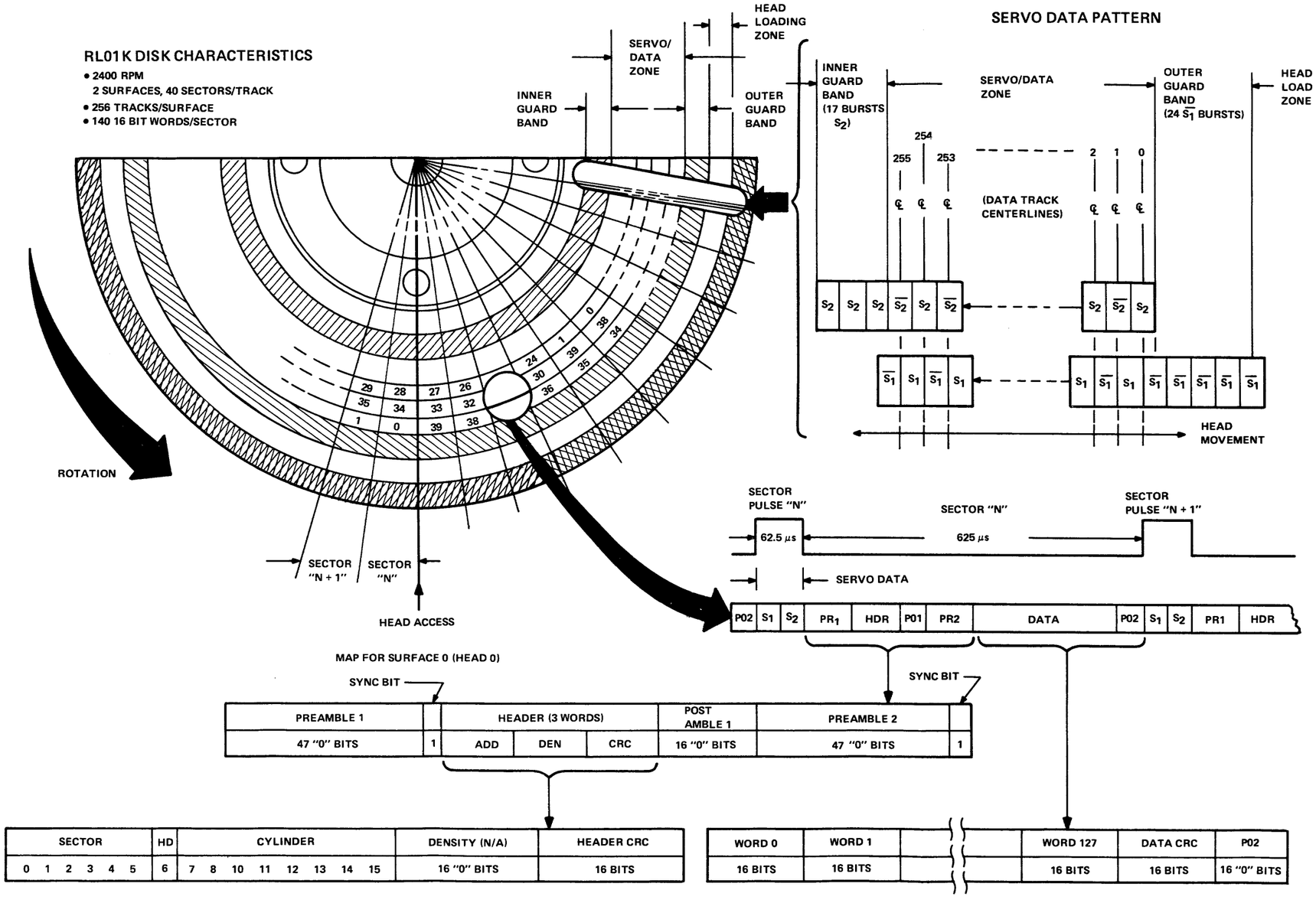


Figure 1-2 RL8-A/RL01K
Disk Cartridge Format

1.3.2 Servo and Header Data

When the disk cartridge is formatted, both servo and header information are prerecorded in each sector. The servo (SV) information is contained in two pulse bursts which occur during each sector pulse. This information identifies the radial position of the heads relative to each of the tracks on the cartridge.

Sector pulses are detected by using transducers to detect the edges of the slots on the disk armature plate.

1.4 REFERENCE DOCUMENTS

The documents listed in Table 1-1 contain information necessary for a complete understanding of the function, installation, operation, programming, and maintenance of the controller and disk drive. Information on the PDP-8/A processor and Omnibus is provided in the PDP-8/A Miniprocessor Handbook.

Table 1-1 Reference Documents

Title	Document No.
RL01 Disk Drive Technical Description Manual	EK-RL01-TD-PRE
PDP-8/A Miniprocessor User's Manual	EK-81002-MM-002
PDP-8/A Operator's Handbook	DEC-8A-HOPHB-A-D
RL01 Disk Subsystem Service Manual	EK-RL01-SV-PRE
RL01 Disk Drive, Illustrated Parts Breakdown	EK-RL01-IPB-PRE
Program Library ZF-233-RB (Paper Tape and Document) ZF-233-RZ (Document Only) ZF-233-PB (Paper Tape Only) ZF-233-FR (Fiche)	

1.5 RL8-A SPECIFICATIONS

Characteristic	Specification
Required Mounting Space	One hex width module slot
Power Requirements	+5 Vdc ± 5% @ 2.5 A +15 Vdc ± 10% @ 150 mA -15 Vdc ± 10% @ 90 mA
No. Drives per Controller	Up to 4
Data Transfer	Single Cycle Data Break (Break priority either 0 or 1)

Characteristic	Specification
Interface Specification	
Output Signals (To Drive)	Power Fail Drive Select 0 Drive Select 1 Write Data Write Gate Drive Command System Clock
Input Signals (From Drive)	Drive Ready Drive Error Sector Pulse Read Data Status Status Clock
Drive Specifications	
Capacity (maximum)	3,481,600 12-bit words 5,242,880 8-bit bytes
Seek Time	55 ms average 15 ms one track 100 ms maximum
Transfer Rate	3.9 μs per word, average 2.0 μs per word, peak (byte mode)
Disk Rotation	2400 revolutions per minute $\pm$ 0.25%
Rotational Latency	12.5 ms average 25 ms peak
No. Recorded Surfaces	2
Number of Cylinders	256
No. Sectors per Track	40
No. Words per Sector	128 in 12-bit mode 256 in 8-bit mode
No. Bits per Word	8- or 12-bit mode
Size	26.52 cm (10.5 in) H $\times$ 42.26 cm (19 in) W $\times$ 63.5 cm (25 in) D Standard 19 inch rack mount
Start Time	40 seconds maximum

Characteristic	Specification
Stop Time	30 seconds maximum
Weight	Approximately 34.02 kg (75 lb)
Power	DC - None AC - 90 to 132 Vac (47 to 63 Hz) 180 to 264 Vac (47 to 63 Hz) 150 W nominal
Environment (Controller Only)	
Operating Temperature	10° C (50° F) to 40° C (104° F)
Humidity	10 to 90% noncondensing

CHAPTER 2
SYSTEM DESCRIPTION

2.1 SCOPE

This chapter describes the RL8-A Controller using a simplified block diagram approach. This includes a discussion of the controller operation in terms of register functions, controller commands, and major signals. A more detailed description of the logic and controls is discussed in Chapter 4.

2.2 GENERAL SYSTEM DESCRIPTION

Figure 2-1 illustrates the basic interface between the Omnibus and the disk drive(s). The Omnibus interface (M8433) contains function and format control logic for controlling data transfers between the processor and the controller. The RL8-A interface also contains logic for receiving and transmitting commands and signals between the CPU and the drive(s).

The RL8-A controller performs four basic functions:

1. Positions the read/write heads.
2. Controls data flow between the Omnibus and the drive(s).
3. Arbitrates status information.
4. Error detection of subsystem (disk and controller) operation.

The basic functions performed by the controller can be lumped under the general headings of the function and format control logic. The function control logic decodes the operation to be performed and then conditions the format control in such a manner that it executes the specified functions. In addition, the format control uses binary counters which address a ROM that provides the necessary output control signals in the proper sequence for command execution. In a typical read/write application, the function control starts to count, sets up various control conditions, reaches a predetermined state, and waits. The format control then takes over, establishes the necessary word boundaries, controls the actual transfer of header information and data, and then reverts operation back to the function control for command completion. The format control is only operative during execution of those commands associated with head positioning and/or the actual transfer of header information or read/write data, in synchronization with the disk drive.

In addition to the standard interface logic, the RL8-A controller contains: a first-in first-out 16-word × 12-bit silo and Omnibus drivers; a write precompensation circuit for offsetting the effects of peak shifting normally associated with modified frequency modulation (MFM) recording; a phase lock loop circuit to compensate for variations in disk speed; a data separation circuit for separating clock and NRZ data signals from the drive MFM encoded serial data; a header compare circuit for searching headers on the recording surfaces; and a cyclic redundancy check (CRC) circuit for detecting header and data recording errors.

2.2.1 Basic Controller Operation

The flowchart in Figure 2-2 depicts the basic sequence of events that occur during execution of all controller commands. After the necessary registers are loaded, the command is decoded and the function control is initiated. Controller logic and error bits are cleared. The command function to be performed is then executed.

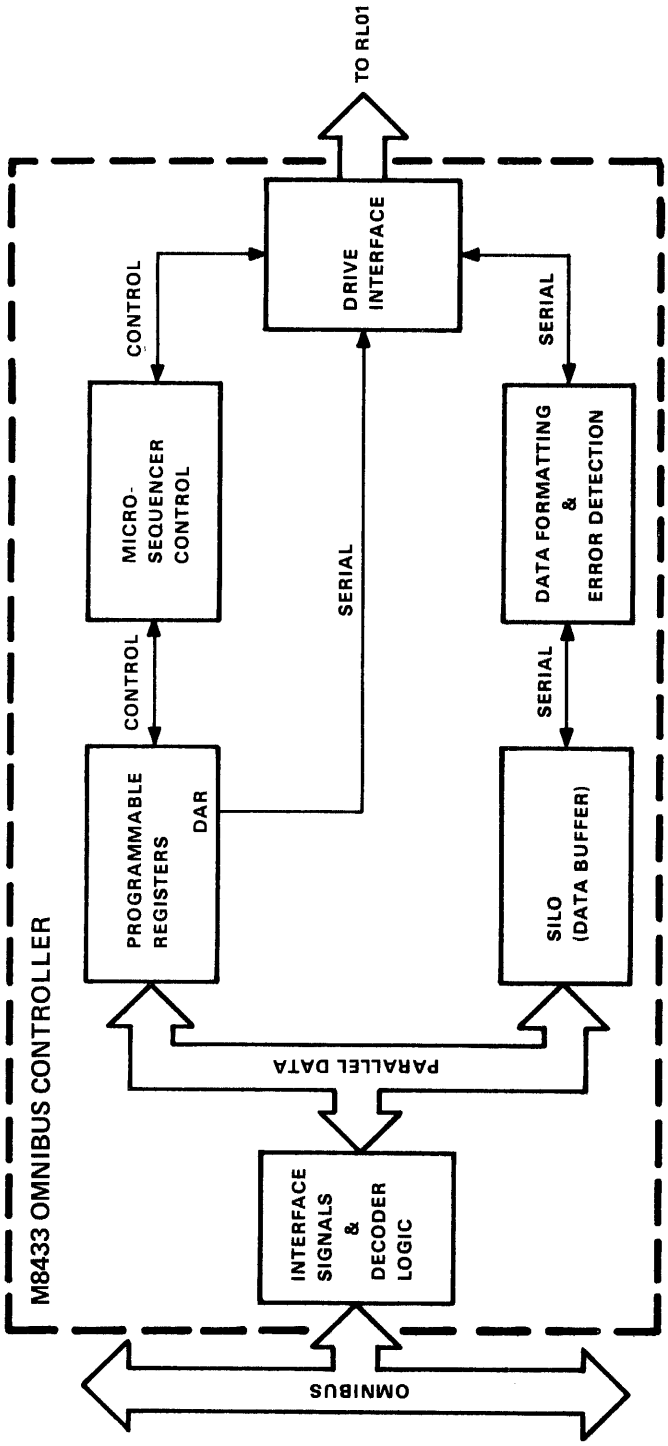


Figure 2-1 RL8-A Simplified Block Diagram

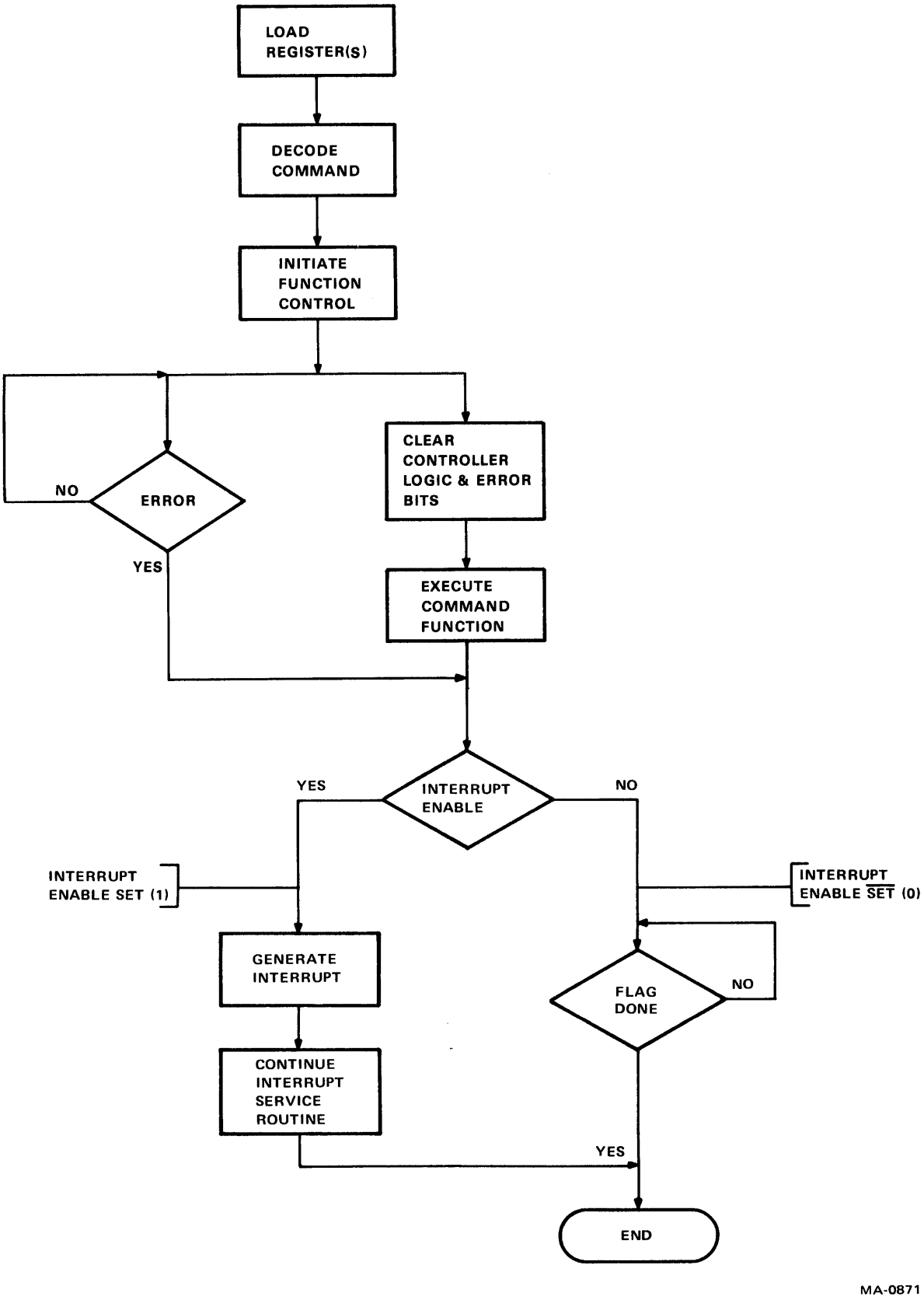


Figure 2-2 Basic Controller Operation Flowchart

When the command function is completed, the done flag is checked and the interrupt enable is checked. If the interrupt is not enabled, a skip function is generated by setting the DONE flag and the command is terminated. If the interrupt is enabled, it causes the CPU to jump (JMS) to location 0 into an interrupt service routine request is made before command termination.

To issue a function command, the CPU places the address and data onto the Omnibus and the controller decodes the information and channels it to the appropriate register. After the desired function command is written into command register B, it initiates the microsequencer routine. The microsequencer control performs a different routine for each of the controller command functions. These routines manipulate the data formatting circuits to format the data. Included in the data formatting function is an error detection feature that uses cyclic redundancy checking (CRC).

Before reading or writing data, it is necessary to make sure that the drive heads are positioned properly on the disk. This is accomplished by first executing a read header command to determine where the heads are located and if the upper or lower head has been selected. If positioned at the desired track address, the read/write command can be executed directly. Otherwise, a software calculation must be performed to determine the cylinder distance that the heads must be moved to the new location. This cylinder-distance information is then transmitted to the selected drive via a seek command.

2.3 PROGRAMMING-INSTRUCTION SET

Input/output transfer (IOTs) instructions are used to initiate the operation of the disk drive(s) and to transfer data. Figure 2-3 illustrates the IOT instruction format. Bits 0–2 contain the op code, which must be a six to specify an IOT instruction. Bits 3–8 contain device selection code and bits 9–11 contain the function code. The instructions are loaded onto the MD lines and decoded by the RL8-A controller. Transfers from the AC to the controller clear the AC after the transfer is complete. Transfers to the AC from the controller clear the AC first and then the transfer takes place.

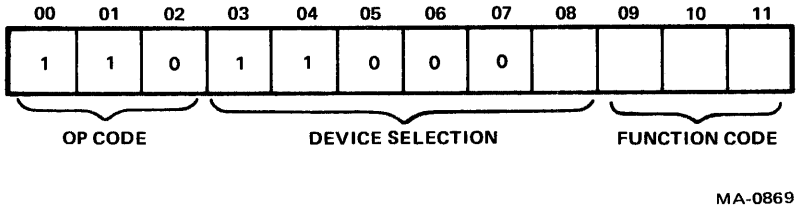


Figure 2-3 RL8-A Instruction Set Format

The skip instructions in the RL8-A instruction set performs the skip and then clears the flag. This means that if a given flag is true, it will be cleared upon completion of the skip IOT. Table 2-1 lists and defines the instruction set for the RL8-A.

Table 2-1 RL8-A Instruction Set

Octal Code*	Mnemonic	Function
6600	RLDC	Clear controller, all registers, AC and flags. (Do not use to terminate a disk function.)
6601	RLSD	Skip on function done. Then clear if set to a one.
6602	RLMA	Load break MA register from AC 0:11
6603	RLCA	Load command register A from AC 0:11
6604	RLCB	Load command register B from AC 0:11, execute command
6605	RLSA	Load sector address from AC 0:5
6606	----	Spare (will clear the AC)
6607	RLWC	Load word count from AC 0:11
6610	RRER	Read error register into AC 0:2, 10:11.
6611	RRWC	Read word count into AC 0:11
6612	RRCA	Read command register A into AC 0:11
6613	RRCB	Read command register B into AC 0:11
6614	RRSA	Read sector address into AC 0:5
6615	RRSI	Read silo word (8-bit) into AC 4:11
6616	----	Spare (doesn't clear AC)
6617	RLSE	Skip on composite error, then clear if set to a one.

*Alternate device code, 62 and 63

2.4 ADDRESSABLE REGISTERS

2.4.1 Command Register A

The command register A is a 12-bit addressable register that contains the cylinder address difference, head select and direction for seek functions (Figure 2-4).

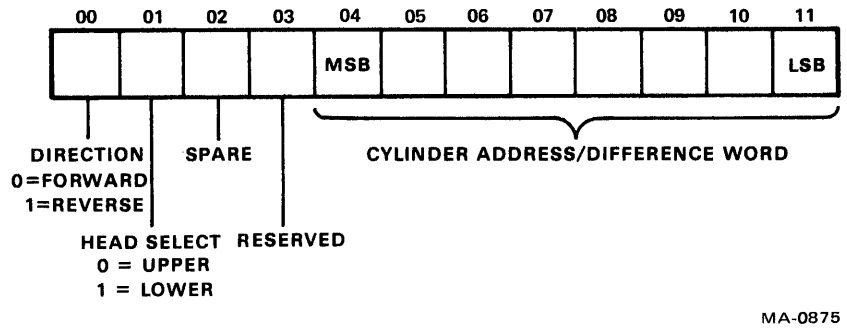


Figure 2-4 Command Register A Format

Bit	Name	Function
AC0	Direction (DIR)	This bit indicates the direction in which a seek is to take place. When the bit is set, the heads move toward the spindle (to a higher cylinder address). When the bit is cleared, the heads move away from the spindle (to a lower cylinder address). The actual distance moved depends on the cylinder address difference (bits 4–11).
AC1	Head Select (HS)	Indicates which head (disk surface) is to be selected. Set = lower, clear = upper.
AC2		Spare
AC3		Reserved
AC4–11	Cylinder Address Difference (DF4:11)	Indicates the number of cylinders the heads are to move on a seek.

2.4.2 Command Register B

The command register B is a 12-bit addressable register that contains the mode, drive number, interrupt enable and the function code (Figure 2-5).

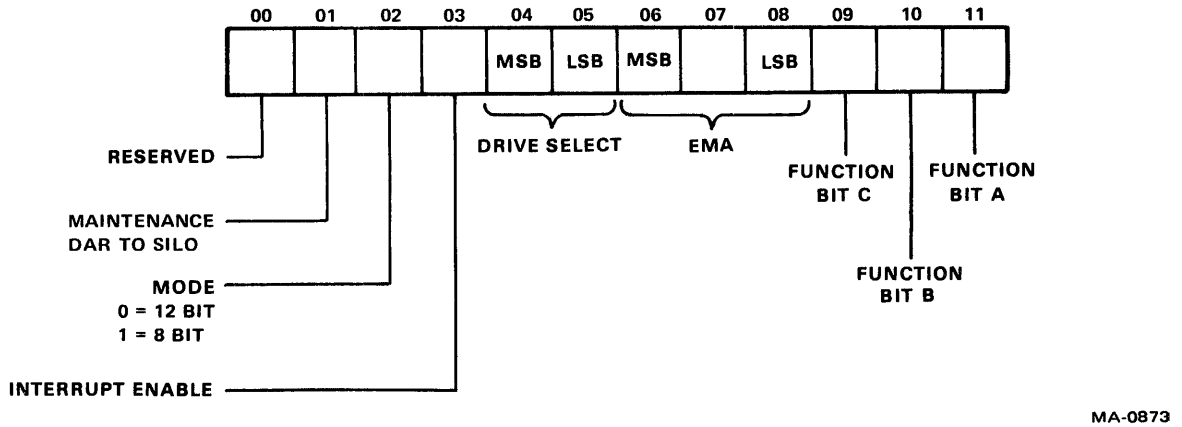


Figure 2-5 Command Register B Format

Bit	Name	Function
AC0		Reserved
AC1	Maintenance	Loop the disk address register (DAR) and load into the silo for maintenance purposes. Bits 1 and 2 of command register B must be set to a one for this function to work correctly. DAR is loaded by using IOT RLCA to load command A (for cylinder address portion) and RLCB to load the function along with the loop-back bit (1) and the mode bit (2). When the function done flag sets to a one, the silo should be read twice to yield two 8-bit words which are the DAR. The first word has the following content: **NOTE** **AC 0:3 are undefined for both words.** AC04 - Cylinder Add. (LSB) AC05 - Head Select, other commands AC06 - 0 AC07 - Head Select for Seek AC08 - A 1 for Reset, otherwise 0 AC09 - Sign Bit AC10 - A 1 for Get Status, otherwise 0 AC11 - Marker Bit (always a 1) The second word has the following content: AC04 - Cylinder Add. (SPARE) AC05 - Cylinder Add. (MSB) AC06 - Cylinder Add. AC07 - Cylinder Add. AC08 - Cylinder Add. AC09 - Cylinder Add. AC10 - Cylinder Add. AC11 - Cylinder Add.
AC2	Mode	When set, this bit indicates that the data field will be 256 8-bit words per sector. When zero, the data field will be truncated to 128 12-bit words per sector (12-bit mode). This bit must be set when doing a GET STATUS or a READ HEADER command is to be executed.
AC3	Interrupt Enable (IE)	When this bit is set, the controller is allowed to interrupt the processor at the conclusion of a normal command or error termination. AC3 is cleared by INIT.
AC4–AC5	Drive Select (DS0, DS1)	These bits determine which drive will communicate with the controller via the drive bus.
AC6, AC7, AC8	Extended Memory Addressed (EMA)	These three bits define the memory field location. This allows 32,768 locations to be addressed for processors having more than 4K of memory.

Bit	Name	Function
AC9–AC10	Function Code	These bits indicate the command to be executed by the controller/disk subsystem.

Bit 9	Bit 10	Bit 11	Command
0	0	0	Maintenance
0	0	1	Reset
0	1	0	Get Status
0	1	1	Seek
1	0	0	Read Header
1	0	1	Write Data
1	1	0	Read Data
1	1	1	Read Data Without Header Check

2.4.3 Disk Status Register

When a get status command is executed, the disk drive returns a status word to the controller. The contents of the disk status register is divided into two words; word 1 and word 2 messages (Figure 2-6).

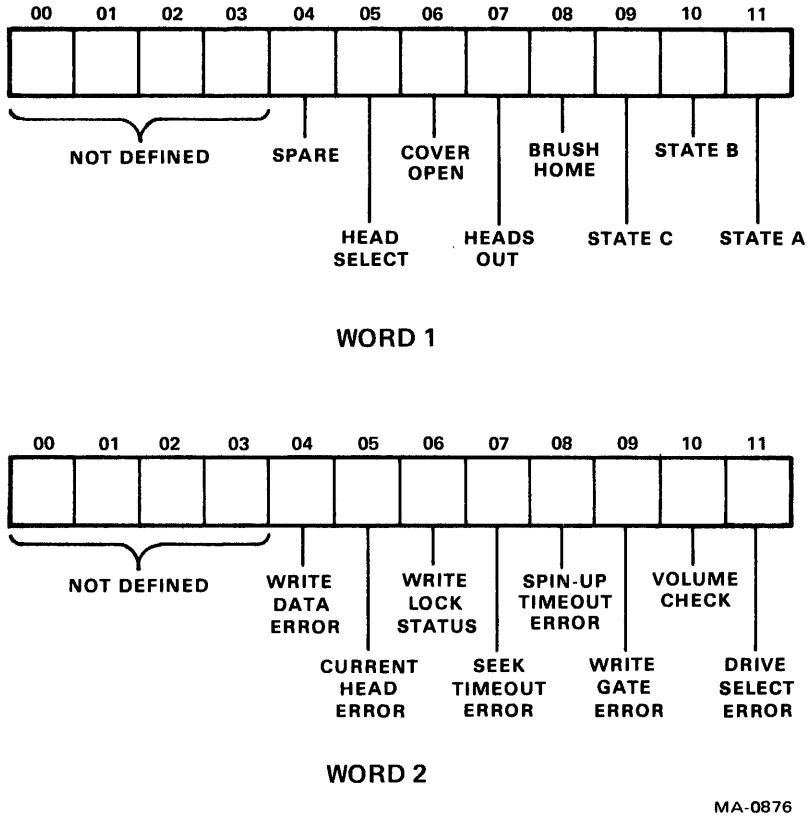


Figure 2-6 Disk Status Register Format

Word No. 1

Bit	Name	Function
AC0–AC3		Undefined
AC4		Spare
AC5	Head Select (HS)	Indicates currently selected head.
AC6	Cover Open (CO)	Asserted when the top cover is open or the dust cover is not in place.
AC7	Heads Out (HO)	Asserted when the heads are over the disk. Cleared when the heads are retracted.
AC8	Brush Home (BH)	Asserted when the brushes are retracted.
AC9–AC11	State Bits	These bits define the state of the disk drive.

State Bit Definitions

Bit C	Bit B	Bit A	Definition
0	0	0	Load State
0	0	1	Spin-up
0	1	0	Load Heads
0	1	1	Brush Cycle
1	0	0	Seek-Track Counting
1	0	1	Seek Linear Mode
1	1	0	Unload Heads
1	1	1	Spin-Down

Word No. 2

Bit	Name	Function
AC0–AC3		Undefined
AC4	Write Data Error (WDE)	When this bit is asserted it indicates that the write gate was on but no transitions were detected on the write data line.
AC5	Current Head Error (CHE)	Indicates write current was detected in the heads when the write gate was not asserted.
AC6	Write Lock (WL)	Indicates the write lock status of the selected drive.
AC7	Seek Time Out Error (SKTO)	Indicates the heads did not come on track within approximately 3 seconds during a seek operation.
AC8	Spin Error (SPE)	Indicates spindle not obtaining up-to-speed in 40 seconds or the spindle is in overspeed (i.e., sector pulses less than 593 apart).

Bit	Name	Function
AC9	Write Gate Error (WGE)	Indicates the drive sensed write gate asserted during sector pulse time, no data pulse within 625 μs, drive write locked or drive not ready.
AC10	Volume Check (VC)	Signals that a new volume (media) has been inserted into the drive or that the power has been cycled down then up on the drive.
AC11	Drive Select Error (DSE)	Indicates one or more drives have the same number.

2.4.4 Header Word(s)

When a read header command is executed, six words will be stored in the silo. The first header word will contain sector address, head select, and cylinder address bit 0 (LSB) information (Figure 2-7). The second word will contain the remaining bits of the cylinder address. Words 3 and 4 will contain 0s and words 5 and 6 will contain the header CRC information. All six words can be extracted by the program using the RRSI IOT.

2.4.5 Word Count Register

When reading or writing data, the word count is loaded with the 2's complement of the number of words to be transferred, and is then incremented by 1 as each word is transferred. The reading or writing operation is terminated when the word count register overflows.

The 12-bit word count register allows up to 4,096 data breaks (DMAs) to take place at one time. This register is loaded with the RLWC IOT from AC0:11 which is the 2's complement of the number of transfers that are specified.

2.5 COMMAND EXECUTION

There are eight controller commands embedded in Command Register B (bits 9–11). Three of these commands are used for initialization and test purposes and the remaining commands control head positioning and the actual transfer of header information and read/write data between the controller and the disk drive(s).

Command Register B Controller Commands

Command	Bit 9	Bit 10	Bit 11
Maintenance	0	0	0
Reset	0	0	1
Get Status	0	1	0
Seek	0	1	1
Read Header	1	0	0
Write Data	1	0	1
Read Data	1	1	0
Read Data (Without Header Check)	1	1	1

2.5.1 Maintenance Command

The maintenance command is used to exercise the controller logic to ensure that it is operating properly (Figure 2-8). This command will test that the command registers, silo, and data paths can function correctly without the disk drive connected.

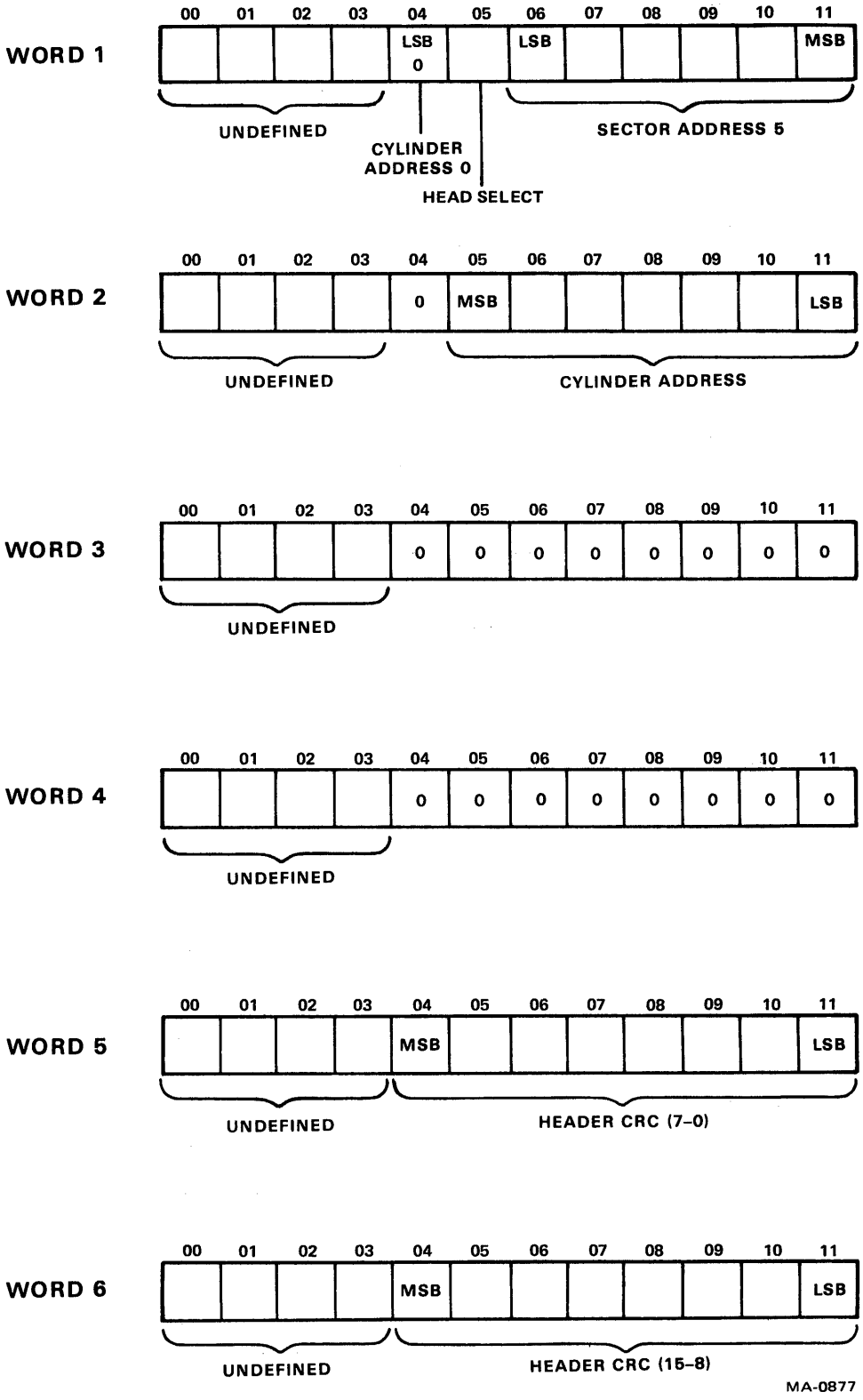


Figure 2-7 Header Words Format

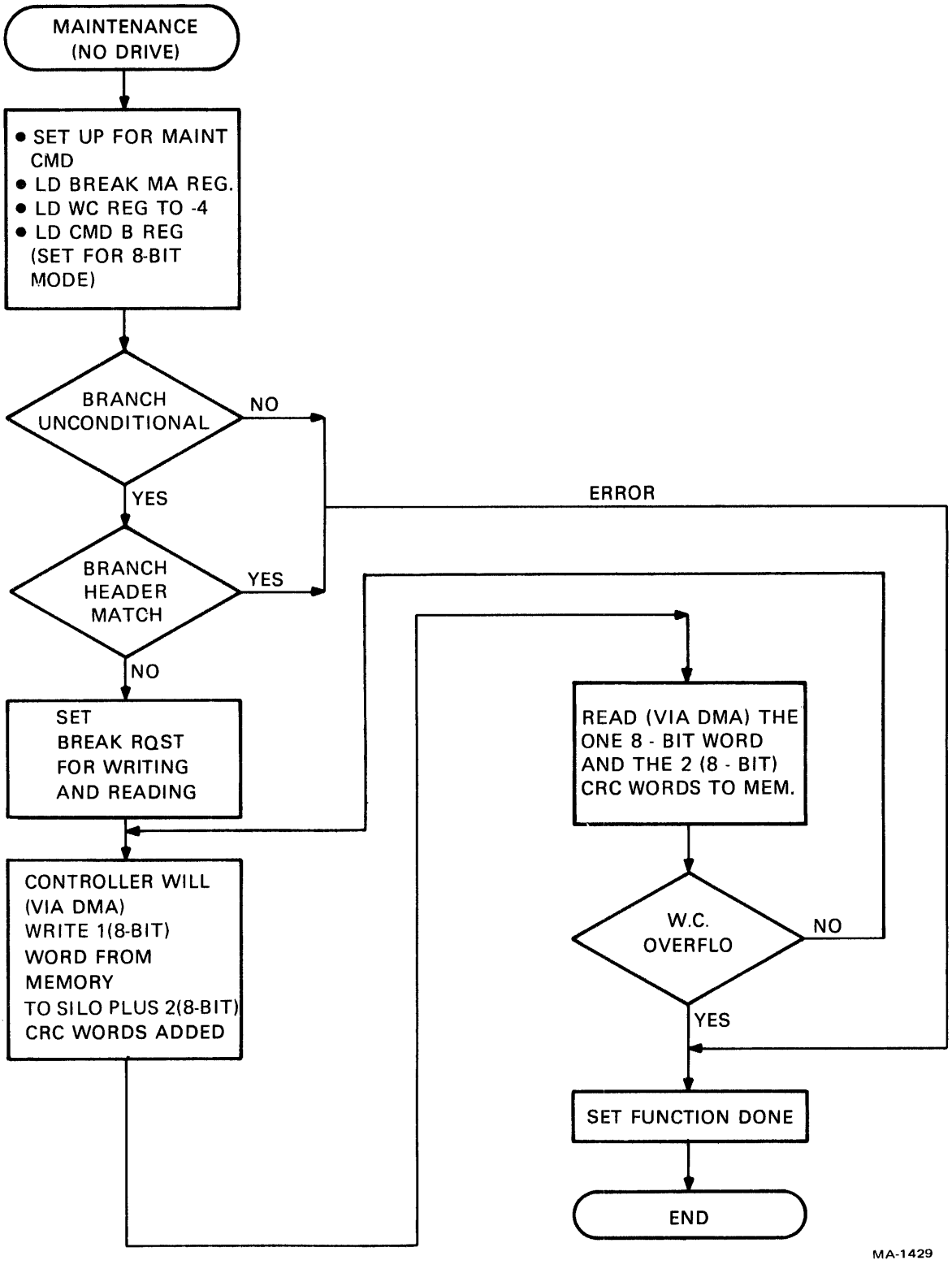


Figure 2-8 Maintenance Command, Operational Sequence

When the MAINT command is decoded the following happens:

1. One DMA will take place out of memory at the location specified by the break MA. A write data function will then be executed out of the silo.

2. The written data will be loaded back into the silo along with two 8-bit words which are the CRC character.

3. DMAs will take place causing the data (8- or 12-bit) word and the two CRC words to be written into memory.

4. This sequence will be repeated until word count overflow. At this time a function done interrupt will occur. A CRC error and function done will immediately raise an interrupt if a CRC error occurs.

2.5.1.1 DAR To Silo Transfers (Maint) – The 16-bit disk address register (DAR) may be loaded into the silo for maintenance purposes. Both bit 1 and 2 of command register B must be set for this function to work properly. The DAR is loaded by using the instruction IOT RLCA to load command register A (for cylinder address portion) and RLCB to load the function along with the loop-back bit and the mode bit. When the function done flag sets, the silo should be read twice to yield two 8-bit words which are the DAR. The first word has the following content:

AC0:3 – Undefined
AC04 – Cylinder Add. (LSB)
AC05 – Head Select, other commands
AC06 – 0
AC07 – Head Select for Seek
AC08 – A 1 for Reset, otherwise 0
AC09 – Sign Bit
AC10 – A 1 for Get Status, otherwise 0
AC11 – Marker Bit (always a 1)

The second word has the following content:

AC0:3 – Undefined
AC04 – Cylinder Add. (Spare)
AC05 – Cylinder Add. (MSB)
AC06 – Cylinder Add.
AC07 – Cylinder Add.
AC08 – Cylinder Add.
AC09 – Cylinder Add.
AC10 – Cylinder Add.
AC11 – Cylinder Add.

2.5.2 Reset Command

When this command is executed, all error bits in the drive are reset. This command does not clear any registers in the controller nor does it cause the heads to move to track 0.

NOTE
The sector address and command register A must be cleared prior to the execution of this command.

2.5.3 Get Status Command

By executing a get status command, it is possible to read the status and errors of the disk drive. The command can also be used to reset errors in the disk drive. When this command is decoded, the drive will send the drive status word to the controller. When the function done interrupt occurs the status words can then be read from the silo into the accumulator by executing two RRSI IOTs.

NOTES

The sector address and command register A must be cleared prior to the execution of this command.

The mode bit (Command Register Bit 2) must be set when loading the Get Status command into command register B.

As shown in Figure 2-9, a status request is sent to the disk drive through a serial drive command line.

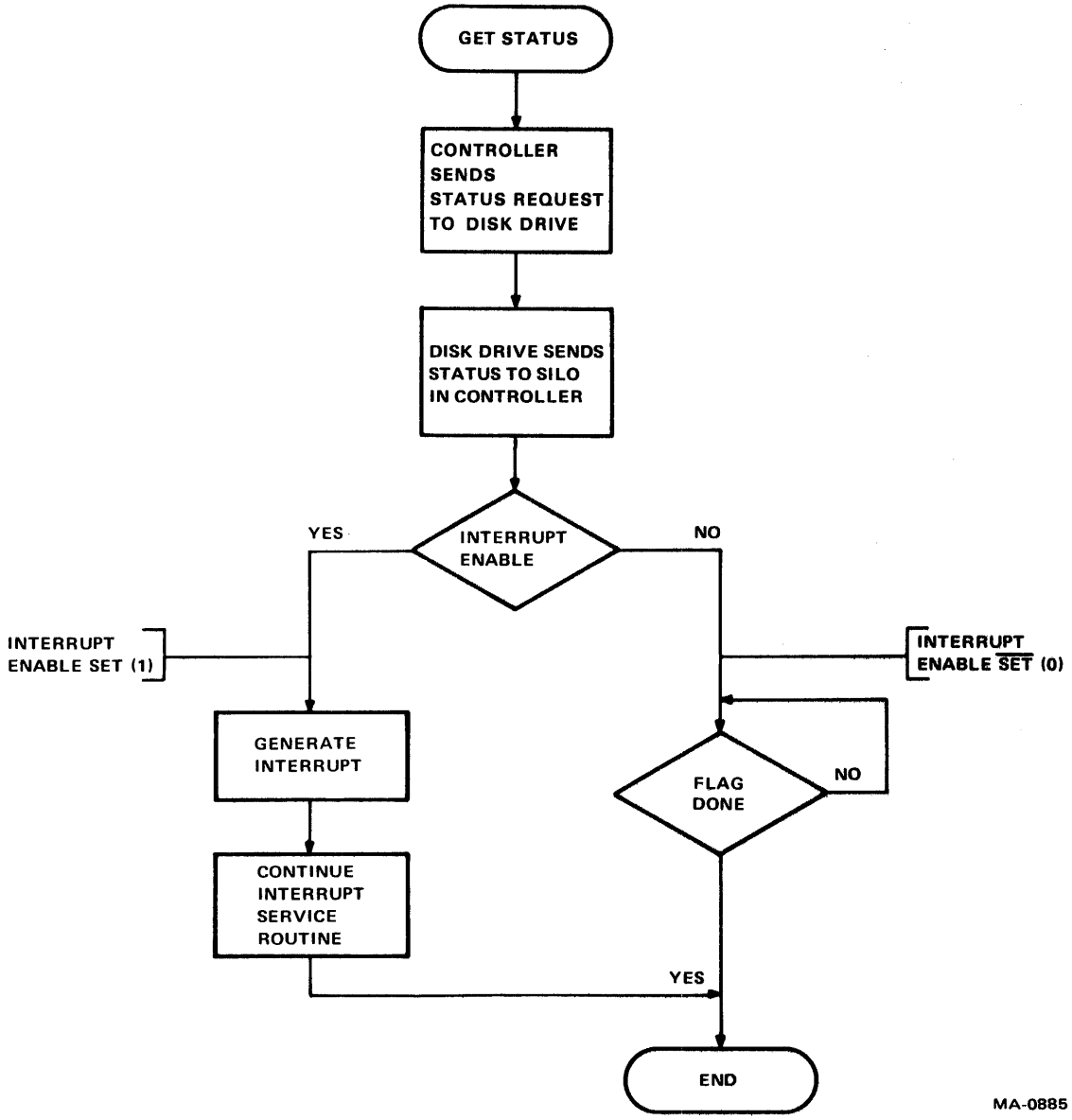


Figure 2-9 Get Status Command, Operational Sequence

The disk drive, in turn, sends back its status through a serial status line. The status information is loaded into the silo and the function is terminated. By reading the silo, drive status is now available. It should be noted that the get status command can be executed any time the controller is ready, even though the drive may not be ready (for example, during a seek or when in the load state).

2.5.4 Seek Command

A seek command involves two basic operations: (1) calculating head selection, cylinder difference, and positioning direction; and (2) transmitting this information from the controller to the drive to initiate a seek.

After the disk address register (DAR) is loaded, the seek command is decoded and the function control is initiated (Figure 2-10). Various circuits and error bits are then cleared. When the drive is ready to accept the seek command, bit 11 (drive ready) of the error register is set, and the controller waits for a sector pulse from the drive. Detection of the sector pulse initiates the format control, whereupon the 16-bit word in the disk address register is shifted out serially on the drive command line to the drive. When the transfer is complete, the format control is terminated, and control reverts back to the function control circuitry. Operation thereafter is the same as for the other commands.

The seek function must be specified and a difference word must be loaded into command register A to enable this function. A function done interrupt will occur when the seek command and difference word have been sent to the disk. If the difference word attempts to move the head past the innermost track (377_8) or the outermost track (0) limits, the head will stop on the track limit.

NOTE

The sector address register must be cleared prior to the execution of this command.

2.5.5 Read Header Command

The function of the read header command is to read the first header encountered on the selected drive, and to store the header information (two header words and one header CRC word) in the silo. The header words can then be extracted from the silo, by reading the silo with two RRSI IOTs to determine the present head position. There are six words loaded into the silo on a read header command: words 1 and 2 = present position, words 3 and 4 = 0s, and words 5 and 6 = CRC information.

When the read header command is decoded, the function control is initiated, and various circuits and error bits are cleared (Figure 2-11). When the drive is ready to accept a command, drive ready is set, and the controller waits for a sector pulse from the drive. When the sector pulse is detected, the format control is initiated, and the read circuits in the phase lock loop/data separator are enabled. Upon detection of the header preamble marker bit, the first header word is directed through the silo. At the same time, the first header word is loaded into the CRC circuit. This operation is repeated for the next two header words. The read circuitry is then disabled, and a header CRC error check is performed. If there is no header CRC error, the function control takes over, and completes the transfer.

When this command is decoded the controller will read the first header encountered on the disk. The function done interrupt will set, indicating that the header words may now be read into the AC. These words (address) can be used to determine current head position or to calculate a difference word to reposition the heads. The header CRC will also be loaded into the controller behind the header words so that reads 5 and 6 of the silo will yield these words for diagnostic purposes.

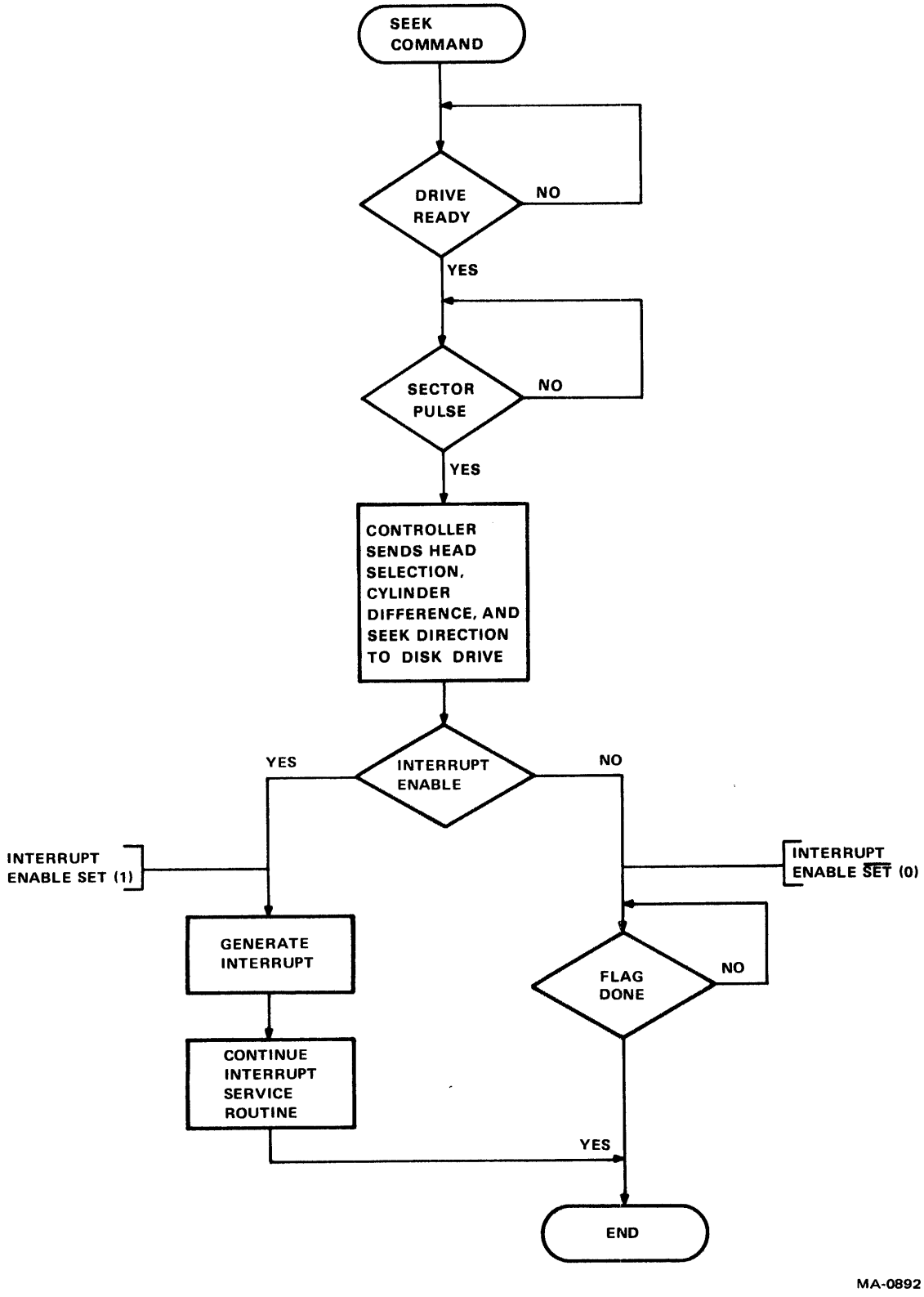


Figure 2-10 Seek Command, Operational Sequence

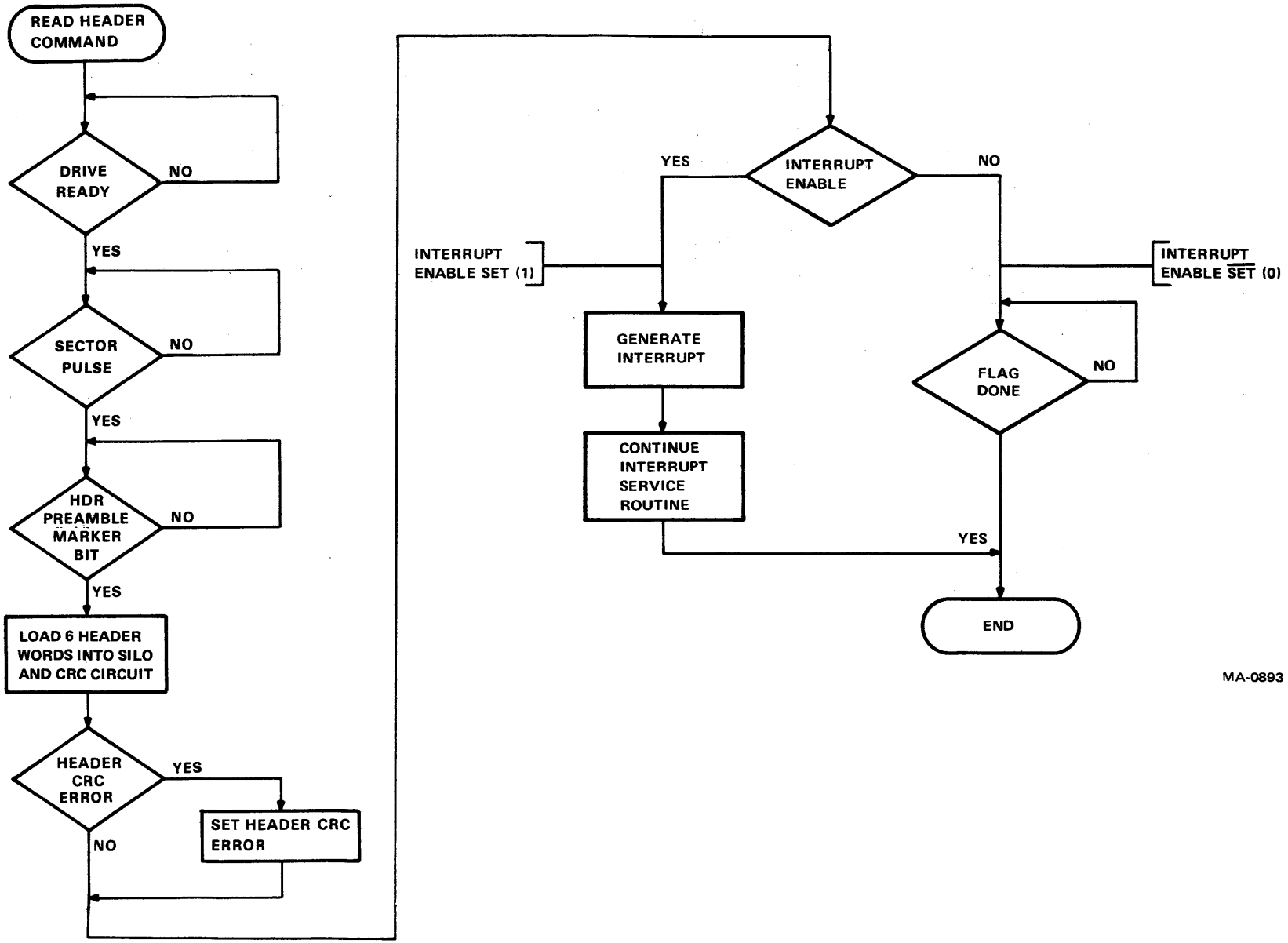


Figure 2-11 Read Header Command, Operational Sequence

2.5.6 Write Data Command

This is a DMA command that transfers parallel data from main memory into the silo and shifts out the data in serial form to the disk drive. Prior to executing the command, it is necessary to set up the following registers:

Break MA register – First memory location from which data will be obtained.
Sector address register – Desired address of first sector to be transferred.
Word count register – Number of data words to be written, entered in 2's complement form.
Command register A – The desired cylinder address from which data is to be transferred.
Command register B – Mode, interrupt, device, EMA function (write).

After the write data command is initiated (Figure 2-12), DMA cycles are executed on the Omnibus immediately. Parallel data words are loaded into the silo. Meanwhile, a search for the correct header is performed by the format control circuit. After a successful header comparison, writing to the disk drive begins. Serial data is shifted out from the silo to the write data encoding circuit before reaching the disk drive.

DMA transfers are continuous until word count overflows. Serial write data is continuous until the silo is emptied. At that point, the command is terminated.

Multiple sector write on one track is executed if the word count is greater than one sector. For a partial sector write, the remaining sector area is zero-filled.

Multiple sector transfers, either read or write, in 12-bit mode should not be attempted because of the odd length word (8-bits) remaining at the end of a given sector. In 8-bit mode, 16 multiple sectors can be transferred.

When this command is decoded the controller begins reading successive header words and comparing them to the sector address register and command register A which contains the cylinder address. When a match is found, the header CRC is checked and if correct, that sector is written with the data starting at the address designated by the break MA register. When writing a partial sector the remainder of the sector will be written with 0s when the word count overflows, an interrupt request from function done will indicate that the command is complete.

2.5.7 Read Data Command

This is a DMA command that transfers serial data from the disk drive into the silo and shifts out the data in parallel form to the main memory. Prior to executing the command, it is necessary to set up the following registers:

Break MA register – First memory location to which data will be transferred.
Sector address register – Desired address of first sector to be transferred.
Word count register – Number of data words to be read, entered in 2's complement form.
Command register A – The desired cylinder address from which data is to be transferred.
Command register B – Mode, interrupt, device, EMA function (read).

After the read data command is initiated (Figure 2-13), a search for the correct header is performed by the format control circuit. After a successful header comparison, DMA cycles are executed on the Omnibus immediately, and parallel data is transferred from the silo to the Omnibus.

DMA transfers are continuous until the word counter overflows. Serial read data is continuous until the DMA is done. At that point, the command is terminated.

When this command is decoded the controller begins reading successive header words and comparing them with the sector address register and command register A which contains the cylinder address and head select information.

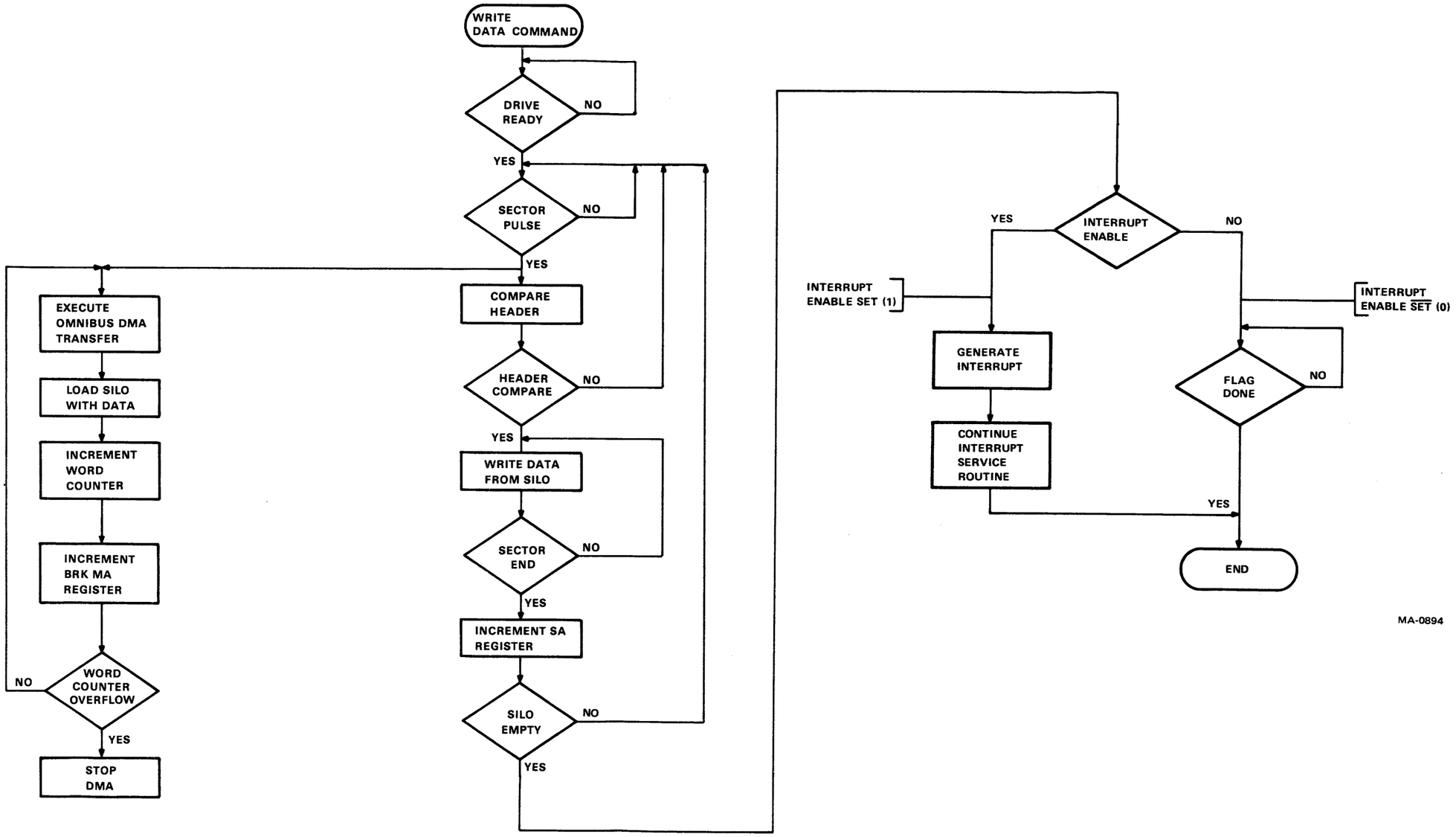


Figure 2-12 Write Data Command, Operational Sequence

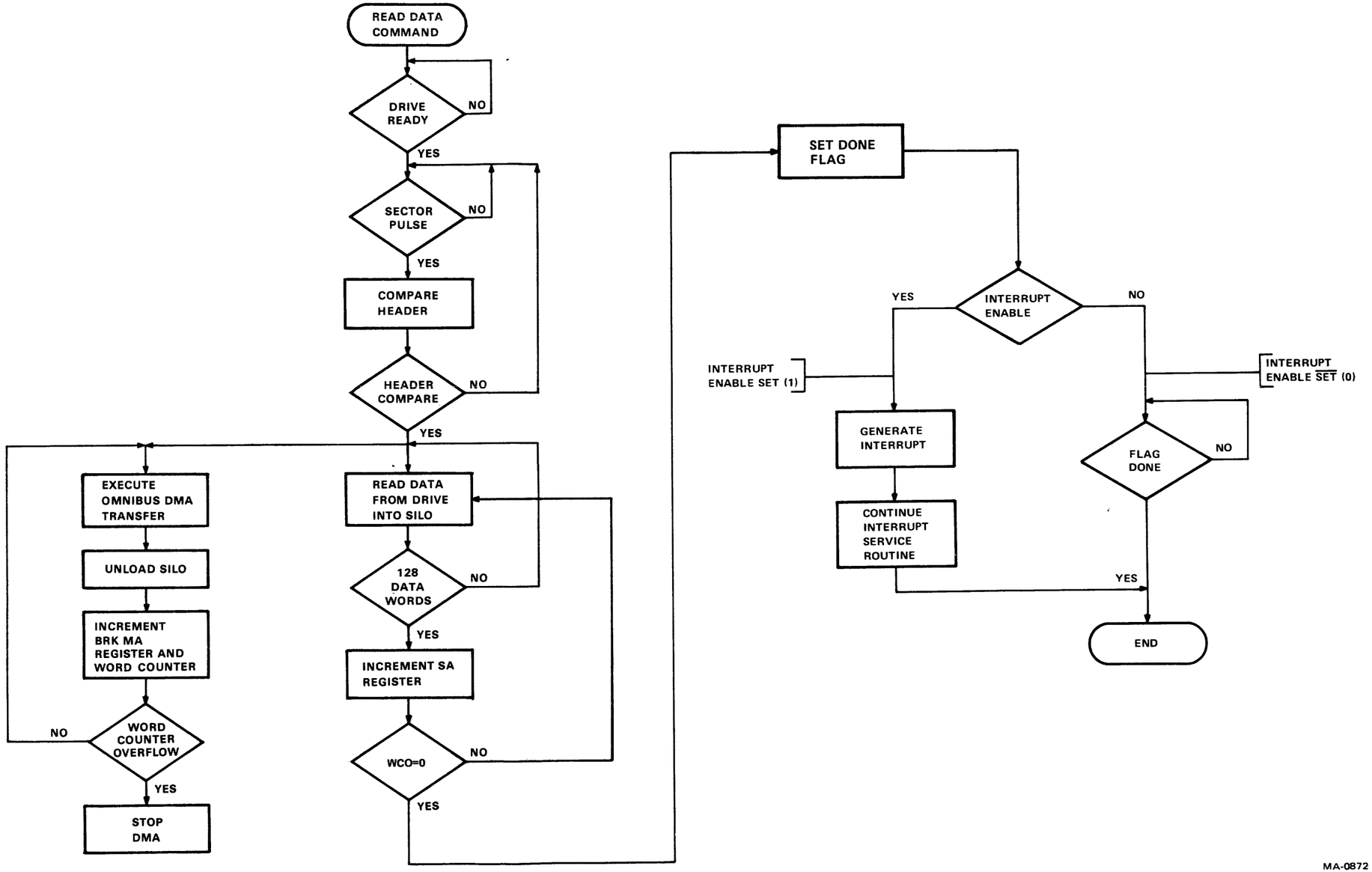


Figure 2-13 Read Data Command, Operational Sequence

When a match is found, the header CRC is checked and if correct, that sector is read. The data being read is placed into memory beginning at the address specified by the break MA register. A function done interrupt will take place at word count overflow time indicating that the transfer is complete.

If a data CRC error occurs, the CRC error flag and function done flag will both be set at the completion of the transfer.

2.5.8 Read Data Without Header Check Command

This command is the same as a read data command, except that no header comparison takes place prior to reading data from the disk drive into the silo (Figure 2-14). The first sector encountered is the one read and shifted into the silo. No check is made for a header CRC error; however, the data is checked for a CRC error.

When this function is decoded the controller reads the data portion of the sector following the next sector pulse. This data is read into the silo then into memory (on second sector pulse) starting at the location specified by the break MA register. The header is neither compared nor checked for CRC errors. Data CRC is checked and the sector address (SA) is incremented. If the word count has not overflowed, the next sector is read, again without checking the header or header CRC. When word count does overflow, a function done interrupt occurs.

2.5.8.1 Track Overrun – If word count overflow has not taken place by the time that the end of the track has been reached, the sector address register will be incremented to 50_8, an illegal sector number, causing an operation incomplete (OPI) time-out error. The sector address register then can be read showing the illegal address.

2.6 I/O TRANSFER OPERATIONS

There are two kinds of I/O transfers that are used to interface the processor with the RL8-A controller: programmed I/O transfers and DMA transfers.

Programmed I/O transfers are executed by IOTs which transfer the contents of the AC to the controller then clears the AC or clears the AC and then transfers the contents of a controller register into the AC.

DMA transfers are the fastest method of transferring data between memory and a device. They can occur between processor bus cycles and do not alter processor status in any way. Blocks of data can be moved at memory speeds via the DMA transfer mode. The read and write data in the controller silo is received and transmitted under DMA control.

Interrupt requests allow the processor to continue a programmed operation without waiting for the controller to become ready to transfer data. When the controller does become ready, it interrupts the processor's background program sequence and causes execution of the controller's service routine. After the controller's service routine has been executed, the background program is restored and program execution resumes at the point where it was interrupted.

2.6.1 Bus Transceivers

These circuits transmit and receive both data and address information on the Omnibus. The device address decoder circuit compares each incoming address with the controller's preset address. When a match is found, the controller is selected.

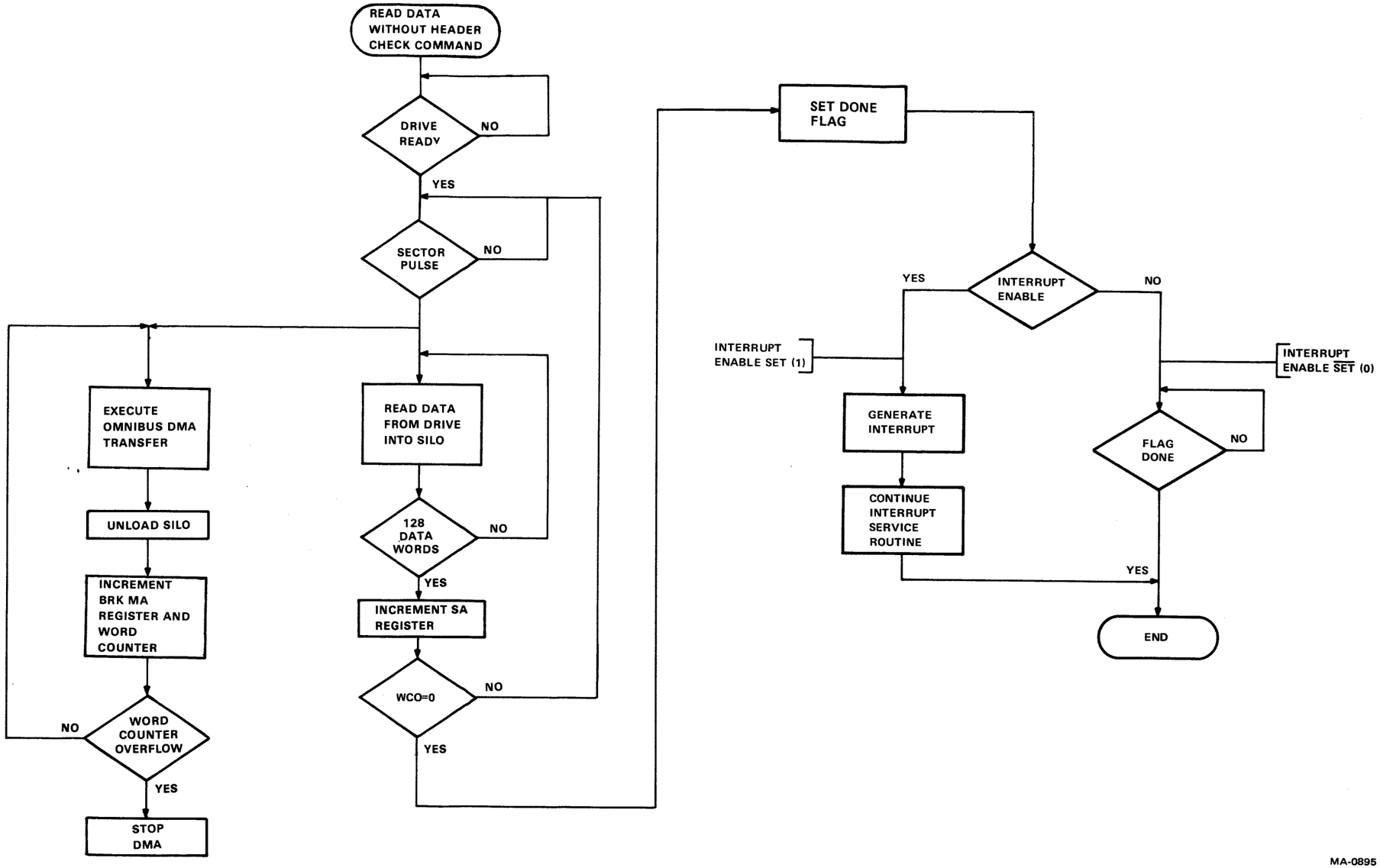


Figure 2-14 Read Data Without Header Check Command, Operational Sequence

2.6.2 Programmable Registers

Logic within the RL8-A controller monitors the MD lines and decodes the information on these lines for major register gating control signals in response to basic instructions. The major registers within the RL8-A that respond to program instructions are as follows.

Command Register A	Contains the desired cylinder address (or difference for a seek) from which data is to be transferred, head selection, and head direction.
Command Register B	Contains the function and format control for the mode (8- or 12-bit), drive select, interrupt enable, extended memory bits and the desired function to be performed.
Break Memory Address	Contains the first memory location to which data will be transferred.
Sector Address Register	Contains the desired address of the first sector to be transferred.
Word Count Register	Contains the number of data words to be transferred to either memory or the disk.
Disk Address Register (DAR)	The DAR contains the first sector and cylinder address where data is to be read or written on the disk. The DAR is also used to store drive command information that is sent to the drive during a seek or get status operation.

2.6.3 Silo

The silo is a first-in first-out (FIFO) memory element that can store up to 16 data words. The silo converts parallel data to serial data for a write operation. The silo also converts serial read data to parallel data.

The silo content is recovered by reading it with the RRSI IOT for the get status and read header commands. This IOT covers the silo data to be transferred to the AC on a word-by-word basis. In other words, to recover the two words of the disk status, two RRSI IOTs must be executed. During read and write operations the silo is emptied and filled under control of the DMA logic.

2.6.4 Microsequencer Logic

The microsequencer first decodes the function command issued, and goes to a particular address in its sequencer ROM. There it finds a routine that corresponds to this particular command. It then proceeds step-by-step to generate all the timing and control signals needed to channel the incoming or outgoing data through all its various flow paths within the controller.

2.6.5 Write Precompensation

This circuit performs two major functions. It encodes digital data into its modified frequency modulation (MFM) form, and it precompensates this data for peak shifting effects.

Modified frequency modulation (Figure 2-15) encoding is a magnetic recording technique used by the RL01 drive. A flux reversal is written on the disk in a center of a bit cell to represent a logical one. To represent two successive logical zeros, a flux reversal is written at this common cell boundary. This recording technique guarantees at least one flux reversal for every two cell bits.

One of the problems associated with magnetic recording is a phenomenon called peak shift. Adjacent flux reversals on a disk appear to be displaced from where they were written.

To offset peak shift, the precompensation logic is used to displace the encoded data pulses in one direction or the other before they are written.

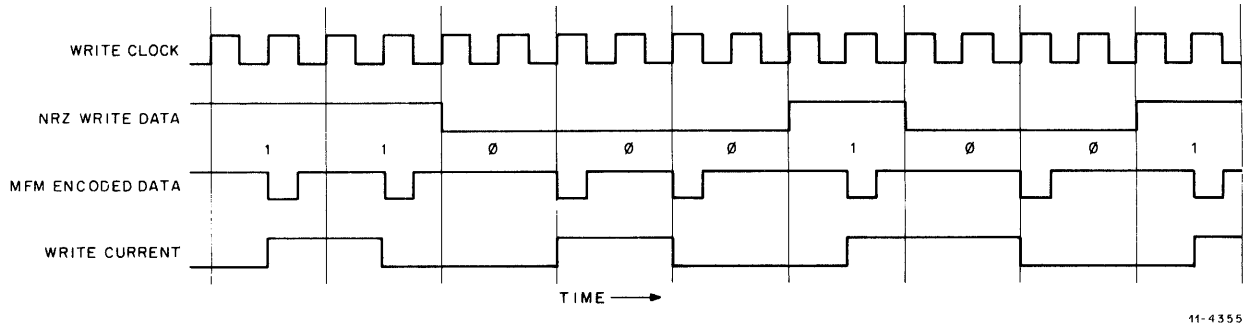


Figure 2-15 Modified Frequency Modulation Waveform Example

2.6.6 Data Separator

The data separator circuit makes use of a phase lock loop oscillator to detect and decode MFM disk data into its digital logical representation. It also generates the timing signals used by the microsequencer to control the read data operations.

2.6.7 CRC Circuit

The cyclic redundancy checker (CRC) is an error detection circuit. For any data written on the disk, a code is generated in the CRC circuit by an internal algorithm. The code is then appended onto the end of each header or sector in the form of a CRC word. When this header or sector is read from the disk, the data is channeled back through the CRC circuit. Any errors introduced into the data or its CRC word will be detected and a CRC error bit will be set in the error register.

2.6.8 Data Source Selector

This circuit allows the multiplexing of different data sources under the control of the microsequencer. There are five different sources of data; CRC data, serial disk address data, serial silo output data, data separator (DS) data, and the write marker pulse.

2.6.9 Header Compare Circuit

The function of the header compare circuit is to compare words coming from the data separator with the serial disk address word coming from the DAR.

This compare is done serially on a bit-by-bit basis. If any pair of bits are not identical, a mismatch signal is generated. If the two words match at the end of a compare, the microsequencer circuit is notified.

CHAPTER 3
INTERFACE DESCRIPTION

3.1 SCOPE

This chapter describes the RL8-A interfaces; from omnibus-to-controller and from controller-to-drive. This includes a description and identification of the interface signals and the operation of the controller registers.

3.2 OMNIBUS INTERFACE SIGNALS

The Omnibus is a parallel bus comprised of 96 signals. The bus is used to connect the processor, memory, peripherals, and options directly into the Omnibus backplane. Figure 3-1 illustrates the Omnibus-controller-drive interface signals and a description of these signals is given here and in succeeding paragraphs.

3.2.1 Omnibus Signal Descriptions

DATA 00–11 – These signals represent the content of the PDP-8/A accumulator register (AC). Information to or from the AC is transferred on the Omnibus data lines. The data bits are strobed into registers in the peripheral (at TP3 or data from a controller register is strobed into the AC) when an IOT instruction is generated. The controller decides data direction by driving the C0 and C1 lines with the appropriate IOT. The data lines are used as the data path between a peripheral controller and memory for DMA transactions into memory during TS2 time.

MD 00–11 – The signals on these lines represent the contents of the memory sense registers. These bits are used during IOT instructions: MD 03–08 carry the device selection code, while MD 09–11 are converted to IOT pulses. The MD lines are also used as the data path for memory to peripheral DMA transactions. The peripheral device should strobe the MD at TP2 time during this DMA transaction.

TS1–TS4 – These signals represent the time states of the CPU. They synchronize operations in the peripheral with those in the computer and perform functions peculiar to the peripheral. They are primarily used in data break timing.

RUN – If this signal is GND, the computer is executing instructions.

SKIP – SKIP is asserted (grounded) by an IOT instruction. It causes the next sequential instruction to be skipped. If the skip bus is asserted during more than one IOT of an I/O instruction, the program skips a corresponding number of instructions. No more than three skips can be made by a single instruction.

INITIALIZE – This 600 ns duration positive pulse is used to clear the link and to clear all flags in peripherals. It is generated at power turn on, and by the clear all flags (CAF) IOT, 6007.

BRK IN PROG – This signal provides indicator information to the console. It is grounded at INT STROBE H leading edge if a break is to take place, and asserts the console BRK IN PROG bit of the major state register.

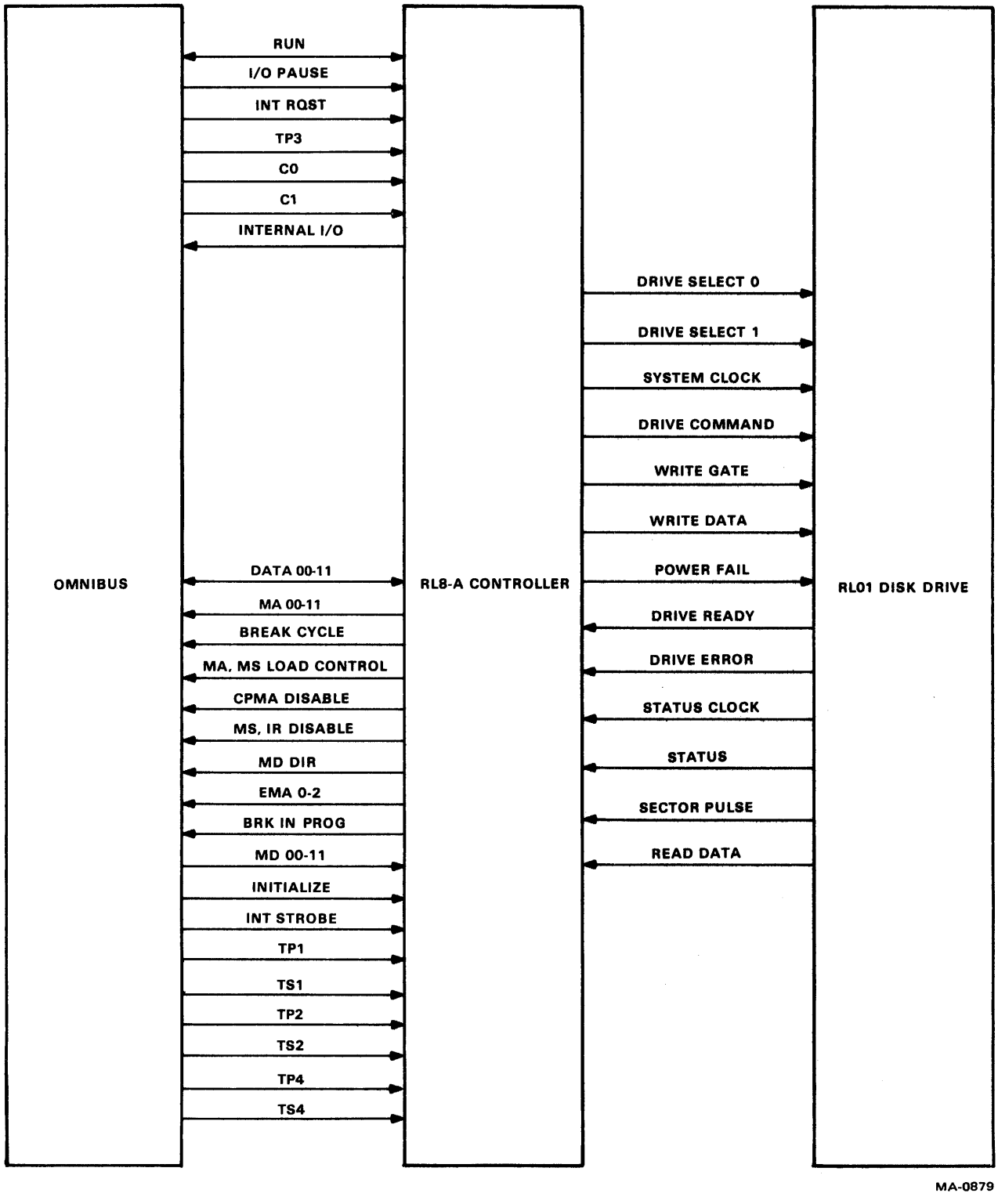


Figure 3-1 Omnibus/Drive Interface Block Diagram

CPMA DISABLE – CPMA DISABLE L is asserted (low) by break devices if data breaks are to occur. It is sampled by the CPU and memory extension control at the leading edge of TP4 H. If CPMA DISABLE L is low at that time, the memory extension control field bits and the CPU memory address bits are removed from EMA<0:2>L and MA<0:11>L. If CPMA DISABLE L is negated (high) at the leading edge of TP4 H, the memory extension field bits and the CPU memory address register are gated to EMA<0:2>L and MA<0:11>L, respectively.

MS, IR DISABLE – When MS, IR DISABLE L is high, the major state and IR flip-flops drive the major state and IR lines on the Omnibus (Paragraph 3.8.1). When MS, IR DISABLE L is asserted (low), the major state and IR lines are not driven by the CPU. Unless some external device asserts a major state, the CPU is then in the DMA state.

MA MS LOAD CONT – When MA, MS LOAD CONT L is negated (high), the CPU and memory extension control function normally. Asserting this line inhibits the loading of new information into the CPU's major state and memory address registers, and into certain control flip-flops of the memory extension control. This signal must not change while TP4 H is high. It is normally changed at TP1 time.

BK CYCLE – Panel information from the data break device. Low indicates that a break data transfer cycle is in progress.

RUN L – This signal is really a CPU state since it indicates that the CPU timing generator is running (when low). It is included in this group of signals because its most important function is as a gating term used to clear all break requests when negated (high).

EXT MEMORY ADD 0–2 – These three lines are used when a KM8-AA memory extension and timeshare interface is included in the PDP-8/A. The peripheral uses the lines to indicate the particular memory field involved in the transfer beyond the first 4K bank of memory.

3.3 DRIVE BUS INTERFACE SIGNALS

The RL8-A communicates with the disk drive(s) via the drive bus interface. The RL8-A can control up to four RL01 disk drives through this interface. The bus is comprised of 12 differential twisted pair signals and a single-ended power fail line. The drive bus signals are described below.

3.3.1 Drive Select

These two lines select one of four disk drives, as determined by bits 4 and 5 of command register B. The drive must be selected before a write gate or serial drive command word is sent to the drive. One drive is always selected even though the controller is idle. Only the selected drive asserts the drive-to-controller interface lines, and these lines are valid after the drive has been selected. A drive will inhibit transmission of a partial sector pulse if it is selected while its sector pulse is asserted.

3.3.2 System Clock (SYS CLK)

This is a free-running clock that shifts the drive command word to the drive and also operates the disk motor servo. Clock frequency is 4.1 MHz ± 0.1%.

3.3.3 Drive Command (DR CMD)

This line is used to transfer control and cylinder address difference information serially to the drive. It is only enabled during seek or get status or reset commands.

3.3.4 Write Gate (WR GATE)

This line enables the write circuits in the selected drive. It must be asserted at the start of preamble PR2, and must precede the first bit of write data. Write gate must not be asserted during a sector pulse; otherwise, a write gate error will be asserted by the drive (bit 9 of word number 2 of the disk status register) and operation terminated. Write gate is removed at the end of postamble P02.

3.3.5 Write Data (WR DATA)
This line contains the serial data, encoded in modified frequency modulation (MFM) pulse form, that is to be written on the disk. The data stream between sector pulses must contain three preamble words (PR2), 128 data words, the data CRC word, and one postamble word (P02).

3.3.6 Power Fail (PWR FAIL)
This signal is received by all drives at all times, regardless of which drive is selected. When the subsystem is first powered up, power fail is not asserted. If AC power is subsequently lost or out of tolerance, power fail is asserted low, in which case the drives unload heads and power down. Return of power causes the drives to power up and load heads over track 0.

3.3.7 Drive Ready (DR RDY)
When asserted, this signal indicates that the selected head is centered on the track, and the drive is ready to receive a command. The signal goes low: (1) after the last bit of a cylinder address difference word has been shifted to the selected drive; (2) when a new head select is transferred to the drive, even though the address difference is zero; and (3) when there is an address difference of zero and no head select change is shifted to the drive.

Drive ready will be low when a drive error occurs. The one exception is when an attempt has been made to write on a write-protected drive. In that case, only drive error will be asserted high.

3.3.8 Drive Error (DR ERR)
This signal is asserted high on certain drive errors. Any attempt to write on a write protected drive also causes the signal to be asserted high. Asserting drive error causes bit 11 of the disk status register to be set. The particular error involved can then be determined by initiating a get status command and reading bits 4, 5, 6, 7, 8, 9, and 10 of the disk register status (word 2).

The drive error latch can be reset by:

1. Manual disk power down/power up

2. Setting bit 3 of the serial drive command (bit 3 of the disk address register during a get status command)

3. Removing the write lock condition via the drive front panel.

3.3.9 Status Clock (STATUS CLK)
This clock is the system clock delayed through drive logic and returned to the controller when a status word is requested. The clock is turned on in sync with the first bit of the status word and remains in sync until: (1) a new drive command marker is received at the input to the drive command shift register, or (2) the drive is deselected.

3.3.10 Status (STATUS)
In response to a get status command, the drive enables status clock and sends the status word to the controller via the status line. This function can be performed even though drive ready is not present (i.e., during power-up or a seek).

3.3.11 Sector Pulse (SEC PLS)
This 62.5 ± 2 μs pulse is asserted high and occurs every 625 μs ± 1% or 40 times per disk revolution. When a drive is initially selected, it must wait until the next full sector pulse is detected before sending the sector pulse to the controller.

3.3.12 Read Data (RD DATA)

This line transfers MFM encoded data from the drive read circuits to the controller. Whenever a drive is selected and drive ready is asserted, read data appears on this line, except when write gate is asserted.

The drive senses the amplitude of the header preamble and sends read data over the read data line 2.5 ± 10.5 μs downstream from where the preamble actually starts.

For reading headers, the VFO loop is phase locked with the arrival of read data after the end of the sector pulse. For the data preamble, the VF0 locks 32 read pulses after the header CRC to avoid transmitting erroneous sync pulses to the clock at the transition between the prerecorded header and the data preamble. Detection of the preamble marker commences after the VFO has had time to phase lock.

CHAPTER 4
FUNCTIONAL DESCRIPTION

4.1 SCOPE

This chapter contains a detailed functional description of the RL8-A controller commands and logic plus a description of the sequence of operations between the Omnibus-controller-drive (Figure 4-1).

4.2 DMA CONTROL

Data breaks (DMA) allow a peripheral to communicate directly with memory, bypassing the CPU. The only CPU register available to the DMA device is the MB; all other CPU registers are preserved. Data breaks may occur between any CPU cycles, but a data break cannot be performed while the CPU is in the midst of an extended I/O cycle because the currently active memory is then supplying information to the bus.

Data breaks take place in the following sequence:

1. At INT STROBE H leading edge, the decision to request a data break is made at the RL8-A controller by strobing its break request flip-flop.

2. A device starting a data break unconditionally asserts CPMA DISABLE L and BREAK IN PROG L until its break request flip-flop is cleared.

3. Data break priority is determined on DATA<0:11>L when TS4 L is asserted. Each data break device is assigned a unique line on DATA<0:11>L, with DATA L being the highest-priority line. Each requesting device asserts its line on DATA<0:11>L, and examines the state of all higher-priority lines to determine if they are all high. For example, a device asserting DATA5 L examines DATA<0:4>L to make sure these lines are all high. The device finding all higher-order lines high proceeds to step 4, all other devices remain in step 3.

4. The device winning the priority test sets its MA control flip-flops at the leading edge of TP4 H. The path from TP4 H to the memory address lines must have as little delay as possible, hence two flip-flops are recommended to provide adequate drive without introducing the delay of a buffer. The MA control flip-flops gate the break address onto the 15 memory address lines and assert MS, IR DISABLE L and (if this cycle is a data exchange cycle) BK CYCLE L.

5. At TP1 H, the active device asserts MA, MS LOAD CONT L.

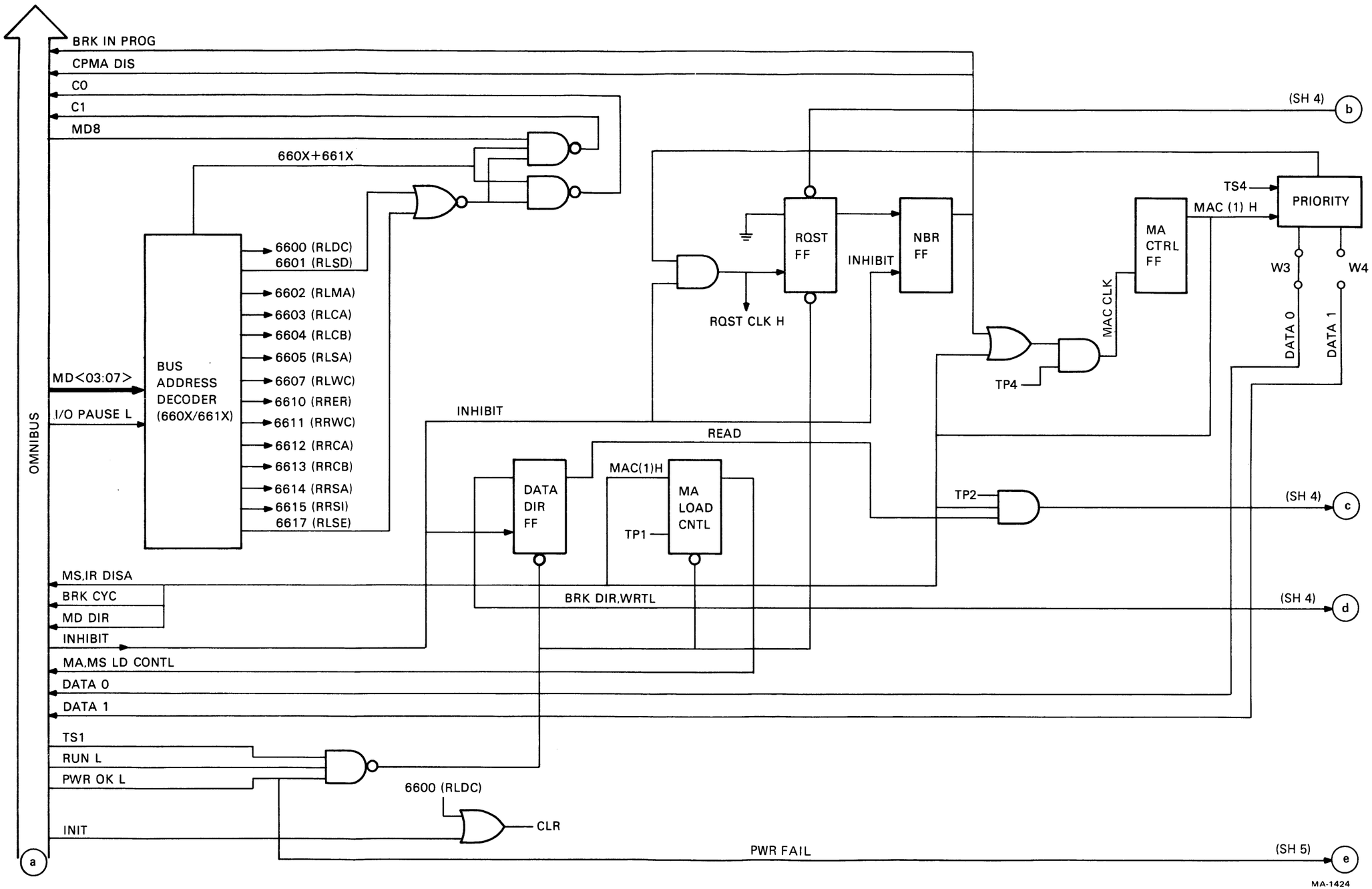


Figure 4-1 RL8-A Controller Detailed Block Diagram (Sheet 1 of 5)

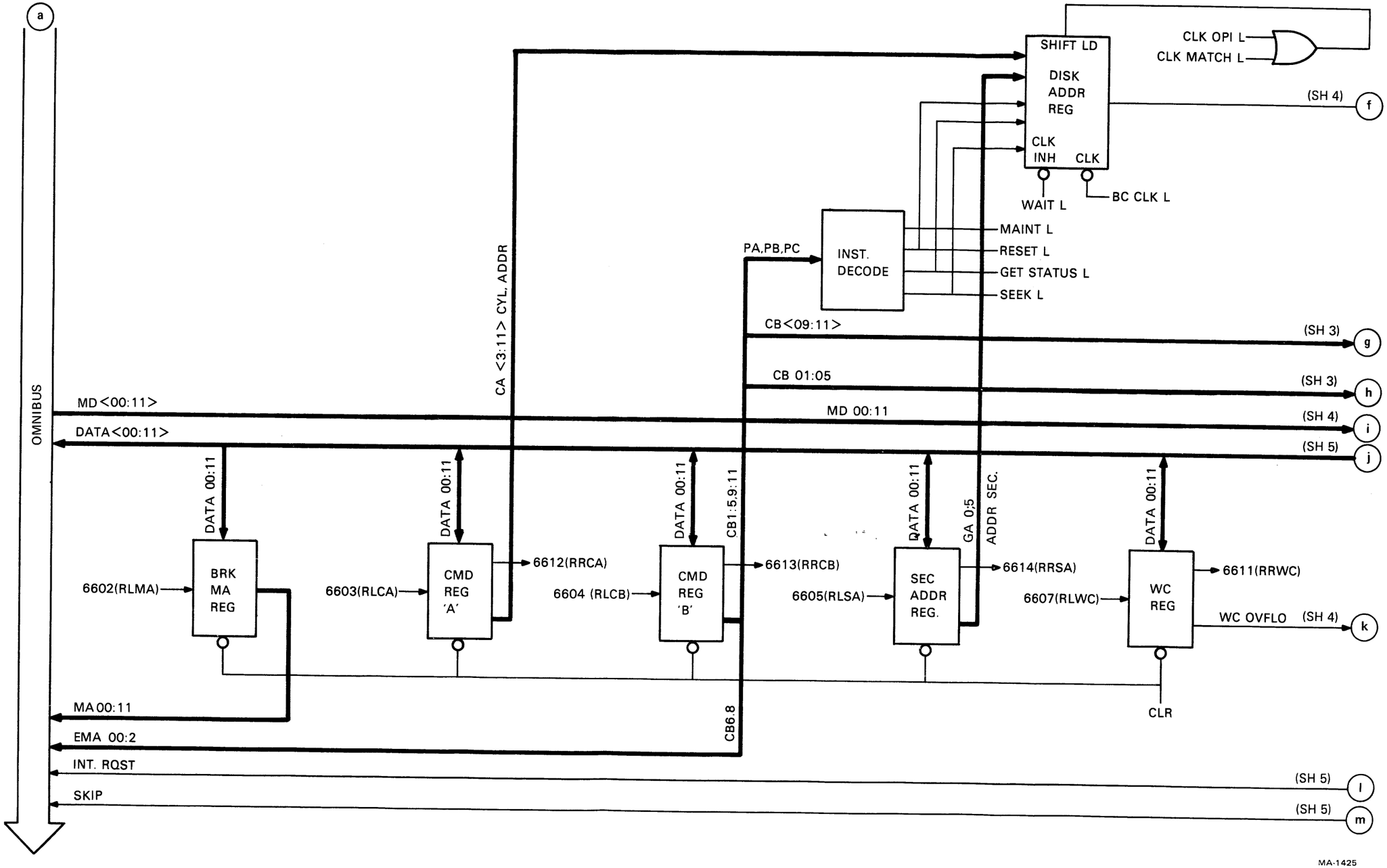


Figure 4-1 RL8-A Controller Detailed Block Diagram (Sheet 2 of 5)

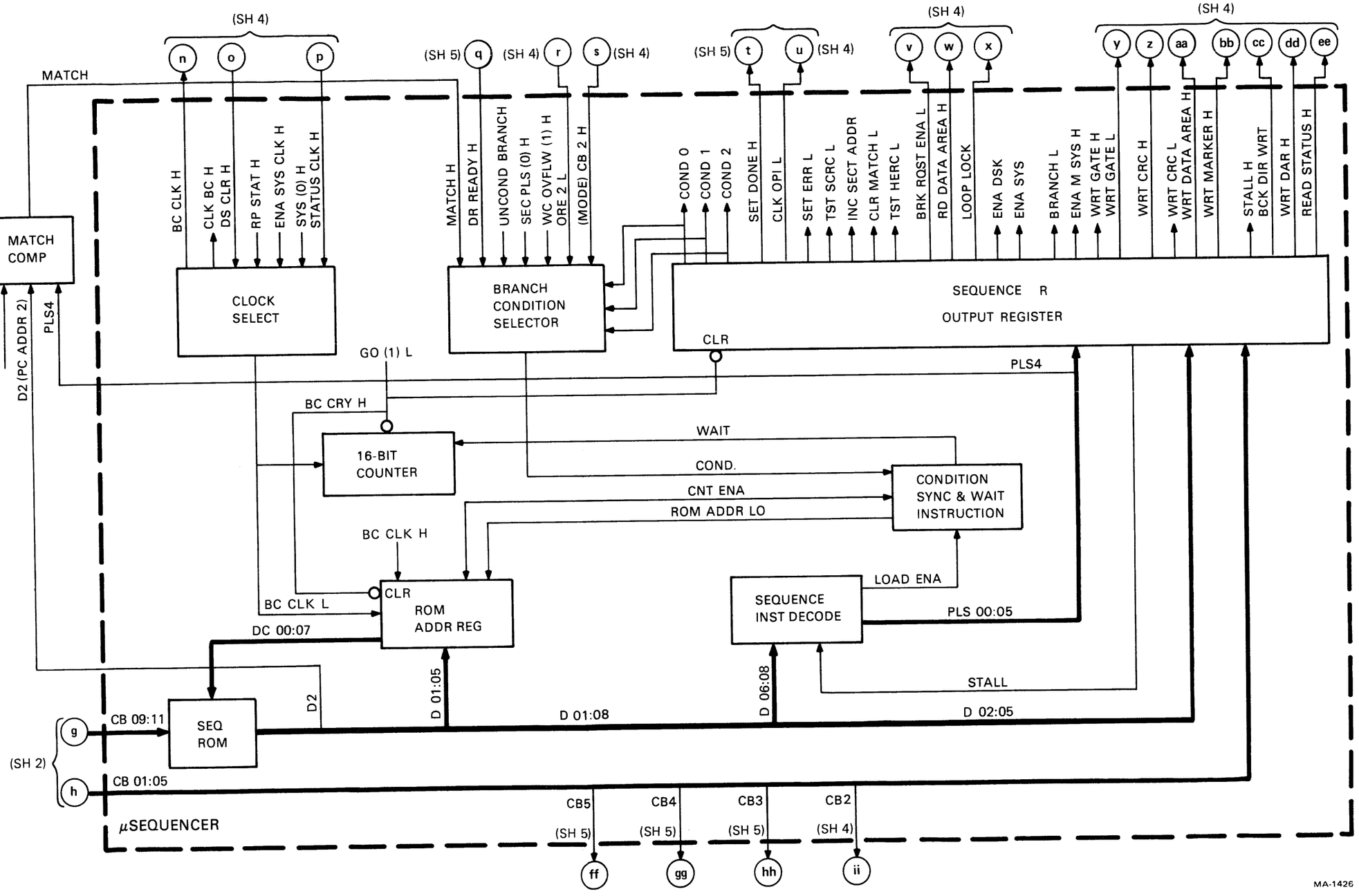


Figure 4-1 RL8-A Controller Detailed Block Diagram (Sheet 3 of 5)

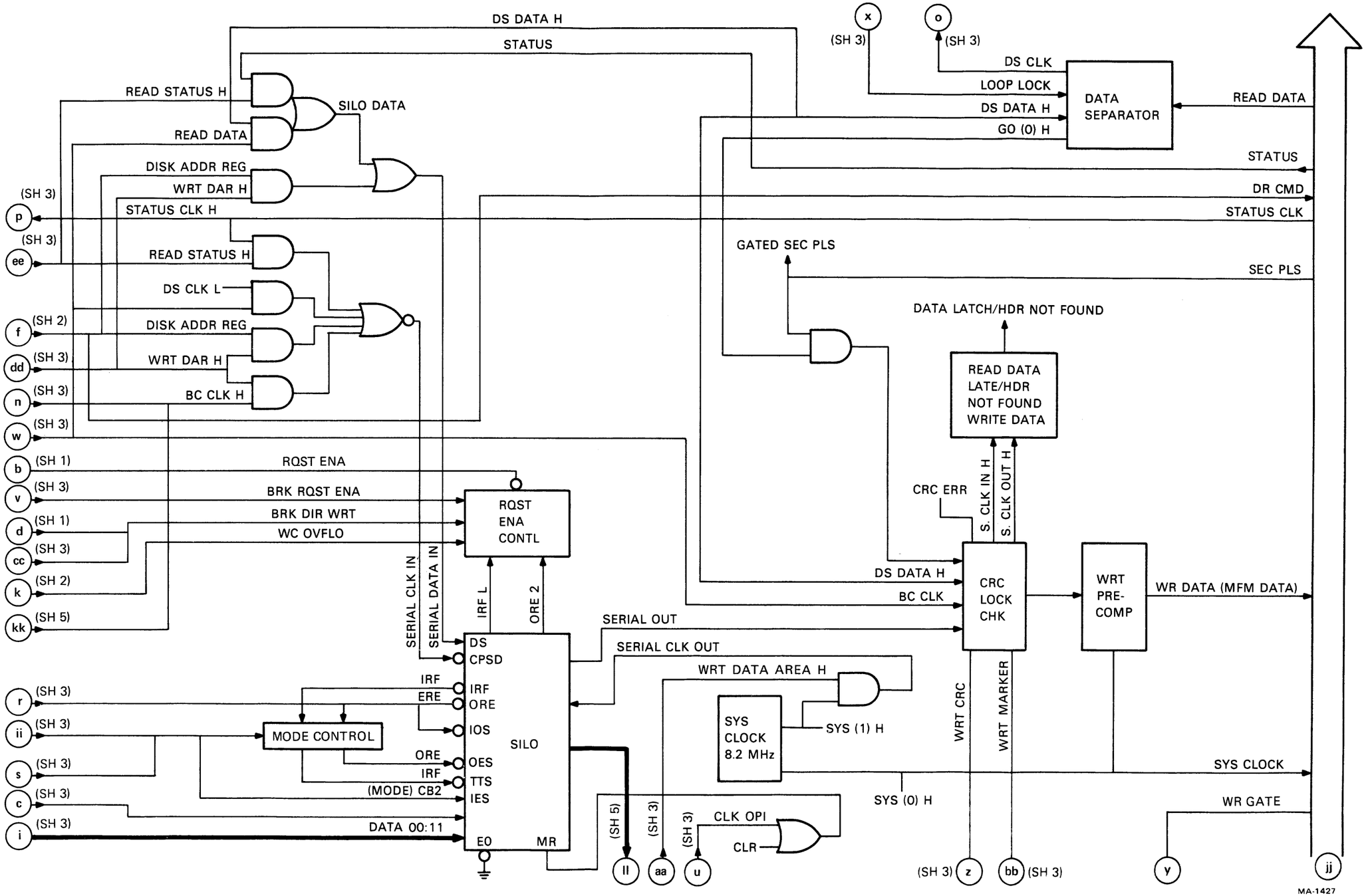


Figure 4-1 RL8-A Controller Detailed Block Diagram (Sheet 4 of 5)

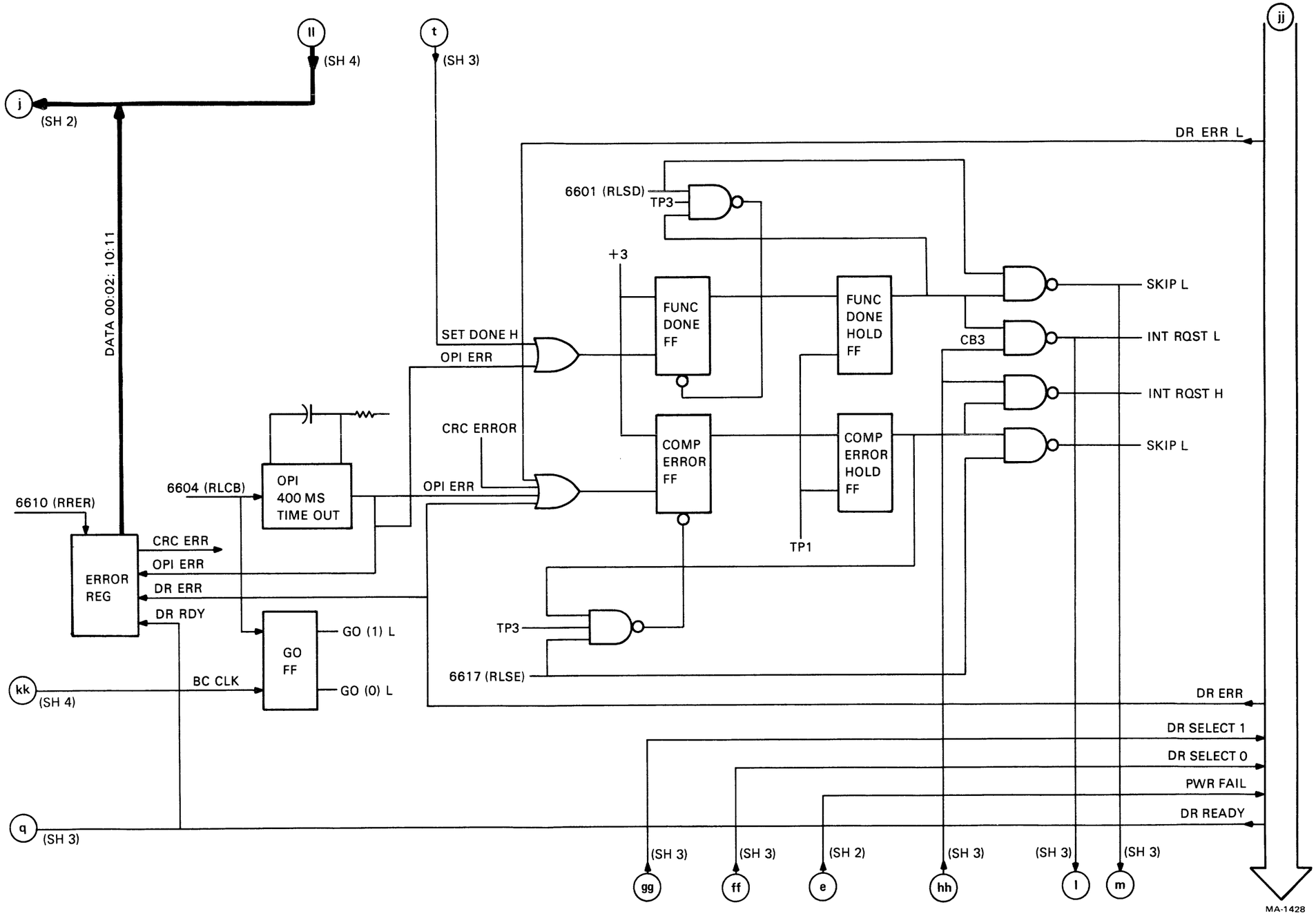


Figure 4-1 RL8-A Controller Detailed Block Diagram (Sheet 5 of 5)

6. For input to memory, the active device merely places its data on DATA<0:11>L during TS2. For output from memory, the device asserts BREAK DATA CONT L during TS2 L and loads its register with the contents of MD<0:11>L at the leading edge of TP2 H, TP3 H, or TP4 H. For add to memory, the device places the data to be added on DATA<0:11>L during TS2, and asserts BREAK DATA CONT L during TS2. The modified information is loaded into the CPU's MB and overflow flip-flop at TP2 H and may be read by the device at TP3 H or TP4 H. (TP3 H is generally used, since OVERFLOW L is valid only at this time.) The data prior to modification may be read at the leading edge of TP2 H.

7. At the completion of a data break, all lines are in the same order and at the same times in which they were asserted.

   DMA latency is the longest machine cycle, plus the time of TS4.

4.3 MICROSEQUENCER DESCRIPTION

The internal operations of the RL8-A controller are under the control of the microsequencer. At the heart of the microsequencer circuit is the sequencer ROM. The ROM contains all the preprogrammed instructions routines to control the eight different command sequences. Each command sequence fits within 200 octal locations of ROM memory space. There is much more to the microsequencer than just the ROM. Figure 4-2 shows all the other functional elements that are incorporated in the microsequencer circuit.

The clock select circuit is used to choose between three possible clock inputs. These are the data separator clock (DS CLK H), the internal crystal system clock (SYS 0H), and drive status clock (Status CLK H)> The data separator clock is used whenever data is being read off the disk and goes through the data separator circuit. The system clock is derived from an 8.2 MHz crystal. It is divided down to 4.1 MHz before it is used to clock drive command words over to the drive. This system clock is used by the microsequencer to generate timing when the DS clock and status clock are not in use. The drive status clock is sent over from the drive during the transmission of a drive status word. The output of the clock select circuit is the BC CLK L signal. This timing signal is used to clock the bit counter, the count enable circuit, the word counter, and the conditional sync and wait instruction circuits.

4.3.1 Branch

The branch condition selector, as its name implies, is used to select one of seven different branch conditions to be monitored. Programmed into the ROM microcode, are certain branch conditions to be looked for. If the given condition is not present, the branch will not take place and the microcode continues on. A good example of this is the sector pulse condition (SEC PLS 0H). During a read header command, the controller must wait for the arrival of the sector pulse before it can start reading the header. Similarly, the drive ready signal (DRIVE READY H) is a prerequisite for most of the commands. It is the function of the branch condition selector to select the actual condition that needs to be monitored.

This selection process is determined by the three condition lines (CONN 2:0 H), that originate at a sequencer output register. When a desired condition is to be monitored, this output register will present the correct code. The three condition lines CONN 2:0 H and assert the BRANCH L signal to the condition sync and wait instruction circuit. The BRANCH L signal will disable the bit counter until one of the branch conditions are received. This is accomplished by the wait instruction circuit. With the arrival of the branch condition signal, WAIT L is asserted allowing the bit count to start counting.

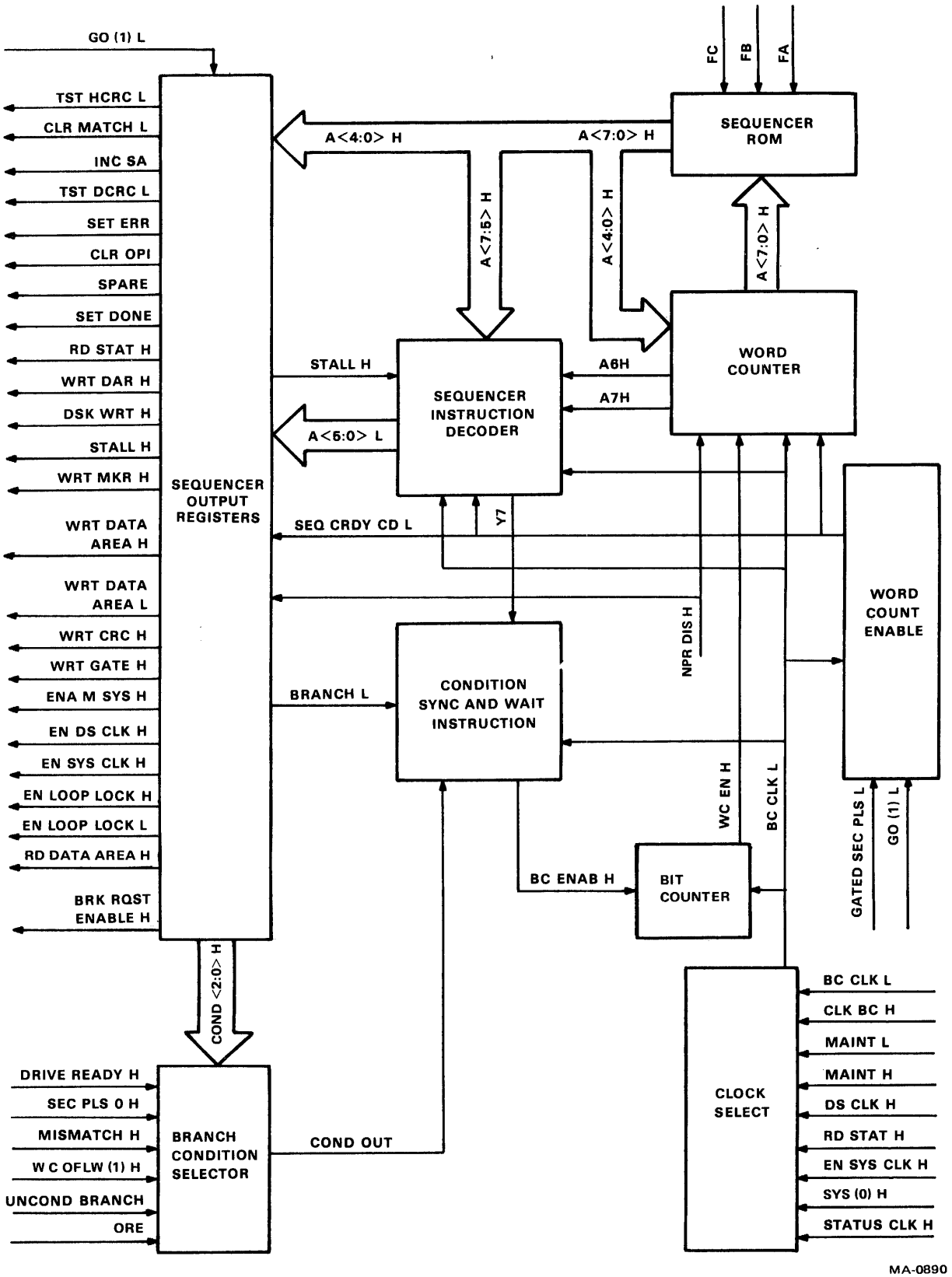


Figure 4-2 Microsequencer Detailed Block Diagram

4.3.2 Skip

The bit counter counts 16 bits per word and overflows. This overflow is indicated by the assertion of BC CRY. This signal drives the ROM address register (word counter), which in turn drives the sequencer ROM allowing it to change only at word boundaries.

The word count enable circuit monitors the controller ready state via the GO flip-flop. If the controller is not in the ready state, it enables the word counter to continue counting.

When the controller becomes ready it clears the word counter.

The word counter is what sequences the ROM through its routine by providing successive addresses. Each command routine is allowed only 200 octal locations. A unique addressing scheme is used here. After the word counter reaches the first 100 octal addresses, a stall instruction is executed. This stall instruction asserts the STALL H signal to the sequencer instruction decoder. The decoder will now wait until the word counter counts to 300 before resuming operation. Octal 300 looks like 100 to the ROM because it doesn't see the most significant bit. Thus when the decoding operation resumes, the ROM will finish its second half of the 200 octal locations. This stall feature is used to count "words" as they are written or read from the disk. The count equals exactly one full sector of data, 2048_{10} bits.

The sequencer ROM program was designed to fit within 1K × 8 memory space. As already mentioned, each of the eight command routines may occupy up to 200 octal locations. Exactly which 200 block being addressed at any given moment, is determined by the three function signals (FA, FB, and FC). They decide the starting address in multiples of 200.

Of the eight output lines from the ROM, the three most significant bits are sent to the instruction decoder circuit. The remaining instruction lines are used to set up a group of six output registers. These output registers provide the majority of the control signals for controller operations.

The major function of the instruction decoder, besides executing the stall instruction, is to select which output register to load the ROM signals into.

4.4 CONTROLLER COMMANDS FUNCTIONAL DESCRIPTION

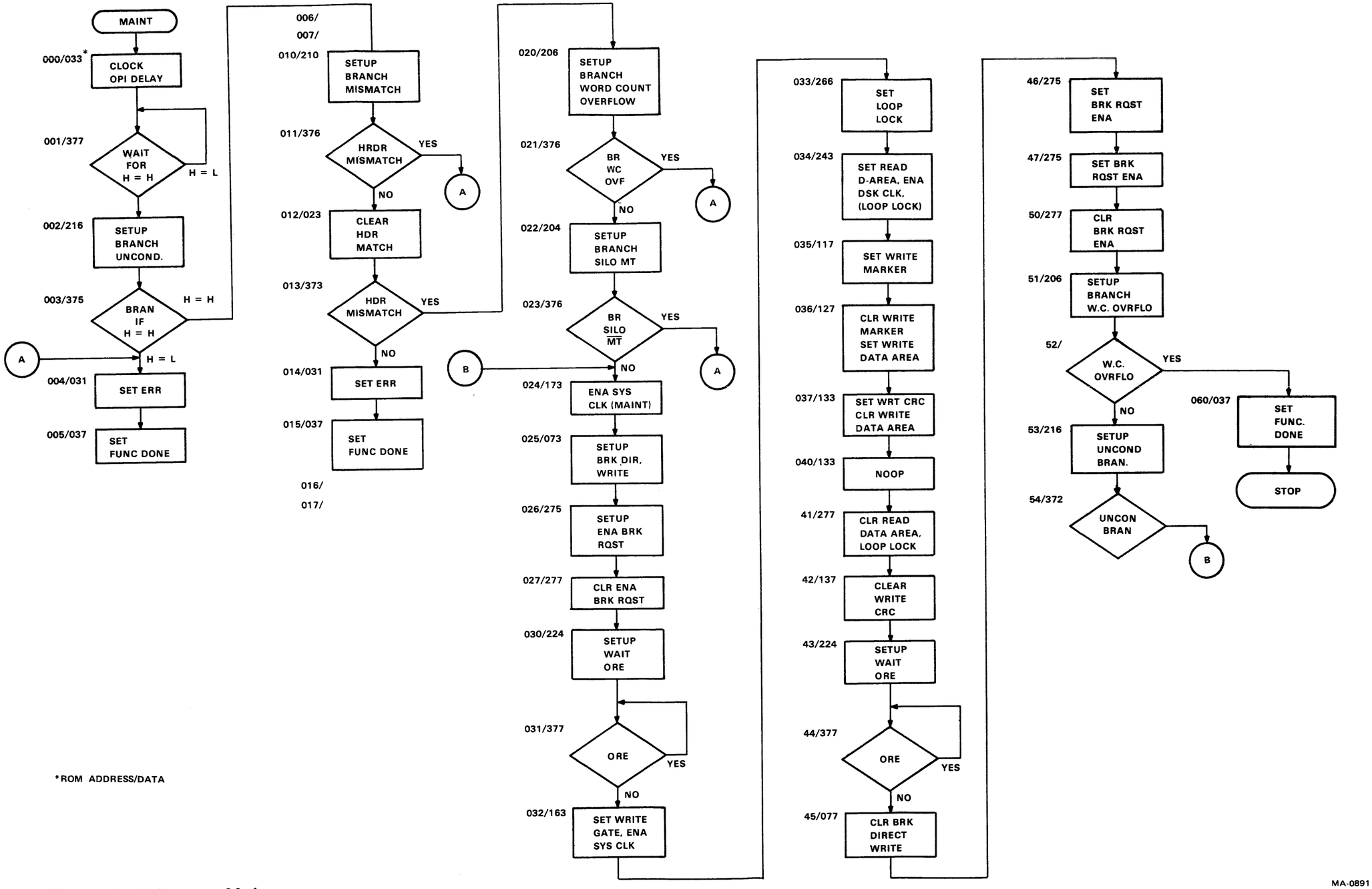
4.4.1 Maintenance Command

The maintenance mode command is used during a diskless diagnostic routine to detect controller malfunctions, or to establish a level of confidence in controller operation. This description of the command operation will follow the maintenance mode functional flow diagram (Figure 4-3). Prior to issuing the maintenance command a buffer area in memory must be set aside to read and write the silo contents.

Registers to be loaded:

Break MA Register
Word Count Register
Command Register B

Upon issuing the maintenance command, the operation incomplete (OPI) timer is started. The microsequencer decodes the command and starts a maintenance routine. This routine begins by enabling a DMA transfer to take place between memory and the controller silo. The silo serializes this one word and sends it through the write precomp and CRC logic. The serial stream is now directed through the data separator logic to recover digital data from the modified frequency modulation (MFM) stream. The digital data is returned to the silo where it is converted to parallel format.



*ROM ADDRESS/DATA

MA-0891

Figure 4-3 Maintenance Mode Flow Diagram

Next, the parallel data word and the CRC words are sent back to memory via the DMA facility. This process is repeated until the word count register overflow terminates DMA requests. When the maintenance command is decoded, one DMA will take place from memory at the location specified by the break MA and a write data function will be executed out of the silo. The written data will be loaded back into the silo along with the CRC character (two 8-bit words). Three DMAs will take place and cause the data word (8- or 12-bit) and the two CRC words to be written into memory. This sequence is repeated until the word count register overflows and sets the function done interrupt. If there is a CRC error, it and function done will immediately cause an interrupt.

As a result of the maintenance test, the following circuits will have been tested: the silo, the registers, the data source selector, the CRC circuit, the write precompensation circuit, the data separator circuit and the silo input and output serializer. Also many of the microsequencer functions (i.e., unconditional branch, header compare, word count overflow, and silo empty) will have been exercised.

4.4.2 Reset Command

When the reset command is executed, all error bits in the drive are reset (Figure 4-4). This command does not clear any registers in the controller nor does it cause the heads to move to track 0. The sector address and command register A must be cleared prior to the execution of this command.

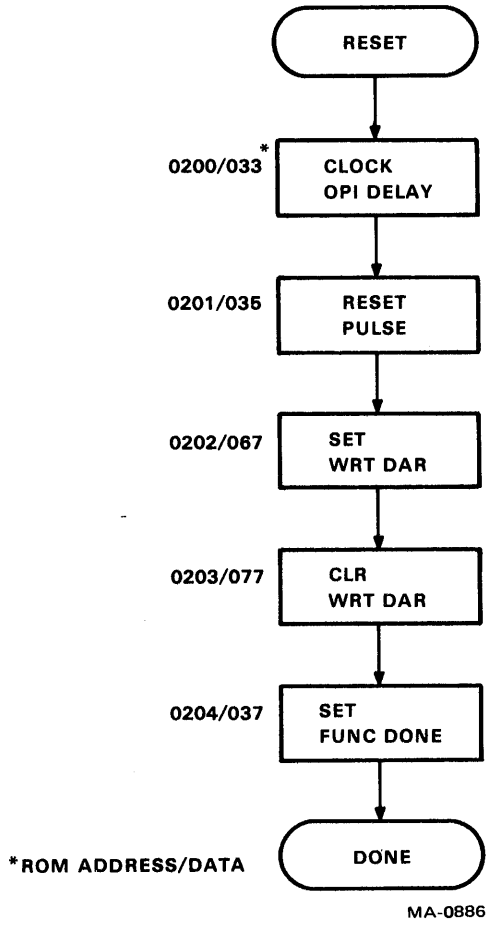


Figure 4-4 Reset Mode Flow Diagram

4.4.3 Get Status Command

The get status command is used to determine the drive's operational status. It involves sending a drive command word out from the controller to the drive and then receiving a status word back from the drive. Refer to Figure 4-5 for the functional flow diagram.

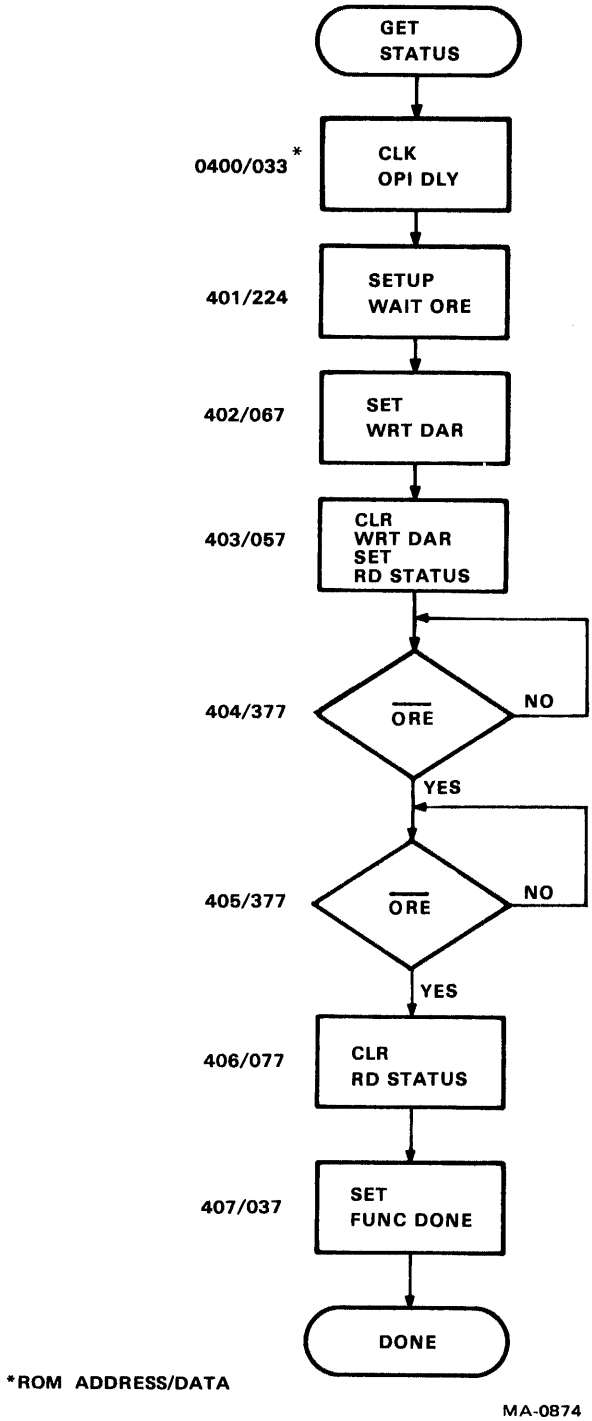


Figure 4-5 Get Status Mode Flow Diagram

The only prerequisite for this command is to know that the controller is in the ready state. Command register B is loaded with the get status command which causes the OPI timer to be initiated and the microsequencer to decode the function command, locate the starting address of the get status routine in the sequencer ROM and proceeds step-by-step through the program.

Serial data out from the DAR is gated directly from the disk bus driver with the send drive command signal from the microsequencer. This drive command word is synchronized with the timing on the system clock line.

Once the drive receives the get status request word, it begins sending back the drive status word. If the reset bit in the get status request word was set, the drive will clear all soft errors (error conditions not still present) first before sending back the drive status. The drive status word is sent back to the controller over the status line in sync with the status clock timing. A READ STATUS signal is generated by the microsequencer that enables the drive status word to reach the silo. The function done is set and the interrupt control circuit will then send an interrupt request to the CPU if the interrupt enable bit is set. The drive status can then be read out of the silo using the RRSI IOT.

4.4.4 Seek Command

The seek command is used to select the drive heads or to reposition them at a new cylinder location. Normally the seek command is preceded by a read header command to obtain new head positioning data. From this information software computes a difference address word that is written into the CA register prior to issuing the seek command. Refer to Figure 4-6 for this function. If the controller is in the ready state, a seek command may be written into command register A. The OPI timer is then initiated and the microsequencer decodes the seek command function. The microsequencer locates the starting address of the seek routine in the sequencer ROM and proceeds to step through the routine.

The next sector pulse that comes along will begin the transfer of the DA register contents to the drive. The difference address word in the DAR will get serialized and pass through the data source selector circuit. A SEND DRIVE COMMAND signal will channel this difference address word out of the controller and to the drive over the drive command line. This drive command word is in sync with the system clock timing. Once this drive command word is received by the drive, the drive then proceeds on its own to reposition the heads.

The controller does not wait for the drive to reposition the heads. After the drive command word is sent, function done is set and the interrupt control circuit issues an interrupt request to the CPU if the interrupt enable bit is set.

4.4.5 Read Header Command

The function of the read header command is to read the first header encountered on the selected drive and store the information in the controller silo. Refer to Figure 4-7 for the functional flow diagram. When the controller is ready, a read header command may be loaded into command register B. At the same time, the OPI timer is initiated, and the microsequencer decodes the read header function. If the drive is ready, the next sector pulse that comes along will initiate the read header sequence. Three header words will be read off the disk and enter the controller pulse discriminator circuit over the READ DATA line. From here, the header words will pass through the data separator circuit and into the silo.

This same information will also enter the data source selector and the CRC circuit to be checked for errors. If a CRC error is detected, the CRC and OPI error bits will be set. If no CRC error is detected function done is set and the interrupt control circuit will send an interrupt request to the CPU if the interrupt enable bit was set.

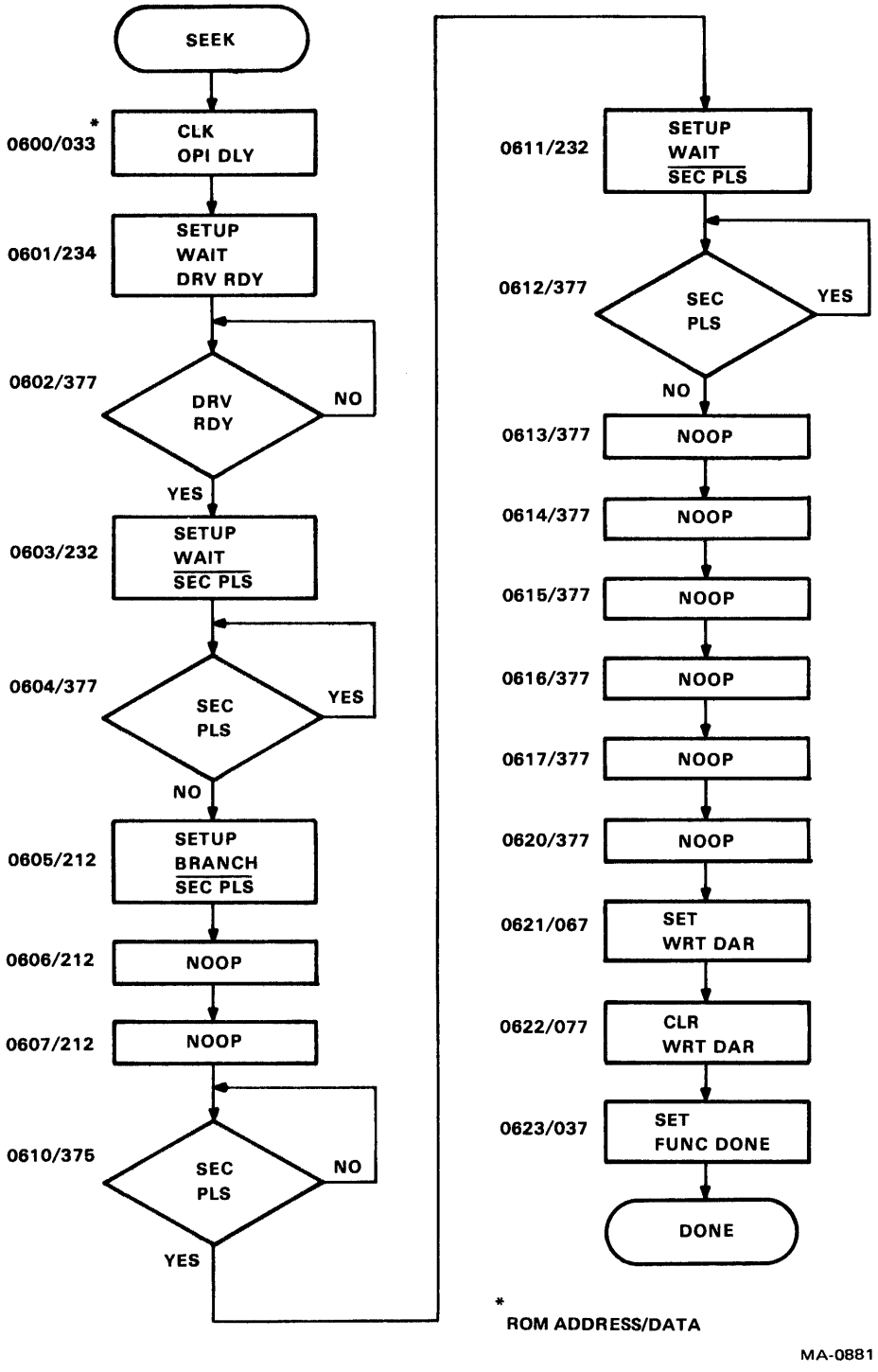


Figure 4-6 Seek Mode Flow Diagram

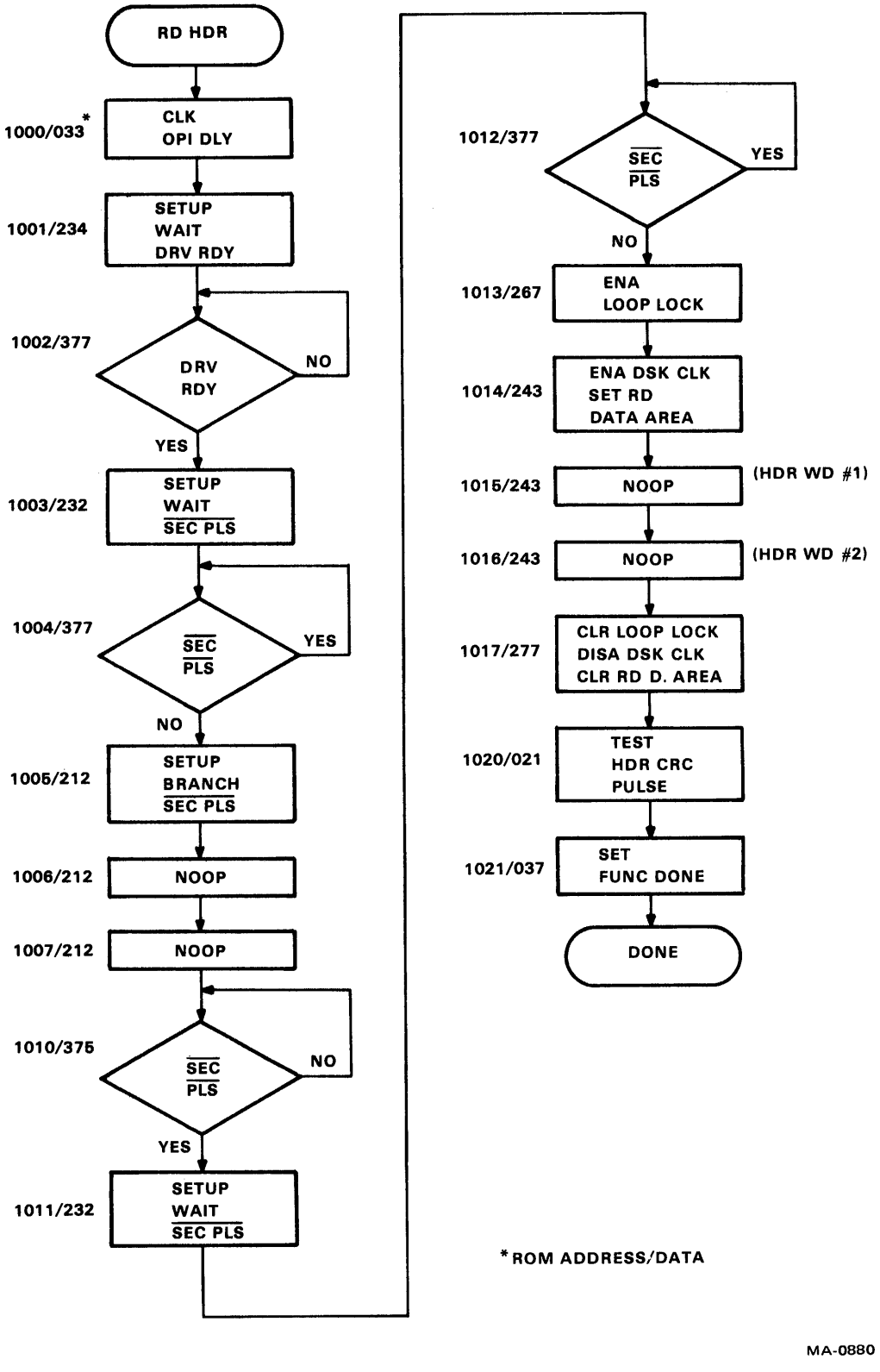


Figure 4-7 Read Header Mode Flow Diagram

4.4.6 Write Data Command

The write data command is used to write data from memory onto the disk. It is normally preceded by read header and seek commands to position the heads over the desired track. The data is written in sector blocks of 128 12-bit words or 256 8-bit words with partial sectors getting zero filled. Refer to Figure 4-8 for the functional flow diagram that illustrates this operation.

Prior to issuing the write data command, the break MA register must be loaded with the address of the first memory location. The word count (WC) stores the number of words to be transferred and the DA register contains the cylinder address of the first sector to be written on the disk. The sector address register contains the address of the first sector to be written.

When the write data command is loaded into command register B, the microsequencer decodes the function command and the OPI timer circuit is initiated. The sequencer locates the starting address of the write data routine in the sequencer ROM and enables the DMA control circuit. The silo now starts filling with data from memory. If the drive is ready, when the next sector pulse comes along, the controller begins reading each header off the disk. As each header enters the controller through the pulse discriminator and data separator circuits, it is then channeled through the data source selector and the header compare circuit. The header that enters the header compare circuit is compared serially with the serial DAR word on a bit for bit basis. The objective is to discover when the header word matches the first disk address stored in the DA register.

The header that entered the data source selector is channeled into the CRC circuit and gets checked for header CRC errors.

Successive headers are read off the disk and compared until a header match is found. Assuming that the header has been checked for CRC errors, a full sector of data will be shifted out of the silo and serialized. This silo serial data passes through the data source selector and the write precompensation circuit. The write gate signal will already be set before the write data is sent to the drive. The write data will be sent over the WRITE DATA line in sync with the system clock timing.

After the write data passes through the data source selector, the CRC data word is appended onto the end of the data block and is also written on the disk. The contents of the SA register is then incremented by one and the whole procedure is repeated again if there are multiple sectors to write.

When all the sectors scheduled have been written, the function done will be set and the interrupt control circuit will send out an interrupt request to the CPU if the interrupt enable bit was set.

4.4.7 Read Data Command

The read data command is used to read data off the disk and place it in memory. Like write data, it also is normally preceded by read header and seek commands. Read data is read off the disk in full sectors. Refer to Figure 4-9 for the functional flow diagram that illustrates this operation.

Prior to issuing the read data command, the break MA register must be loaded with the address of the first memory location. The word count (WC) stores the number of words to be transferred, command register A and sector address contain the cylinder and sector address of the first sector to be read off the disk.

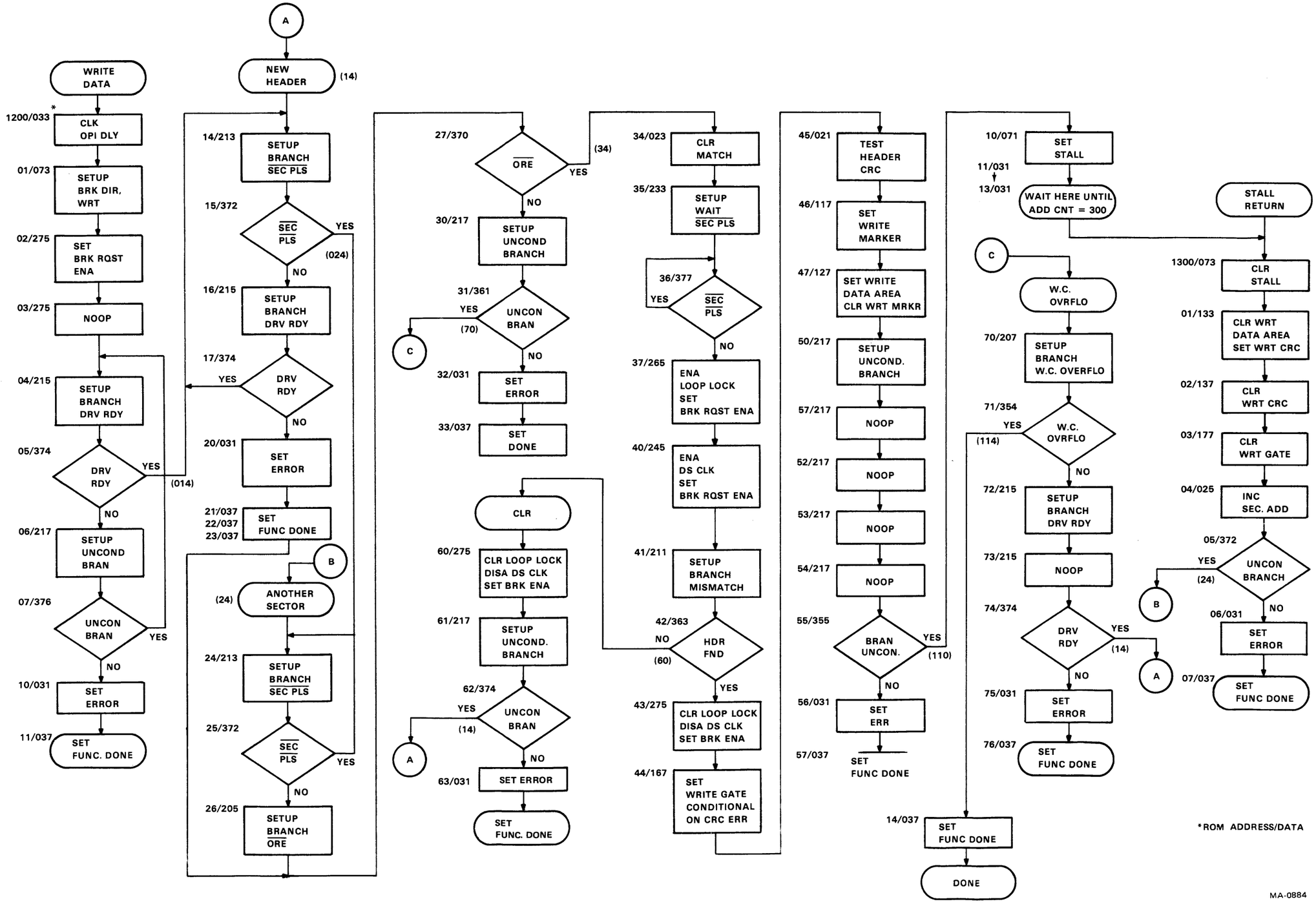


Figure 4-8 Write Data Mode Flow Diagram

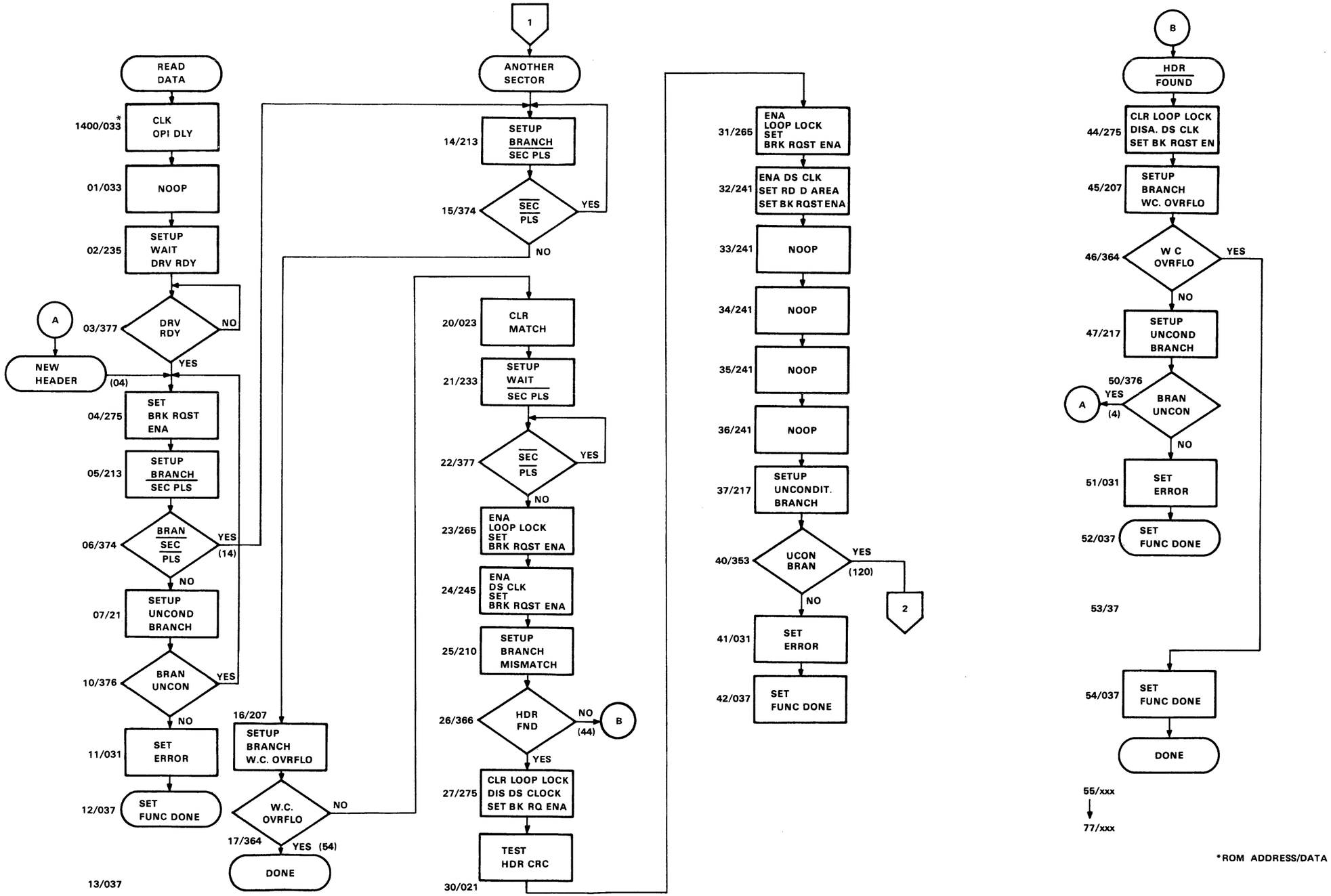


Figure 4-9 Read Data Mode Flow Diagram
(Sheet 1 of 2)

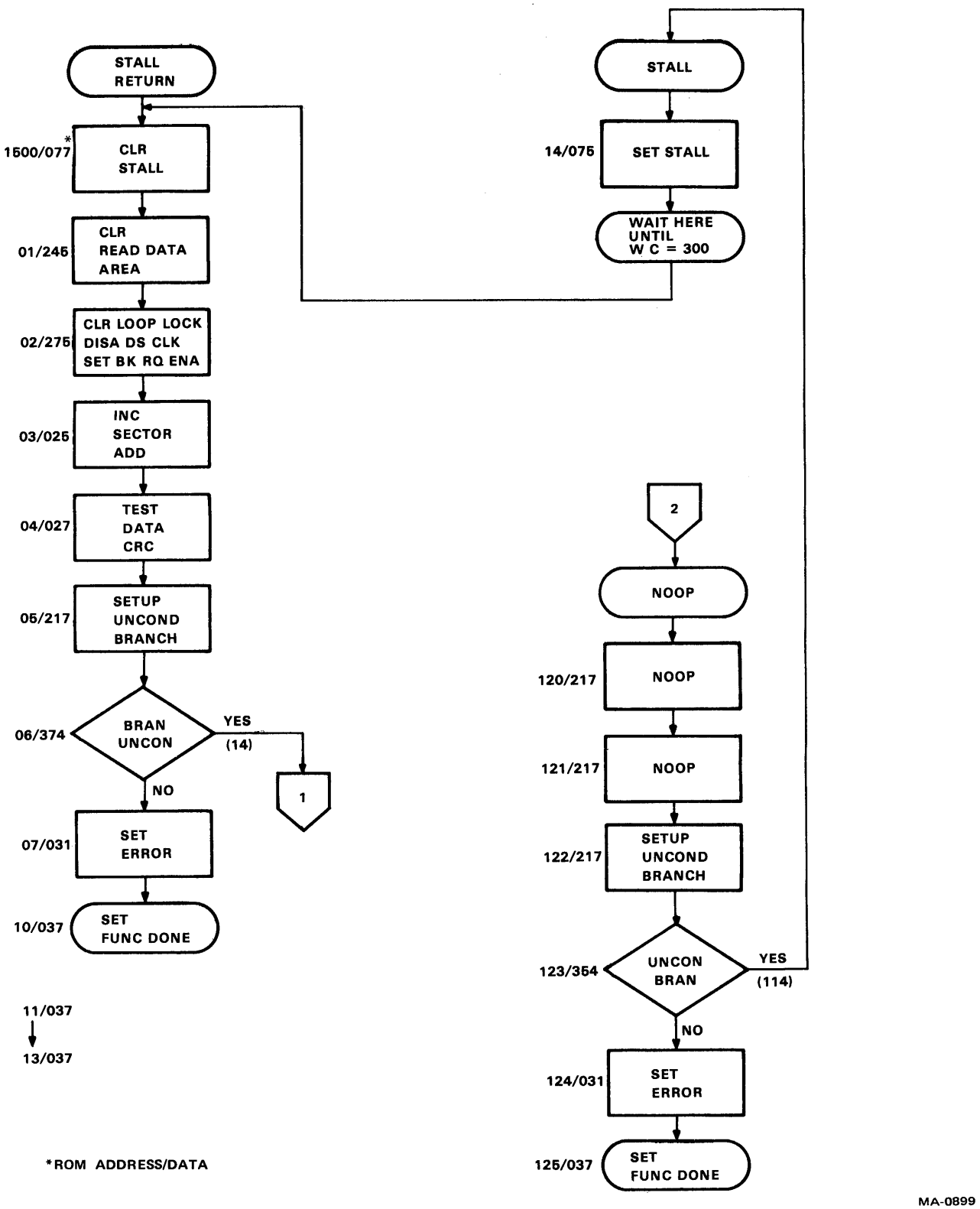


Figure 4-9 Read Data Mode Flow Diagram
(Sheet 2 of 2)

When the read data command is loaded into command register B, the microsequencer decodes the function command and the OPI timer circuit is initiated. The sequencer locates the starting address of the read data routine in the sequencer ROM and enables the DMA control circuit. When the next sector pulse comes along, the controller begins reading each header off the disk. As each header enters the controller through the pulse discriminator and data separator circuits, it is then channeled through the data source selector and header compare circuit. In the header compare circuit it is compared serially with the serial DAR word on a bit-for-bit basis. The objective is to discover when the header word matches the first disk address stored in the DA register. The header that entered the data source selector is channeled into the CRC circuit and gets checked for header CRC errors.

Successive headers are read off the disk and compared until a header match is found. If the silo is ready and the header CRC is good, 128 data words are read off the disk. The read data enters the controller through the pulse discriminator and data separator circuits to end up in the silo. The silo is simultaneously transferring read data to memory under DMA control.

After a sector is read, the SA register is incremented and the whole procedure is repeated if there are multiple sector reads. The data CRC word is checked at the end of each sector.

When the last sector has been read, function done is set and the interrupt control circuit sends an interrupt request to the CPU if the interrupt enable bit was set.

4.4.8 Read Data W/O Header Check Command

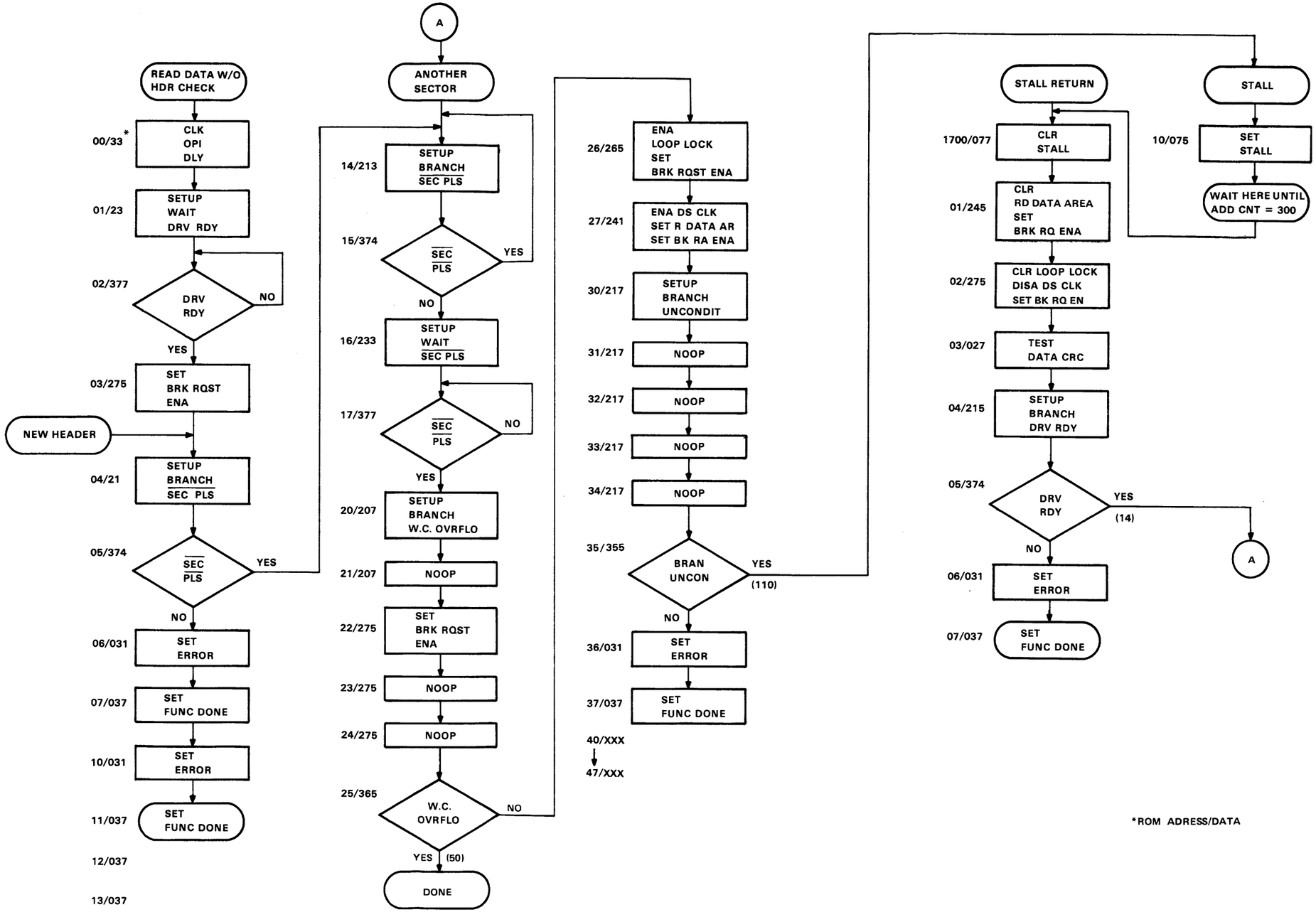
The read data without header check command is a special command used to recover data from sectors with bad header information. Refer to Figure 4-10 for the functional flow diagram that illustrates this operation.

Prior to issuing this command, system software must locate the sector preceding the bad sector by performing successive read header commands. Then the break MA register is loaded with the address of the first memory location. The word count (WC) is loaded with the number of words to be transferred.

The read data without header check command, device select, and interrupt enable bits can now be written into command register B to start the command operation.

The OPI timer is initiated and the microsequencer locates the operational routine to be used. If the drive is ready, the command routine will enable the DMA control circuit and wait for the next sector pulse to come along. Once a sector pulse is detected, 128 data words will be read from the bad sector on the disk into the controller. They will enter the controller pulse discriminator and data separator circuits on their way to the silo. The data will be checked by the CRC circuit for any errors.

Simultaneously, silo data is being transferred to memory under DMA control. If multisector reads are to be performed, the operation is repeated on the next sector pulse. If only one sector is to be read, or the multisector read is complete, function done is set and the interrupt control circuit sends an interrupt request to the CPU.



MA-0887

Figure 4-10 Read Data W/O HDR CK Mode Flow Diagram

4.5 SILO STORAGE AND CONTROL

The 9403 silo is a 16-word × 12-bit low-power, bipolar, FIFO memory device that can accept and output data in either serial or parallel format (refer to Figures 4-11 and 4-12). The silo contains a 4-bit input register with serial data inputs, parallel data inputs, and mode control inputs and outputs for handshaking. An automatic priority scheme is built into the 9403 configuration which assures that a slow device automatically dominates the configuration regardless of its location. This automatic priority scheme is essentially based on a ripple effect within the silo.

The 16-word × 12-bit silo is a fall-through stack that physically (internally) consists of four flip-flops per row. If data is loaded into the first row of flip-flops and the second row is empty, the stack control logic transfers data from the first row to the second. If the third row is empty, data will fall through into the third row, and so on.

Parallel data is entered into the input register using the data input lines and the LOAD SILO H as the clock pulse. In disc controllers, parallel loading is used during write operations to the disk. A DMA request can be generated whenever the input register is empty. LOAD SILO H direct sets and clears the inputs of input register flip-flops. Serial data is entered using the SERIAL DATA INPUT line which is clocked by SERIAL CLK IN H. Data input occurs on the high-to-low transition when CB2 H (mode) is true. During a read operation from the disk, when the output register is full, a DMA transfer can be requested.

When the input register receives four data bits (serial or parallel), the status flag, input register full output IRF L goes low and transfers the data from the input register into the first stack location, provided that this location is empty. As soon as the data is transferred, and LOAD SILO H goes low, the control logic attempts to initialize the input register so that it can accept another word from the input lines. The silo has been altered from its basic operation in such a way that, when a data word is received by the input register, the word automatically enters the stack and "falls through" towards the output to the empty location nearest the output.

As soon as a data word appears in the output register, output register empty output (ORE L) goes high to indicate the presence of data. Because the output enable (pin 17) is connected to ground, the tristate buffers are enabled and data is available on the output lines. Data can be extracted from the output register either serially or in parallel. The SILO SER DATA is the serial data output, and SER CLK OUT H as the associated clock. As soon as the last data bit is shifted out, ORE 2L goes low, designating an empty output register.

The ORE L (output register empty) going low allows new data to be loaded from the stack into the output register. The silo is configured in such a manner that as soon as the last data bit is shifted out, additional data is requested from the stack.

For parallel output, the ENA DATA L going high allows data to be loaded into the output register and output on the parallel data lines. A high on ENA DATA L disables ORE 2L which temporarily disables the serial data out line.

The silo is cleared by a high on CLR H master reset input. This level causes the output flags IRF 2L and ORE 2L to indicate an empty condition. The CLR H does not clear data flip-flops in the stack or output register; it only initializes the stack control logic.

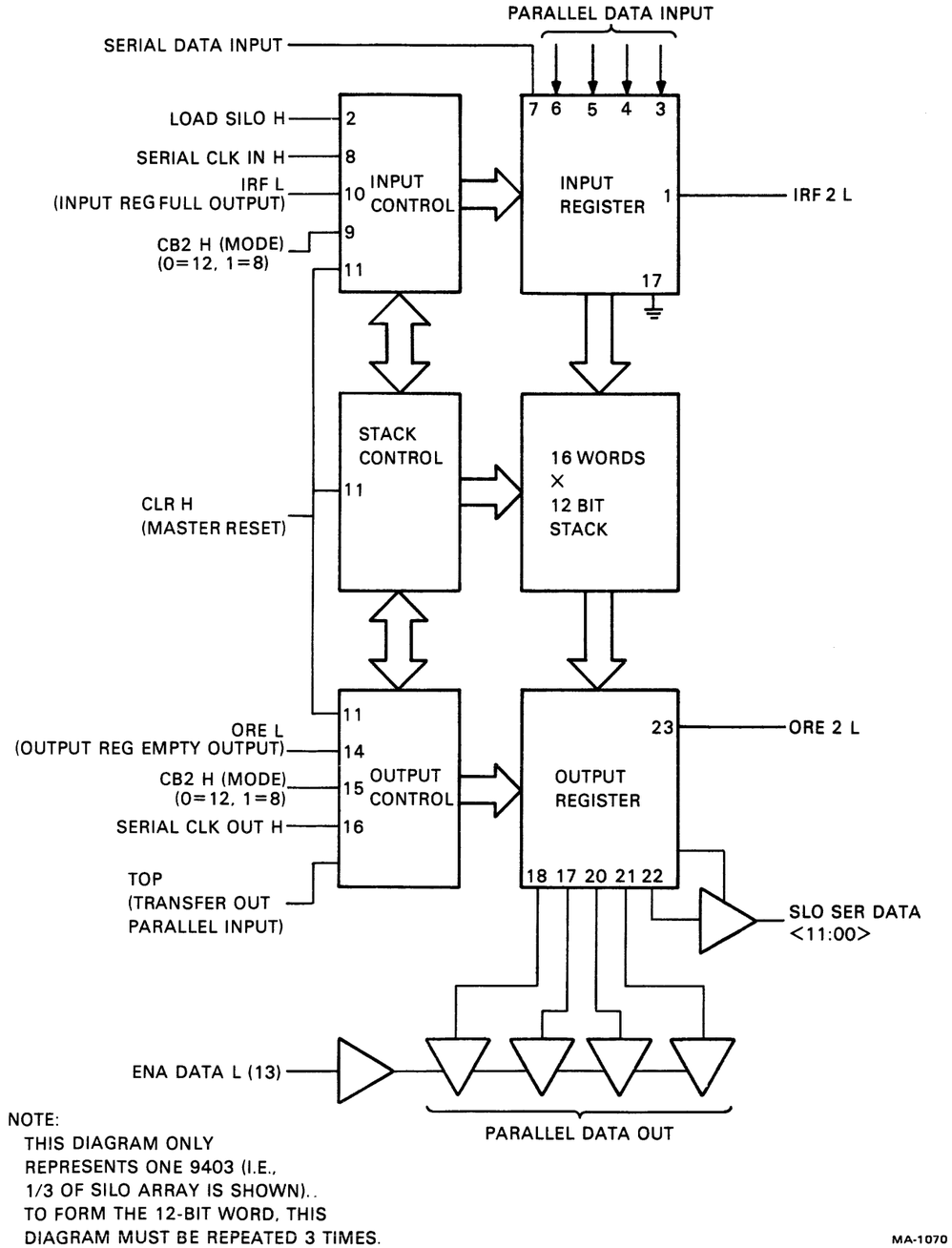


Figure 4-11 Silo Functional Block Diagram

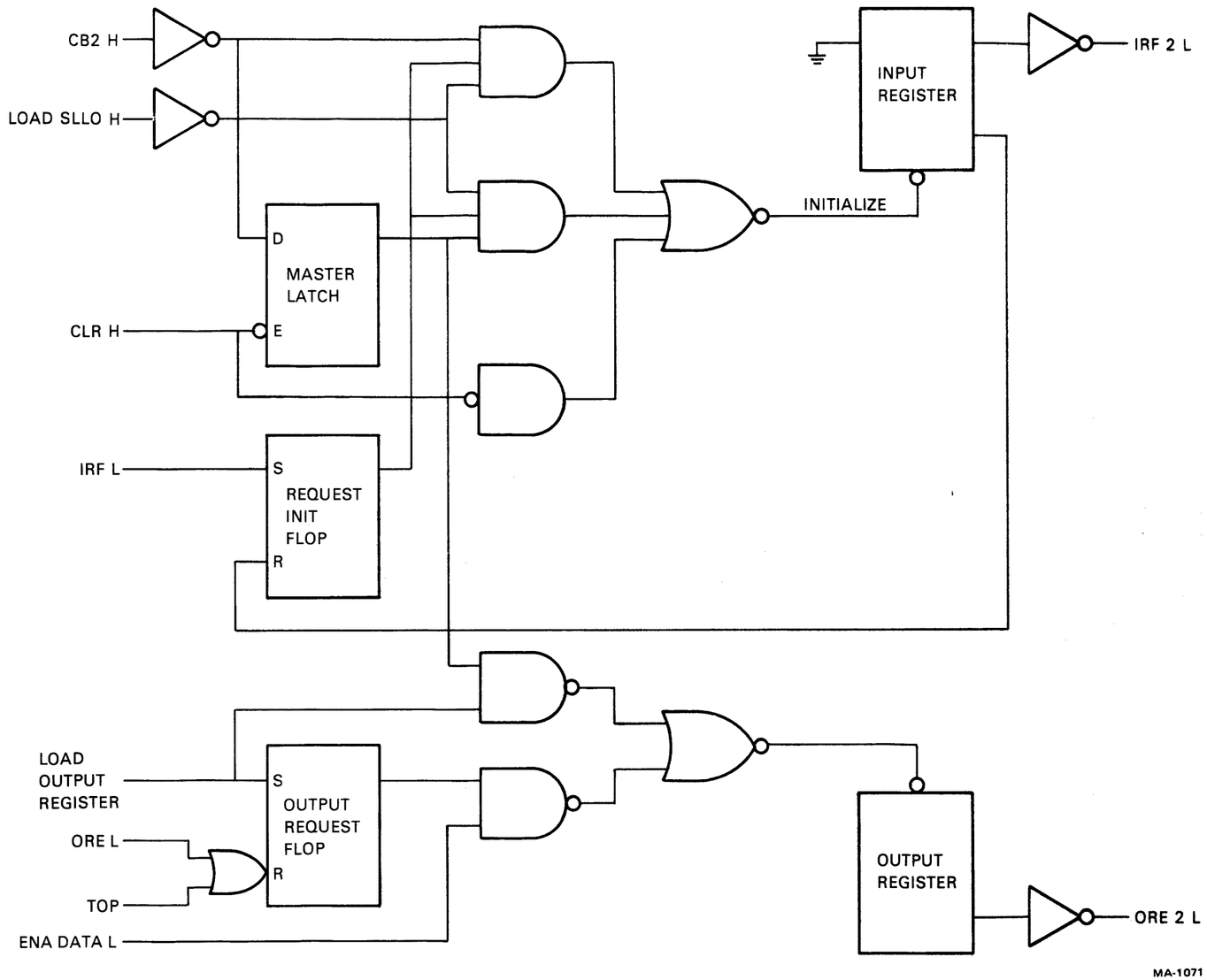


Figure 4-12 Silo Conceptual Overview

4.6 WRITE PRECOMPENSATION CIRCUIT DESCRIPTION

The write precompensation circuit has the function of time shifting the data pulses to precompensate for the affects peak shift. Figure 4-13 shows the precompensation circuit detailed block diagram.

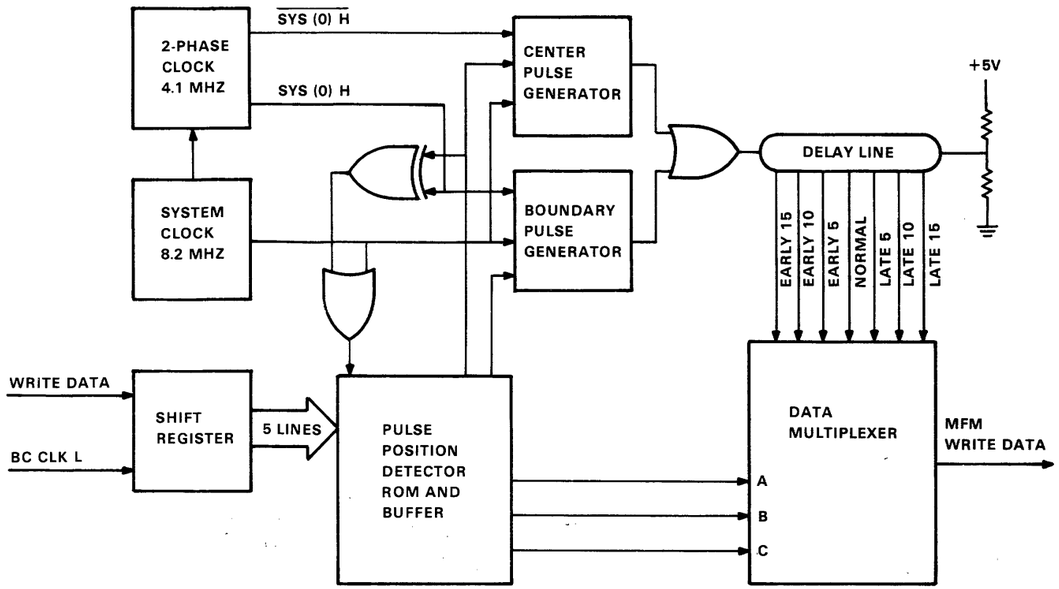


Figure 4-13 Write Encode and Precompensation Circuit

The write data is serially shifted into a 5-bit shift register. The shift register output is then fed into a pulse position detector ROM. This ROM can examine five serial data bits at a time. Internally, the ROM is preprogrammed to encode NRZ data into MFM data. The bit pattern from the shift register that addresses the ROM will determine the ROM output instructions. This instruction does two things. Two bits of the instruction go into generating either a bit cell boundary pulse or a bit cell center pulse. A boundary pulse is generated when encoding two successive logical zeros. A center pulse is generated to encode a logical one. The other three bits of the instruction select the delay line input path to the data multiplexer. Various bit patterns can advance or retard the leading edge of the MFM data pulse by as much as 15 ns. This is the time shifting that accomplishes the precompensation for peak shift phenomena.

4.7 DATA SEPARATOR AND PHASE LOCKED LOOP CIRCUITS

The basic problem with the recovery of MFM encoded data is the absence of a regularly recurring reference. Since the data pulse may occur either at the center or boundary of a bit cell, its location remains unpredictable for random data patterns. The only consistent pattern that may be used as the basis for data recovery is the fact that MFM encoding guarantees there will be at least one flux reversal on the disk for every 2-bit cells. This fundamental frequency is what makes it feasible to use phase-locked loop techniques to form a self-clocking data recovery system.

4.7.1 Phase Locked Loop

The phase locked loop (Figure 4-14) is a closed-loop circuit that provides an output that is in phase and frequency lock with the input signal. Its output frequency is twice that of the incoming read data bit rate.

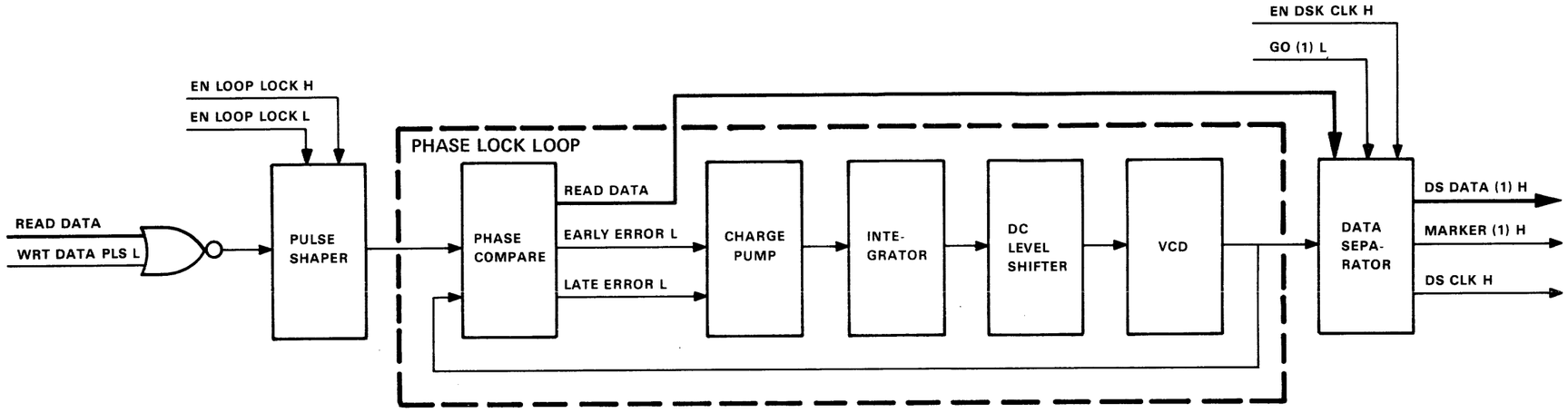


Figure 4-14 Data Separator and PLL Block Diagram

The input data into the pulse shaper can come from two different sources. The WRT DATA PLS L line provides a data path between the write precompensation circuit and the pulse shaper. This data path is used during the maintenance command as a means of testing internal controller circuit functions.

The READ DATA line is the data path followed when reading data off the disk. The read data pulses are standardized in the pulse shaper to a uniform 60 ns pulse width and applied to one input of the phase comparator. The other input to the phase comparator is the output of the voltage controlled oscillator (VCO). In the phase comparator, the phase of the data pulse is compared with that of the VCO clock as shown in Figure 4-15. As illustrated, data early and data late pulses are directly related to phase error. These error outputs are applied to the charge pump which converts them into a net current pulse that charges a capacitor. The capacitor integrates the current waveform and generates the small error offset voltages needed to control the VCO frequency and maintain loop lock. The only purpose of the DC level shifter is to place the integrator output voltage within a range that is usable by the VCO input.

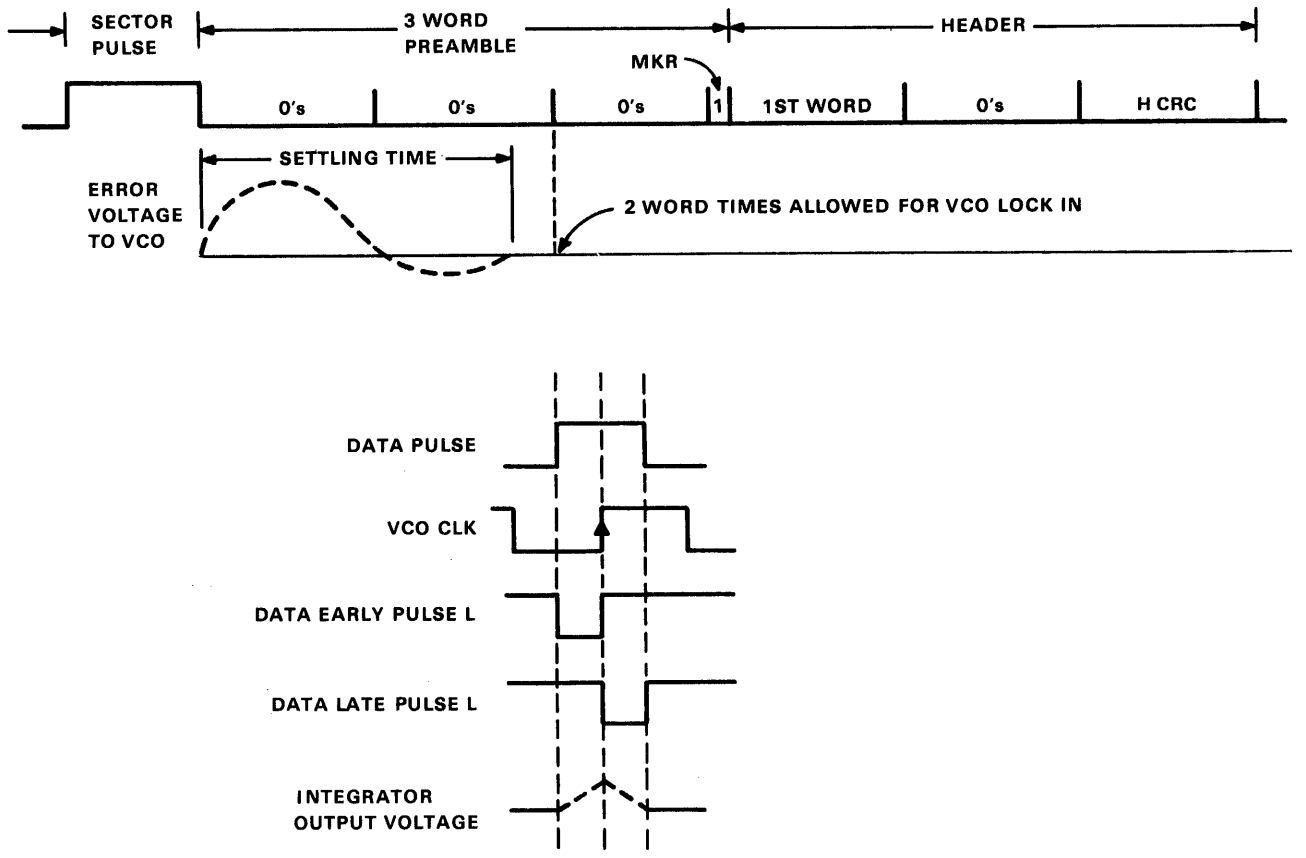


Figure 4-15 PLL Timing Relationships

Figure 4-16 illustrates the relationship between the read data and the phase lock loop settling time. The phase lock loop is guaranteed to lock onto the read data frequency with two word times.

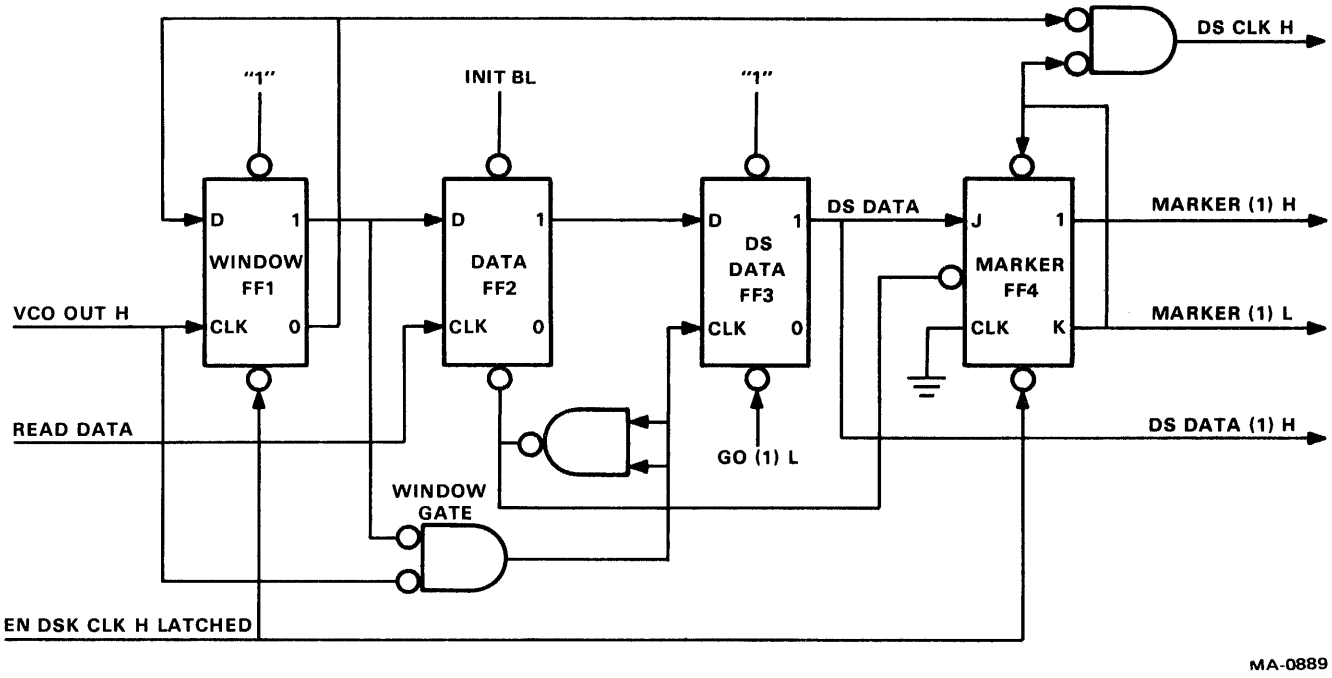


Figure 4-16 Data Separator Circuit

4.7.2 Data Separator

The data separator is used to separate the digital data from the MFM encoded data. A detailed block diagram of the data separator circuit is shown in Figure 4-17.

When a read header command is decoded and the first sector pulse is detected, header preamble PR1 appears on the READ DATA line from the selected drive. LOOP LOCK is asserted and enables the preamble 0s to enter the phase lock loop.

The PLL is given two word times to allow it enough time to lock onto the bit rate frequency. Then EN DSK CLK H is asserted to enable the data separator circuit. Consequently, when the marker bit at the end of the preamble PR1 is detected, DS DATA is asserted. When the clock input to the MARKER flip-flop is asserted, the flip-flop is latched in the one state, asserting MARKER (1) H. At this point, the controller is no longer synchronized to the system clock, but to the data separator DS CLK. Consequently, whenever the output of the window flip-flop goes high, DS CLK will also go high. Thereafter, when the bit in a header or data word is a "1", DS DATA is asserted.

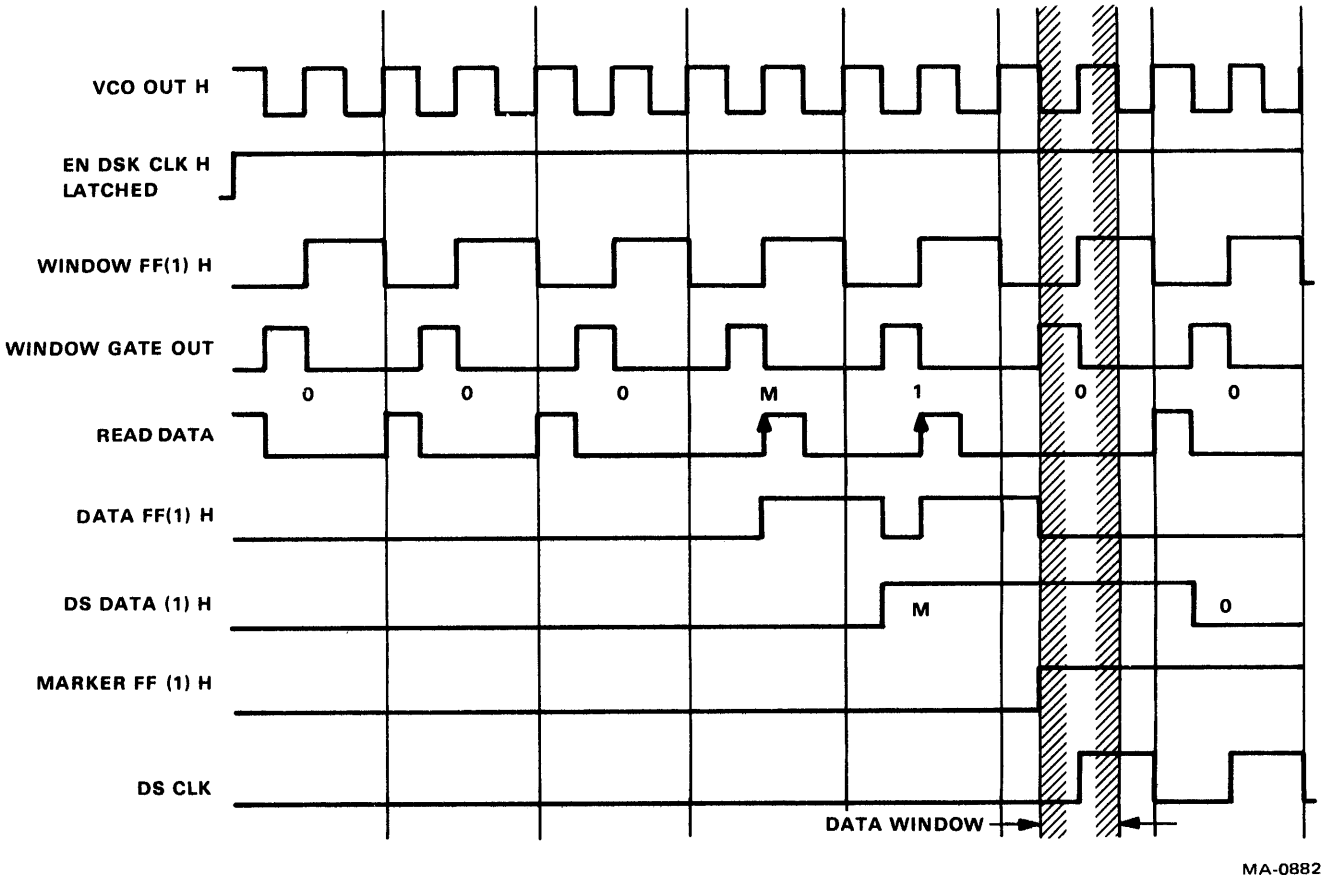


Figure 4-17 Data Separator Waveforms

APPENDIX A
BOOT PROGRAM INFORMATION

A.1 GENERAL

Essentially, two new ROMs have been created which replace the TA8-E boot program. The new ROMs contain the RL8-A boot which provides the program for the RL8-A/RL01 disk subsystem. The two ROMs (listed below) are part of the RL8-A shipping package and are to be installed on the KM8-A (M8317) module on-site.

ROM P/N	Location
23.465A2	E82 (ROM 1)
23.469A2	E87 (ROM 2)

Installation of these ROMs requires switches 1 and 2 on the M8317 module to be set as shown below. Reference engineering drawing CS-M8317 or the *PDP-8/A Miniprocessor User's Manual* (EK-8A002-MM).

Program	S2-5	S2-6	S2-7	S2-8	S1-1	S1-2	S1-3	ROM St Add	MEM St Add
RL8A	off	on	off	off	off	on	off	272	0001

RL8-A OMNIBUS CONTROLLER
TECHNICAL MANUAL
EK-RL8A-TM-001

Reader's Comments

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgment is it complete, accurate, well organized, well written, etc.? Is it easy to use? _____

_____

_____

_____

What features are most useful? _____

_____

_____

_____

What faults or errors have you found in the manual? _____

_____

_____

_____

Does this manual satisfy the need you think it was intended to satisfy? _____

Does it satisfy *your* needs? _____ Why? _____

_____

_____

_____

☐ Please send me the current copy of the *Technical Documentation Catalog*, which contains information on the remainder of DIGITAL's technical documentation.

Name _____ Street _____
Title _____ City _____
Company _____ State/Country _____
Department _____ Zip _____

Additional copies of this document are available from:

Digital Equipment Corporation
444 Whitney Street
Northboro, Ma 01532
Attention: Communications Services (NR2/M15)
Customer Services Section

Order No. **EK-RL8A-TM-001**

— — — — — — — — — — — — Fold Here — — — — — — — — — — —

— — — — — — — — — — Do Not Tear - Fold Here and Staple — — — — — — — — —

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

Digital Equipment Corporation
Technical Documentation Department
Maynard, Massachusetts 01754