

DICK BAILEY

digital

RLV11 CONTROLLER TECHNICAL DESCRIPTION MANUAL





000000



000000



Copyright © 1978 by Digital Equipment Corporation
1st Edition, March 1978

RLV11 Controller Technical Description Manual

Printed in U.S.A.
This document was set on DIGITAL's DECset-1000
computerized typesetting system.

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts:

DEC	DECtype	DEC
DECMM	DECUS	DECsystem-10
DECsystem-10	DIGITAL	DECsystem-20
DECsystem-20	MASSBUS	TYPISET-11
	UNIBUS	

Copyright © 1977, 1978 by Digital Equipment Corporation

The material in this manual is for informational purposes and is subject to change without notice.

Digital Equipment Corporation assumes no responsibility for any errors which may appear in this manual.

Printed in U.S.A.

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

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts:

DEC	DECtape	PDP
DECCOMM	DECUS	RSTS
DECsystem-10	DIGITAL	TYPESET-8
DECSYSTEM-20	MASSBUS	TYPESET-11
		UNIBUS

CONTENTS

		Page
CHAPTER 1 INTRODUCTION		
1.1	PURPOSE AND SCOPE OF MANUAL	1-1
1.2	GENERAL DESCRIPTION	1-1
1.3	SUBSYSTEM CONFIGURATION	1-1
1.4	PHYSICAL DESCRIPTION OF CONTROLLER	1-1
1.4.1	Bus Interface Module	1-1
1.4.1.1	Bus/Vector Address Switches	1-5
1.4.2	Drive Module	1-5
1.4.3	Module Slot Location	1-5
1.5	ADDRESSABLE REGISTERS	1-5
1.6	CONTROLLER COMMANDS	1-8
1.7	REFERENCE DOCUMENTS	1-9
1.8	CONTROLLER SPECIFICATIONS	1-9
CHAPTER 2 SYSTEM LEVEL DESCRIPTION		
2.1	GENERAL	2-1
2.2	SYSTEM BLOCK DIAGRAM	2-1
2.2.1	LSI-11 Microcomputer	2-1
2.2.2	RLV11 Controller	2-2
2.2.2.1	LSI-11 Bus Interface	2-2
2.2.2.2	Programmer Interface	2-2
2.2.2.3	Data Buffer	2-2
2.2.2.4	Control Microsequencer	2-2
2.2.2.5	Data Formatter	2-2
2.2.2.6	Drive Bus Interface	2-2
2.2.3	RL01 Disk Drive	2-4
2.2.3.1	Drive Features	2-4
2.2.3.2	Disk Formatting	2-4
2.2.3.3	Bad Sector File	2-6
2.3	TYPICAL OPERATIONAL SEQUENCES	2-6
2.3.1	Commands Independent of Others	2-6
2.3.1.1	Maintenance Command	2-6
2.3.1.2	Get Status Command	2-6
2.3.1.3	Read Header Command	2-6
2.3.2	Commands Dependent on Others	2-6
2.3.2.1	Seek Command	2-6
2.3.2.2	Write Data Command	2-8

CONTENTS

		Page
2.3.2.3	Write Check Command.....	2-8
2.3.2.4	Read Data Command.....	2-8
2.3.2.5	Read Data Without Header Check Command.....	2-8
2.4	MAINTENANCE FUNCTION.....	2-8
2.5	GET STATUS FUNCTION.....	2-10
2.6	READ HEADER FUNCTION.....	2-11
2.7	SEEK FUNCTION.....	2-12
2.8	WRITE DATA FUNCTION.....	2-13
2.9	WRITE CHECK FUNCTION.....	2-14
2.10	READ DATA FUNCTION.....	2-14
2.11	READ DATA WITHOUT HEADER CHECK FUNCTION.....	2-14
CHAPTER 3	INTERFACE LEVEL DESCRIPTION	
3.1	GENERAL.....	3-1
3.2	LSI-11 BUS.....	3-2
3.3	DRIVE BUS INTERFACE SIGNALS.....	3-6
3.3.1	General.....	3-6
3.3.2	Drive Bus Signal Descriptions.....	3-6
3.3.2.1	Drive Select (DRV SEL 0,1).....	3-6
3.3.2.2	System Clock (SYS CLK).....	3-6
3.3.2.3	Drive Command (DR CMD).....	3-7
3.3.2.4	Write Gate (WR GATE).....	3-7
3.3.2.5	Write Data (WR DATA).....	3-7
3.3.2.6	Power Fail (PWR FAIL).....	3-7
3.3.2.7	Drive Ready (DR RDY).....	3-7
3.3.2.8	Drive Error (DR ERR).....	3-7
3.3.2.9	Status Clock (STATUS CLK).....	3-7
3.3.2.10	Status (STATUS).....	3-7
3.3.2.11	Sector Pulse (SEC PLS).....	3-7
3.3.2.12	Read Data (RD DATA).....	3-8
3.3.3	Drive Bus Dialogue.....	3-8
3.4	ADDRESSABLE REGISTERS.....	3-15
3.4.1	Control Status Register (CSR).....	3-15
3.4.2	Bus Address Register (BAR).....	3-18
3.4.3	Disk Address Register (DAR).....	3-18
3.4.3.1	DAR During a Seek Command.....	3-18
3.4.3.2	DAR During Read or Write Data Command.....	3-19
3.4.3.3	DAR During a Get Status Command.....	3-20
3.4.4	Multipurpose Register (MPR).....	3-21
3.4.4.1	MPR During a Get Status Command.....	3-21
3.4.4.2	MPR During a Read Header Command.....	3-22
3.4.4.3	MPR During Read/Write Data Commands.....	3-23

CONTENTS

		Page
3.5	I/O TRANSFER OPERATIONS.....	3-23
3.5.1	Programmed I/O Transfers	3-26
3.5.1.1	Writing Controller Registers.....	3-26
3.5.1.2	Reading Controller Registers.....	3-26
3.5.2	DMA I/O Transfers	3-26
3.5.3	Interrupt-Driven I/O Transfers	3-26
3.6	BUS SIGNAL TIMING	3-26
CHAPTER 4	FUNCTIONAL LEVEL DESCRIPTION	
4.1	GENERAL.....	4-1
4.2	CONTROLLER SIMPLIFIED BLOCK DIAGRAM.....	4-1
4.2.1	Bus Interface Module (M8014).....	4-1
4.2.1.1	Bus Control Functions	4-1
4.2.1.2	Bus Transceivers.....	4-1
4.2.1.3	Programmable Registers.....	4-3
4.2.1.4	FIFO.....	4-3
4.2.2	Drive Module (M8013).....	4-3
4.2.2.1	Microsequencer Logic	4-3
4.2.2.2	Write Precompensation	4-3
4.2.2.3	Data Separator.....	4-4
4.2.2.4	CRC Circuit	4-4
4.2.2.5	Data Source Selector	4-4
4.2.2.6	Header Compare Circuit	4-4
4.3	MAINTENANCE COMMAND FUNCTIONAL FLOW DIAGRAM.....	4-4
4.4	GET STATUS FUNCTIONAL FLOW DIAGRAM	4-9
4.5	READ HEADER FUNCTION.....	4-9
4.6	SEEK FUNCTIONAL FLOW DIAGRAM	4-14
4.7	WRITE DATA FUNCTIONAL FLOW DIAGRAM	4-14
4.8	WRITE CHECK FUNCTIONAL FLOW DIAGRAM.....	4-20
4.9	READ DATA FUNCTIONAL FLOW DIAGRAM.....	4-20
4.10	READ DATA WITHOUT HEADER CHECK FUNCTIONAL FLOW DIAGRAM	4-26
CHAPTER 5	UNIT LEVEL DESCRIPTION	
5.1	GENERAL.....	5-1
5.2	LSI-11 BUS CONTROL CIRCUIT	5-1
5.2.1	Bus Transceivers.....	5-1
5.2.2	Register Protocol.....	5-1
5.2.2.1	Writing a Register	5-1
5.2.2.2	Reading a Register	5-5
5.2.3	DMA Control	5-5

CONTENTS

		Page
5.2.4	Interrupt Control	5-9
5.2.5	Non-Existent Memory (NXM) Timer Circuit	5-10
5.2.6	Operation Incomplete (OPI) Timer Circuit	5-12
5.2.7	BAR/WC.....	5-13
5.2.8	DAR and Serializer	5-15
5.2.9	CSR and Error Bits.....	5-15
5.3	FIFO STORAGE AND CONTROL	5-18
5.4	MICROSEQUENCER	5-23
5.4.1	Clock Select Circuit	5-23
5.4.2	Branch Condition Selector.....	5-23
5.4.3	Condition Synchronizer.....	5-23
5.4.4	Wait and Branch Control Circuit.....	5-23
5.4.5	CRDY Synchronizer	5-25
5.4.6	Word Counter	5-25
5.4.7	Sequencer ROM.....	5-25
5.4.8	ROM Buffer	5-25
5.4.9	Instruction Decoder.....	5-25
5.4.10	Sequencer Stall Instruction	5-25
5.5	WRITE ENCODE AND PRECOMPENSATION CIRCUIT	5-25
5.5.1	MFM Encoding.....	5-25
5.5.2	Peak Shift Phenomenon.....	5-27
5.5.3	Write Precompensation Circuit.....	5-27
5.6	DATA SEPARATOR AND PLL	5-27
5.6.1	Phase-Locked Loop (PLL).....	5-27
5.6.2	Data Separator.....	5-30

FIGURES

Figure No.	Title	Page
1-1	RL01/RLV11 Subsystem Configuration	1-2
1-2	RLV11 Bus Interface Module (M8014).....	1-3
1-3	Address Switching Scheme	1-4
1-4	RLV11 Drive Module (M8013)	1-6
1-5	H9273 Backplane Priority Structure	1-7
2-1	System Block Diagram	2-1
2-2	RLV11 Controller Major Functional Blocks	2-3
2-3	Sector Format	2-5
2-4	Sequential Relationship of Controller Commands	2-7
2-5	Simplified Maintenance Operation	2-9

FIGURES (CONT)

Figure No.	Title	Page
2-6	Simplified Get Status Operation	2-10
2-7	Simplified Read Header Operation	2-11
2-8	Simplified Seek Operation	2-12
2-9	Simplified Write Data Operation	2-13
2-10	Simplified Write Check Operation	2-15
2-11	Simplified Read Data Operation.....	2-16
2-12	Simplified Read Data Without Header Check Operation	2-17
3-1	Controller Interface Signals.....	3-1
3-2	LSI-11 Bus Signals	3-2
3-3	Drive Bus Signals	3-6
3-4	Get Status Drive Bus Sequence.....	3-8
3-5	Read Header Drive Bus Sequence.....	3-9
3-6	Seek Drive Bus Sequence.....	3-10
3-7	Write Data Drive Bus Sequence.....	3-11
3-8	Write Check Drive Bus Sequence.....	3-12
3-9	Read Data Drive Bus Sequence	3-13
3-10	Read Data Without Header Check Drive Bus Sequence.....	3-14
3-11	Control Status Register	3-15
3-12	Bus Address Register.....	3-18
3-13	DAR Seek Command.....	3-18
3-14	DAR Read/Write Data Command.....	3-19
3-15	DAR Get Status Command.....	3-20
3-16	MPR Status Word.....	3-21
3-17	MPR Three Header Words.....	3-22
3-18	MPR Used As Word Counter.....	3-23
3-19	Register Summary	3-24
3-20	DATO Bus Cycle.....	3-27
3-21	DATI Bus Cycle	3-28
3-22	DMA Request/Grant Sequence	3-29
3-23	Interrupt Request/Acknowledge Sequence	3-30
4-1	RLV11 Controller Simplified Block Diagram	4-2
4-2	Maintenance Command Functional Block Diagram.....	4-5
4-3	Maintenance Command Flowchart	4-6
4-4	Results of Maintenance Command.....	4-8
4-5	Get Status Functional Block Diagram	4-10
4-6	Get Status Command Flowchart	4-11
4-7	Read Header Functional Block Diagram	4-12
4-8	Read Header Command Flowchart	4-13
4-9	Seek Functional Block Diagram	4-15
4-10	Seek Command Flowchart	4-16
4-11	Write Data Functional Block Diagram	4-17
4-12	Write Data Command Flowchart	4-18
4-13	Write Check Functional Block Diagram	4-21
4-14	Write Check Command Flowchart	4-22

FIGURES (CONT)

Figure No.	Title	Page
4-15	Read Data Functional Block Diagram.....	4-24
4-16	Read Data Command Flowchart.....	4-25
4-17	Read Data Without Header Check Functional Block Diagram.....	4-27
4-18	Read Data Without Header Check Command Flowchart.....	4-28
5-1	LSI-11 Bus Control Circuit.....	5-2
5-2	CSR Read/Write Diagram.....	5-3
5-3	Bus Protocol Timing for Writing the CSR.....	5-4
5-4	Protocol Timing for Reading the CSR.....	5-6
5-5	DMA Control Circuit.....	5-7
5-6	Single Word DMA Transfer to FIFO.....	5-8
5-7	Interrupt Control Circuit.....	5-9
5-8	Interrupt Timing Sequence.....	5-10
5-9	NXM Timer Circuit.....	5-11
5-10	NXM Timing Sequence.....	5-11
5-11	OPI Timer Circuit.....	5-12
5-12	BAR/WC Block Diagram.....	5-13
5-13	BAR/WC Operational Truth Table.....	5-14
5-14	DAR Block Diagram.....	5-16
5-15	CSR Block Diagram.....	5-17
5-16	RLV11 FIFO Control Circuit.....	5-19
5-17	Disk Read FIFO Timing.....	5-21
5-18	Disk Write FIFO Timing.....	5-22
5-19	Microsequencer Detailed Block Diagram.....	5-24
5-20	MFM Encoding.....	5-26
5-21	Peak Shift Waveform.....	5-28
5-22	Write Encode and Precompensation Circuit.....	5-29
5-23	Data Separator and PLL Block Diagram.....	5-29
5-24	PLL Timing Relationships.....	5-31
5-25	Data Separator Circuit.....	5-32
5-26	Data Separator Waveforms.....	5-33

TABLES

Table No.	Title	Page
1-1	Controller Addressable Registers	1-8
1-2	Controller Commands.....	1-8
1-3	Reference Documents	1-9
1-4	RLV11 Controller Specifications.....	1-9
3-1	LSI-11 Bus Signals	3-3
3-2	Register Addresses	3-15
5-1	Error Status Related Signals	5-18

TABLES

Table No.	Title	Page
1-1	Controller Addressable Register	1-4
1-2	Controller Commands	1-4
1-3	Reference Documents	1-9
1-4	RLVII Controller Specifications	1-9
2-1	LSI-11 Bus Signals	2-3
2-2	Register Address	2-12
2-3	Error Status Related Signals	2-18

CHAPTER 1 INTRODUCTION

1.1 PURPOSE AND SCOPE OF MANUAL

This manual contains a physical and technical description of the RLV11 Disk Controller. It is intended to aid Field Service and user maintenance personnel in servicing the RLV11. It provides the controller functional theory of operation and a logic block description that correlates with the field maintenance print set.

1.2 GENERAL DESCRIPTION

The RL01/RLV11 disk subsystem is a random-access, mass storage system that stores data in fixed-length blocks on preformatted disk cartridges. The disk subsystem consists of one RLV11 controller and from one to four RL01 disk drives. Each drive can store 5.24 million bytes. In the maximum configuration, the subsystem storage capability approaches nearly 21 million bytes.

The RLV11 controller provides the control functions for the disk subsystem. It consists of two quad modules that mount on the backplane inside the PDP-11/03L cabinet or any backplane adhering to the LSI-11 bus specification, DEC STD 160, and the H9273 backplane specification. The controller is the interface between the LSI-11 bus and the disk drive.

1.3 SUBSYSTEM CONFIGURATION

In the minimum subsystem configuration, there is only one disk drive attached to the drive bus. In the maximum configuration, the RLV11 controller can support up to four RL01 disk drives (Figure 1-1). The design is such that the controller can communicate with two or more drives in a time-shared fashion. Connections between controller and drives are made in daisy-chain arrangement.

1.4 PHYSICAL DESCRIPTION OF CONTROLLER

The RLV11 controller consists of two quad-height modules that insert into the LSI-11 microcomputer system backplane. Each module has some unique features such as switches, jumpers, trimpots, and connectors, which are explained in the following paragraphs.

1.4.1 Bus Interface Module

The bus interface module (M8014) contains the logic circuits that perform the following major functions.

- LSI-11 bus interface functions
- Programmable registers
- First-in/first-out (FIFO) data storage and control circuits

Figure 1-2 shows the component side of M8014, which includes the locations of the bus address switches, the vector address switches, and the connector finger assignments. Figure 1-3 illustrates the address switching scheme.

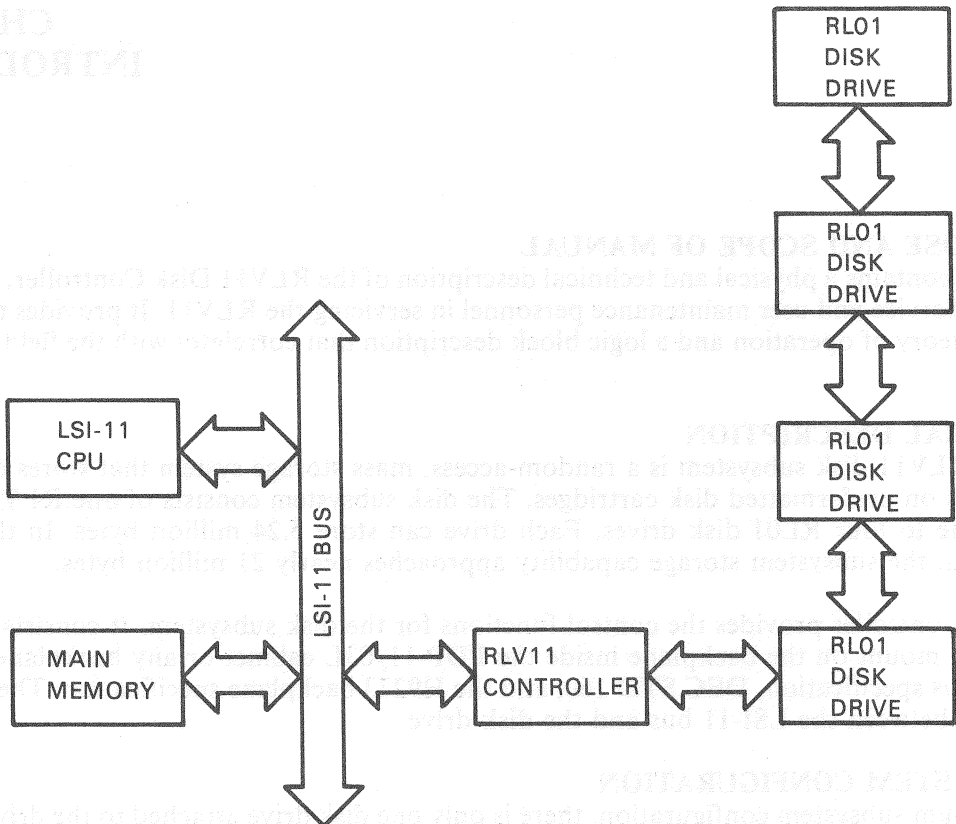
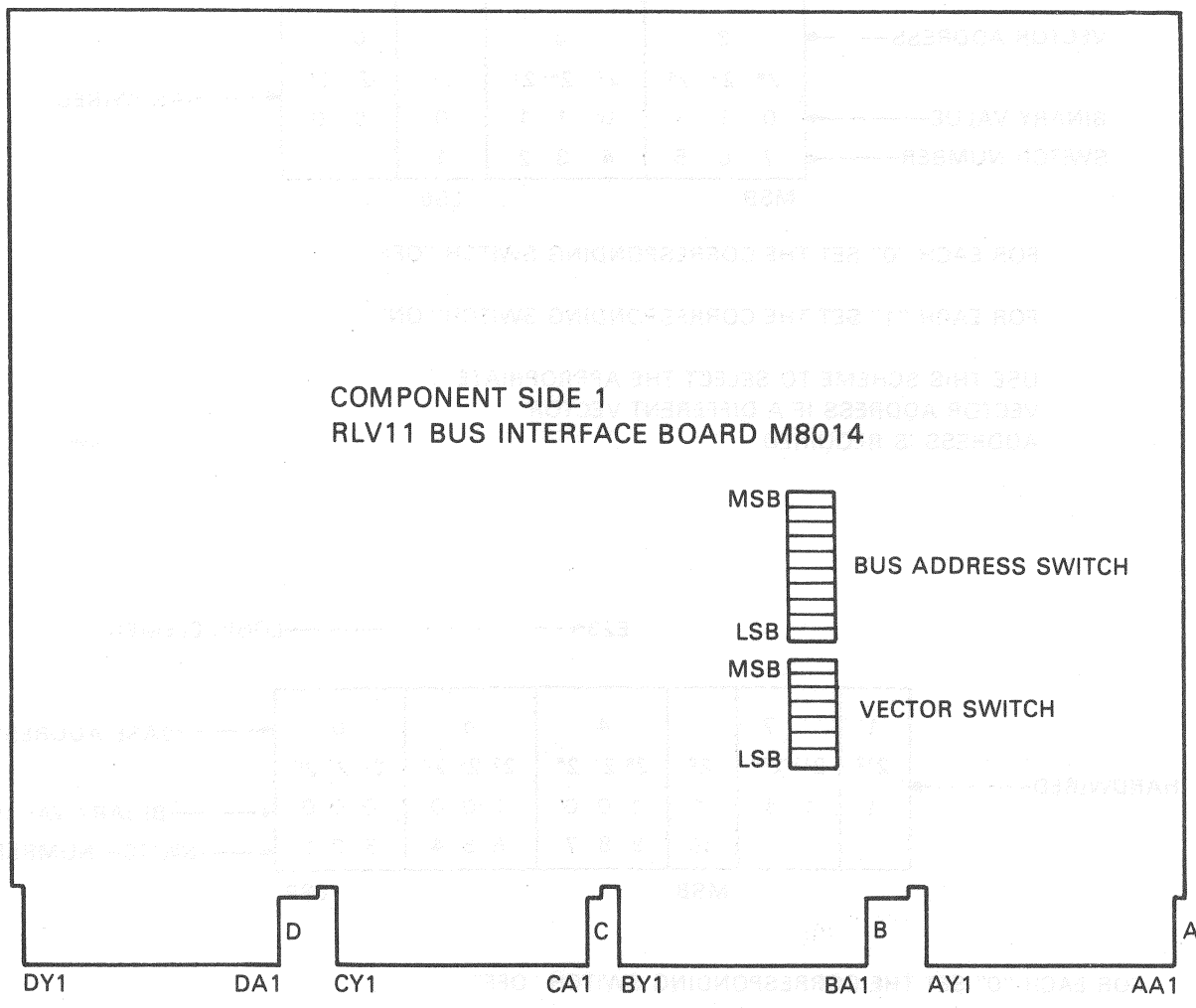


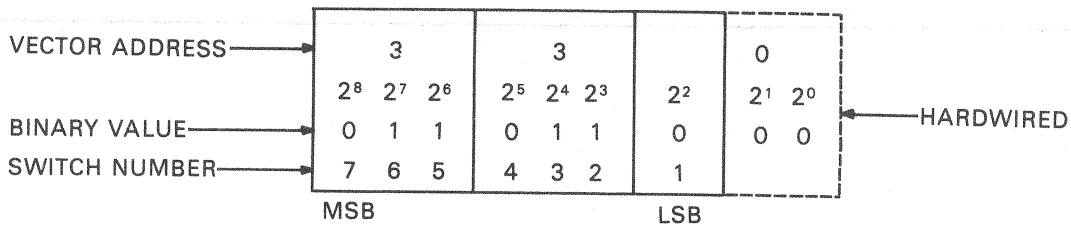
Figure 1-1 RL01/RLV11 Subsystem Configuration



MA-1168

Figure 1-2 RLV11 Bus Interface Module (M8014)

E22 ← LOGIC ELEMENT



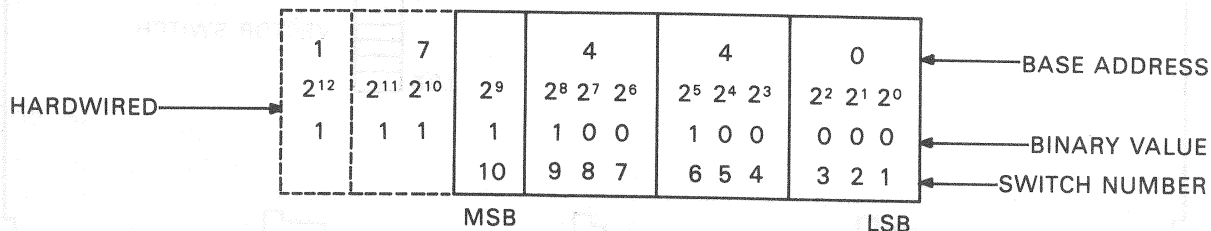
FOR EACH "0" SET THE CORRESPONDING SWITCH "OFF"

FOR EACH "1" SET THE CORRESPONDING SWITCH "ON"

USE THIS SCHEME TO SELECT THE APPROPRIATE VECTOR ADDRESS IF A DIFFERENT VECTOR ADDRESS IS REQUIRED

MA-0562

E23 ← LOGIC ELEMENT



FOR EACH "0" SET THE CORRESPONDING SWITCH "OFF"

FOR EACH "1" SET THE CORRESPONDING SWITCH "ON"

USE THIS SCHEME TO SELECT THE APPROPRIATE BASE ADDRESS IF A DIFFERENT BASE ADDRESS IS REQUIRED

MA-0563

Figure 1-3 Address Switching Scheme

1.4.1.1 Bus/Vector Address Switches – The bus address switch group is used to set up the device base address to be used. It is normally factory preset to 7440. This means that the device control status register (CSR) has an address of 174400 and the multipurpose register (MPR) has an address of 174406. The ON and OFF positions are labeled. The ON position is the logical 1 or true state.

The vector address switch group is used to point to the service routine address used when servicing a controller interrupt. It is factory preset for an address of 330.

1.4.2 Drive Module

The drive module (M8013) contains the circuitry that performs the following major functions.

- Data formatting and error-detecting circuits
- Control microsequencer and timing circuits
- Drive bus interface

Figure 1-4 shows the component side of M8013, which includes the location of jumpers, trimpot, connectors, and voltage-controlled oscillator (VCO) monitoring point.

The control microsequencer circuit can be implemented either with an erasable-programmable ROM (EPROM), IC DEC part no. 23-000B7, or with a masked ROM, IC part no. 23-000E2. Whichever memory chip is used requires that two different jumpers be factory installed. Figure 1-4 shows the location of jumpers W1 through W4.

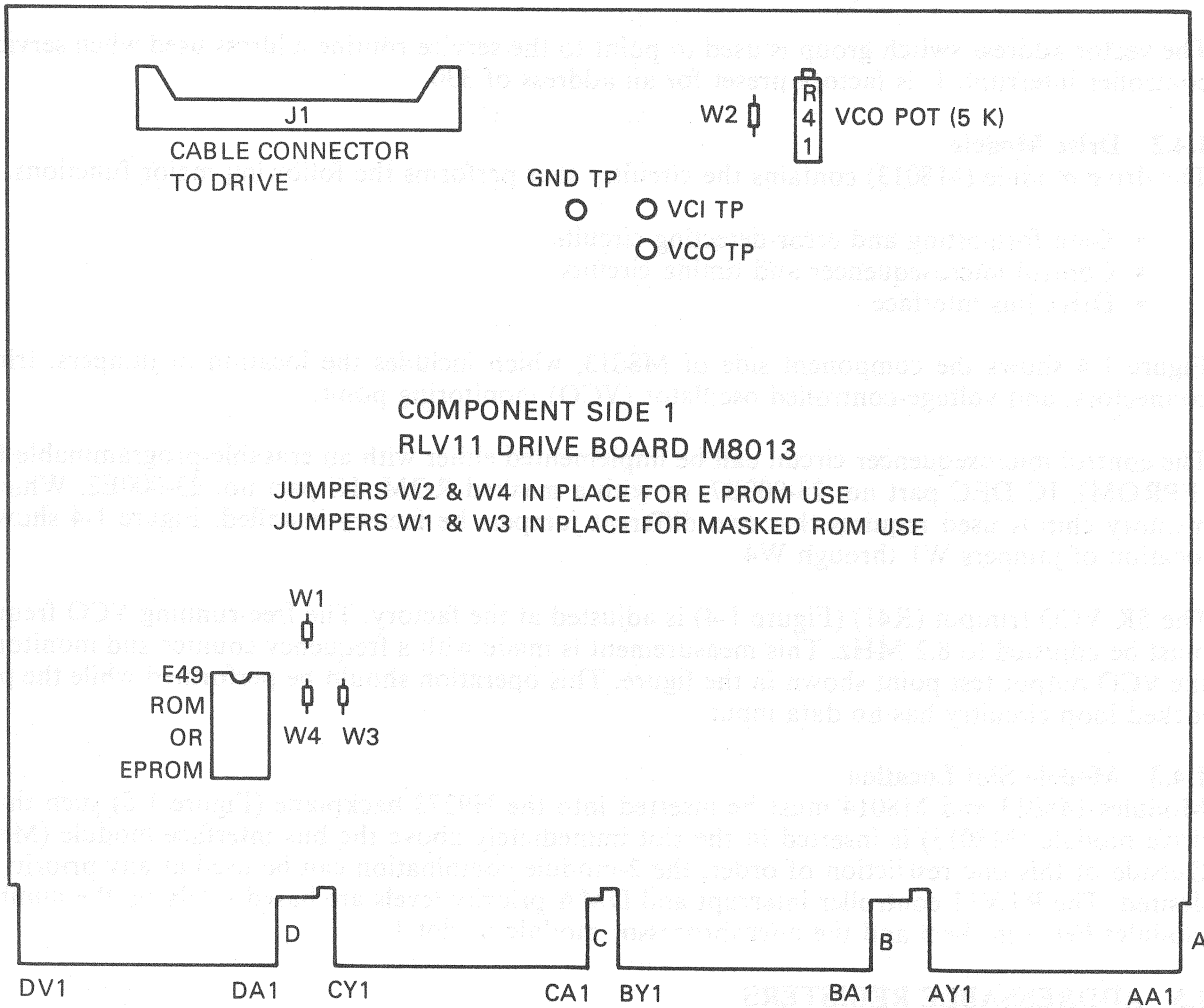
The 5K VCO trimpot (R41) (Figure 1-4) is adjusted at the factory. The free-running VCO frequency must be adjusted to 8.2 MHz. This measurement is made with a frequency counter and monitored at the VCO output test point shown in the figure. This operation should be performed while the phase-locked loop circuitry has no data input.

1.4.3 Module Slot Location

Modules M8013 and M8014 must be inserted into the H9273 backplane (Figure 1-5) such that the drive module (M8013) is inserted in the slot immediately above the bus interface module (M8014). Outside of this one restriction of order, the 2-module combination can be used at any priority level desired. The RLV11 controller interrupt and DMA priority levels are based solely on the number of modules between them and the microprocessor module in slot 1.

1.5 ADDRESSABLE REGISTERS

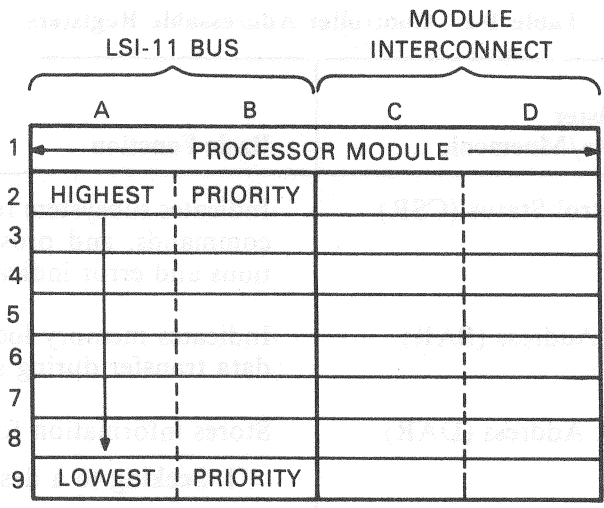
The controller contains four LSI-11 bus word-addressable registers. Each register has a read/write capability. The contents of two of the registers, the disk address register (DAR) and the multipurpose register (MPR), can have different meanings, depending on which command is being executed. Table 1-1 lists the register addresses and describes the basic functions of each. A complete description of the register bit configurations and functions is in Chapter 3 of this manual.



NOTE:
JUMPERS ARE 0-OHM COMPOSITION RESISTORS.

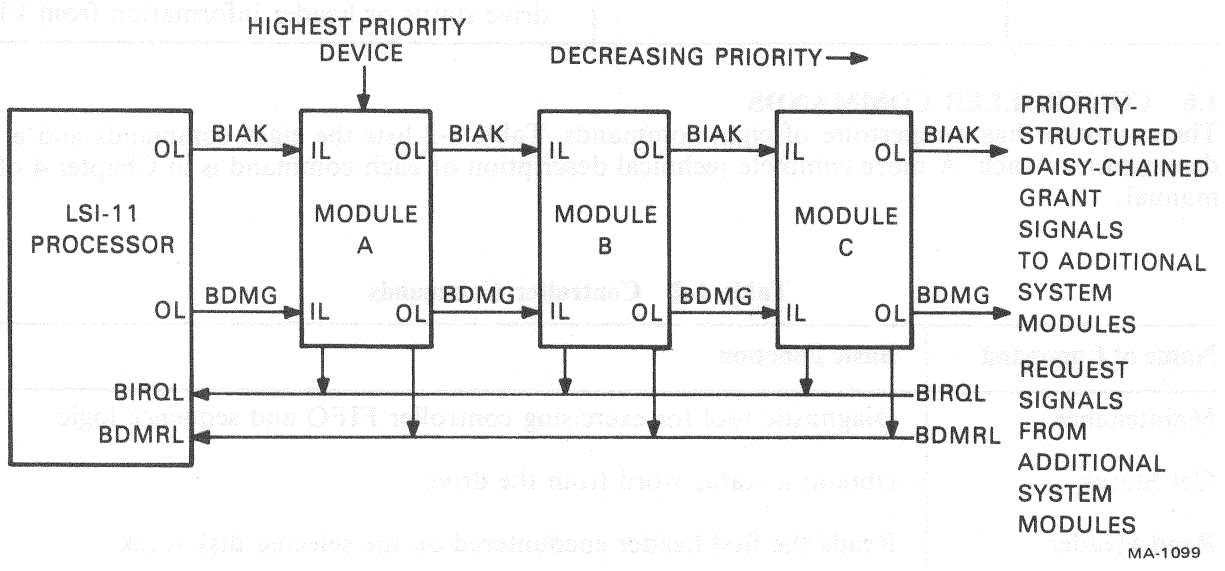
MA-1167

Figure 1-4 RLV11 Drive Module (M8013)



MA-1100

a. Module Side View of 9-Slot Backplane



b. Priority Interconnection

Figure 1-5 H9273 Backplane Priority Structure

Table 1-1 Controller Addressable Registers

Standard Address (Octal)	Register Name/Mnemonic	Basic Function
174400	Control Status (CSR)	Indicates subsystem ready condition, holds drive commands, and provides overall control functions and error indications.
174402	Bus Address (BAR)	Indicates memory location involved in a DMA data transfer during a read or write operation.
174404	Disk Address (DAR)	Stores information for: <ol style="list-style-type: none"> 1. Seeking to a desired track 2. Selecting sectors to be transferred during read/write operations. Also used to request a drive status message.
174406	Multipurpose (MPR)	Functions as a DMA word counter when transferring read/write data between LSI-11 bus and controller. Acts as a storage buffer when reading drive status or header information from FIFO.

1.6 CONTROLLER COMMANDS

The controller has a repertoire of eight commands. Table 1-2 lists the eight commands and a brief description of each. A more complete technical description of each command is in Chapter 4 of this manual.

Table 1-2 Controller Commands

Name of Command	Basic Function
Maintenance	Diagnostic tool for exercising controller FIFO and sequence logic
Get Status	Obtains a status word from the drive.
Read Header	Reads the first header encountered on the selected disk track.
Seek	Causes drive to seek to desired cylinder and/or select a new head.
Write Data	Writes data from memory to selected disk track.
Write Check	Reads newly written block of data and compares it with original data in memory.
Read Data	Reads data from selected disk track into memory.
Read Data Without Header Check	Reads serial data from selected disk track without performing header comparison. (This enables data recovery in case a header is not found.)

1.7 REFERENCE DOCUMENTS

The documents listed in Table 1-3 contain information necessary to understand the function, installation, operation, programming, and maintenance of the RLV11 controller and RL01 disk drive.

Table 1-3 Reference Documents

Title	Document No.
RL01 Disk Subsystem User's Manual	EK-RL01-OP
RL01 Disk Drive Technical Description Manual	EK-RL01-TD
RL01 Disk Subsystem Service Manual	EK-RL01-SV
RL01 Disk Subsystem Preventive Maintenance	EK-RL01-PM
RL01 Disk Drive Illustrated Parts Breakdown	EK-RL01-IP

1.8 CONTROLLER SPECIFICATIONS

The performance, power, and physical specifications for the RLV11 controller are listed in Table 1-4.

Table 1-4 RLV11 Controller Specifications

Characteristic	Specification
Required Mounting Space	Two quad-height modules that mount in an LSI-11 backplane with CD interconnections.
Power Requirements	+5 Vdc \pm 5% @ 6.5 A max +12 Vdc \pm 5% @ 1 A max
No. Drives/Controller	Up to 4
No. of LSI-11 Bus Addressable Registers	4
Device Base Address	774 400 (octal), switch selectable
Device Interrupt Vector	000 330 (octal), switch selectable
Data Transfer Rates	40-Sector (16-bit data words): 4.9 μ s/word (average) drive to controller, controller to memory 3.9 μ s/word (peak) drive to controller 2.0 μ s/word (peak) controller to memory
Error Detection Capability	Cyclic redundancy checking (CRC) on data and headers
Maximum Cable Length between Controller and Last Drive	30 m (100 ft)
Nonoperating Environment	
Temperature/Humidity	-40° to 66° C (-40° to 151° F) and 90% to 95% relative humidity

Table 1-4 RLV11 Controller Specifications (Cont)

Characteristic	Specification
Altitude	9.1 km (30,000 ft)
Shock Test	Individual packages are designed to withstand half-sine shock pulses of 40 Gpk and 30 ± 10 ms duration.
Vibration	
Vertical Axis Excitation	1.40 Grms overall from 10–300 Hz Power Spectral Density: 0.029 g ² /Hz from 10–50 Hz with 8 dB/octave rolloff from 50–300 Hz
Longitudinal and Lateral Axis Excitation	0.68 Grms overall from 10–200 Hz Power Spectral Density: 0.007 g ² /Hz from 10–50 Hz with 8 dB/octave rolloff from 50–200 Hz
Operating Environment	
Ambient Temperature	5° to 50° C (41° to 122° F)
Relative Humidity	10% or less to 95% with maximum wet bulb, 32° C (90° F) and minimum dew point, 2° C (36° F)
Altitude	2.4 km (8000 ft)
Vibration	5–50 Hz 0.004 DA 50–500 Hz 0.50 Gpk 500–50 Hz 0.50 Gpk 50–5 Hz 0.004 DA Sweep rate of 1 octave/minute
Shock Test	The product operates while a half-sine shock pulse of 10 Gpk and 10 ± 3 ms duration is applied once in either direction of three orthogonal axes.

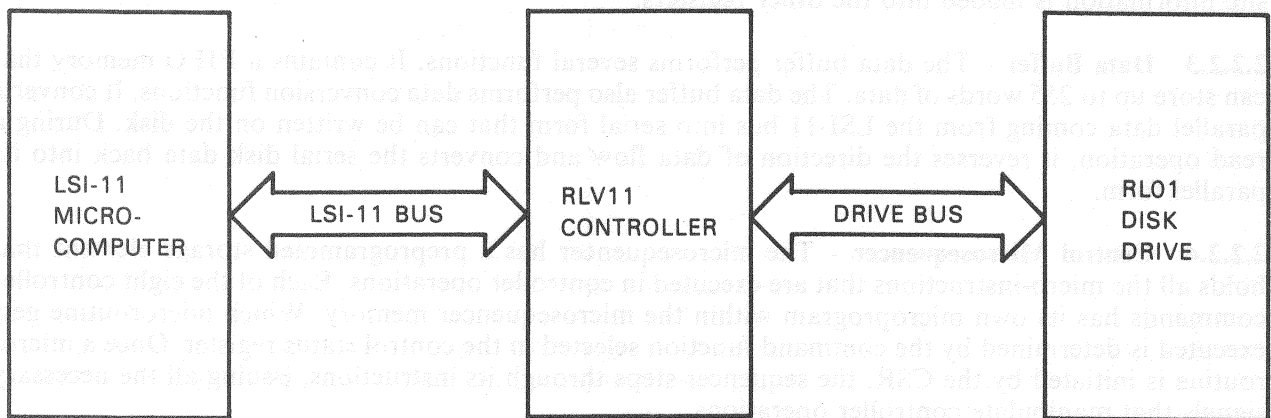
CHAPTER 2 SYSTEM LEVEL DESCRIPTION

2.1 GENERAL

This chapter provides a simplified block diagram discussion of overall system operation. The various controller commands are discussed in a general way. A more detailed description of the controller commands is in Chapter 4.

2.2 SYSTEM BLOCK DIAGRAM

In its maximum configuration, the controller can support up to four RL01 disk drives, but needs only one drive in its minimum configuration. In this chapter, the minimum configuration illustrated in Figure 2-1 is used to demonstrate system concepts.



MA-1112

Figure 2-1 System Block Diagram

2.2.1 LSI-11 Microcomputer

The use of large-scale-integration (LSI) technology enables DIGITAL to put a central processor and 4K of RAM memory on one quad-height printed circuit module. Starting with the basic LSI-11 processor module and its growing list of options, many LSI-11 microcomputer systems can be configured.

The microcomputer is the most intelligent system element and exercises overall system control. During program execution, it continually monitors interrupt requests from its terminals and mass storage devices, and provides service as required.

In the RL01 disk-based microcomputer system, the data storage capability of the system can be extended to over 20 megabytes. To use this data base effectively, the microcomputer must be able to read or write data on the disks easily.

2.2.2 RLV11 Controller

The RLV11 controller is the interface between the RL01 disk drives and the LSI-11 microcomputer. Once activated by the microcomputer, the RLV11 transfers data between the disk and the main memory. The major functional sections of the RLV11 controller are shown in Figure 2-2.

2.2.2.1 LSI-11 Bus Interface – The RLV11 controller communicates with the LSI-11 microcomputer by means of the LSI-11 bus. This bus provides a bidirectional path for data and control information. To receive and respond to this information, the controller contains an LSI-11 bus interface circuit. It performs such functions as encoding and decoding bus addresses, regulating bus timing for interrupts and direct memory access (DMA) requests, and controlling data transfers to main memory.

2.2.2.2 Programmer Interface – Essential to executing controller commands is a set of programmable registers. These registers can be written into or read by the CPU and provide the programmer with an interface to the controller. In this case, the programmer interface consists of four registers: bus address register (BAR), disk address register (DAR), control status register (CSR), and multipurpose register (MPR). These registers can be addressed like any other CPU memory location. Of the four registers, the CSR is always written last because it initiates the microsequencer operation after all the prerequisite information is loaded into the other registers.

2.2.2.3 Data Buffer – The data buffer performs several functions. It contains a FIFO memory that can store up to 255 words of data. The data buffer also performs data conversion functions. It converts parallel data coming from the LSI-11 bus into serial form that can be written on the disk. During a read operation, it reverses the direction of data flow and converts the serial disk data back into its parallel form.

2.2.2.4 Control Microsequencer – The microsequencer has a preprogrammed storage element that holds all the micro-instructions that are executed in controller operations. Each of the eight controller commands has its own microprogram within the microsequencer memory. Which microroutine gets executed is determined by the command function selected in the control status register. Once a microroutine is initiated by the CSR, the sequencer steps through its instructions, issuing all the necessary signals that manipulate controller operations.

2.2.2.5 Data Formatter – The data formatter circuits condition the serial data leaving or coming to the disk in a special way. When writing on the disk, the serial data must pass through a write pre-compensation circuit. Here the data is converted to its modified frequency modulated (MFM) form. The data pulses are also modulated in time to precompensate for a peak shifting effect that occurs in magnetic recording.

Data read off the disk passes through a phase-locked loop oscillator and data separator circuit. It is here that disk data is converted back from MFM to its digital word format.

2.2.2.6 Drive Bus Interface – The drive bus interface is used to transmit and receive drive bus signals. These transceivers are used to drive differential pair signals. The drive bus provides the means of communication between the RL01 disk drive and the RLV11 controller.

2.2.3.1 Drive Features - The RLV11 disk drive employs the "servo-in-data" concept. This concept permits the detection of head positioning and track counting information from pulses embedded within the data record. In effect, each read of the head seeking to a desired track becomes its own servo transport.

The disk drive consists essentially of a spindle motor, head positioner, drive electronics, power supply, control console, and front panel assembly. The storage medium for the drive is a four-floppy disk cartridge with a total data storage capacity of approximately 0.5 megabyte.

2.2.3.2 Disk Formatting - The RLV11 disk cartridge contains a single disk with two recording surfaces. Each recording surface has a total of 330 tracks. When the top and bottom surfaces are combined, they yield 330 cylinders.

Each track on a recording surface is subdivided into 40 equal-length sectors, which are further subdivided into fields. Each sector contains a total of 140 words of 16 bits each (Figure 2-2). For the top only 130 of the 140 words contain data.

When the disk cartridge is formatted in the factory, both servo and header and raised preamble and postamble fields are pre-recorded in each sector. The servo information is contained in two fields. The first field contains the track address. The second field contains the radial position of the head.

Header Preamble P01 - These three words precede the header information and contain 47 bits followed by a marker bit to indicate the start of valid information.

Header - This field contains three words of 16 bits each. The first word identifies the drive head (upper or lower), the cylinder address (1 of 330), and the sector address (1 of 40). The second word is all 0's. The third word is the header CRC check word. This check word is pre-recorded on the track, as are the other two header words. During a read of the header, the header is checked for recording errors. If one is detected, a header CRC error is flagged.

Header Preamble P02 - This field contains 16 bits. It separates the header and data fields to protect the header information from damage when the current is switched on in the next sector.

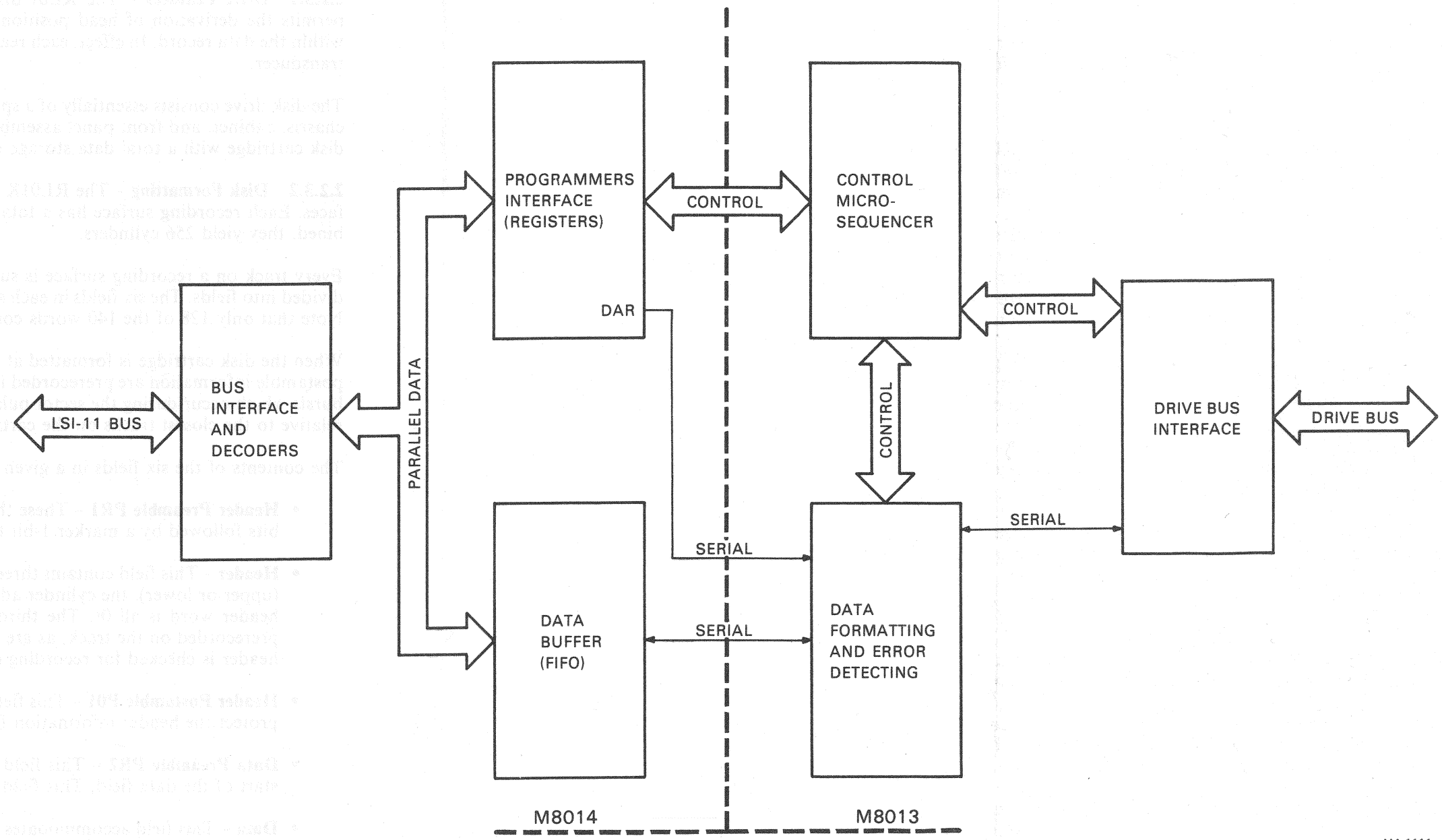
Header Preamble P03 - This field contains 47 bits followed by a marker bit to indicate the start of the data field. This field is written whenever data is recorded.

Data - This field communicates a block of 130 16-bit data words (2048 bits) followed by a 16-bit auto-check word. When writing data to a drive, a data CRC word is generated in the controller and appended to the 138-word data block. The contents of the CRC word are written to the contents of the data block. When reading the data from the disk, the data and CRC words are checked in the same controller channel. Detection of a data recording error results in a data CRC flag.

Data Preamble P04 - This field contains 16 bits. When recording data, write errors are detected at the end of the data field. Data CRC information will not be detected.

In order to reduce long start-up times, the RLV11 disk drive employs a "servo-in-data" concept. This concept permits the detection of head positioning and track counting information from pulses embedded within the data record. In effect, each read of the head seeking to a desired track becomes its own servo transport.

The disk drive consists essentially of a spindle motor, head positioner, drive electronics, power supply, control console, and front panel assembly. The storage medium for the drive is a four-floppy disk cartridge with a total data storage capacity of approximately 0.5 megabyte.



MA-1111

Figure 2-2 RLV11 Controller Major Functional Blocks

2.2.3 RL01 Disk Drive

2.2.3.1 Drive Features – The RL01 disk drive employs the “servo-in-data” concept. This concept permits the derivation of head positioning and track counting information from pulses embedded within the data record. In effect, each read/write head seeking to a desired track becomes its own servo transducer.

The disk drive consists essentially of a spindle motor, head positioner, drive electronics, power supply, chassis, 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 megabytes.

2.2.3.2 Disk Formatting – The RL01K data cartridge contains a single disk with two recording surfaces. Each recording surface has a total of 256 tracks. When the top and bottom surfaces are combined, they yield 256 cylinders.

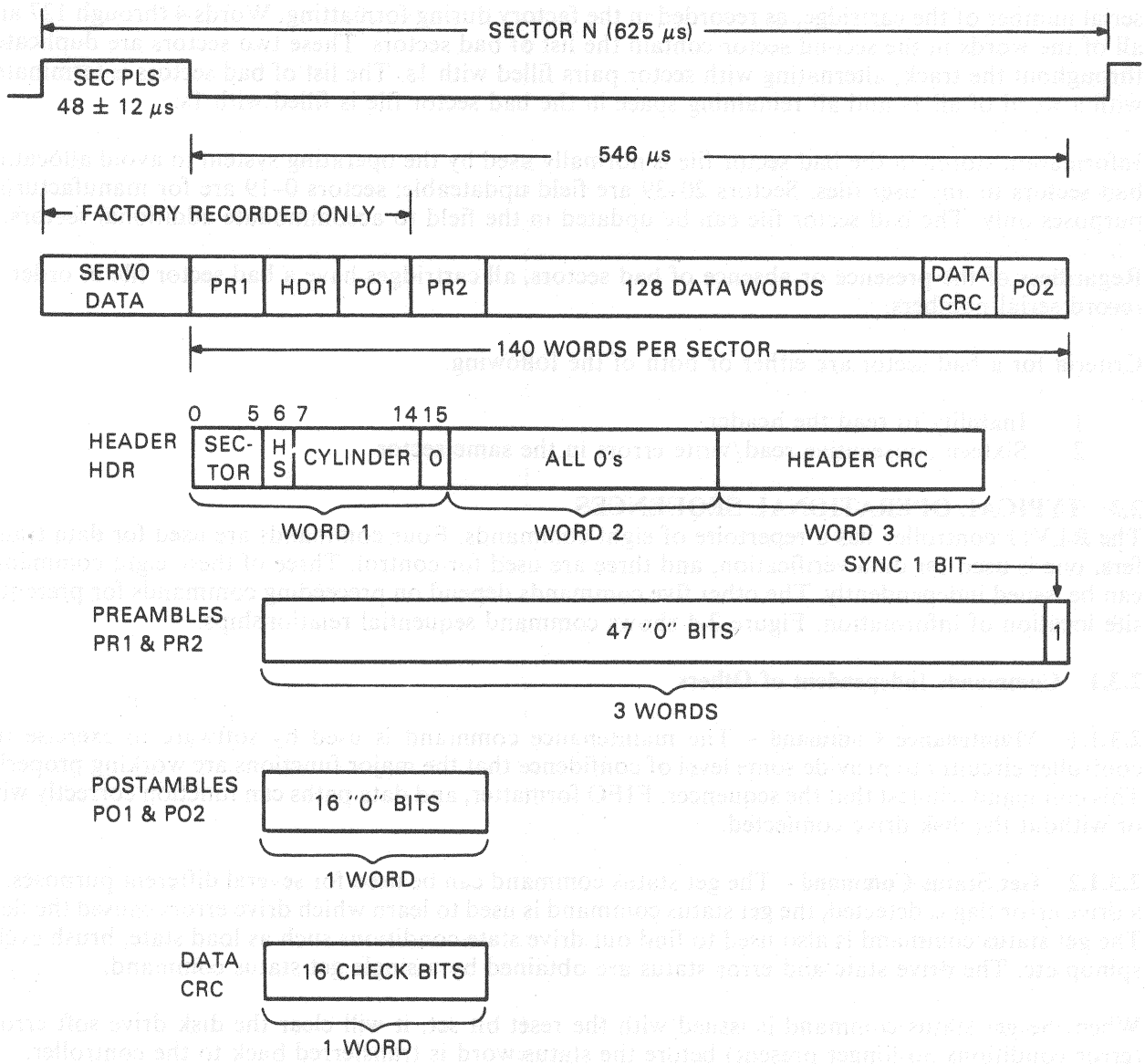
Every track on a recording surface is subdivided into 40 equal-length sectors, which are further subdivided into fields. The six fields in each sector contain a total of 140 words of 16 bits each (Figure 2-3). Note that only 128 of the 140 words contain data.

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

The contents of the six fields in a given sector include the following.

- **Header Preamble PR1** – These three words precede the header information and contain 47 0-bits followed by a marker 1-bit to indicate the start of valid information.
- **Header** – This field contains three words of 16 bits each. The first word identifies the drive head (upper or lower), the cylinder address (1 of 256), and the sector address (1 of 40). The second header word is all 0s. The third word is the header CRC check word. This check word is prerecorded on the track, as are the other two header words. During a read of the header, the header is checked for recording errors; if one is detected, a header CRC error is flagged.
- **Header Postamble P01** – This field contains 16 0-bits. It separates the header and data fields to protect the header information from damage when write current is switched on in the head.
- **Data Preamble PR2** – This field contains 47 0-bits followed by a marker 1-bit to indicate the start of the data field. This field is rewritten whenever data is recorded.
- **Data** – This field accommodates a block of 128 16-bit data words (2048 bits) followed by a 16-bit data check word. When writing data to a drive, a data CRC word is generated in the controller and appended to the 128-word data block. The contents of the CRC word vary with the contents of the data block. When reading the data from the drive, the data block and CRC word are checked in the same controller circuit. Detection of a data recording error results in a data CRC flag.
- **Data Postamble P02** – This field consists of 16 0-bits. When recording data, write current turnoff is delayed until the end of this field so that data CRC information will not be disturbed.

To ensure a sector boundary required by mechanical tolerances, fixed time delays are introduced between the leading and trailing edges of the sector pulses and the adjoining preamble and postamble pulses.



MA-1101

Figure 2-3 Sector Format

2.2.3.3 Bad Sector File – The bad sector file is the list of bad sectors on the cartridge, and is recorded on the lower surface of the disk at track 255 (decimal). The track contains 40 sectors with 128 data words each. Contents of the first sector of the bad sector file are as follows: Words 0 and 1 contain the serial number of the cartridge, as recorded in the factory during formatting. Words 4 through 127 and all of the words in the second sector contain the list of bad sectors. These two sectors are duplicated throughout the track, alternating with sector pairs filled with 1s. The list of bad sectors is terminated with a word of all 1s and all remaining space in the bad sector file is filled with 1s.

Information stored in the bad sector file is normally used by the operating system to avoid allocating bad sectors to any user files. Sectors 20–39 are field updateable; sectors 0–19 are for manufacturing purposes only. The bad sector file can be updated in the field to accommodate additional sectors.

Regardless of the presence or absence of bad sectors, all cartridges have a bad sector file in order to record serial numbers.

Criteria for a bad sector are either or both of the following.

1. Inability to read the header
2. Sixteen consecutive read/write errors in the same sector

2.3 TYPICAL OPERATIONAL SEQUENCES

The RLV11 controller has a repertoire of eight commands. Four commands are used for data transfers, one is used for data verification, and three are used for control. Three of these eight commands can be issued independently. The other five commands depend on preceding commands for prerequisite location of information. Figure 2-4 shows command sequential relationships.

2.3.1 Commands Independent of Others

2.3.1.1 Maintenance Command – The maintenance command is used by software to exercise the controller circuitry to provide some level of confidence that the major functions are working properly. This command will test that the sequencer, FIFO formatter, and data paths can function correctly with or without the disk drive connected.

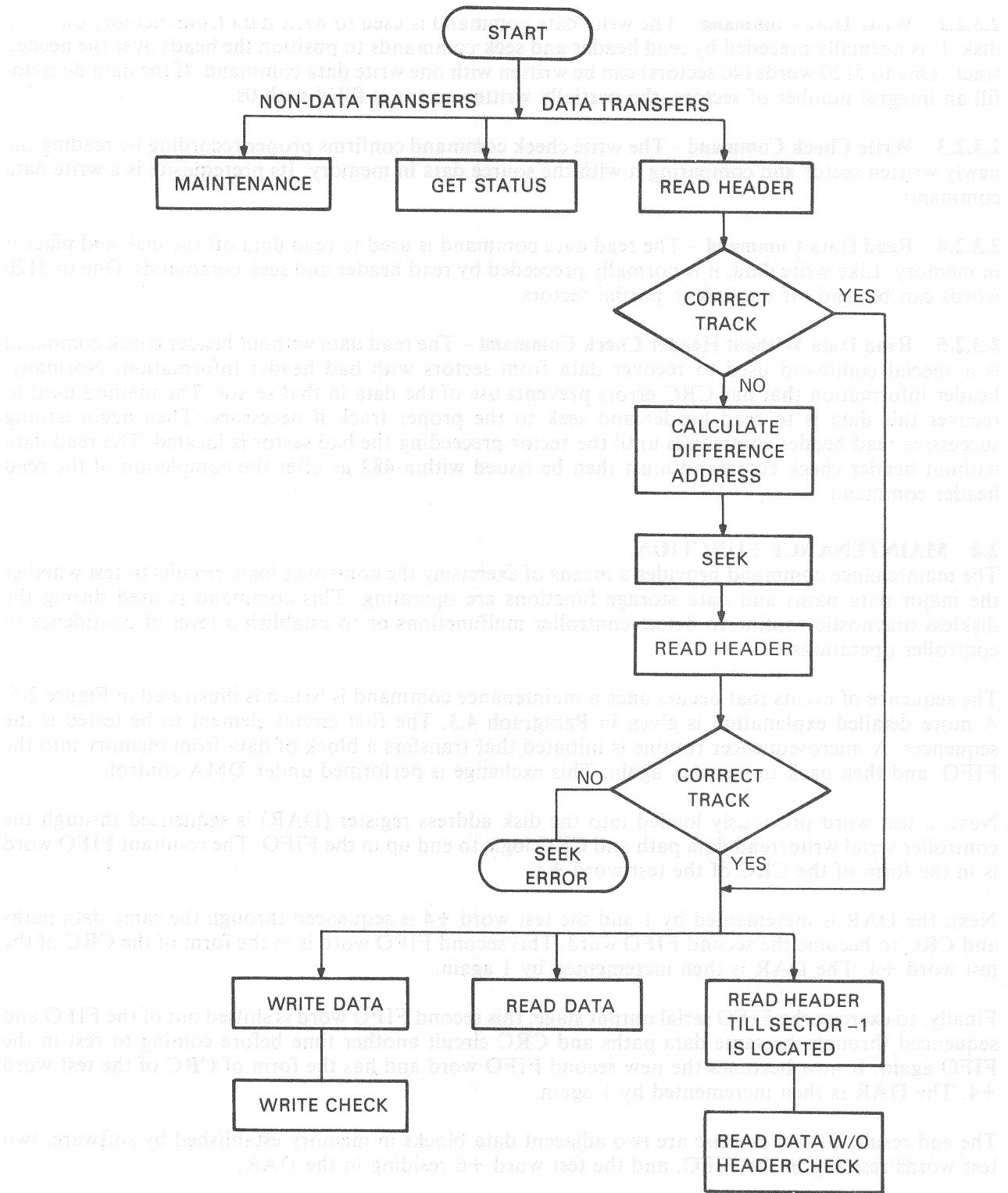
2.3.1.2 Get Status Command – The get status command can be used for several different purposes. If a drive error flag is detected, the get status command is used to learn which drive errors caused the flag. The get status command is also used to find out drive state conditions such as load state, brush cycle, spinup etc. The drive state and error status are obtained by a single get status command.

When the get status command is issued with the reset bit set, it will clear the disk drive soft errors (error conditions no longer present) before the status word is transferred back to the controller.

2.3.1.3 Read Header Command – The read header command will read the first header encountered on the selected drive. The header provides cylinder, sector, and selected head information. The information provided by the read header command forms the basis for software calculating a difference address to be used for a seek command.

2.3.2 Commands Dependent on Others

2.3.2.1 Seek Command – The seek command is used to select the read/write heads or to reposition them at a new cylinder location. The seek command is normally preceded by a read header command so that new head positioning data can be obtained. The seek command is usually followed by write data or read data commands in normal usage. In cases of trying to recover bad sectors, a different sequence may be used. The drive head selected during a seek remains selected after the seek is complete.



MA-1169

Figure 2-4 Sequential Relationship of Controller Commands

2.3.2.2 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 needed track. One to 5120 words (40 sectors) can be written with one write data command. If the data does not fill an integral number of sectors, the partially written sector is filled with 0s.

2.3.2.3 Write Check Command – The write check command confirms proper recording by reading the newly written sector and comparing it with the source data in memory. Its prerequisite is a write data command.

2.3.2.4 Read Data Command – The read data command is used to read data off the disk and place it in memory. Like write data, it is normally preceded by read header and seek commands. One to 5120 words can be read off the disk in partial sectors.

2.3.2.5 Read Data Without Header Check Command – The read data without header check command is a special command used to recover data from sectors with bad header information. Normally, header information that has CRC errors prevents use of the data in that sector. The method used to recover this data is to read header and seek to the proper track if necessary. Then begin issuing successive read header commands until the sector preceding the bad sector is located. The read data without header check command must then be issued within 482 μ s after the completion of the read header command.

2.4 MAINTENANCE FUNCTION

The maintenance command provides a means of exercising the controller logic circuits to test whether the major data paths and data storage functions are operating. This command is used during the diskless diagnostic routine to detect controller malfunctions or to establish a level of confidence in controller operations.

The sequence of events that occurs once a maintenance command is issued is illustrated in Figure 2-5. A more detailed explanation is given in Paragraph 4.3. The first circuit element to be tested is the sequencer. A microsequencer routine is initiated that transfers a block of data from memory into the FIFO, and then back to memory again. This exchange is performed under DMA control.

Next, a test word previously loaded into the disk address register (DAR) is sequenced through the controller serial write/read data path and CRC logic to end up in the FIFO. The resultant FIFO word is in the form of the CRC of the test word +3.

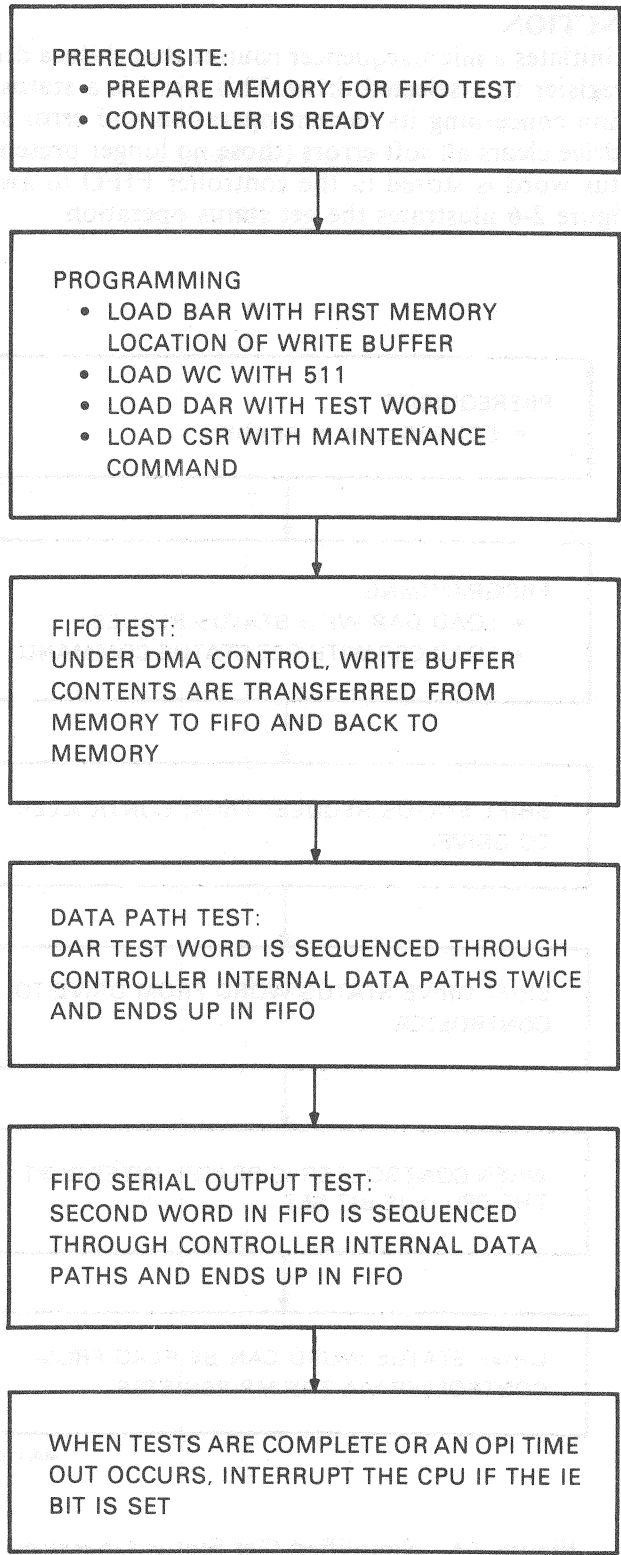
Next, the DAR is incremented by 1 and the test word +4 is sequenced through the same data paths and CRC to become the second FIFO word. This second FIFO word is in the form of the CRC of the test word +4. The DAR is then incremented by 1 again.

Finally, to exercise the FIFO serial output stage, this second FIFO word is shifted out of the FIFO and sequenced through the same data paths and CRC circuit another time before coming to rest in the FIFO again. It now becomes the new second FIFO word and has the form of CRC of the test word +4. The DAR is then incremented by 1 again.

The end results of this exercise are two adjacent data blocks in memory established by software, two test words residing in the FIFO, and the test word +6 residing in the DAR.

One of the data blocks in memory, the write FIFO buffer, occupies 256 locations. The other, the read FIFO buffer, occupies 255 locations.

The two words residing in the FIFO can be accessed via the multipurpose register. The first word is a test of the data paths. The second word is a test of the data paths plus the FIFO serial output stage. These words are read into the CPU and software checked for malfunctions.

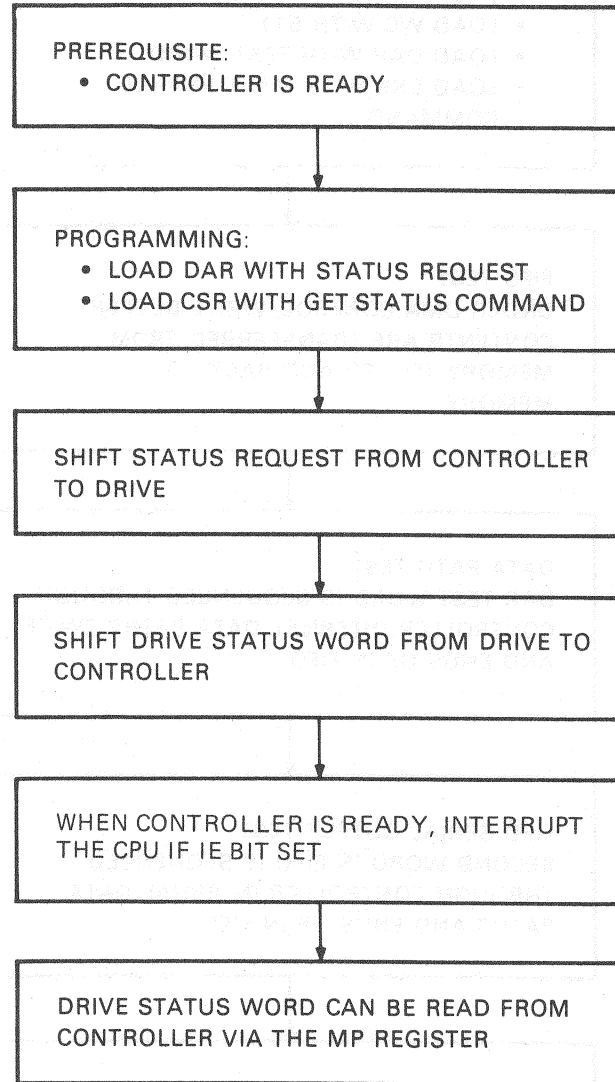


MA-1110

Figure 2-5 Simplified Maintenance Operation

2.5 GET STATUS FUNCTION

The get status command initiates a microsequencer routine that shifts a drive command word from the controller disk address register to a selected drive. This word is a status request word that asks the drive to return information concerning its current operation and error status. If the reset bit in the status request is set, the drive clears all soft errors (those no longer present) before sending back drive status. The returned status word is stored in the controller FIFO to await later access through the multipurpose register. Figure 2-6 illustrates the get status operation.



MA-1109

Figure 2-6 Simplified Get Status Operation

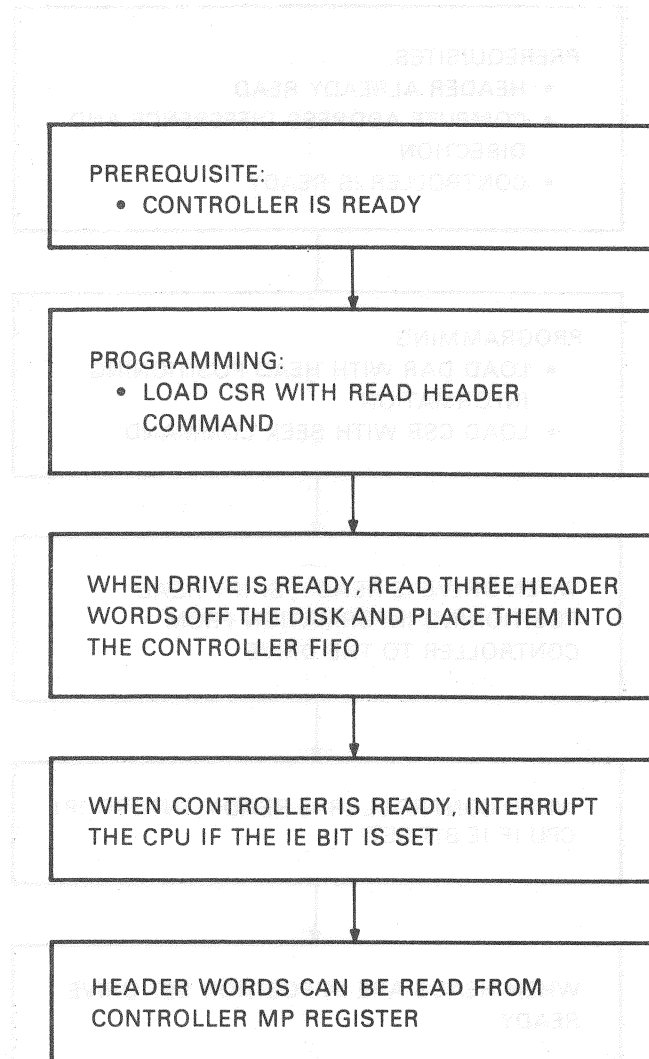
One prerequisite for issuing the get status command is a knowledge that the controller is in the ready state. It is important to note that the drive does not have to be ready (for example, during a seek or when in the load state) to issue a get status command.

The only programming prerequisite is that the status request word must be loaded first into the disk address register before issuing the get status command.

2.6 READ HEADER FUNCTION

The function of the read header command is to read the first header encountered on the selected drive, and to store the three header words in the FIFO. One or more header words can then be extracted from the FIFO by reading the multipurpose register. Extracting the first header word alone provides sufficient head positioning information to permit software calculation of cylinder difference for a subsequent seek operation to a new track address.

The only prerequisite for issuing the read header command is a knowledge that the controller is in the ready state. Figure 2-7 illustrates the read header operation.



MA-1108

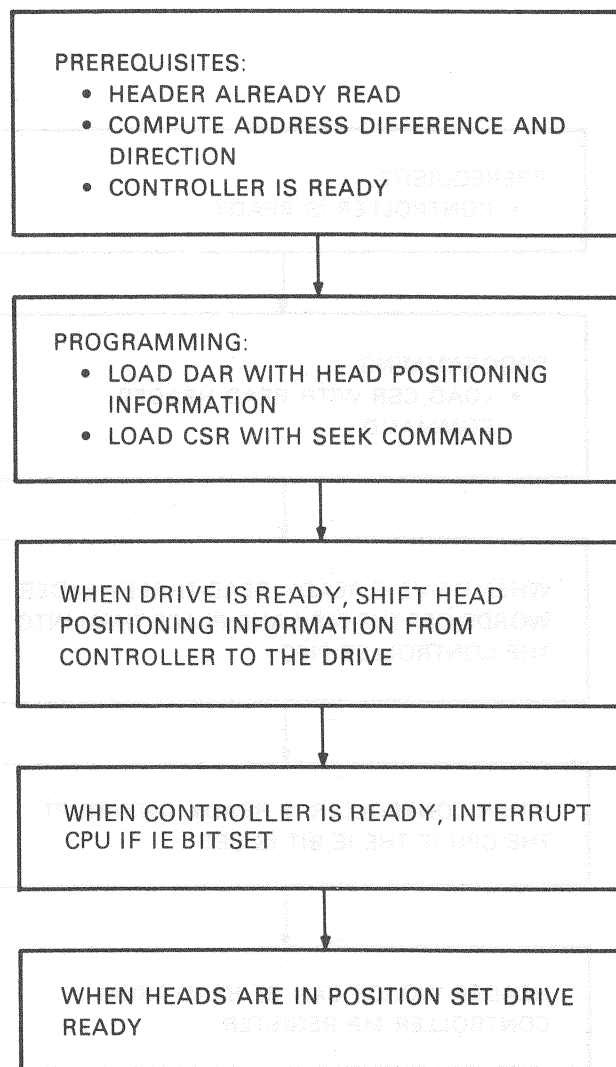
Figure 2-7 Simplified Read Header Operation

2.7 SEEK FUNCTION

The seek command initiates a microsequencer routine that shifts a drive command word from the controller disk address register to the drive. This drive command word contains head positioning information that includes the cylinder distance to be moved, the direction of movement, and the head to be selected for the next data transfer operation. Once this positioning information is received by the drive, the heads seek to the new track location. Figure 2-8 illustrates a simplified seek operation.

There are several prerequisites for issuing a seek command. First, a read header must have already been issued to know where the heads are located presently. Once this is known, the software must calculate the cylinder difference information needed by the drive to reposition the heads. Then, before issuing the seek command, the software must know that the controller is in the ready state.

The only programming prerequisite is to load the disk address register with the head positioning information prior to issuing the seek command.

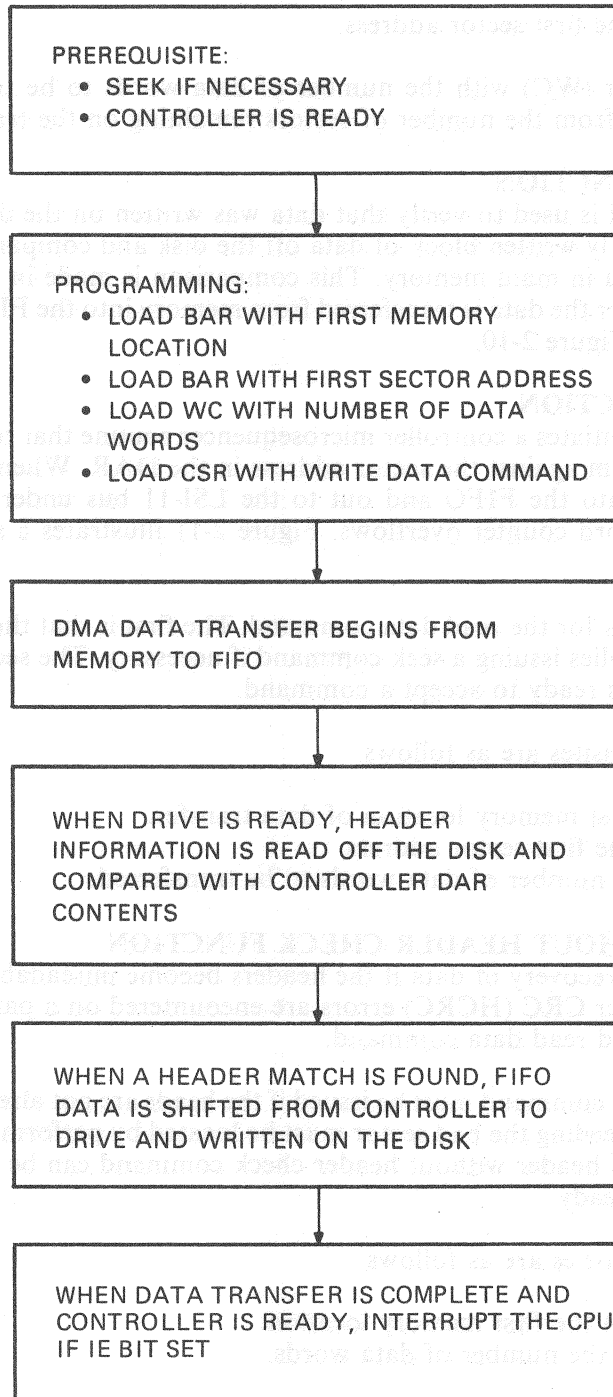


MA-1107

Figure 2-8 Simplified Seek Operation

2.8 WRITE DATA FUNCTION

The write data command initiates a microsequencer routine that enables the controller DMA circuitry. The controller eventually becomes LSI-11 bus master and data words are loaded into the FIFO. When the drive is ready, header information is continually read off the disk and compared with the sector address stored in the DAR. Once a header match is found and the FIFO contains at least 128 words, the FIFO data is written on the disk in successive sectors until the word counter overflows. For partial sector writes, the remaining sector area is filled with 0s. Figure 2-9 illustrates a simplified write data operation.



MA-1106

Figure 2-9 Simplified Write Data Operation

Prior to the write data command, the heads must already be located at the correct track. This implies issuing a seek command if necessary. Also, the software must know that the controller is in the ready state before issuing a command.

The programming prerequisites are as follows.

- Load BAR with first memory location of data transfer.
- Load DAR with the first sector address.
- Load word counter (WC) with the number of data words to be transferred. The maximum number is derived from the number of sectors remaining on the track.

2.9 WRITE CHECK FUNCTION

The write check command is used to verify that data was written on the disk correctly. It is accomplished by reading the newly written block of data off the disk and comparing it with the contents of the source data buffer area in main memory. This comparison is made in the serial header compare circuit in the controller after the data is transferred from memory into the FIFO. The basic sequence of the operation is given in Figure 2-10.

2.10 READ DATA FUNCTION

The read data command initiates a controller microsequencer routine that reads successive headers off the disk and compares them against the sector address in the DAR. When a header match is found, disk data is transferred into the FIFO and out to the LSI-11 bus under DMA control. The data transfer ends when the word counter overflows. Figure 2-11 illustrates a simplified read data operation.

There are two prerequisites for the read data command. The first is that the heads must be located at the correct track. This implies issuing a seek command if necessary. The second is that software must know that the controller is ready to accept a command.

The programming prerequisites are as follows.

- Load BAR with first memory location of data transfer.
- Load DAR with the first sector address.
- Load WC with the number of data words to be transferred.

2.11 READ DATA WITHOUT HEADER CHECK FUNCTION

This command allows the recovery of data if the headers become unreadable (Figure 2-12). If header not found (HNF) or header CRC (HCRC) errors are encountered on a particular sector, data is not recoverable by the standard read data command.

To recover this data, a seek command must be issued if the heads are not already located on the correct track. Then the sector preceding the bad sector must be located by performing successive read header commands. Finally, a read header without header check command can be issued to recover the next sector if the controller is ready.

The programming prerequisites are as follows.

- Load the BAR with the first memory location.
- Load the WC with the number of data words.

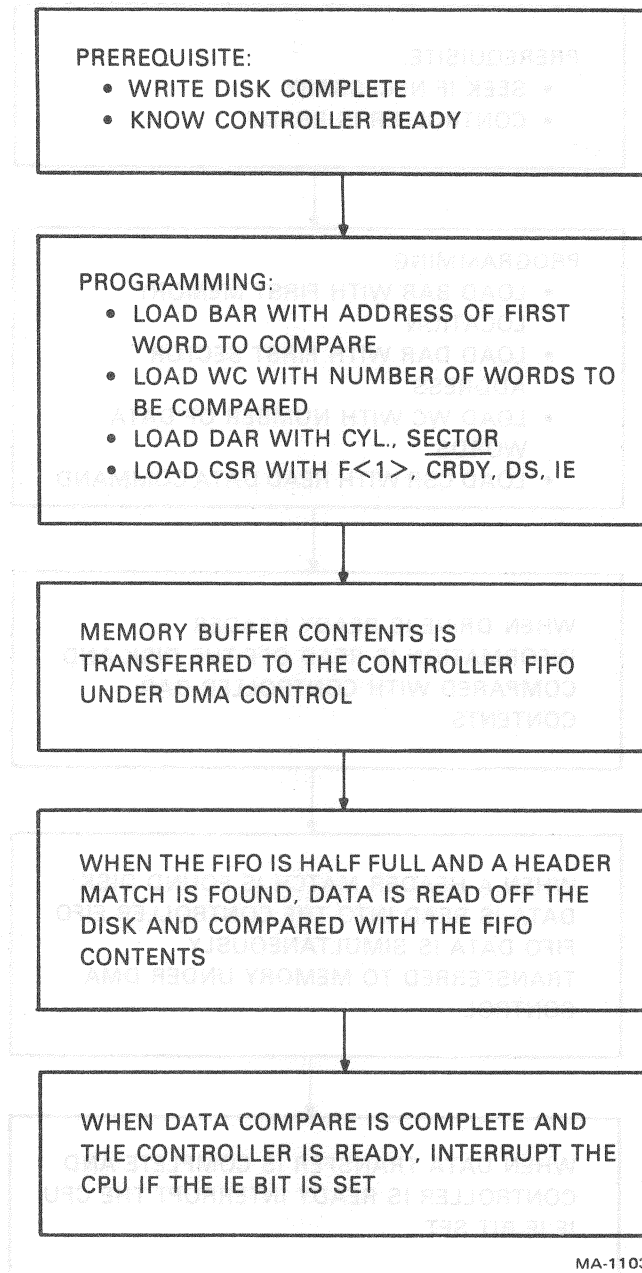
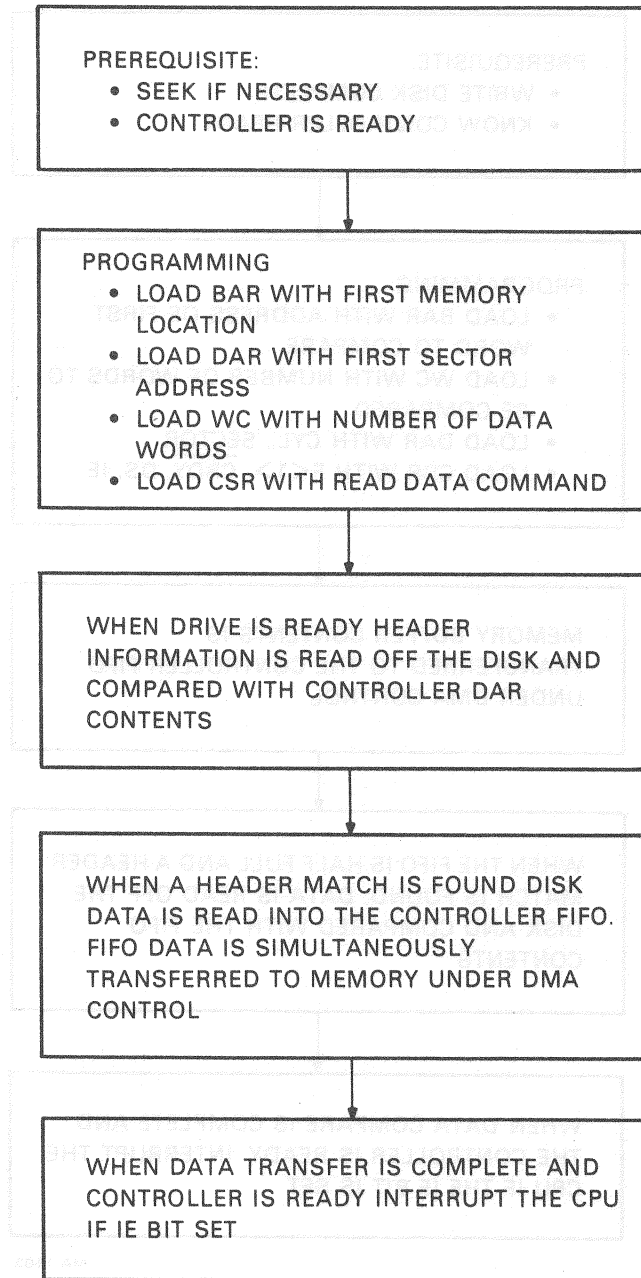
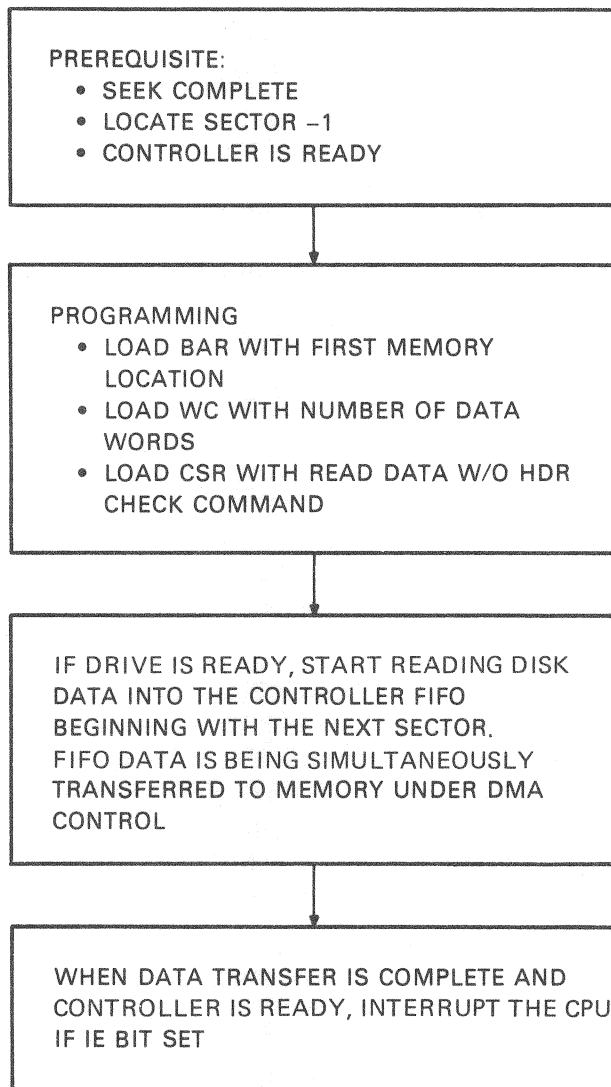


Figure 2-10 Simplified Write Check Operation



MA-1104

Figure 2-11 Simplified Read Data Operation



MA-1105

Figure 2-12 Simplified Read Data Without Header Check Operation

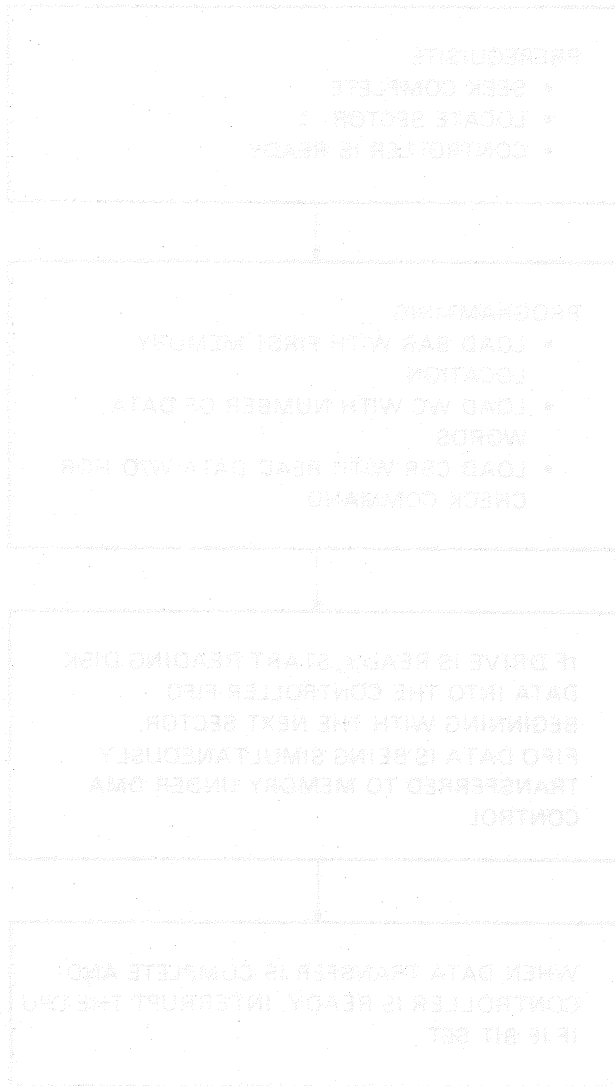


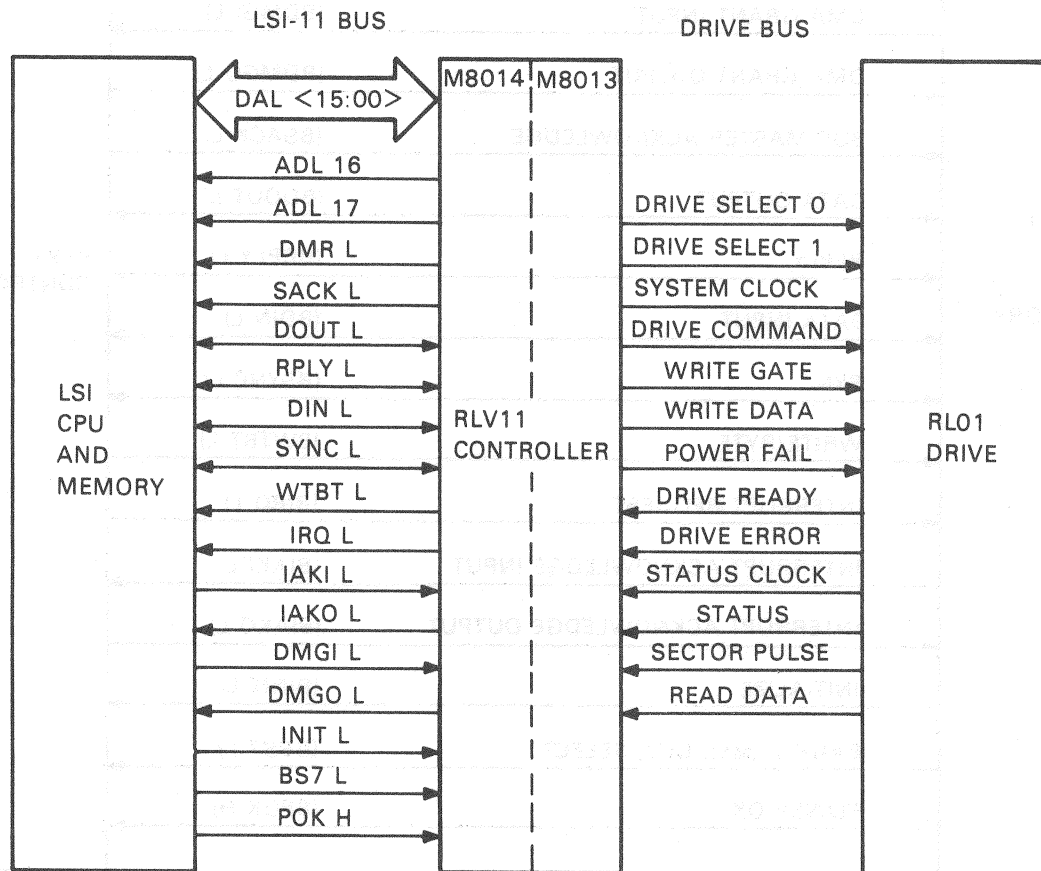
Figure 5-12 Simplified Read Data Without Header Check Operation

CHAPTER 3 INTERFACE LEVEL DESCRIPTION

3.1 GENERAL

The RLV11 controller is serviced by two separate bus organizations (Figure 3-1).

The first bus system is called the LSI-11 bus and permits the controller to enter into dialogue with the CPU, memory, and other peripherals on the bus. The LSI-11 bus is described in Paragraph 3.2. If further information is needed, refer to the *Microcomputer Handbook* published by Digital Equipment Corporation.



MA-1102

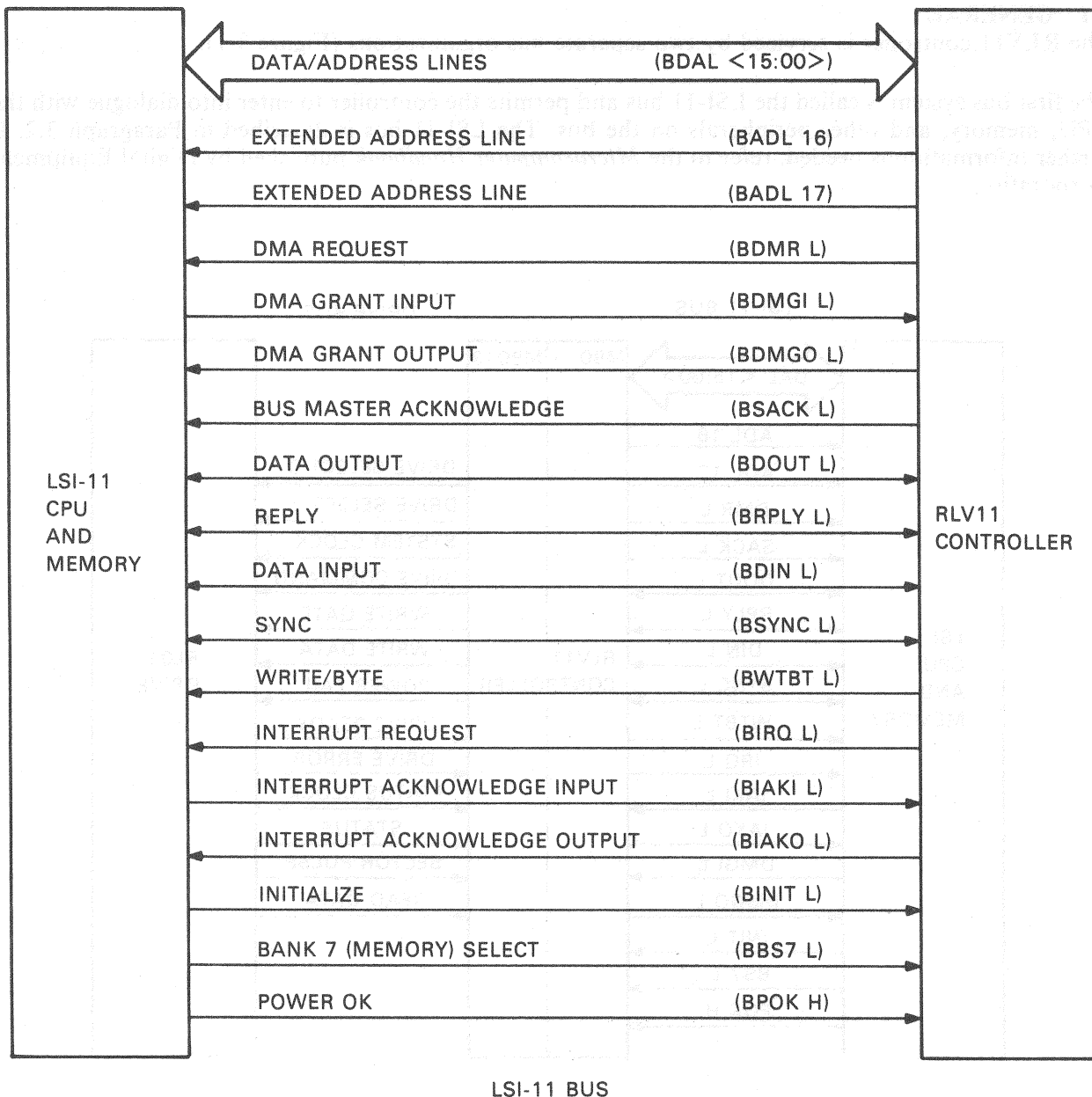
Figure 3-1 Controller Interface Signals

The second bus system servicing the controller is called the drive bus. This bus permits the controller and disk drives to communicate with one another. The drive bus signals are described in Paragraph 3.3.

Access to the disk subsystem is gained through the controller addressable registers. These registers are described in detail in Paragraph 3.4.

3.2 LSI-11 BUS

The LSI-11 bus signals used to control the RLV11 controller are shown in Figure 3-2. Note that some of the LSI-11 bus signals are bidirectional and permit 2-way communication. A complete list of all LSI-11 bus signals and their descriptions is provided in Table 3-1.



LSI-11 BUS

MA-1116

Figure 3-2 LSI-11 Bus Signals

Table 3-1 LSI-11 Bus Signals

Bus Pin	Mnemonic	Description
AC1 AD1	BAD16 BAD17	Extended address bits (not implemented)
AJ1	GND	Ground – System signal ground and dc return.
AM1	GND	Ground – Signal ground and dc return.
AN1	BDMR L	Direct Memory Access (DMA) Request – The RLV11 asserts this signal to request bus mastership. The processor arbitrates bus mastership between itself and all DMA devices on the bus. If the processor is not bus master (it has completed a bus cycle and BSYNC L is not being asserted by the processor), it grants bus mastership to the controller by asserting BDMGO L. The RLV11 responds by negating BDMR L and asserting BSACK L. DMA is used for transferring words in and out of the FIFO.
AT1	GND	Ground – System signal ground and dc return.
BA1	BDCOK H	DC Power OK – Power supply-generated signal that is asserted when there is sufficient dc voltage available to sustain reliable system operation.
BB1	BPOK H	Power OK – Asserted by the power supply when primary power is normal. RLV11 buffers this signal and drives the Drive Bus Power Fail signal. Also used to inhibit DMA transfers so that the power-fail program can be executed by the CPU without interference from the RLV11 DMA transfers.
BJ1	GND	Ground – System signal ground and dc return.
BM1	GND	Ground – System signal ground and dc return.
BN1	BSACK L	This signal is asserted by the RLV11 in response to the processor's BDMGO L signal, acknowledging that the RLV11 is bus master.
BT1	GND	Ground – System signal ground and dc return.
BV1	+5	+5 V Power – +5 Vdc system power.
AA2	+5	+5 V Power – Normal +5 Vdc system power.
AC2	GND	Ground – System signal ground and dc return.
AD2	+12	+12 V Power – +12 Vdc system power.
AE2	BDOUT L	Data Output – DBOUT, when asserted, implies that valid data is available on BDAL0 – 15 L and that an output transfer to the CPU or memory is taking place. BDOUT L is deskewed with respect to data on the bus. The recipient must assert BRPLY L to complete the transfer.

Table 3-1 LSI-11 Bus Signals (Cont)

Bus Pin	Mnemonic	Description
AF2	BRPLY L	Reply – BRPLY L is asserted in response to BDIN L or BDOUT L and during IAK transactions. It is generated by RLV11 to indicate that it has input data available on the BDAL bus or that it has accepted output data from the bus.
AH2	BDIN L	Data Input – BDIN L is used for two types of bus operations: <ol style="list-style-type: none"> 1. When asserted during BSYNC L time, BDIN L implies an input transfer from the CPU or memory and requires a response (BRPLY L). BDIN L is asserted when the RLV11 is ready to accept data as bus master. 2. When asserted without BSYNC L, it indicates that an interrupt operation is occurring.
AJ2	BSYNC L	Synchronize – BSYNC L is asserted by the bus master device to indicate that it has placed an address on BDAL0 – 15 L. The transfer is in process until BSYNC L is negated.
AK2	BWTBT L	Write/Byte – BWTBT L is asserted during the leading edge of BSYNC L to indicate that an output sequence is to follow (DATO) rather than an input sequence.
AL2	BIRQ L	Interrupt Request – RLV11 asserts this signal when its interrupt enable and interrupt request flip-flops are set. This signal informs the processor that the RLV11 has completed a function or it is ready to accept a new function. If the processor's PS word bit 7 is 0, the processor responds by acknowledging the request by asserting BDIN L and BIAKO L.
AM2 AN2	BIAKI L BIAKO L	Interrupt Acknowledge Input and Interrupt Acknowledge Output – This is an interrupt acknowledge signal that is generated by the processor in response to an interrupt request (BIRQ L). The processor asserts BIAKO L, which is routed to the BIAKI L pin of the RLV11. If it is requesting an interrupt, it inhibits passing BIAKO L. If it is not asserting BIRQ L, RLV11 passes BIAKI L to the next (lower priority) device via its BIAKO L pin and the lower priority device's BIAKI L pin.
AP2	BBS7 L	Bank 7 Select – The bus master asserts BBS7 L when an address in the upper 4K bank (address in the 28–32K range) is placed on the bus. BSYNC L is then asserted and BBS7 L remains active for the duration of the addressing portion of the bus cycle. The base address of the RLV11 is in this bank.

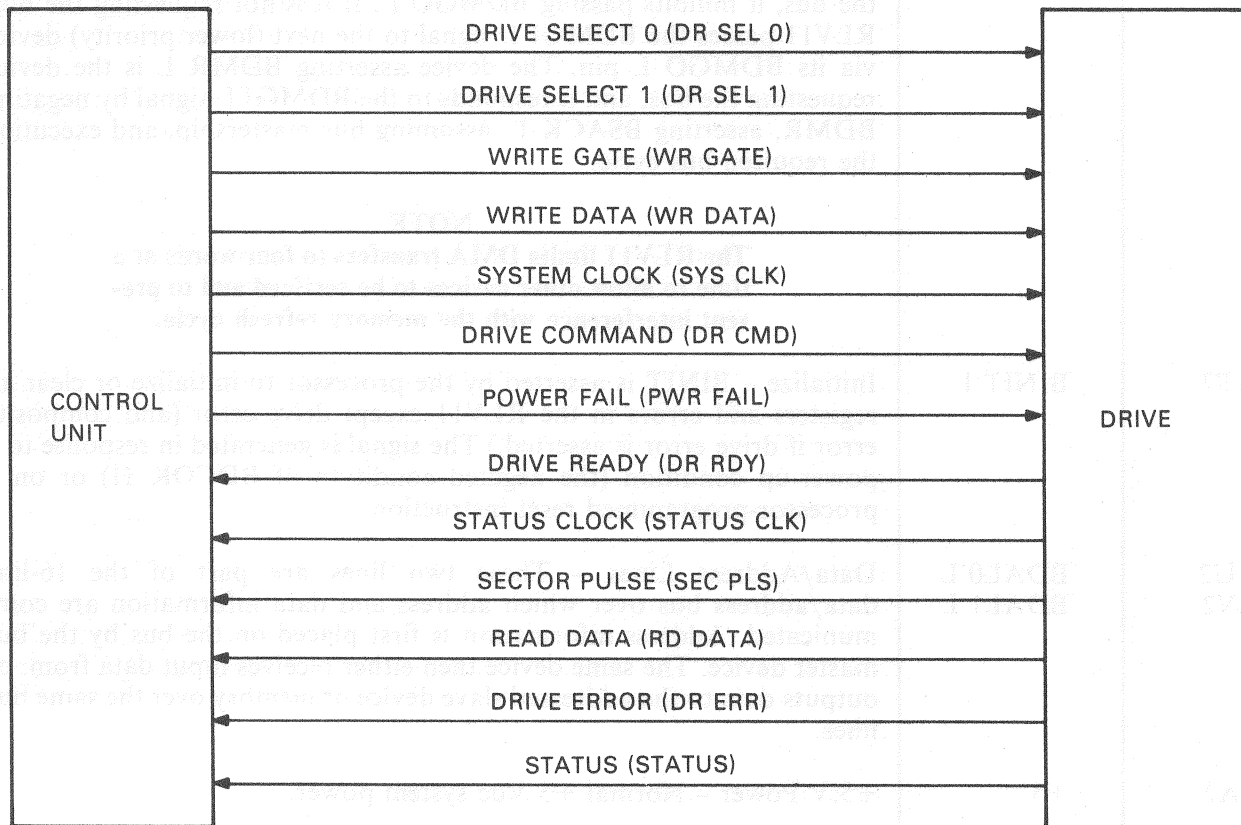
Table 3-1 LSI-11 Bus Signals (Cont)

Bus Pin	Mnemonic	Description
AR2 AS2	BDMGI L BDMGO L	DMA Grant-Input and DMA Grant-Output - This is the processor-generated daisy-chained signal that grants bus mastership to the highest priority DMA device along the bus. The processor generates BDMGO L, which is routed to the BDMGI L pin of the RLV11. If it is requesting the bus, it inhibits passing BDMGO L. If it is not requesting the bus, RLV11 passes the BDMGI L signal to the next (lower priority) device via its BDMGO L pin. The device asserting BDMR L is the device requesting the bus, and it responds to the BDMGI L signal by negating BDMR, asserting BSACK L, assuming bus mastership, and executing the required bus cycle. NOTE The RLV11 limits DMA transfers to four words at a time to allow other devices to be serviced and to prevent interference with the memory refresh cycle.
AT2	BINIT L	Initialize - BINIT is asserted by the processor to initialize or clear all registers and errors in the RLV11 except drive error (and composite error if drive error is asserted.) The signal is generated in response to a power-up condition (the negated condition of BDCOK H) or on a processor-programmed reset instruction.
AU2 AV2	BDAL0 L BDAL1 L	Data/Address Lines - These two lines are part of the 16-line data/address bus over which address and data information are communicated. Address information is first placed on the bus by the bus master device. The same device then either receives input data from, or outputs data to the addressed slave device or memory over the same bus lines.
BA2	+5	+5 V Power - Normal +5 Vdc system power.
BC2	GND	Ground - System signal ground and dc return.
BD2	+12	+12 V Power - +12 V system power.
BE2 BF2 BH2 BJ2 BK2 BL2 BM2 BN2 BP2 BR2 BS2 BT2 BU2 BV2	BDAL2 L BDAL3 L BDAL4 L BDAL5 L BDAL6 L BDAL7 L BDAL8 L BDAL9 L BDAL10 L BDAL11 L BDAL12 L BDAL13 L BDAL14 L BDAL15 L	Data/Address Lines - These 14 lines are part of the 16-line data/address bus previously described.

3.3 DRIVE BUS INTERFACE SIGNALS

3.3.1 General

The drive bus is the avenue of communication between the RLV11 controller and the RL01 disk drives. Over this bus, the controller can operate one or up to four disk drives. The bus is composed of 12 differential signals and a single-wire power-fail line. The drive bus signals are shown in Figure 3-3.



MA-0580

Figure 3-3 Drive Bus Signals

3.3.2 Drive Bus Signal Descriptions

3.3.2.1 Drive Select (DRV SEL 0,1) – These two lines select one of four disk drives, as determined by bits 8 and 9 of the CSR. 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 newly selected drive will inhibit transmission of a partial sector pulse if it is selected while its sector pulse is asserted.

3.3.2.2 System Clock (SYS CLK) – This clock shifts the drive command word to the drive and also provides a timing reference for the disk motor servo. Clock frequency is 4.1 MHz.

3.3.2.3 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 commands.

3.3.2.4 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 10 of status word) and operation terminated. Write gate is removed at the end of postamble P02.

3.3.2.5 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).

3.3.2.6 Power Fail (PWR FAIL) – This signal is provided by the RLV11 and reflects the state of power in the logic box holding the RLV11 modules. This signal is received by all drives at all times, regardless of which drive is selected. While the RLV11 controller is supplied sufficient power, PWR FAIL is negated high. If power is lost or out of tolerance, PWR FAIL is asserted low, in which case the drives unload heads and spin down. Return of power causes the drives to spin up and load heads over track 0.

3.3.2.7 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 or supply read data. The signal is negated any time a disk address difference word is sent even though no seek or head change occurs. It will also be negated if the head drifts off track or a drive error occurs. DR RDY returns after a circuit timeout or at the end of a long seek.

DR RDY will be negated when a drive error occurs except when an attempt has been made to write on a write-protected drive or if volume check is set. In either case, only DR ERR will be asserted.

3.3.2.8 Drive Error (DR ERR) – This signal is asserted on certain drive errors. Any attempt to write on a write-protected drive also causes the signal to be asserted. Asserting DR ERR causes bits 14 and 15 of the CSR to be set. The particular error involved can then be determined by initiating a get status command and reading bits 10, 11, 12, 14, and 15 of the status word.

The drive error can be removed by:

1. Setting bit 3 of the serial drive command (bit 3 of the disk address register during a get status command)
2. Removing the write lock condition via the drive's front panel followed by step 1.

3.3.2.9 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 on until: (1) a new drive command marker is received at the input to the drive command shift register, or (2) the drive is deselected.

3.3.2.10 Status (STATUS) – In response to a get status command, the drive enables STATUS CLK and sends the status word to the controller via the status line. This function can be performed even though DR RDY is not present (i.e., during spinup or a seek).

3.3.2.11 Sector Pulse (SEC PLS) – This 48 μ s pulse is asserted high and occurs every 625 μ s or 40 times per disk revolution. When a drive is initially selected, it must wait until the next full SEC PLS is detected before sending the SEC PLS to the controller. This pulse is used to indicate the beginning of a sector. The next preamble encountered marks the beginning of the header.

3.3.2.12 Read Data (RD DATA) – This line transfers MFM encoded data from the drive read circuits to the controller. Whenever a drive is selected and DR RDY is asserted, RD DATA appears on this line, except when WR GATE is asserted.

The drive senses the amplitude of the header preamble and sends RD DATA over the RD DATA line $2.5 \pm 0.5 \mu\text{s}$ downstream from where the preamble actually starts.

For reading headers, the VFO loop is phase locked with the arrival of RD DATA after the end of the SEC PLS. For the data preamble, the VFO continues to lock on RD DATA following the header. Detection of the preamble marker is enabled at the beginning of the third word of the data preamble.

3.3.3 Drive Bus Dialogue

In this section typical drive bus sequences are illustrated for those controller commands that issue drive bus signals. Seven of the eight controller commands communicate with the disk drive as shown in Figures 3-4 through 3-10. Only maintenance does not.

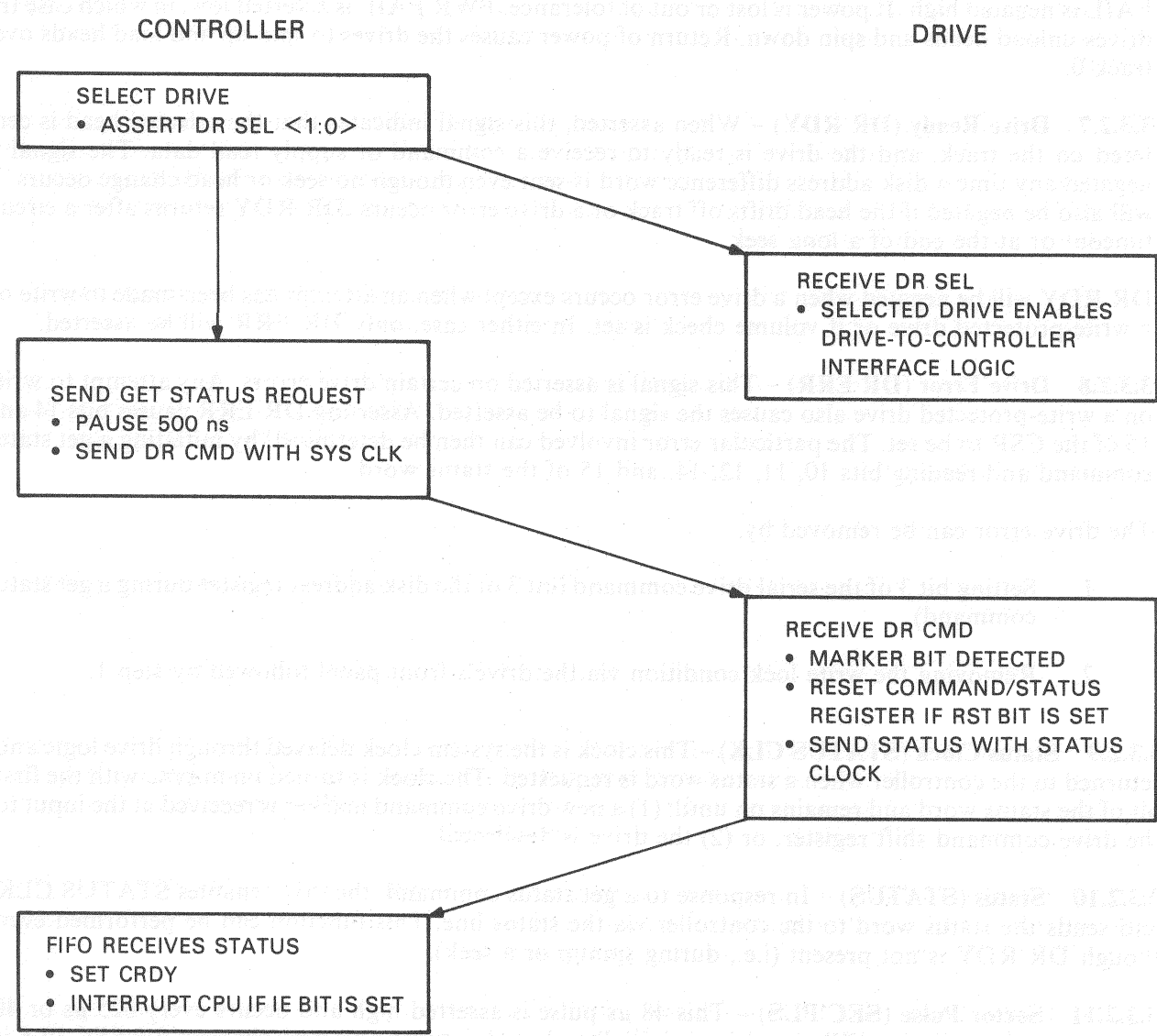
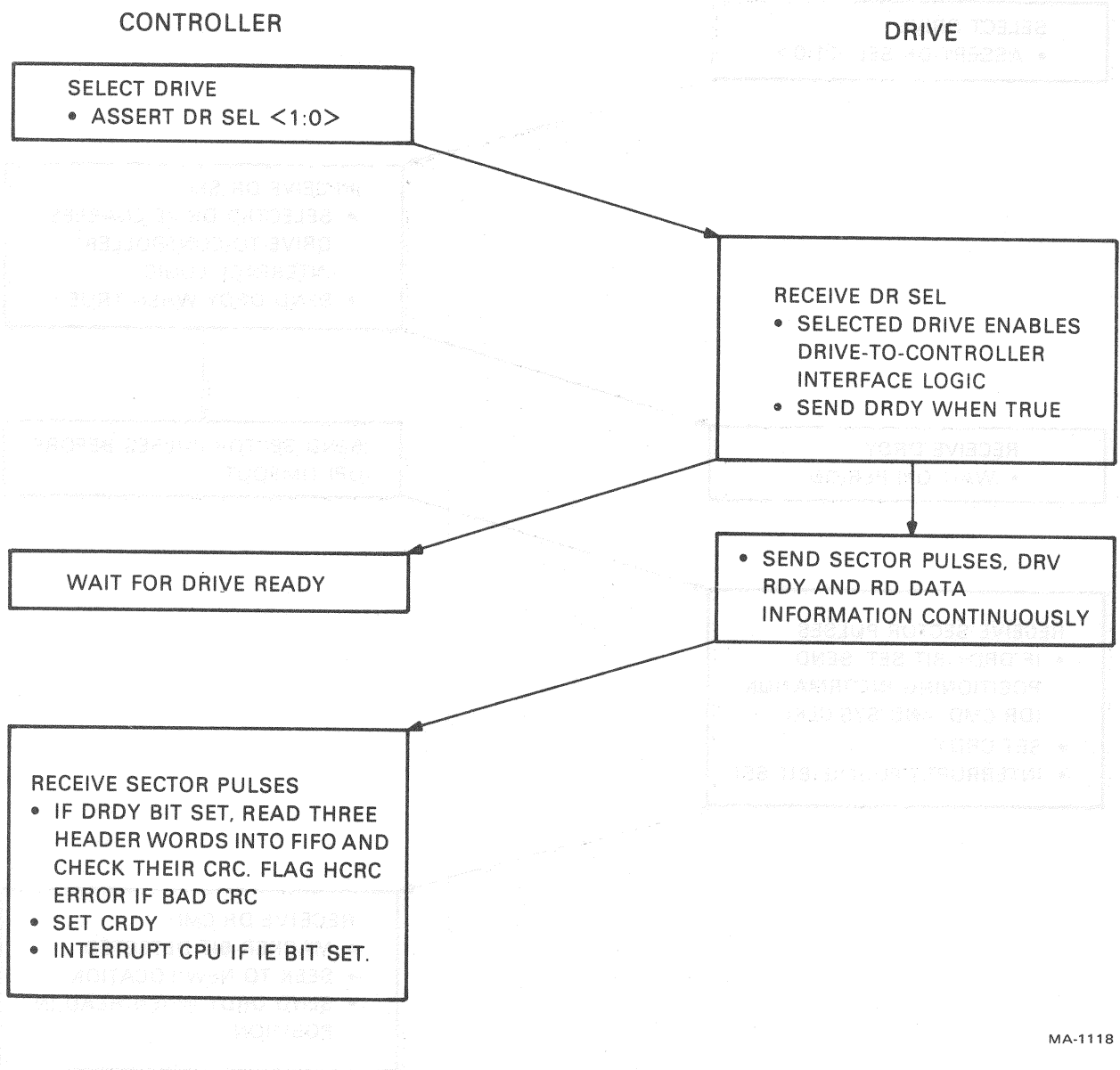


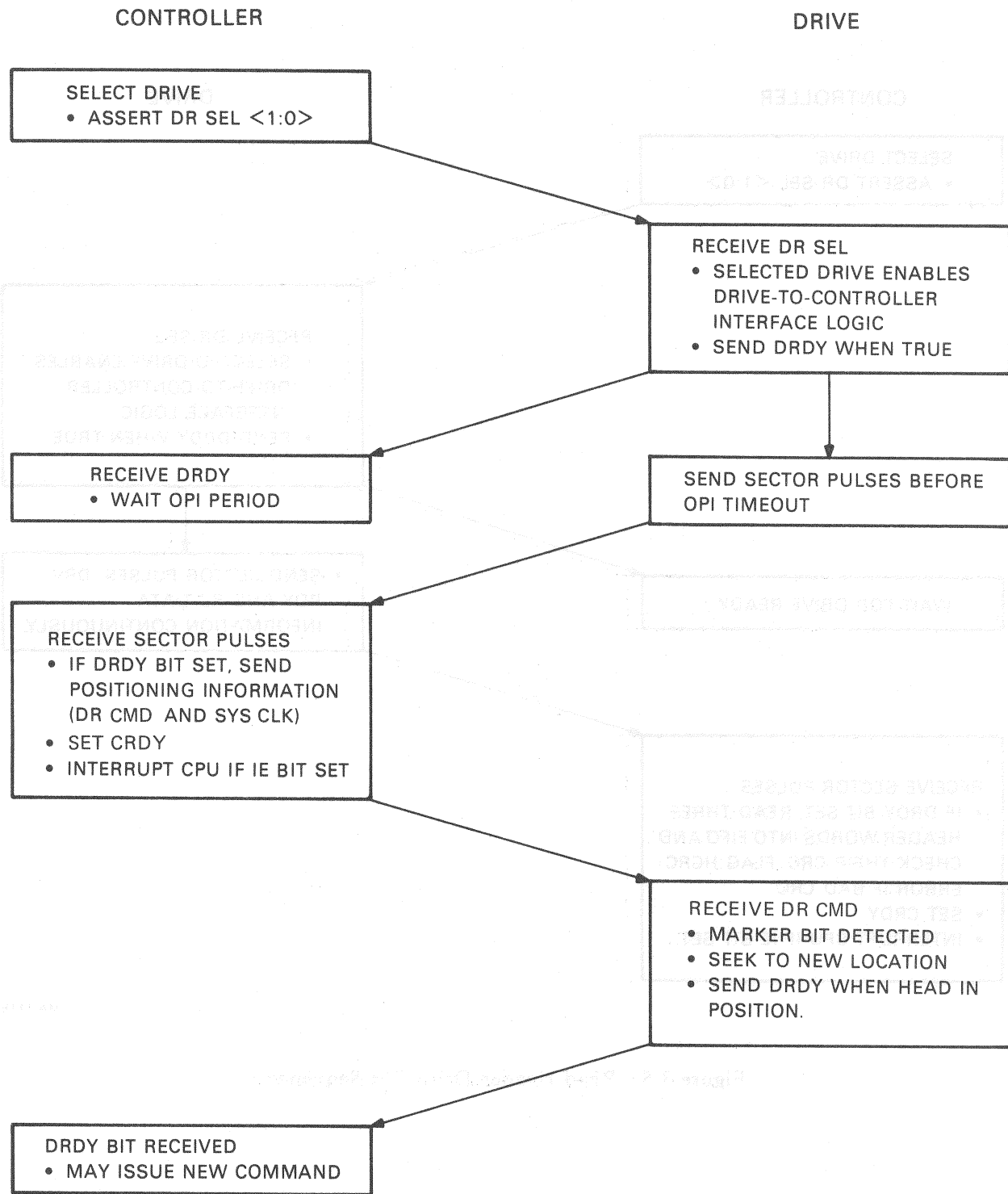
Figure 3-4 Get Status Drive Bus Sequence

MA-1117



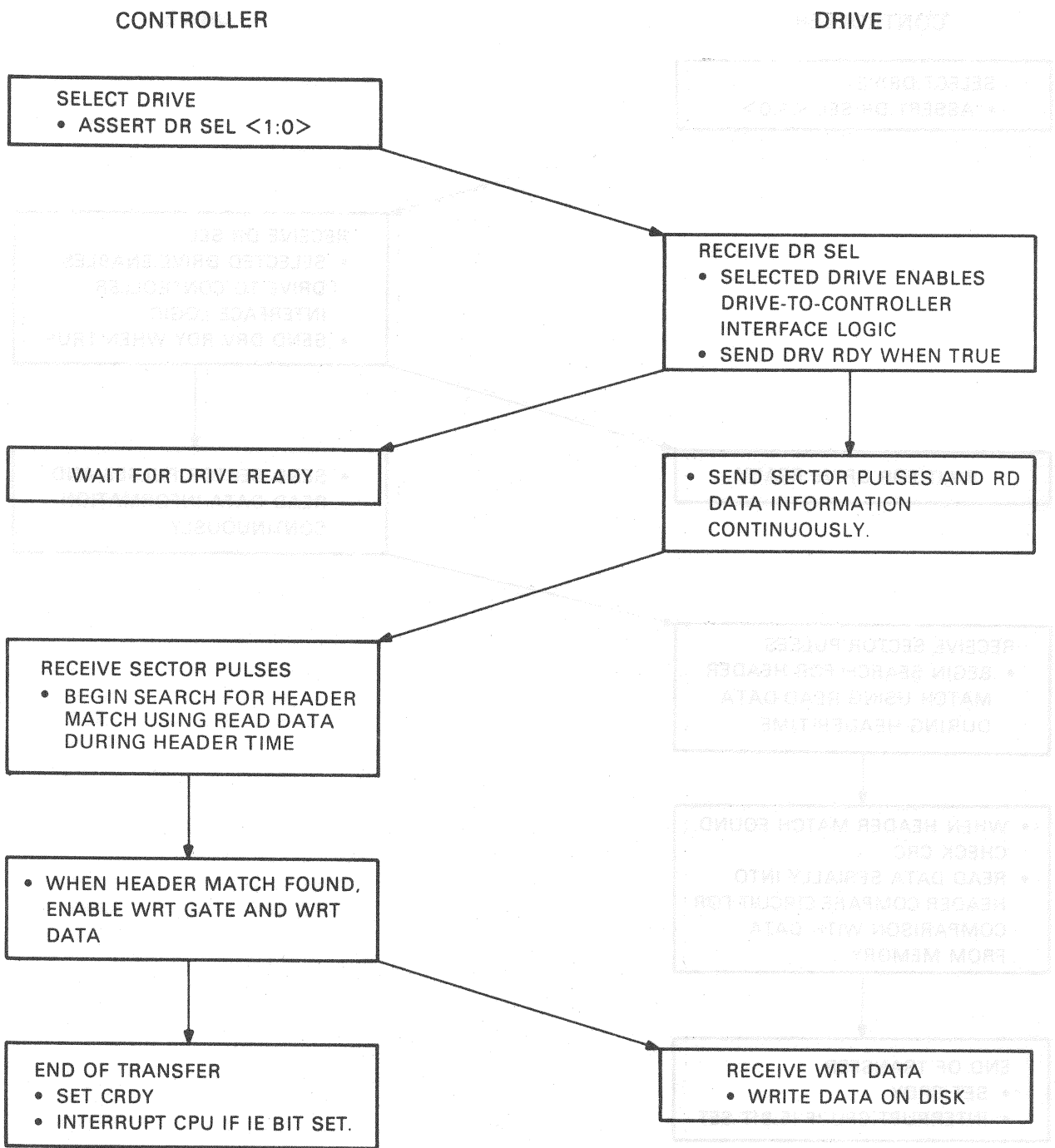
MA-1118

Figure 3-5 Read Header Drive Bus Sequence



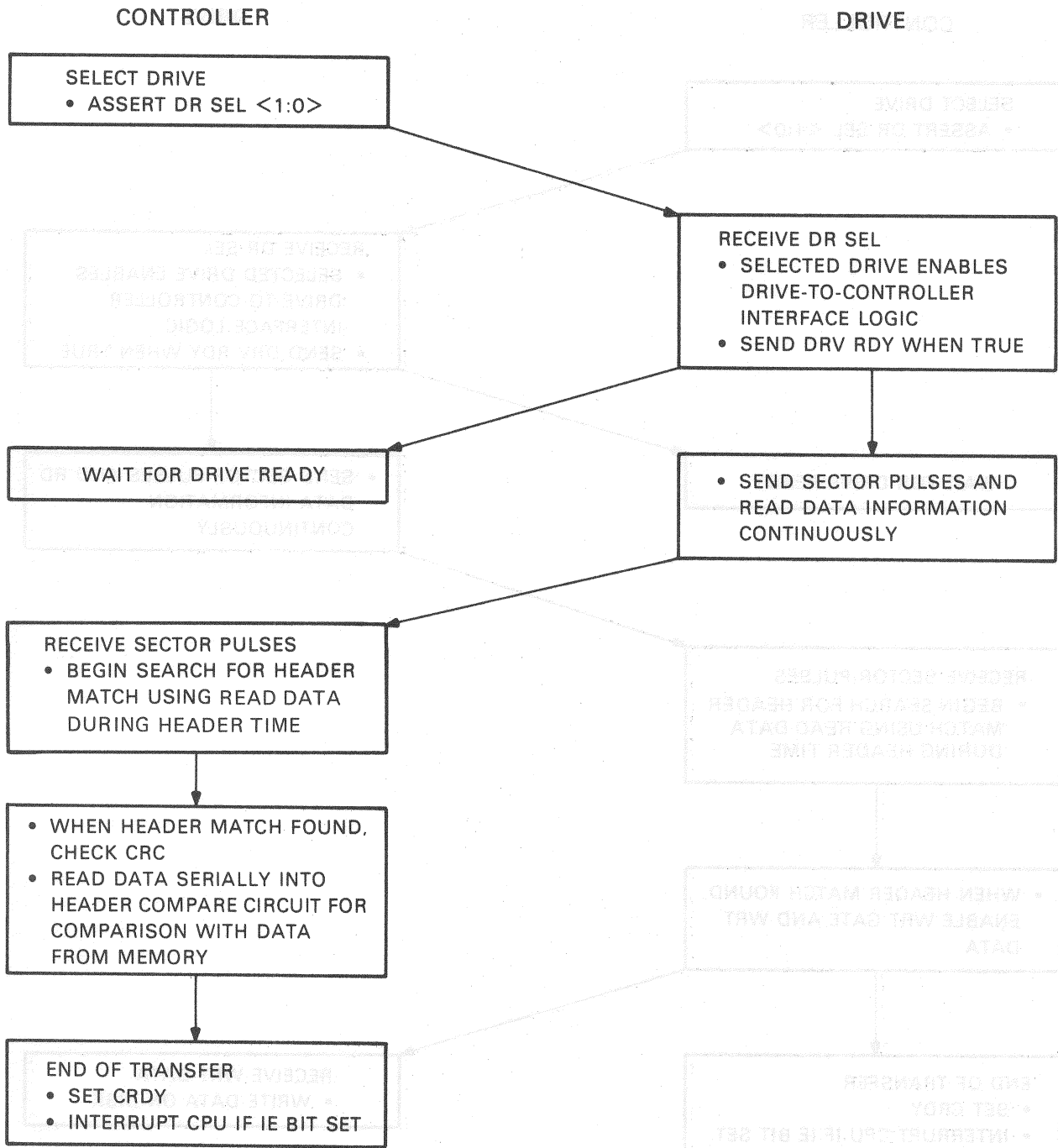
MA-1146

Figure 3-6 Seek Drive Bus Sequence



MA-1145

Figure 3-7 Write Data Drive Bus Sequence

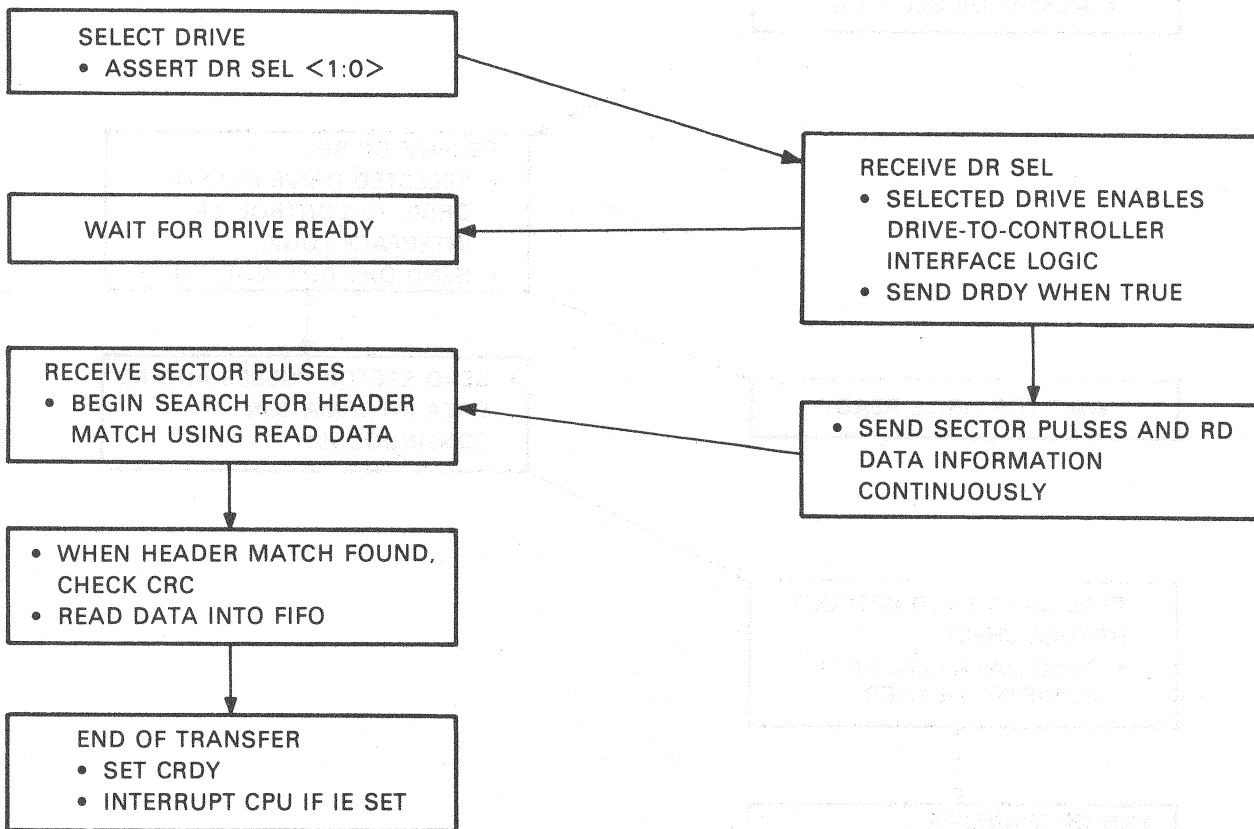


MA-1144

Figure 3-8. Write Check Drive Bus Sequence

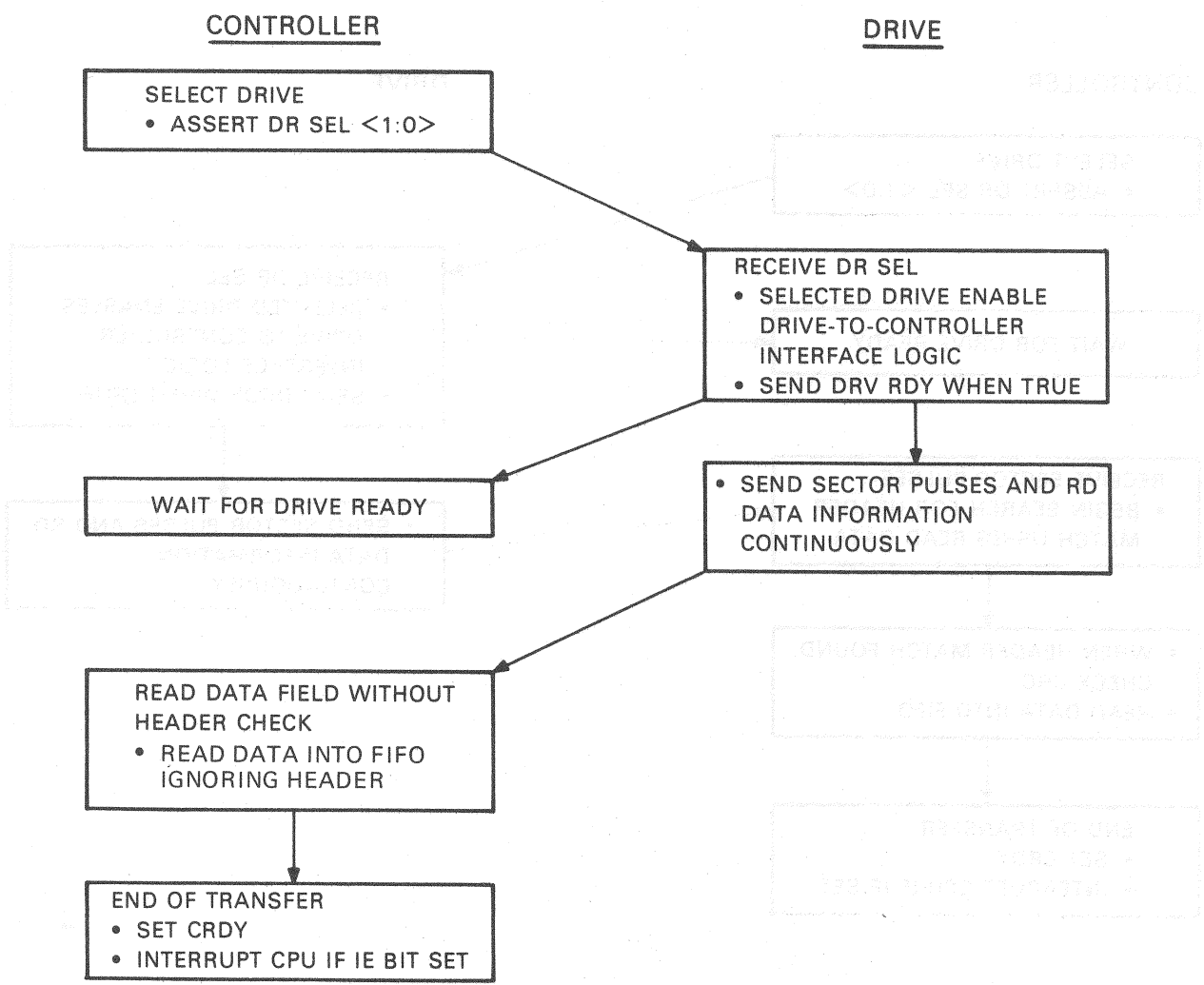
CONTROLLER

DRIVE



MA-1166

Figure 3-9 Read Data Drive Bus Sequence



MA-1113

Figure 3-10 Read Data Without Header Check Drive Bus Sequence

3.4 ADDRESSABLE REGISTERS

The RLV11 controller has four addressable registers that are used to store data and control information. These registers can be accessed like any location in memory except that they may not be read or written while the controller is busy. In the LSI-11 microcomputer systems, the upper 4K of address space, between 28K and 32K, is reserved for I/O device addresses. Within this range each RLV11 register has a unique address assigned as shown in Table 3-2.

Table 3-2 Register Addresses

LSI-11 Bus Address						Register Name
1	7	4	4	0	0	Control Status Register
1	7	4	4	0	2	Bus Address Register
1	7	4	4	0	4	Disk Address Register
1	7	4	4	0	6	Multipurpose Register

See Figure 3-19 at the end of this section for a register summary.

3.4.1 Control Status Register (CSR)

The control status register (Figure 3-11) is a 16-bit word-addressable register with a standard address of 174 400. Bits 1 through 9 can be read or written; the other bits can only be read.

When the LSI-11 bus is initialized (BINIT L), bits 1-6 and 8-13 are cleared, and bit 7 is set. Bit 0 is set whenever the selected drive is in the ready condition; otherwise the bit is cleared. Bit 14 is cleared as long as there is no drive error; otherwise the bit is set and remains set until the drive error is corrected or the drive error is cleared by a get status command. Bit 15 is set only when there is a drive or controller error (bits 10-14).

At the beginning of each controller command, error bits 10-13 are automatically cleared. At the completion of each controller command, bit 7 is automatically set. (Bit 7 is also set if an error is detected during command execution.)

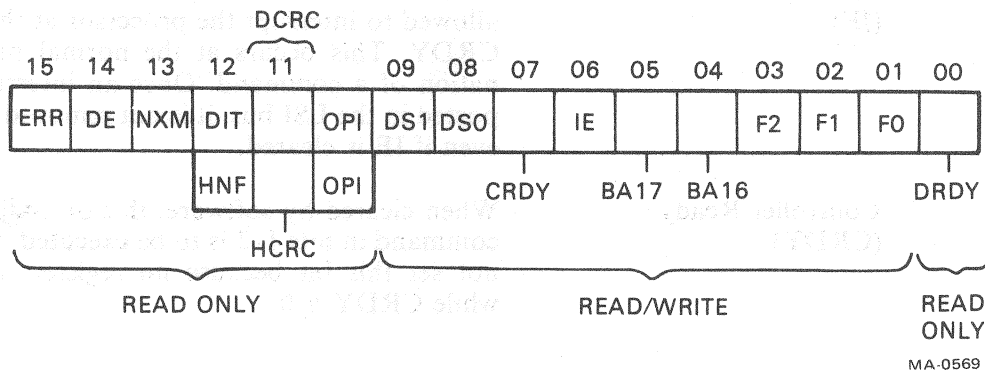


Figure 3-11 Control Status Register

Bit(s)	Name	Function
0	Drive Ready (DRDY)	When set, this bit indicates that the selected drive is ready to receive a command or supply valid read data. The bit is cleared when a seek operation is initiated and set when the seek operation is completed.
1-3	Function Code	These bits are set by software to indicate the command to be executed:

F2	F1	F0	Command	Octal Code
0	0	0	Maintenance Mode	0
0	0	1	Write Check	1
0	1	0	Get Status	2
0	1	1	Seek	3
1	0	0	Read Header	4
1	0	1	Write Data	5
1	1	0	Read Data	6
1	1	1	Read Data Without Header Check	7

Command execution starts when CRDY (bit 7) of the CSR is cleared by software. In a sense, then, bit 7 can be considered a negative GO bit.

4-5	Bus Address Extension Bits (BA16, BA17)	Two upper-order bus address bits. Read and written as bits 4 and 5 of the CSR, they function as address bits 16 and 17 of the BAR.
6	Interrupt Enable (IE)	When this bit is set by software, the controller is allowed to interrupt the processor at the assertion of CRDY. This occurs at the normal or error termination of a command. Once an interrupt request is posted in the LSI bus, it is not removed until serviced even if IE is cleared.
7	Controller Ready (CRDY)	When cleared by software, this bit indicates that the command in bits 1-3 is to be executed. Software cannot set this bit because no registers are accessible while CRDY is 0.

Bits	Name	Function
8-9	Drive Select (DS0, DS1)	These bits determine which drive will communicate with the controller via the drive bus.
10	Operation Incomplete (OPI)	When set, this bit indicates that the current command was not completed within the OPI timer period.
11	Data CRC (DCRC) or Header CRC (HCRC)	If OPI (bit 10) is cleared and bit 11 is set, the CRC error occurred on the data (DCRC). If OPI (bit 10) is set and bit 11 is also set, the CRC error occurred on the header (HCRC).

NOTE

Cyclic redundancy checking is done only on the desired header. It is performed on the first and second header words, even though the second header word is always 0.

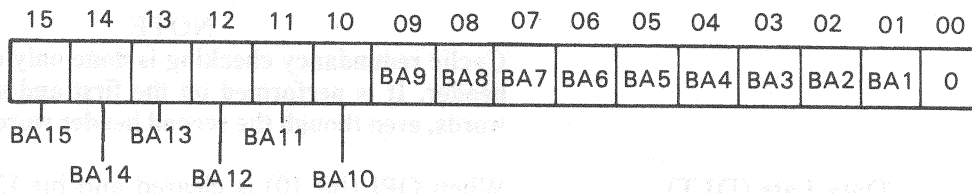
12	Data Late (DLT) or Header Not Found (HNF Error)	<p>When OPI (bit 10) is cleared and bit 12 is set, it indicates that a data late condition occurred on a read without header check operation. The FIFO was more than half full and the controller was unable to transfer the next sequential sector.</p> <p>When OPI (bit 10) is set and bit 12 is also set, it indicates that a timeout occurred while the controller was searching for the correct sector to read or write (no header compare).</p>
13	Non-Existent Memory (NXM)	When set, this bit indicates that during a DMA data transfer, the memory location addressed did not respond within 14 ms.
14	Drive Error (DE)	<p>This bit is buffered from the drive error interface line. When set, it indicates that the selected drive has flagged an error, the source of which can be determined by executing a get status command.</p> <p>One way to clear the drive error is to reset the drive error register (e.g., by setting bit 3 of the get status command word).</p>
15	Composite Error (ERR)	When set, this bit indicates that one or more of the error bits (bits 10-14) is set. When an error occurs, the current operation terminates and an interrupt routine is initiated if the interrupt enable bit (bit 6 of the CSR) is set.

3.4.2 Bus Address Register (BAR)

The bus address register (Figure 3-12) is a 16-bit word-addressable register with an address of 174 402. Bits 0 through 15 can be read or written; bit 0 should normally be written 0. Expansion bits 16 and 17 are programmable via bits 4 and 5 of the CSR.

The bus address register indicates the memory location involved in the DMA data transfer during a read or write operation. The contents of the BAR are automatically incremented by 2 as each word is transferred between system memory and controller.

Clearing of the BAR is accomplished by executing a BUS INIT.



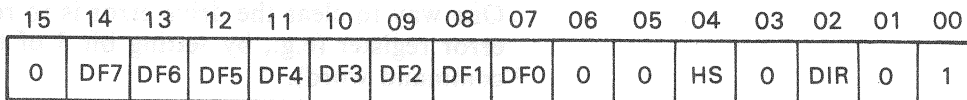
MA-0570

Figure 3-12 Bus Address Register

3.4.3 Disk Address Register (DAR)

The disk address register is a 16-bit read/write word-addressable register with an address of 174 404. Its contents can have one of three meanings, depending on the function being performed. Clearing of this register is accomplished by executing a BUS INIT.

3.4.3.1 DAR During a Seek Command – To perform a seek function, it is necessary to provide address difference, head select, and head directional information to the selected drive as indicated in Figure 3-13.

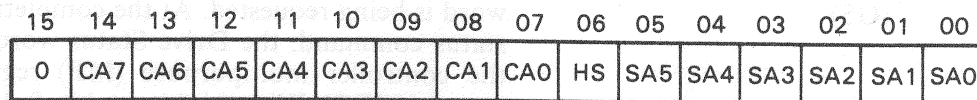


MA-0571

Figure 3-13 DAR Seek Command

Bits	Name	Function
0	Marker (MRKR)	Must be a 1.
1		Must be a 0, indicating to the drive that a seek command is being requested and that the remaining bits in the register will contain the seek specifications.
2	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 7-14).
3		Must be a 0.
4	Head Select (HS)	Indicates which head (disk surface) is to be selected. Set = lower, clear = upper.
5-6		Reserved
7-14	Cylinder Address Difference (DF<07:00>)	Indicates the number of cylinders the heads are to move on a seek.
15		Must be a 0.

3.4.3.2 DAR During Read or Write Data Command – For a read, write, or write check operation, the DAR is loaded with the address of the first sector to be transferred. Thereafter, as each adjoining sector is transferred, the DAR is automatically incremented by 1 (Figure 3-14). If the DAR increments to the non-existent sector address 50_h, an OPI timeout will occur. The drive must then seek to a new track if the transfer is to continue.



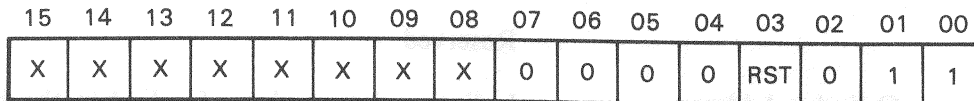
MA-0572

Figure 3-14 DAR Read/Write Data Command

Bit(s)	Name	Function
0-5	Sector Address (SA<5:0>)	Address of one of the 40 sectors on a track. (Octal range is 0 to 47.)
6	Head Select (HS)	Indicates which head (disk surface) is to be selected. Set = lower; clear = upper.
7-14	Cylinder Address (CA<7:0>)	Address of one of the 256 cylinders. (Octal range is 0 to 377.)
15		Must be a 0.

3.4.3.3 DAR During a Get Status Command – After the get status command is deposited in the CSR, it is the DAR's responsibility to get the command transferred to the drive. Therefore, the DAR must also be programmed along with the CSR to do the get status command.

For a get status command, the DAR register bits must be programmed as follows (Figure 3-15).



MA-0573

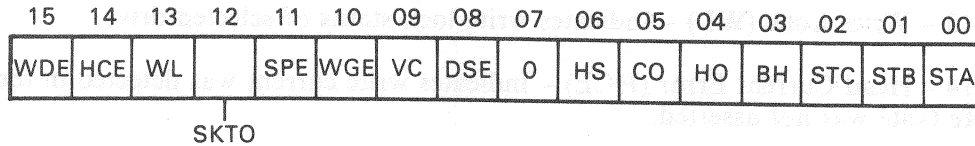
Figure 3-15 DAR Get Status Command

Bit(s)	Name	Function
0	Marker (MRKR)	Must be a 1.
1	Get Status (GS)	Must be a 1, indicating to the drive that its status word is being requested. At the completion of the get status command, the Drive Status word is read into the controller multipurpose (MP) register (output stage of FIFO). With this bit set, bits 8-15 are ignored by the drive.
2		Must be a 0.
3	Reset (RST)	When this bit is set, the drive clears its error register of soft errors before sending a status word to the controller.
4-7		Must be a 0.
8-15		Not used.

3.4.4 Multipurpose Register (MPR)

The MPR is two registers bearing the same base address. When writing into that location, the word counter accepts the data. When reading from that location, the FIFO output buffer provides the data.

3.4.4.1 MPR During a Get Status Command – When a get status command (Figure 3-16) is executed and a status word is returned to the controller, the contents of the MPR (FIFO output stage) are defined as follows.



MA-0574

Figure 3-16 MPR Status Word

- Bits 0–2 – State (C:A) (ST <C:A>) – These bits define the state of the drive.

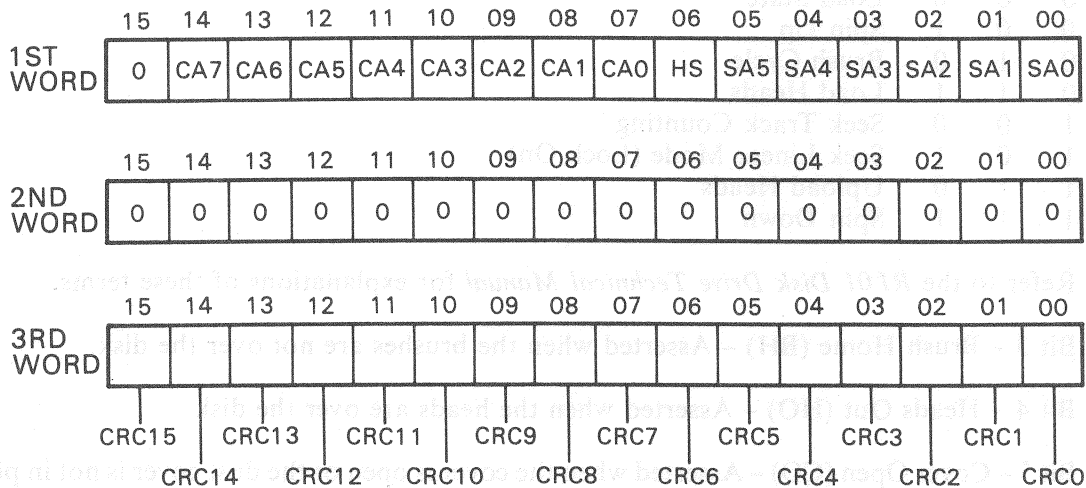
C	B	A	
0	0	0	Load State
0	0	1	Spin Up
0	1	0	Brush Cycle
0	1	1	Load Heads
1	0	0	Seek Track Counting
1	0	1	Seek Linear Mode (Lock On)
1	1	0	Upload Heads
1	1	1	Spin Down

Refer to the *RL01 Disk Drive Technical Manual* for explanations of these terms.

- Bit 3 – Brush Home (BH) – Asserted when the brushes are not over the disk.
- Bit 4 – Heads Out (HO) – Asserted when the heads are over the disk.
- Bit 5 – Cover Open (CO) – Asserted when the cover is open or the dust cover is not in place.
- Bit 6 – Head Select (HS) – Indicates the currently selected head.
- Bit 7 – Reserved – Will be 0.
- Bit 8 – Drive Select Error (DSE) – Indicates multiple drive selection is detected.
- Bit 9 – Volume Check (VC) – VC is set every time the drive goes into load heads state. This asserts a drive error at the controller but not on the front panel. VC is an indication that program does not really know which disk is present until it has read the serial number and bad sector file. (The disk might have been changed while the heads were unloaded.)

- Bit 10 – Write Gate Error (WGE) – Indicates the drive sensed that write gate was asserted when sector pulse was asserted, or write gate was set with the drive not ready, or the drive was write locked.
- Bit 11 – Spin Error (SPE) – Indicates the spindle not reaching speed in the required time, or over speeding.
- Bit 12 – Seek Time Out (SKTO) – Indicates the heads did not come on track in the required time during a seek command.
- Bit 13 – Write Lock (WL) – Indicates write lock status of selected drive.
- Bit 14 – Head Current Error (HCE) – Indicates write current was detected in the heads when Write Gate was not asserted.
- Bit 15 – Write Data Error (WDE) – Indicates write gate was asserted but no transitions were detected on the write data line.

3.4.4.2 MPR During a Read Header Command – When a read header command is executed, three words will be stored in the multipurpose register (FIFO output buffer). The first header word will contain sector address, head select, and cylinder address information. The second word will contain all 0s. The third word will contain the header CRC information. All three words are readable by the main program (Figure 3-17).



MA-0575

Figure 3-17 MPR Three Header Words

3.4.4.3 MPR During Read/Write Data Commands – When transferring data via DMA, the MPR functions as a word counter and is loaded by program with the 2's complement of the number of words to be transferred. It is then incremented by 1 by the controller as each word is transferred. The reading or writing operation generally is terminated when the word counter overflows. The word counter can keep track of from one data word to the full 40-sector count of 5120 data words (decimal). The maximum number of words that can be transferred in a single operation is limited by the number of sectors available to be written in the track (Figure 3-18).

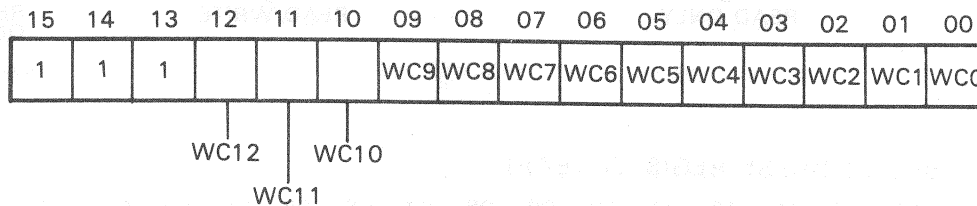


Figure 3-18 MPR Used As Word Counter

Bit(s)	Name	Function
0-12	Word Count (WC<12:0>)	2's complement of total number of words to be transferred.
13-15		Must be a 1 for word count in correct range.

Figure 3-19 is a summary of the registers discussed in this paragraph.

3.5 I/O TRANSFER OPERATIONS

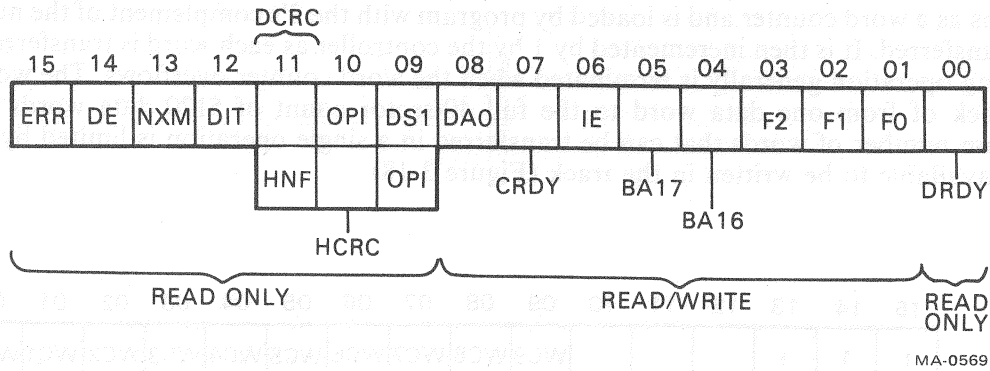
There are three kinds of I/O transfers that are used to interface the processor with the RLV11 controller. They are programmed I/O transfers, DMA transfers, or interrupt-driven transfers.

Programmed I/O transfers are executed by single- or double-operand PDP-11 instructions. By including the device's address as the effective source or destination address, the user specifies the transfer as an input or output operation. Programmed I/O allows information to be transferred between the RLV11 addressable registers and LSI-11 memory locations and CPU registers. The transfer of each word requires the execution of a PDP-11 instruction.

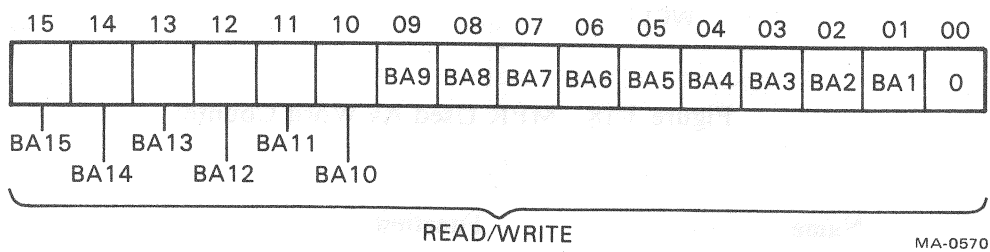
DMA transfers, on the other hand, require only a few programmed I/O transfers to set control information. Then a large block of data can be moved to or from memory without any support from the processor. DMA transfers are the fastest method of transferring data between memory and a device. They can occur between processor bus cycles and do not alter processor status in any way. Blocks of data can be moved at speeds that are not limited by CPU instruction execution via the DMA transfer mode. The read and write data in the controller FIFO is received and transmitted under DMA control.

Interrupt-driven transfers allow the processor to continue a programmed operation without waiting for the controller to become ready. When the controller becomes 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.

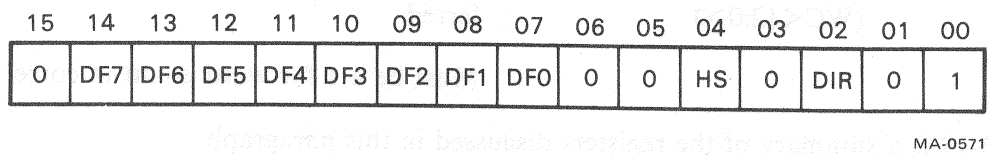
CONTROL STATUS REGISTER (CSR)



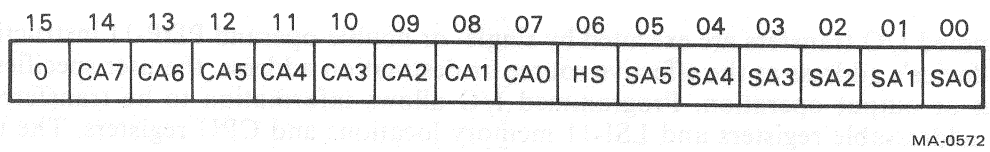
BUS ADDRESS REGISTER (BAR)



DAR DURING SEEK COMMAND



DAR DURING READING OR WRITING DATA COMMANDS



DAR DURING GET STATUS COMMAND

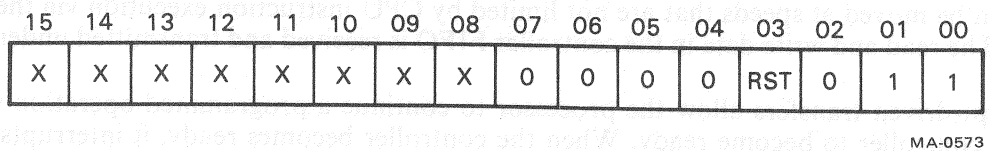
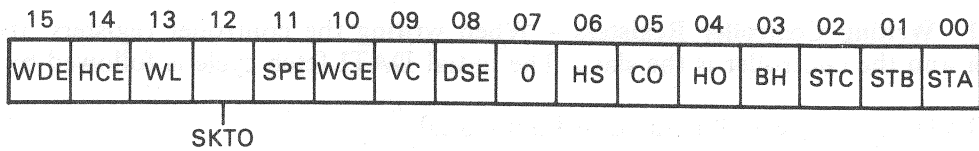


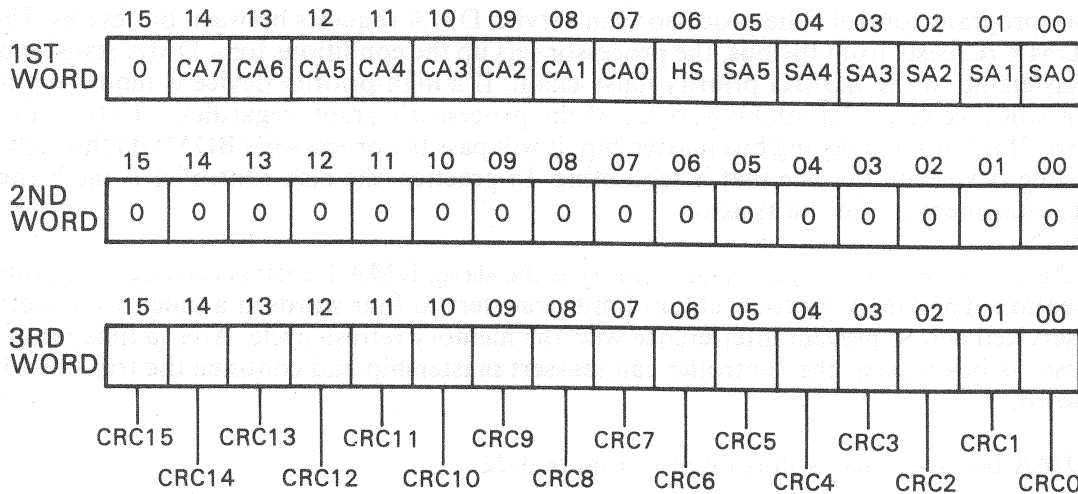
Figure 3-19 Register Summary (Sheet 1 of 2)

MPR DURING GET STATUS COMMAND



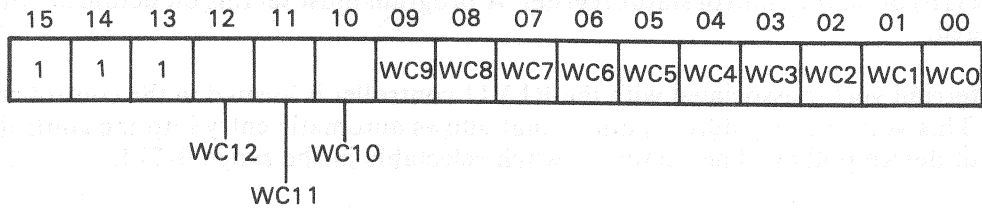
MA-0574

MPR DURING READ HEADER COMMAND



MA-0575

MPR DURING READ/WRITE COMMANDS FOR WORD COUNT



MA-0576

Figure 3-19 Register Summary (Sheet 2 of 2)

3.5.1 Programmed I/O Transfers

Every processor instruction requires one or more I/O operations. The first operation required is a data input transfer (DATI), which fetches an instruction from memory at the location addressed by the program counter. This operation is called a DATI bus cycle. If the controller is referenced, additional DATI, or data output transfer (DATO) bus cycles are required.

3.5.1.1 Writing Controller Registers – When writing the controller registers, the CPU is the bus master and the controller is the slave. The initial DATI fetch cycle is followed by a DATO cycle.

The DATO bus cycle is illustrated in Figure 3-20.

3.5.1.2 Reading Controller Registers – When reading the controller registers, the CPU is bus master and the controller is the slave. The CPU performs a DATI cycle (Figure 3-21) to obtain the data from the RLV11 registers. The DATI cycle is a result of a CPU programmed instruction which addresses the controller registers.

3.5.2 DMA I/O Transfers

Direct memory access (DMA) is used to transfer data between the controller FIFO and memory without program control. The processor can service DMA requests between bus cycles. Upon receiving BDMR requests from the bus, the processor sets up the conditions for a DMA transfer by granting bus mastership to the BDMG priority daisy-chain. If a high-priority device is requesting bus mastership, it will receive it and inhibit passage of the processor's grant, regardless of other lower priority requests. If it is not requesting bus mastership, it will pass the processor's BDMGO through other non-requesting devices to the one that is requesting. In practice, the disk controller is the highest priority device, after memory, in the system.

Once the controller is bus master and memory is the slave, DMA transfers can occur without processor intervention. The DMA protocol circuit limits transfers to four words at a time to allow other devices to be serviced and to prevent interference with the memory refresh cycle. After a timeout of 4 μ s, if the processor is bus master, the controller can reassert mastership and continue the transfer with another four words.

The DMA bus sequence is illustrated in Figure 3-22.

3.5.3 Interrupt-Driven I/O Transfers

Interrupts are requests made by the controller that cause the processor to temporarily suspend its present program sequence to execute the controller service routine. The controller can interrupt the processor only when its interrupt control circuit is enabled. This circuit is enabled by an interrupt enable (IE) bit in the control status register. A program must set this bit before an interrupt request can be issued.

An interrupt vector associated with the RLV11 controller is located in the controller interface/control logic. This vector is an address pointer that allows automatic entry into the controller service routine without device polling. The vector is switch-selectable in the range 0-774.

The interrupt request sequence is illustrated in Figure 3-23. The controller requests interrupt service by asserting BIRQ L. The processor acknowledges the interrupt request by asserting BDIN L followed by BIAKO L. The first device on the bus receives this daisy-chained signal at its BIAKI L input. If it is not requesting service, it passes the signal via its BIAKO L output to the next device, and so on, until the requesting device receives the signal. The requesting device responds by asserting BRPLY L and placing its interrupt vector on the data/address bus lines BDAL <0-15> L. Automatic entry to the service routine is then executed by the processor.

3.6 BUS SIGNAL TIMING

Diagrams illustrating the bus timing requirements between the LSI-11 CPU and the RLV-11 controller, given in general master/slave device terms, may be found in the *Microcomputer Handbook* published by Digital Equipment Corporation.

BUS MASTER (PROCESSOR)

SLAVE (CONTROLLER)

ADDRESS RLV11

- ASSERT BDALO-15 L WITH ADDRESS 774400 AND
- ASSERT BBS7 L (IF ADDRESS IS IN THE 28 -32K RANGE)
- ASSERT BWTBT L (WRITE CYCLE)
- ASSERT B SYNC L

DECODE ADDRESS

- DECODE BASE ADDRESS
- STORE DAL 1,2 FOR RLV11 REGISTER SELECTION

OUTPUT DATA

- REMOVE THE ADDRESS FROM BDALO-15 L AND NEGATE BBS7 L AND BWTBT L
- PLACE DATA ON BDALO-15 L
- ASSERT BDOUT L

TAKE DATA

- RECEIVE DATA FROM BDAL LINES
- ASSERT BRPLY L

TERMINATE OUTPUT TRANSFER

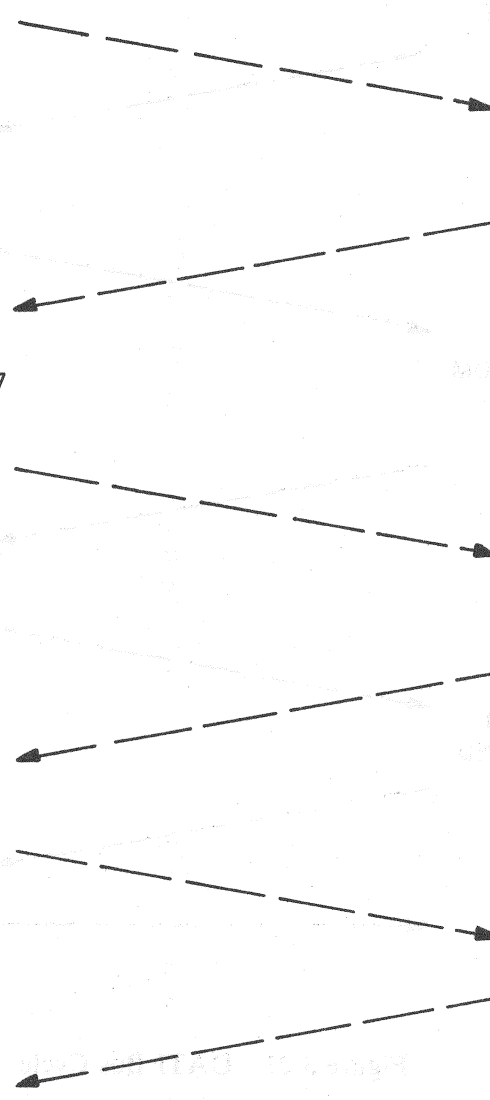
- REMOVE DATA FROM BDALO-15 L AND NEGATE BDOUT L

OPERATION COMPLETED

- TERMINATE BRPLY L

TERMINATE BUS CYCLE

- NEGATE BSYNC L



11-3140

Figure 3-20 DATO Bus Cycle

SLAVE (CONTROLLER)

BUS MASTER (PROCESSOR)

BUS MASTER (PROCESSOR)

SLAVE (CONTROLLER)

ADDRESS RLV11

- ASSERT BDALO-15 L WITH ADDRESS AND
- ASSERT BBS7 IF THE ADDRESS IS IN THE 28 - 32K RANGE
- ASSERT BSYNC L

- ASSERT BBS7 IF ADDRESS IN THE 28-32K RANGE
- ASSERT BWRITE
- ASSERT BTYPE



DECODE ADDRESS

- DECODE BASE ADDRESS
- STORE DAL 1,2 FOR REGISTER SELECTION

REQUEST DATA

- REMOVE THE ADDRESS FROM BDALO-15 L AND NEGATE BBS7 L
- ASSERT BDIN L

- REMOVE ADDRESS FROM BDALO-15 L AND NEGATE BBS7
- ADD DATA
- PLACE DATA ON BDALO-15 L
- ASSERT BWRITE



INPUT DATA

- PLACE DATA ON BDALO-15 L
- ASSERT BRPLY L

TERMINATE INPUT TRANSFER

- ACCEPT DATA AND RESPOND BY NEGATING BDIN L

- TERMINATE OUTPUT TRANSFER
- REMOVE DATA FROM BDALO-15 L AND NEGATE BWRITE



OPERATION COMPLETED

- TERMINATE BRPLY L

TERMINATE BUS CYCLE

- NEGATE BSYNC L



11-3138

Figure 3-21 DATI Bus Cycle

**LSI-11 PROCESSOR
(MEMORY IS SLAVE)**

**BUS MASTER
(CONTROLLER)**

GRANT BUS CONTROL

- NEAR THE END OF THE CURRENT BUS CYCLE (BRPLY L IS NEGATED), ASSERT BDMGO L AND INHIBIT NEW PROCESSOR GENERATED BYSYNC L FOR THE DURATION OF THE DMA OPERATION.

TERMINATE GRANT SEQUENCE

- NEGATE BDMGO L AND WAIT FOR DMA OPERATION TO BE COMPLETED

RESUME PROCESSOR OPERATION

- ENABLE PROCESSOR-GENERATED BSYNC L (PROCESSOR IS BUS MASTER) OR ISSUE ANOTHER GRANT IF BDMR L IS ASSERTED.

REQUEST BUS

- ASSERT BDMR L

ACKNOWLEDGE BUS MASTERSHIP

- RECEIVE BDMG
- WAIT FOR NEGATION OF BSYNC L AND BRPLY L
- ASSERT BSACK L
- NEGATE BDMR L

EXECUTE A DMA DATA TRANSFER (RLV11 IS BUS MASTER)

- ADDRESS MEMORY AND TRANSFER UP TO 4 WORDS OF DATA AS DESCRIBED FOR DATI, OR DATO BUS CYCLES
- RELEASE THE BUS BY TERMINATING BSACK L (NO SOONER THAN NEGATION OF LAST BRPLY L) AND BSYNC L.
- WAIT 4 μ s OR UNTIL ANOTHER FIFO TRANSFER IS PENDING BEFORE REQUESTING BUS AGAIN.

11-3141

Figure 3-22 DMA Request/Grant Sequence

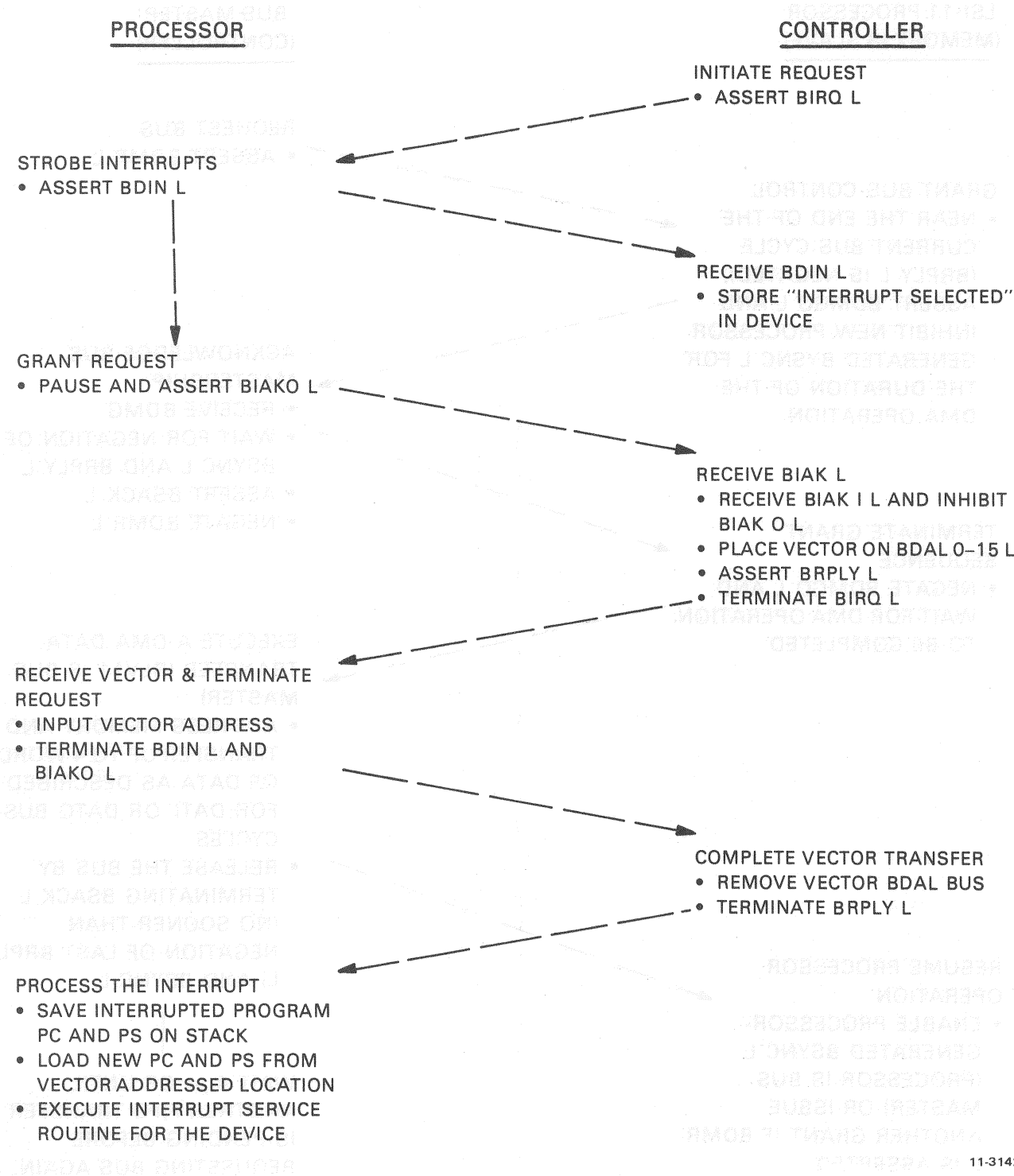


Figure 3-23 Interrupt Request/Acknowledge Sequence

CHAPTER 4 FUNCTIONAL LEVEL DESCRIPTION

4.1 GENERAL

This chapter introduces simplified controller block diagrams that illustrate the data paths followed by each command operation. A more detailed circuit description is provided in Chapter 5.

4.2 CONTROLLER SIMPLIFIED BLOCK DIAGRAM

The controller simplified block diagram is illustrated in Figure 4-1. The diagram shows all the major functional blocks that are used by the various command operations. These functional blocks are shown located on their appropriate modules.

4.2.1 Bus Interface Module (M8014)

Board M8014 contains all the LSI-11 bus related circuitry. Items such as the bus control circuit, bus transceivers and decoders, programmable registers, and FIFO circuitry are located here.

4.2.1.1 Bus Control Functions – The bus control block consists of five separate functional units.

- Register Protocol Circuit – This circuit selects which controller register to be read or written and supplies the required control signals for loading and reading.
- Interrupt Control Circuit – This circuit sends out a bus interrupt request to the CPU when the controller has completed a command operation and the interrupt enable bit is set. It also passes or blocks the CPU interrupt acknowledge along the priority daisy-chain and produces control signals for the generation of RPLY and vector data.
- OPI Circuit – This circuit is the operation incomplete (OPI) timer. This timer is initiated upon issuing a controller command. If the command sequence is not completed within the 490 ms, nominal, OPI timeout period, an OPI error bit is set in the control status register. The controller ready bit is also set and the CPU receives an interrupt request if enabled.
- DMA Control Circuit – The direct memory access (DMA) circuit coordinates the timing of the controller FIFO during DMA data exchanges with memory.
- NXM Circuit – The non-existent memory (NXM) circuit is a timer used when the controller is attempting to read or write from memory. It is initialized by the Bus SYNC signal and gives the memory device 10 μ s to reply to a controller Data In (DIN) or Data Out (DOUT) signal. If the reply (BRPLY) is not received within 10 μ s, the NXM error bit is set in the CSR. NXM timeout can also occur because of a failure at the controller or drive.

4.2.1.2 Bus Transceivers – These circuits transmit and receive both data and address information on the bus. The address decoder circuit compares each incoming address with the controller's preset base address. When a match is found, the register protocol circuit is enabled.

4.2.1.3 Programmable Registers

Control Status Register (CSR) – The CSR is a holding register for command control information such as drive select, function to be performed, interrupt enable, and extended address bits. It also indicates drive ready and error conditions.

Bus Address Register (BAR) – The BAR contains the 16-bit memory address to which the next DMA transfer is to be made. It is incremented by two under control of the DMA control circuit at the end of each DMA transfer.

Disk Address Register (DAR) – The DAR contains the next sector address where data is to be read or written on the disk. It is incremented by one at the end of each sector read or written. The DAR is also used to store drive command information that is sent to the drive during a seek or get status operation.

Multipurpose Register (MPR) – The MPR is not a single physical entity like the other registers. It consists of two separate registers, the word counter and the FIFO output buffer, both bearing the same base address.

When writing the MPR, the data word is loaded into the word counter (WC) register. The WC register contains the number of data words remaining to be transferred under DMA control. The WC register is incremented by 1 under control of the DMA control circuit at the end of each DMA transfer.

When reading the MPR, the data word is read from the FIFO output buffer. After a read header command, it contains the header words. After a get status, it contains disk drive status.

4.2.1.4 FIFO – The FIFO is a first-in/first-out silo-type memory element that can store up to 256 data words. When full, it holds two sectors of data. A FIFO serializer circuit converts the FIFO parallel data into the serial form needed for writing to the disk. Similarly, the serial data read from the disk is converted to parallel form through the same serializer circuit.

The FIFO contents can be recovered by reading the current data word in the MPR. For example, to recover three FIFO words requires three successive readings of the MPR. During disk read and write operations, the FIFO is emptied and filled under control of the DMA logic.

4.2.2 Drive Module (M8013)

Board M8013 contains all the controller timing and sequencing logic plus the data formatting circuits needed to read and write the disk.

4.2.2.1 Microsequencer Logic – The microsequencer first decodes the function command by using three function bits to point to an address in its sequencer ROM. There it finds a routine that corresponds to the command issued. It then proceeds to generate the timing and control signals needed to channel the incoming or outgoing data through all its various paths within the controller.

4.2.2.2 Write Precompensation – This circuit performs two major functions. It encodes digital data into its MFM form, and it precompensates this data for peak shifting effects.

MFM 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 1. To represent two successive logical 0s, 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 track 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 the opposite direction as the expected peak shift before they are written.

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

4.2.2.4 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 through the CRC circuit. Any errors introduced into the data or its CRC word are detected and a CRC error bit is set in the CSR.

4.2.2.5 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 FIFO output data, data separator (DS) data, and the write marker pulse.

4.2.2.6 Header Compare Circuit – The function of the header compare circuit is to compare the first header word 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 is not identical, a mismatch signal is generated. At the end of a compare, the result is available to the microsequencer.

4.3 MAINTENANCE COMMAND FUNCTIONAL FLOW DIAGRAM

The maintenance 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 follows the maintenance command functional flow diagrams (Figures 4-2 and 4-3). Prior to issuing the maintenance command, a buffer area in memory must be set aside to read and write the contents. The controller registers must be loaded by program with the following contents.

- BAR with address of first memory buffer location
- WC register with 2's complement of 511₁₀
- DAR with test word
- CSR with F<0>, CRDY, IE

Upon issuing the maintenance command and clearing the CRDY bit, the OPI timer is started. The microsequencer decodes the command and starts a maintenance routine. Two internal tests are performed and the DAR is incremented after each. Then, by enabling a DMA transfer to take place between memory and the controller FIFO, 256 words are transferred from the memory write test buffer into the FIFO. Once the FIFO is full, 255 words are transferred into the memory read test buffer area previously prepared. The DAR is now incremented a third time. Throughout MAINT, checks are made and if an error occurs, the function stops with ERR set. The DAR is incremented as the test proceeds. This incrementing serves as a trace to determine the failing internal test.

Next, the test word +3 that was initially loaded into the DAR is channeled through the data source selector and into the CRC circuit. A CRC word is generated from this test word and sent through the data source selector again. This CRC of the test word then passes through the write precompensation circuit and the data separator circuit to eventually end up in the FIFO.

The contents of the DAR is then incremented by one and becomes test word +4. This new test word follows the same path as the preceding test word and ends up as the second word in the FIFO. At this point, the FIFO holds the following contents.

Word	FIFO
1st	CRC of test word +3
2nd	CRC of test word +4

The contents of the DAR is now incremented once again and becomes test word +5.

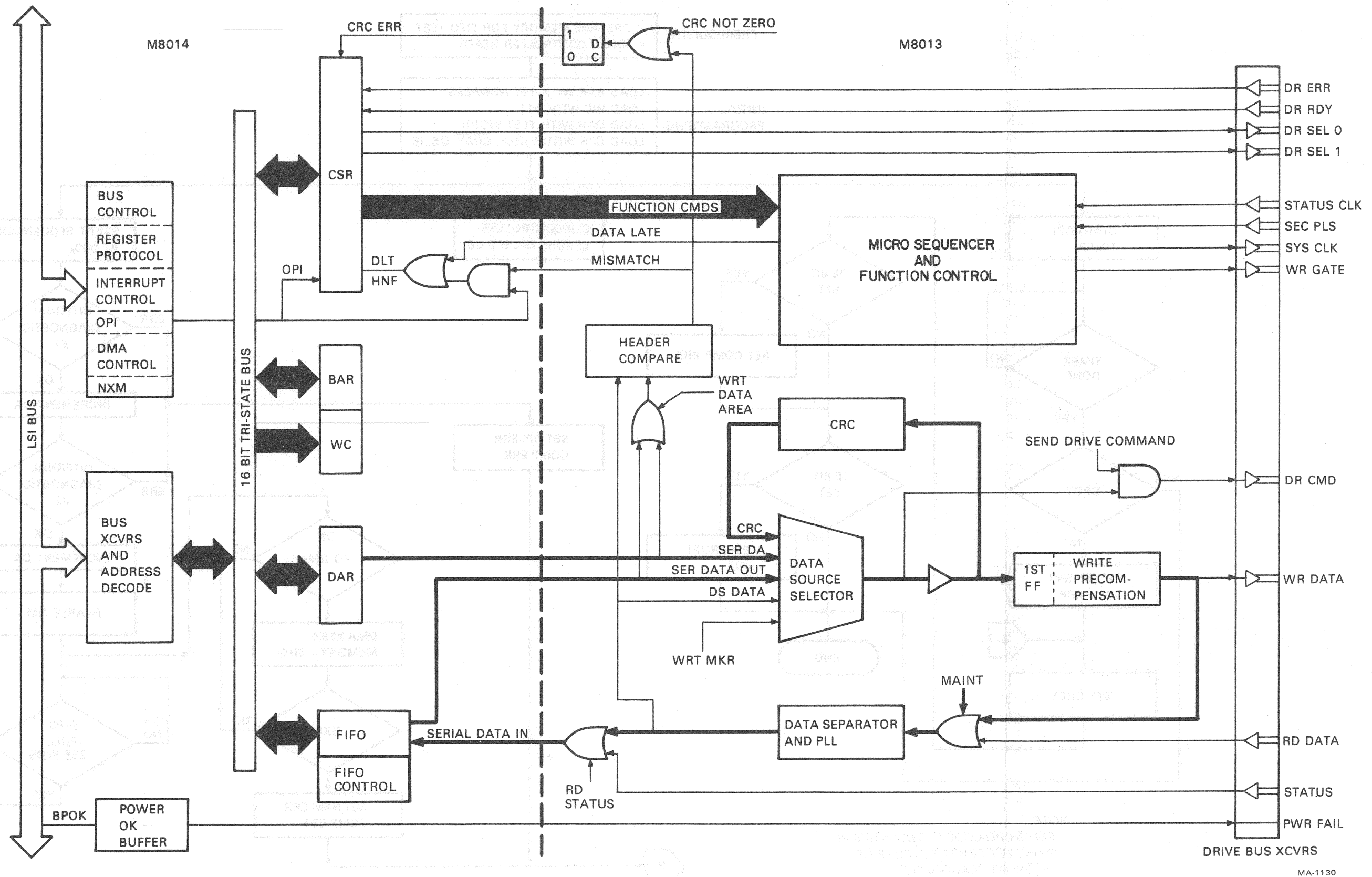
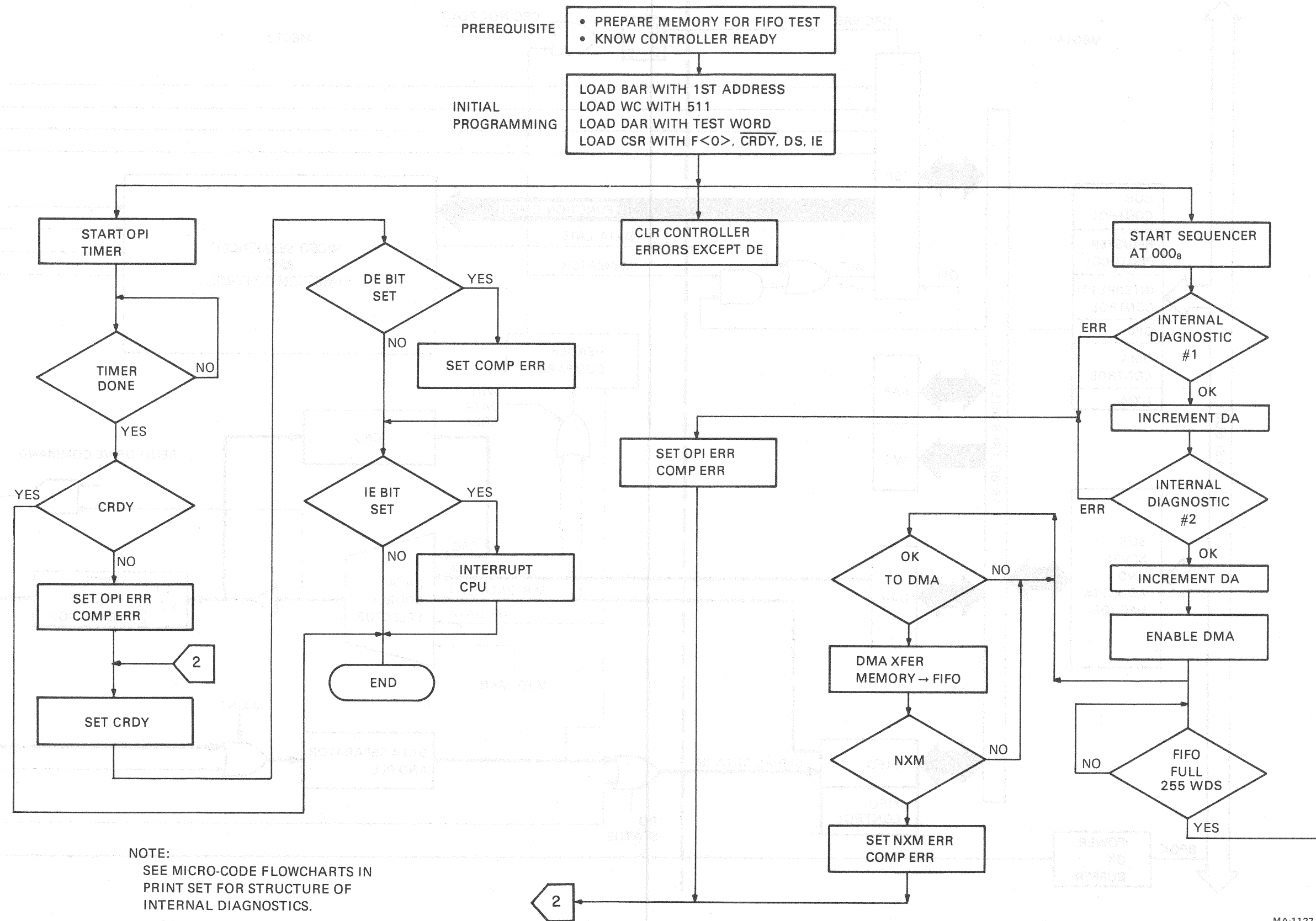


Figure 4-2 Maintenance Command Functional Block Diagram



NOTE:
SEE MICRO-CODE FLOWCHARTS IN
PRINT SET FOR STRUCTURE OF
INTERNAL DIAGNOSTICS.

Figure 4-3 Maintenance Command Flowchart
(Sheet 1 of 2)

MA-1127

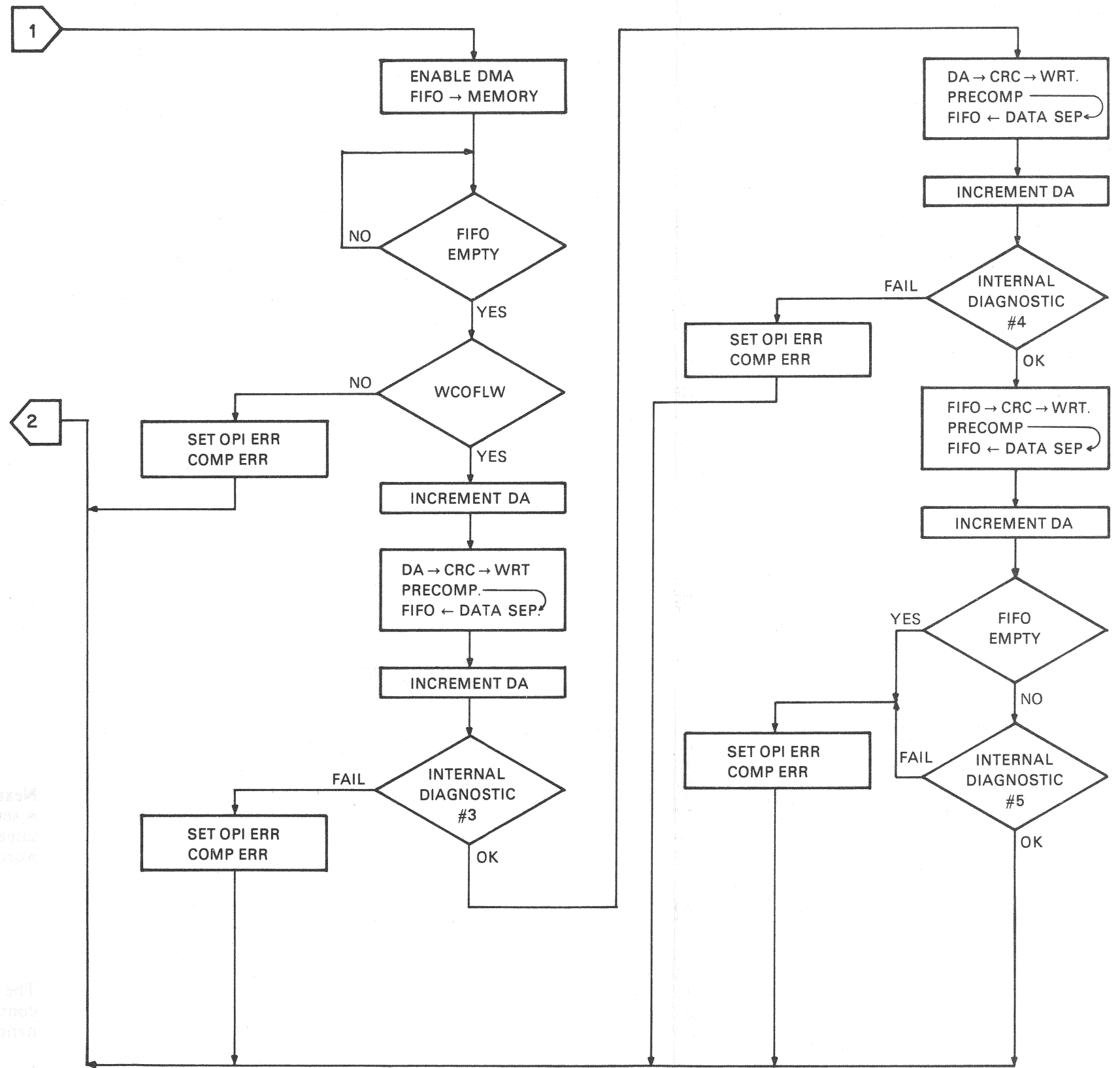
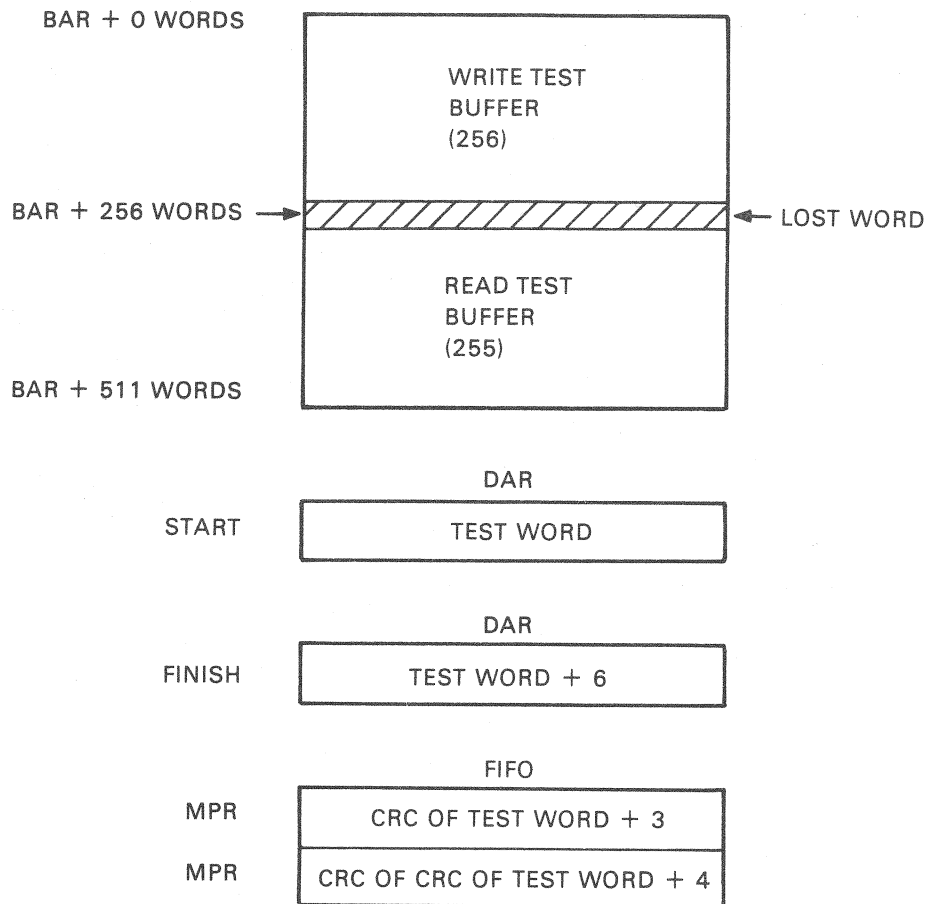


Figure 4-3 Maintenance Command Flowchart
(Sheet 2 of 2)

MA-1128



MA-1165

Figure 4-4 Results of Maintenance Command

Next, the second word in the FIFO (CRC of test word +4) is removed from the FIFO and serialized. It is sent through the data source selector, the CRC, and data source selector again, etc. It follows the same data path as the two previous words and ends up back in the FIFO as the new second FIFO word. At this point, the FIFO holds the following contents.

Word	FIFO
1st	CRC of test word +3
2nd	CRC of CRC of test word +4

The contents of the DAR is then incremented by one for the 6th time to become test word +6. The controller ready bit is then set and the CPU receives an interrupt request. This completes the maintenance command operation. Figure 4-4 shows all the results of this test in memory, FIFO, and DAR.

As a result of this maintenance test, the following circuits are tested: the FIFO, the registers, the data source selector, the CRC circuit, the match circuit, the write precompensation circuit, the data separator circuit, and the FIFO input and output serializer. Also, many of the microsequencer functions are exercised.

4.4 GET STATUS FUNCTIONAL FLOW DIAGRAM

The get status command is used to discover the current drive operational status. It involves sending a drive command word from the controller to the drive and then receiving a status word back from the drive. Refer to Figures 4-5 and 4-6 for the functional flow diagrams.

The only prerequisite for this command is to know that the controller is in the ready state. The DAR is then loaded with a status request word, and the get status command is written into the CSR. When the CSR controller ready bit is cleared, the OPI timer is initiated and the microsequencer decodes the function command. The microsequencer is pointed to the starting address of the get status routine in the sequencer ROM and proceeds step by step through the program.

First, the microsequencer tests the validity of the get status request word in the DAR by monitoring the marker bit and get status bit of the DAR contents. These two bits (DA0 and DA1) must be set or the command operation will be aborted and an OPI timeout error will occur. If these two conditions are met, the DAR contents are serialized and sent through the data source selector circuit into the first stage of the write precompensation circuit. No CRC and no write precompensation operation will be performed on this status request word. A send drive command signal is issued that allows the status request word to be picked off the first flip-flop in the write precompensation circuit and sent to the drive over the drive command line. This drive command word is synchronized with the timing on the system clock line.

Once the drive receives the entire 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 clears all soft errors (error conditions not still present) 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 FIFO. The controller ready bit is set and the interrupt control circuit then sends an interrupt request to the CPU if the interrupt enable bit is set. The drive status can then be read out of the FIFO via the MPR.

4.5 READ HEADER FUNCTION

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 FIFO. Refer to Figures 4-7 and 4-8 for the functional flow diagrams.

When the controller is ready, a read header command may be written into the CSR. At the same time, the controller ready bit is cleared, 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 initiates the read header sequence. Three header words are read off the disk and enter the controller pulse generator circuit over the read data line. From here the header words pass through the data separator circuit and into the FIFO.

This same information also enters 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 are set. If no CRC error is detected, the controller ready bit is set and the interrupt control circuit sends an interrupt request to the CPU if the interrupt enable bit was set. The two header words and the header CRC are then available to be read from the multipurpose register.

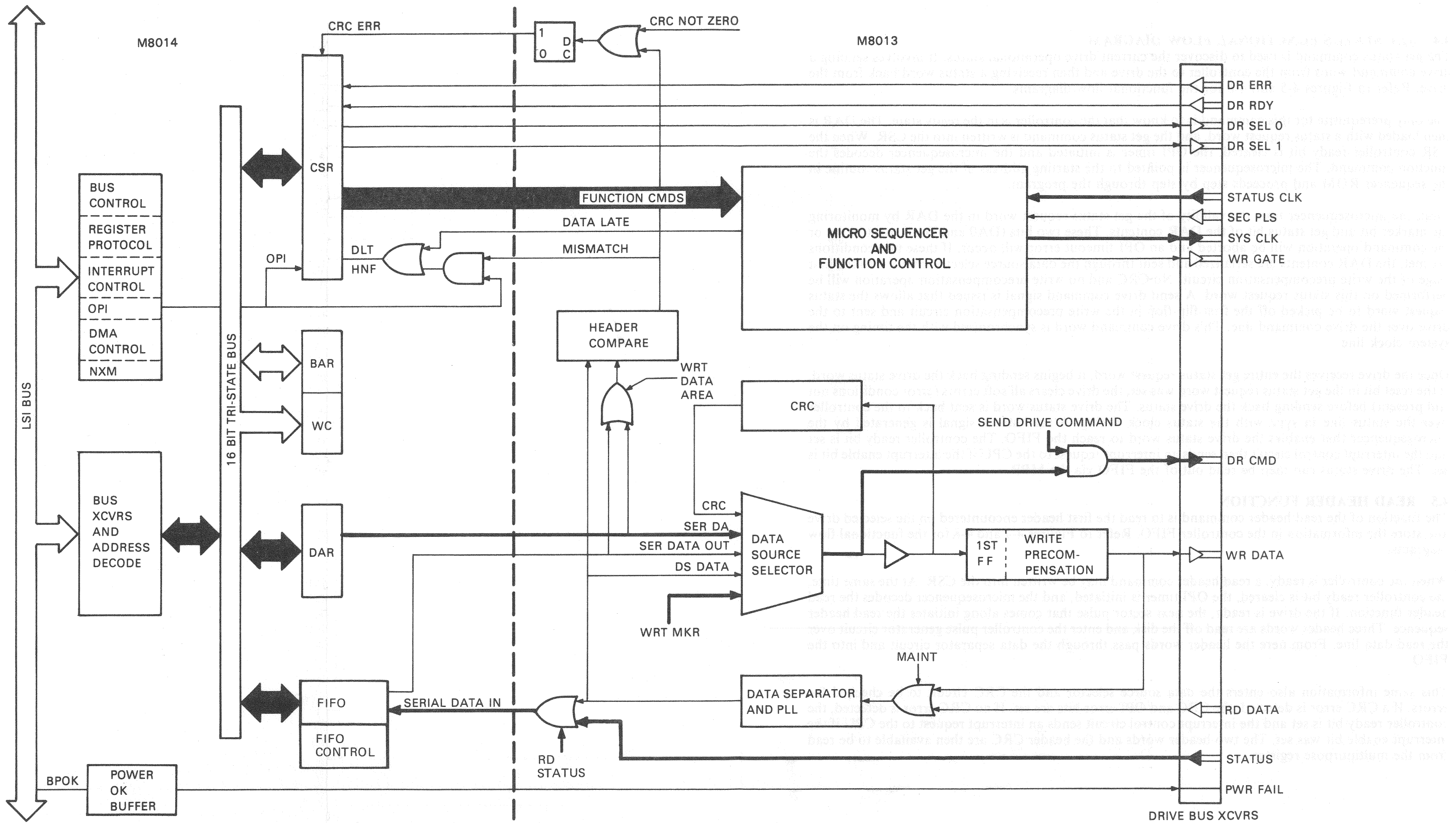
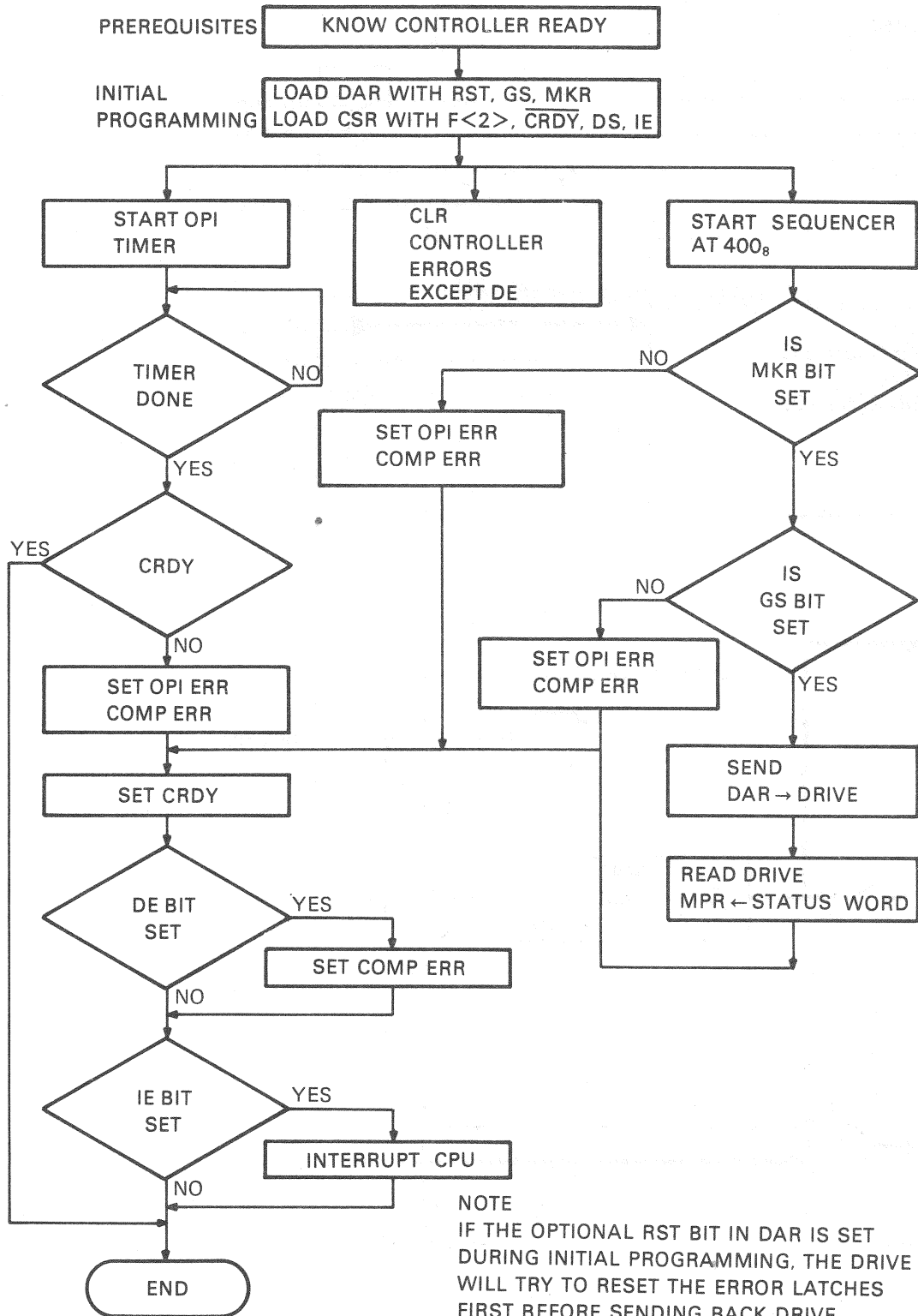


Figure 4-5 Get Status Functional Block Diagram



MA-1163

Figure 4-6 Get Status Command Flowchart

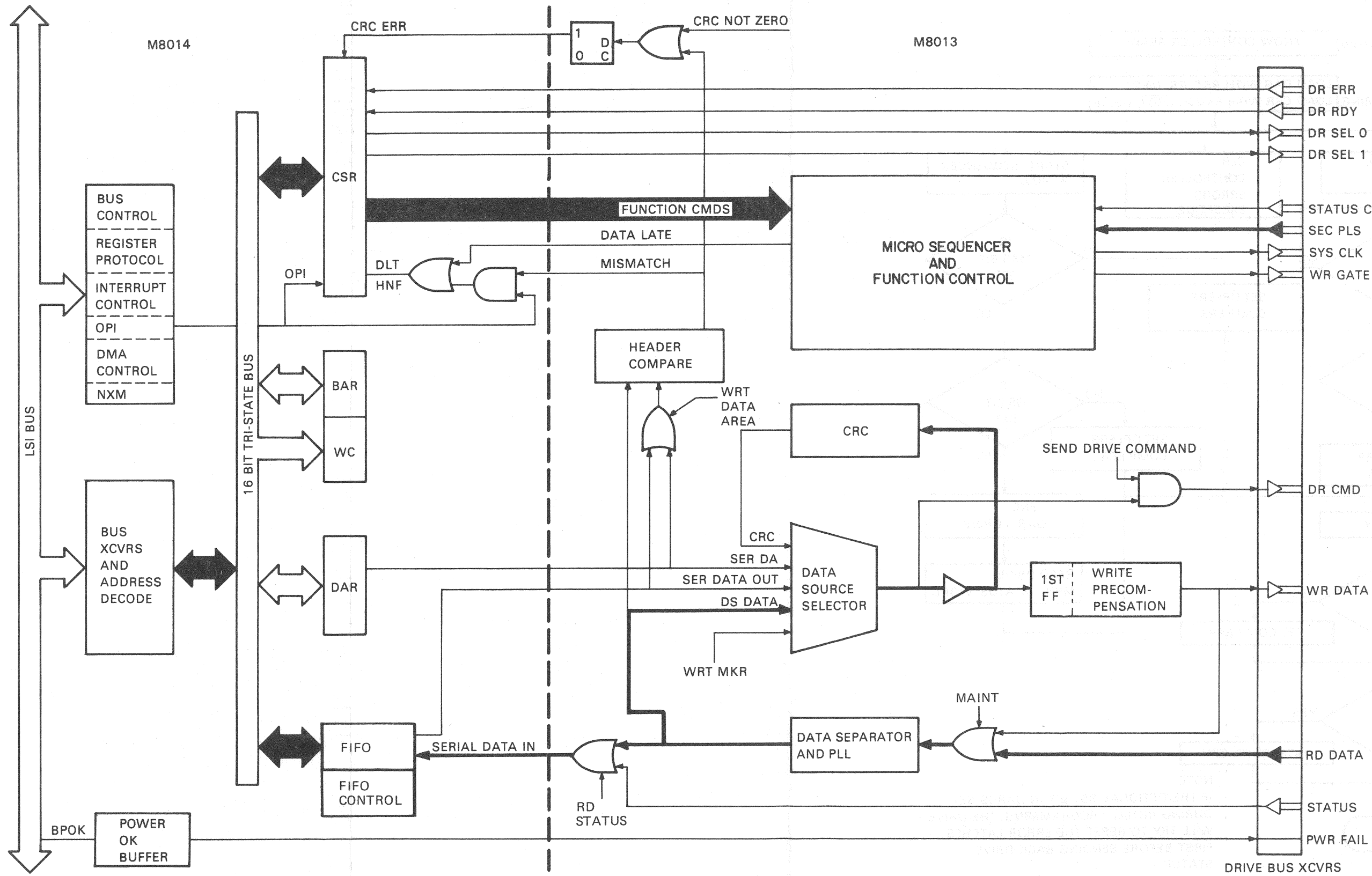


Figure 4-7 Read Header Functional Block Diagram

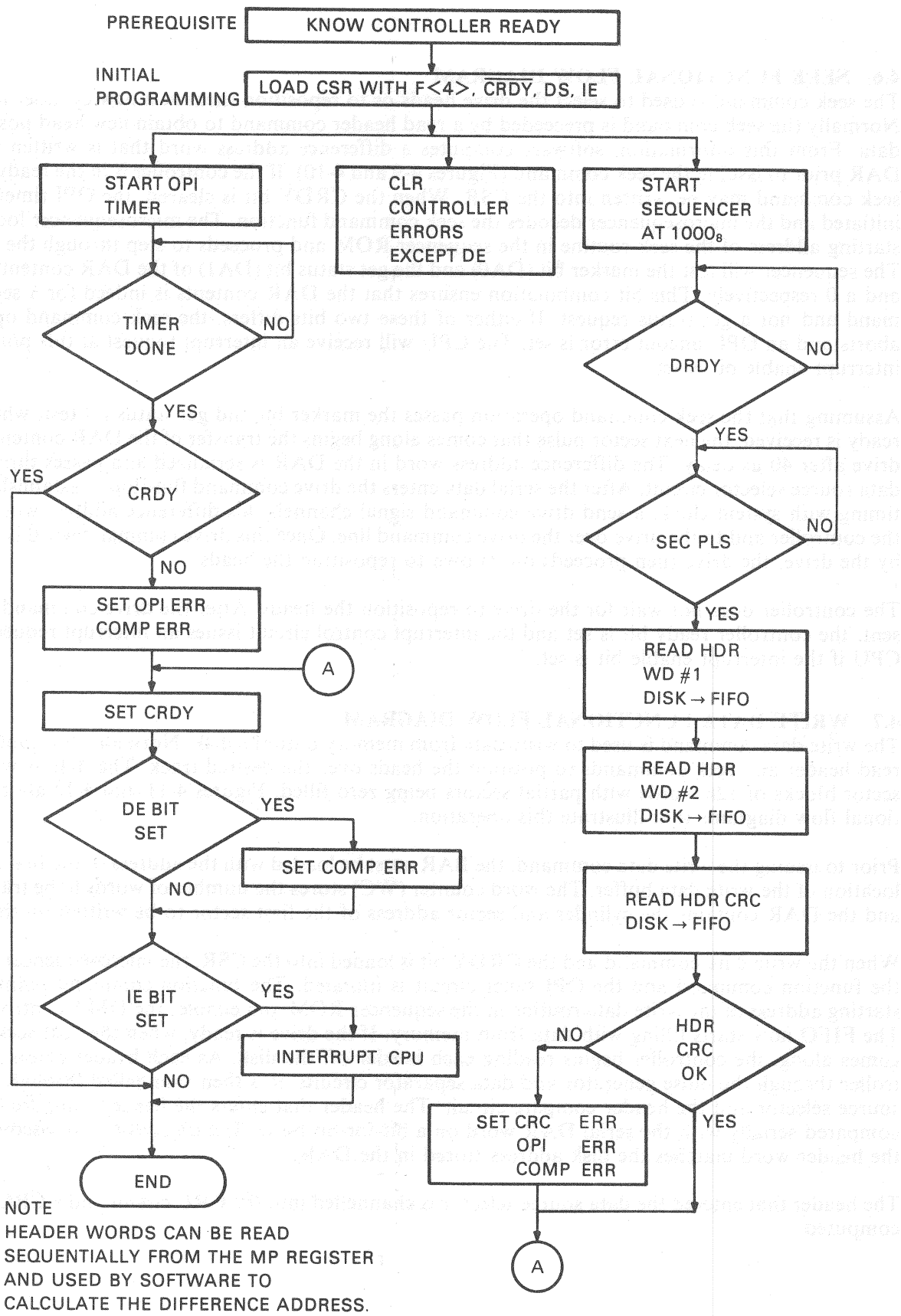


Figure 4-8 Read Header Command Flowchart

MA-1119

4.6 SEEK FUNCTIONAL FLOW DIAGRAM

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 DAR prior to issuing the seek command (Figures 4-9 and 4-10). If the controller is in the ready state, a seek command may be written into the CSR. When the CRDY bit is cleared, 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 sequencer will test the marker bit (DA0) and the get status bit (DA1) of the DAR contents for a 1 and a 0 respectively. This bit combination ensures that the DAR contents is indeed for a seek command and not a get status request. If either of these two bits differs, the seek command operation aborts and an OPI timeout error is set. The CPU will receive an interrupt request at this point if the interrupt enable bit is set.

Assuming that the seek command operation passes the marker bit and get status bit test, when drive ready is received, the next sector pulse that comes along begins the transfer of the DAR contents to the drive after 40 μ s delay. The difference address word in the DAR is serialized and passes through the data source selector circuit. After the serial data enters the drive command flip-flop to establish correct timing with system clock, a send drive command signal channels this difference address word out of the controller and to the drive over the drive command line. 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, the controller ready bit is set and the interrupt control circuit issues an interrupt request to the CPU if the interrupt enable bit is set.

4.7 WRITE DATA FUNCTIONAL FLOW DIAGRAM

The write data command is used to write data from memory onto the disk. Normally, it is preceded by read header and seek commands to position the heads over the desired track. The data is written in sector blocks of 128 words with partial sectors being zero filled. Figures 4-11 and 4-12 are the functional flow diagrams that illustrate this operation.

Prior to issuing the write data command, the BAR must be loaded with the address of the first memory location of the write data buffer. The word counter (WC) stores the number of words to be transferred and the DAR contains the cylinder and sector address of the first sector to be written on the disk.

When the write data command and the CRDY bit is loaded into the CSR, the microsequencer decodes the function command and the OPI timer circuit is initiated. The function command points to the starting address of the write data routine in the sequencer ROM and enables the DMA control circuit. The FIFO 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 generator and data separator circuits, it is then channelled 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 disk address stored in the DAR.

The header that entered the data source selector is channelled into the CRC circuit and a CRC word is computed.

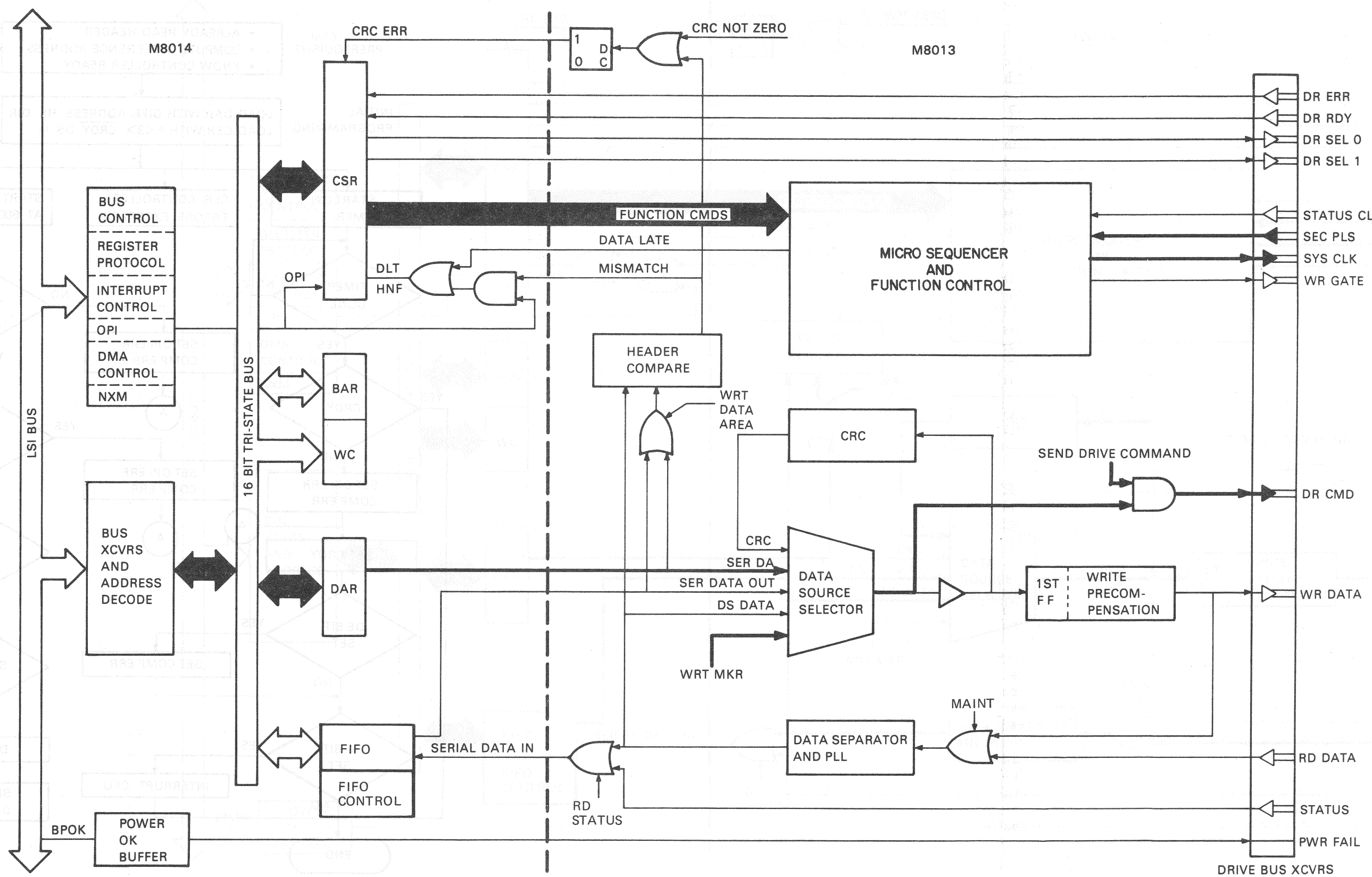
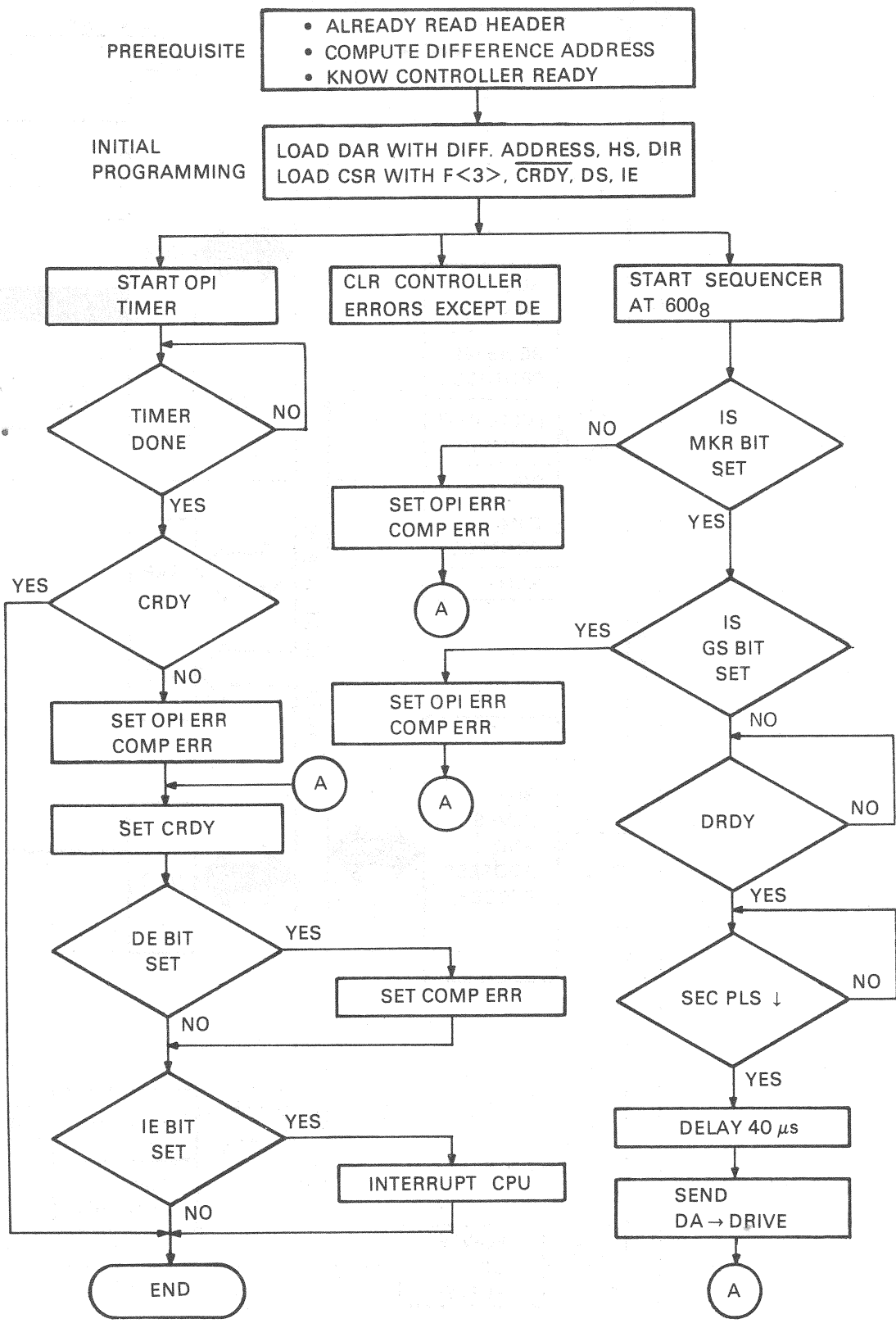


Figure 4-9 Seek Functional Block Diagram



MA-1164

Figure 4-10 Seek Command Flowchart

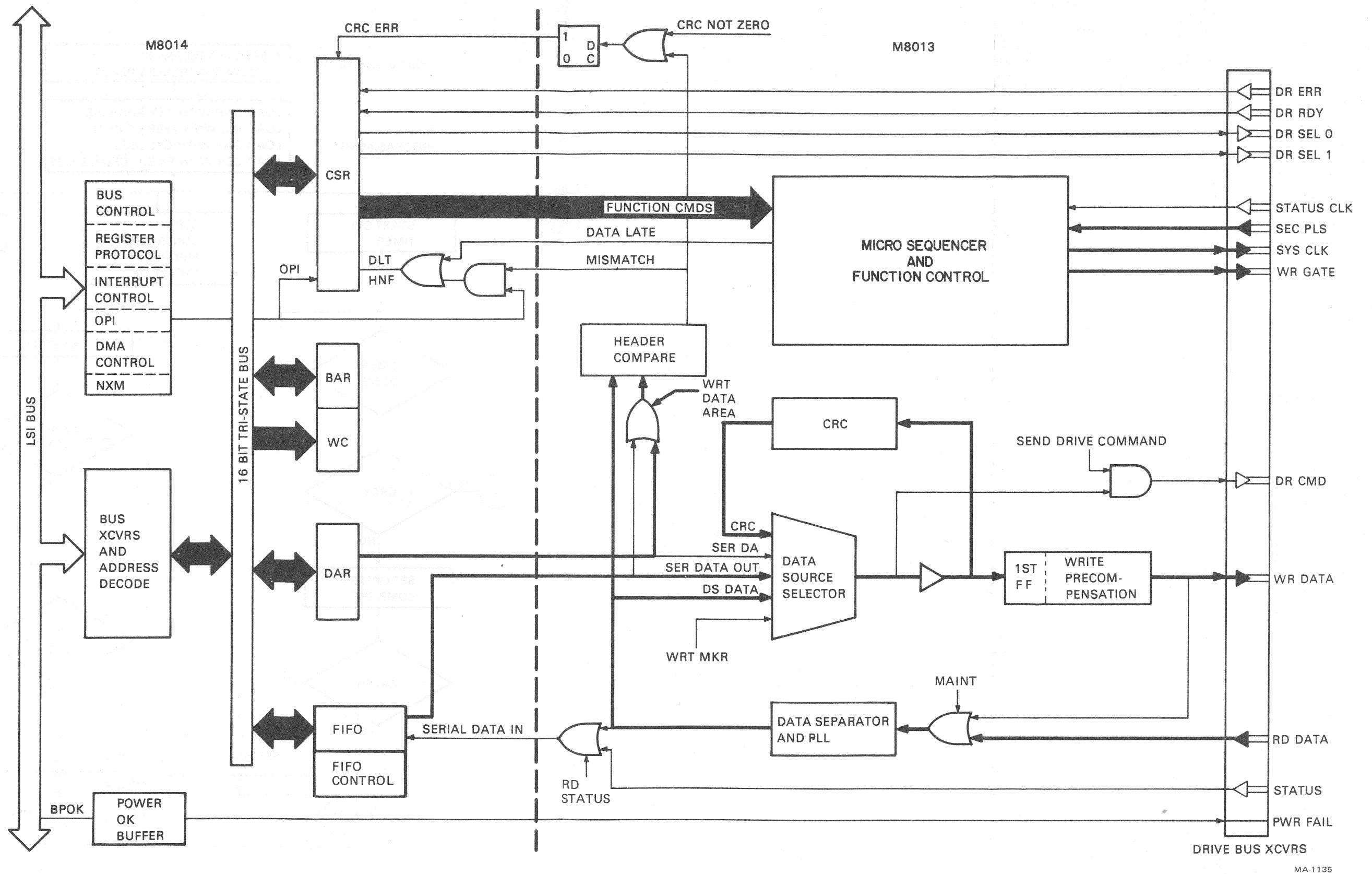
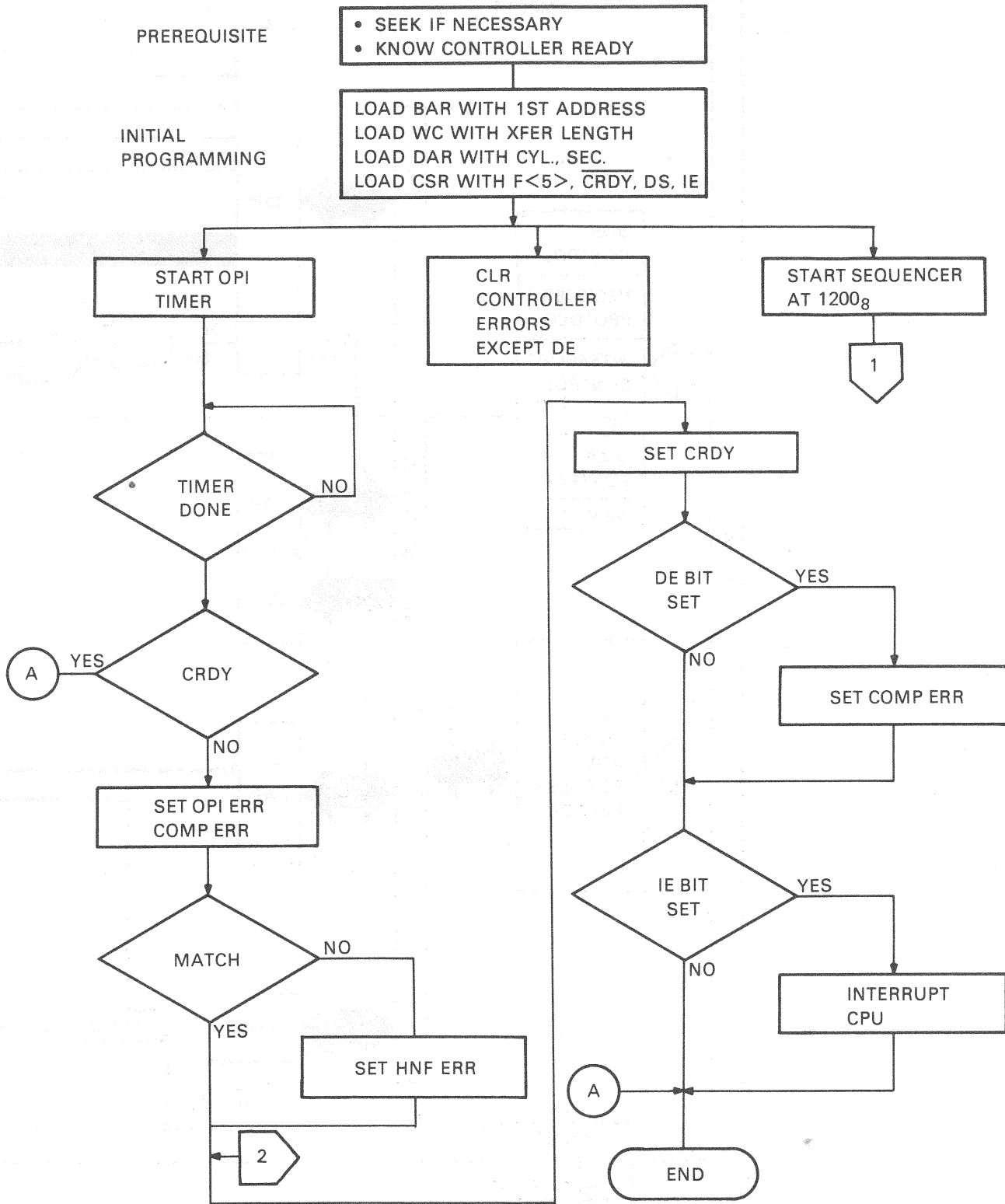


Figure 4-11 Write Data Functional Block Diagram



MA-1154

Figure 4-12 Write Data Command Flowchart (Sheet 1 of 2)

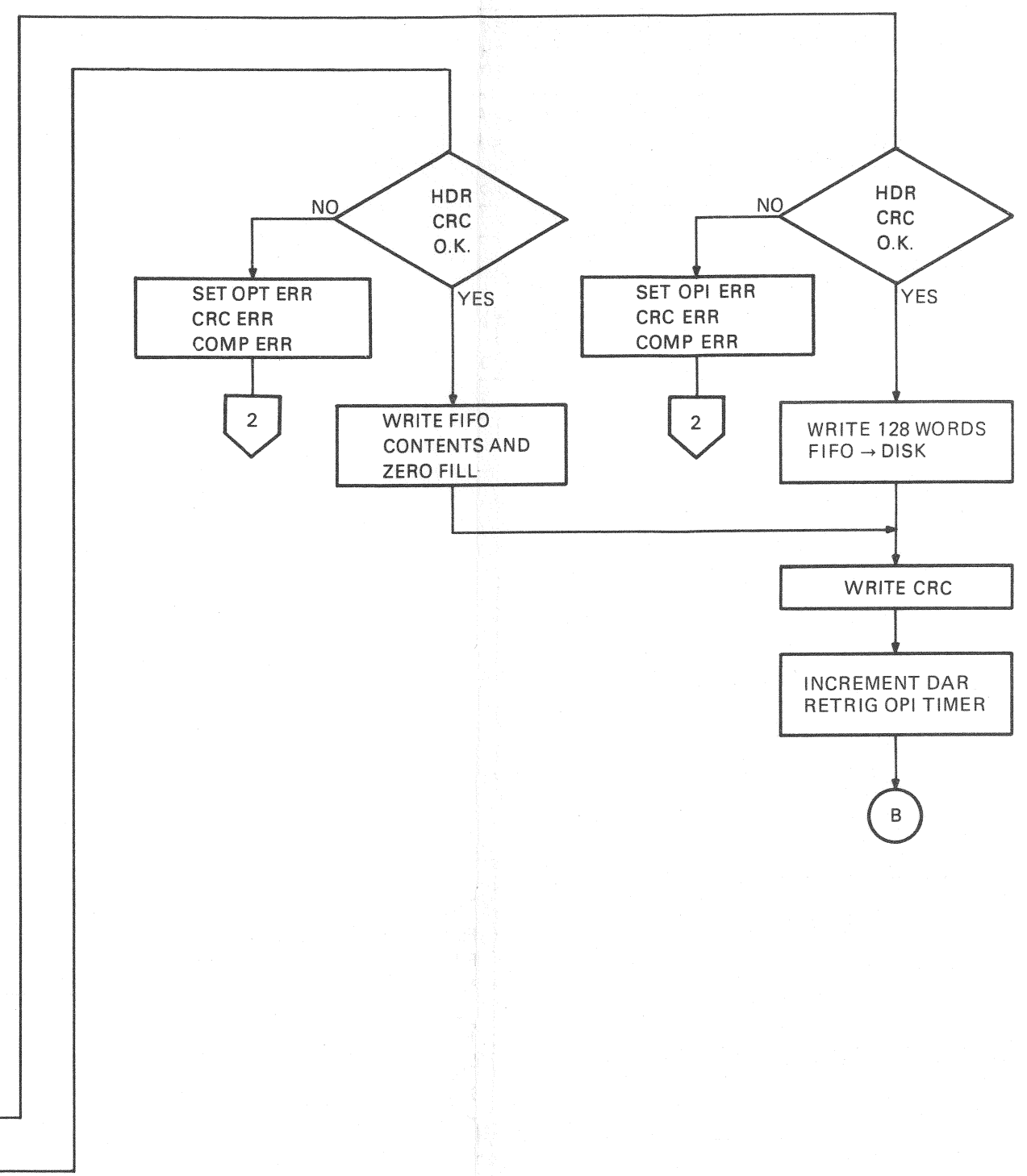
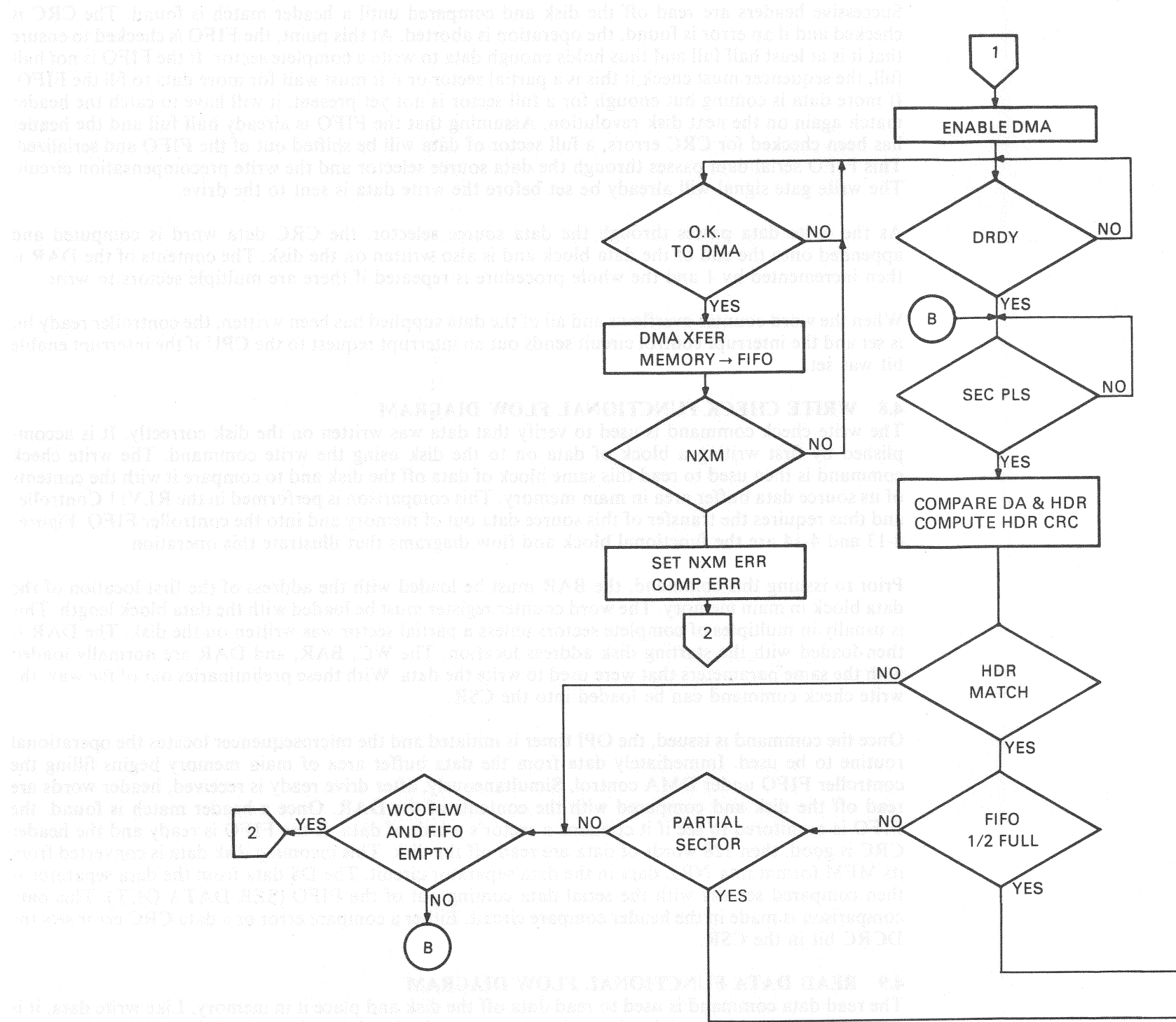


Figure 4-12 Write Data Command Flowchart (Sheet 2 of 2)

Successive headers are read off the disk and compared until a header match is found. The CRC is checked and if an error is found, the operation is aborted. At this point, the FIFO is checked to ensure that it is at least half full and thus holds enough data to write a complete sector. If the FIFO is not half full, the sequencer must check if this is a partial sector or if it must wait for more data to fill the FIFO. If more data is coming but enough for a full sector is not yet present, it will have to catch the header match again on the next disk revolution. Assuming that the FIFO is already half full and the header has been checked for CRC errors, a full sector of data will be shifted out of the FIFO and serialized. This FIFO 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.

As the write data passes through the data source selector, the CRC data word is computed and appended onto the end of the data block and is also written on the disk. The contents of the DAR is then incremented by 1 and the whole procedure is repeated if there are multiple sectors to write.

When the word counter overflows and all of the data supplied has been written, the controller ready bit is set and the interrupt control circuit sends out an interrupt request to the CPU if the interrupt enable bit was set.

4.8 WRITE CHECK FUNCTIONAL FLOW DIAGRAM

The write check command is used to verify that data was written on the disk correctly. It is accomplished by first writing a block of data on to the disk using the write command. The write check command is then used to read this same block of data off the disk and to compare it with the contents of its source data buffer area in main memory. This comparison is performed in the RLV11 Controller and thus requires the transfer of this source data out of memory and into the controller FIFO. Figures 4-13 and 4-14 are the functional block and flow diagrams that illustrate this operation.

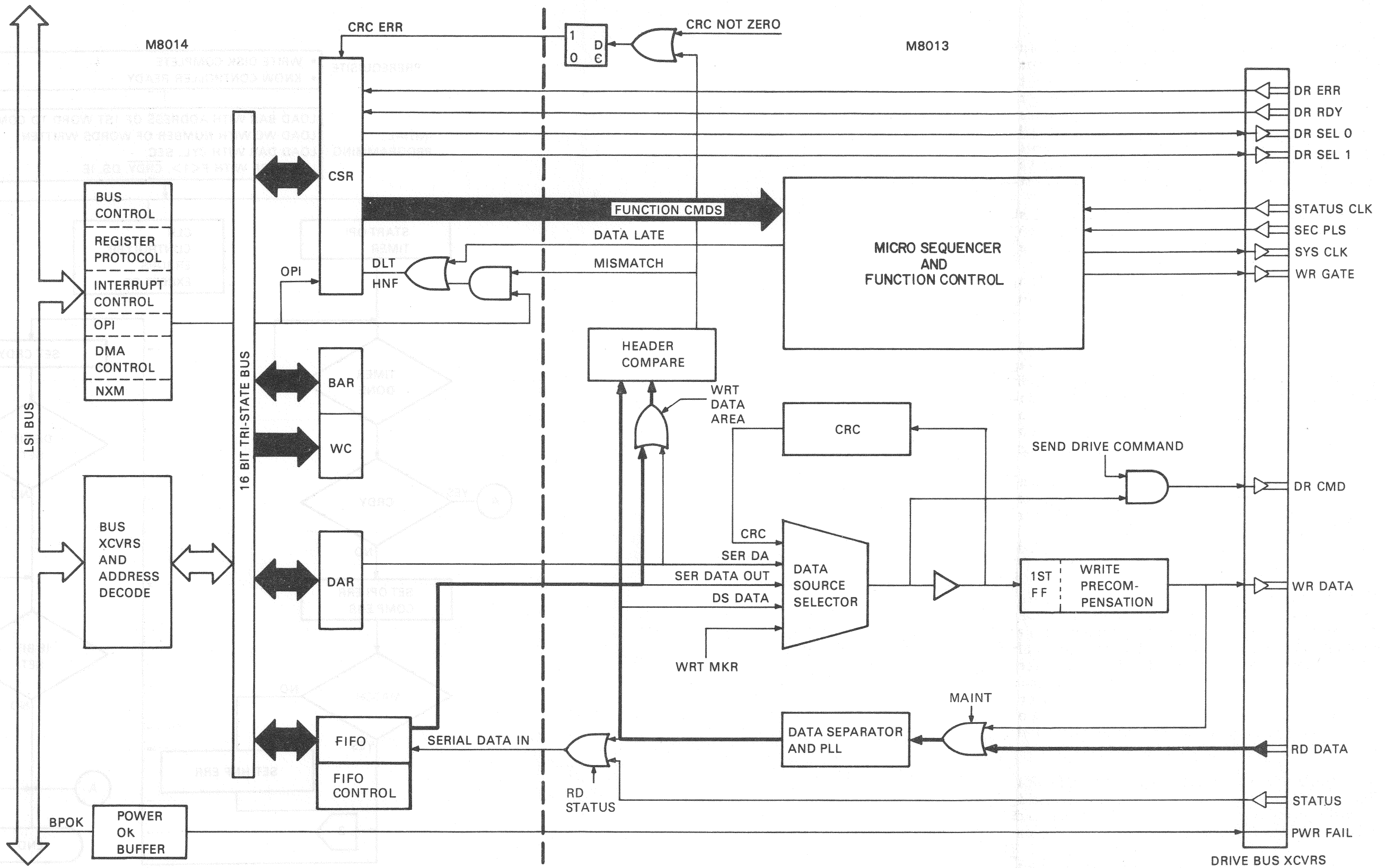
Prior to issuing this command, the BAR must be loaded with the address of the first location of the data block in main memory. The word counter register must be loaded with the data block length. This is usually in multiples of complete sectors unless a partial sector was written on the disk. The DAR is then loaded with the starting disk address location. The WC, BAR, and DAR are normally loaded with the same parameters that were used to write the data. With these preliminaries out of the way, the write check command can be loaded into the CSR.

Once the command is issued, the OPI timer is initiated and the microsequencer locates the operational routine to be used. Immediately data from the data buffer area of main memory begins filling the controller FIFO under DMA control. Simultaneously, after drive ready is received, header words are read off the disk and compared with the contents of the DAR. Once a header match is found, the FIFO is monitored to see if it contains a sector's worth of data. If the FIFO is ready and the header CRC is good, then 128 words of data are read off the disk. This incoming disk data is converted from its MFM format into NRZ data in the data separator circuit. The DS data from the data separator is then compared serially with the serial data coming out of the FIFO (SER DATA OUT). This data comparison is made in the header compare circuit. Either a compare error or a data CRC error sets the DCRC bit in the CSR.

4.9 READ DATA FUNCTIONAL FLOW DIAGRAM

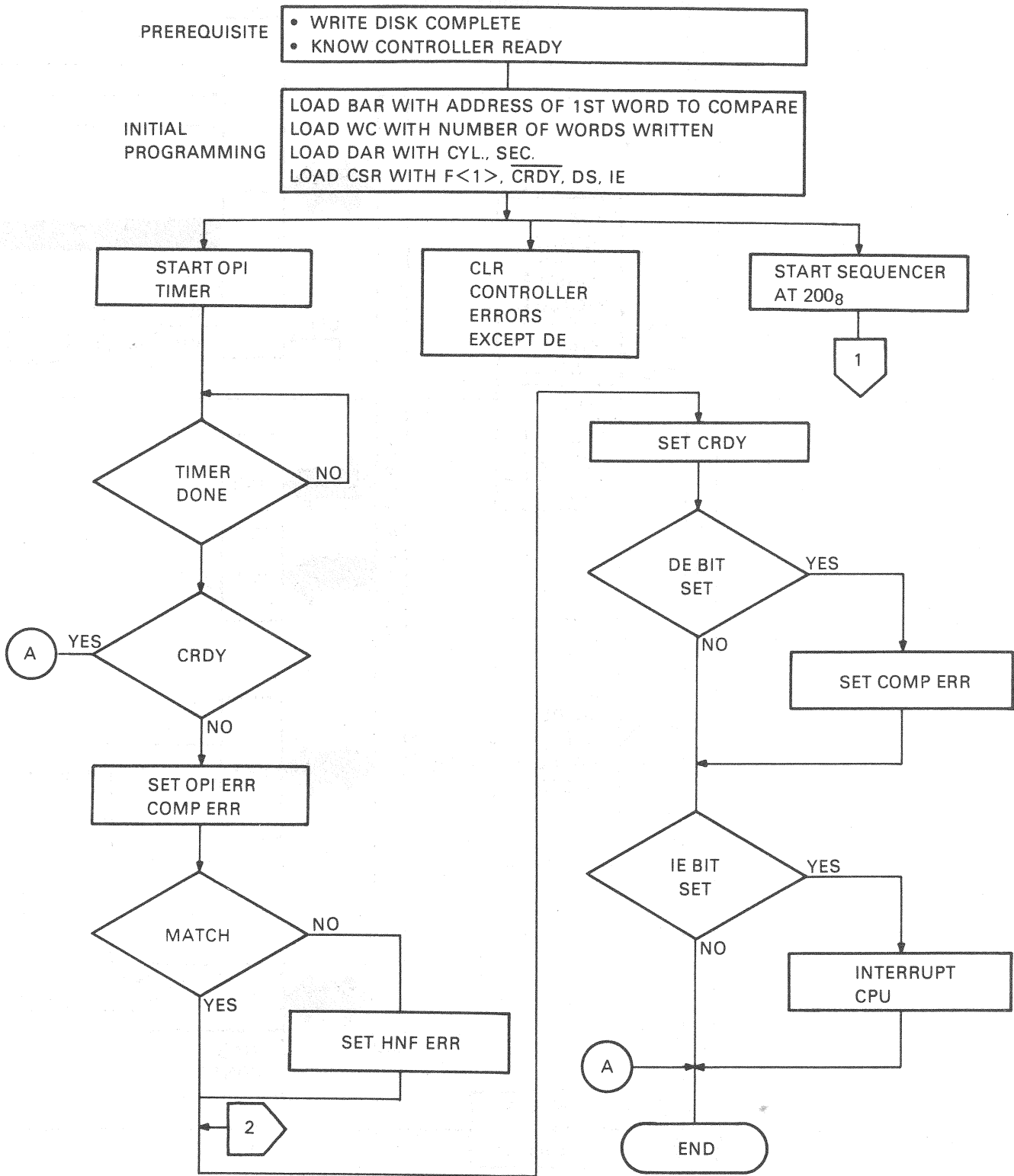
The read data command is used to read data off the disk and place it in memory. Like write data, it is normally preceded by read header and seek commands. Read data is read off the disk in full sectors. Figures 4-15 and 4-16 are the functional flow diagrams that illustrate this operation.

Prior to issuing the read data command, the BAR must be loaded with the address of the first memory location of the read buffer. The word counter stores the number of words to be transferred and the DAR contains the cylinder and sector address of the first sector to be read off the disk.



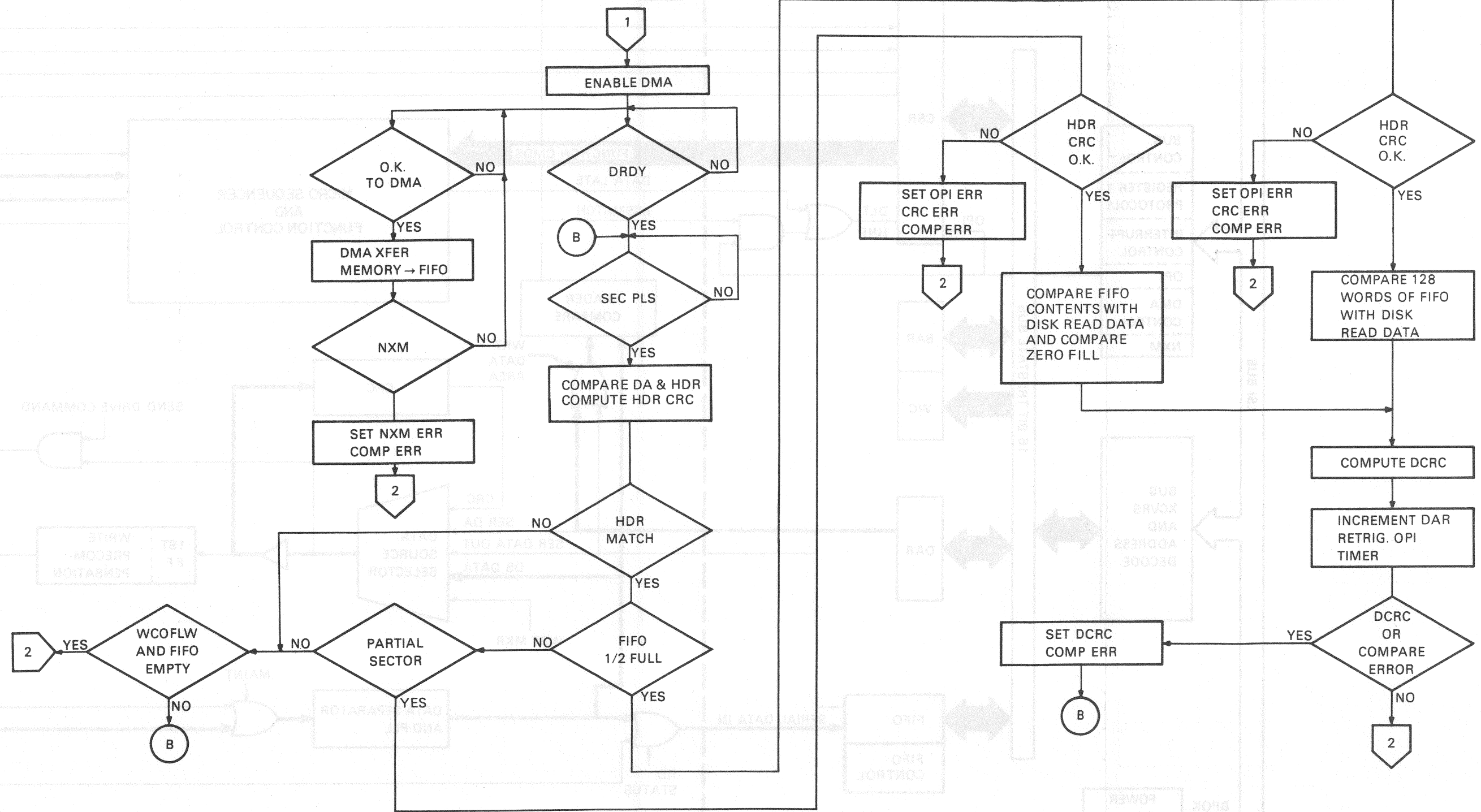
MA-1129

Figure 4-13 Write Check Functional Block Diagram



MA-1152

Figure 4-14 Write Check Command Flowchart (Sheet 1 of 2)



MA-1153

Figure 4-14 Write Check Command Flowchart (Sheet 2 of 2)

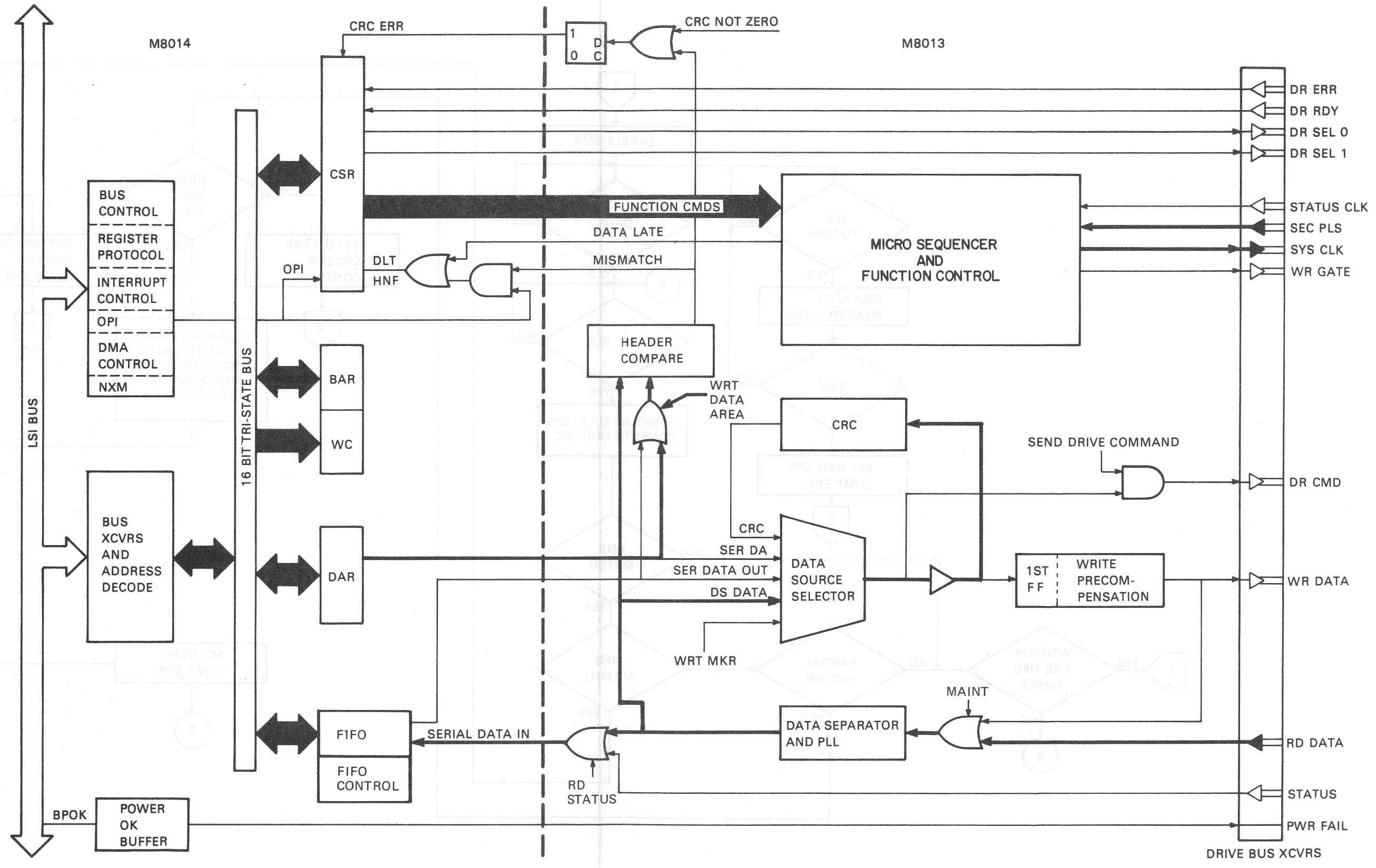


Figure 4-15 Read Data Functional Block Diagram

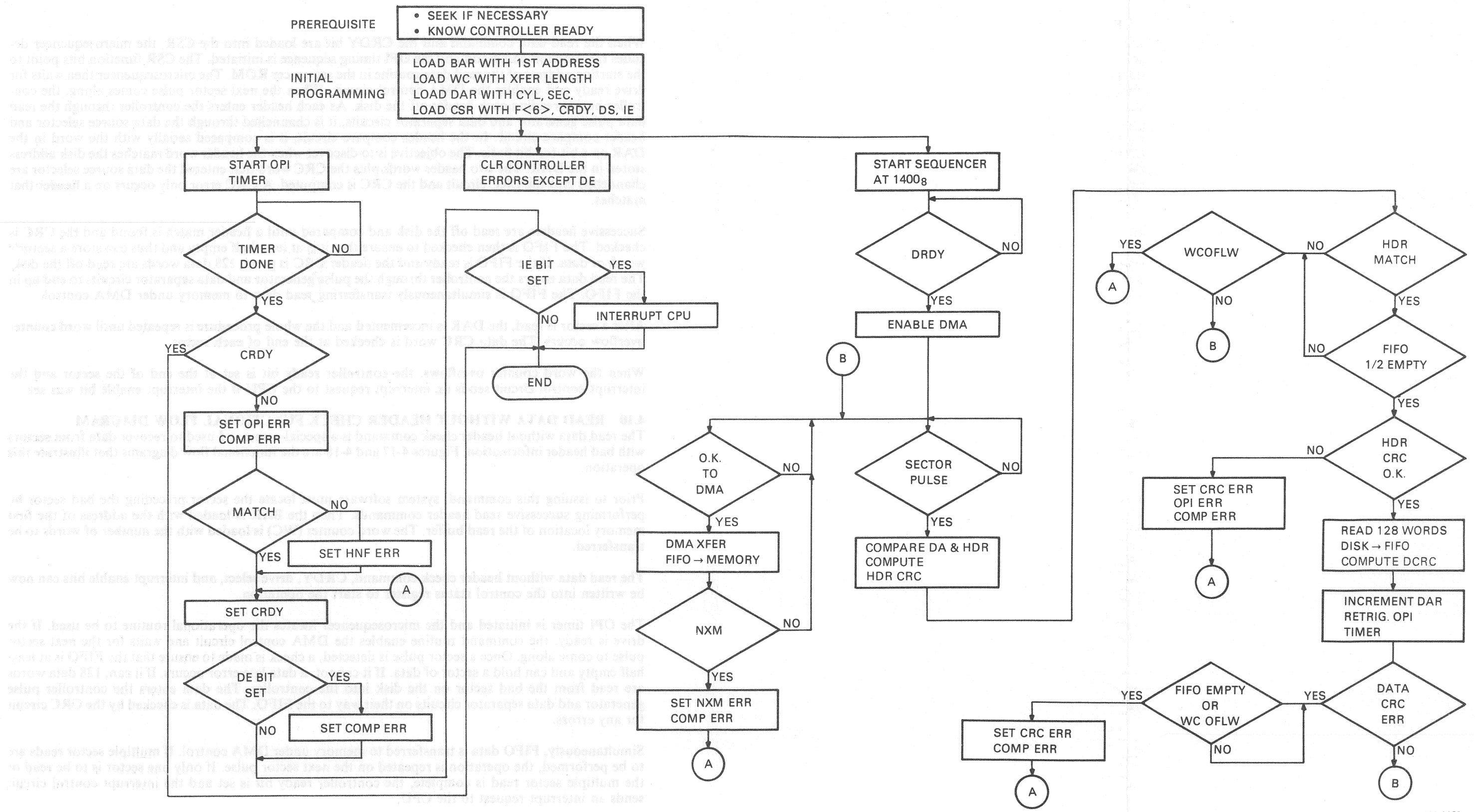


Figure 4-16 Read Data Command Flowchart

When the read data command and the CRDY bit are loaded into the CSR, the microsequencer decodes the function command and the OPI timing sequence is initiated. The CSR function bits point to the starting address of the read data routine in the sequencer ROM. The microsequencer then waits for drive ready 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 read data pulse generator and data separator circuits, it is channelled through the data source selector and header compare circuit. In the header compare circuit, it is compared serially with the word in the DAR on a bit-for-bit basis. The objective is to discover when the header word matches the disk address stored in the DAR. The two header words plus the CRC word that entered the data source selector are channelled into the CRC circuit and the CRC is computed. A CRC error only occurs on a header that matches.

Successive headers are read off the disk and compared until a header match is found and the CRC is checked. The FIFO is then checked to ensure that it is at least half empty and thus can store a sector's worth of data. If the FIFO 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 generator and data separator circuits to end up in the FIFO. The FIFO is simultaneously transferring read data to memory under DMA control.

After a sector is read, the DAR is incremented and the whole procedure is repeated until word counter overflow occurs. The data CRC word is checked at the end of each sector.

When the word counter overflows, the controller ready bit is set at the end of the sector and the interrupt control circuit sends an interrupt request to the CPU if the interrupt enable bit was set.

4.10 READ DATA WITHOUT HEADER CHECK FUNCTIONAL FLOW DIAGRAM

The read data without header check command is a special command used to recover data from sectors with bad header information. Figures 4-17 and 4-18 are the functional flow diagrams that illustrate 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 BAR is loaded with the address of the first memory location of the read buffer. The word counter (WC) is loaded with the number of words to be transferred.

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

The OPI timer is initiated and the microsequencer locates the operational routine to be used. If the drive is ready, the command routine enables the DMA control circuit and waits for the next sector pulse to come along. Once a sector pulse is detected, a check is made to ensure that the FIFO is at least half empty and can hold a sector of data. If it cannot, a data late error occurs. If it can, 128 data words are read from the bad sector on the disk into the controller. The data enters the controller pulse generator and data separator circuits on their way to the FIFO. The data is checked by the CRC circuit for any errors.

Simultaneously, FIFO data is transferred to memory under DMA control. If multiple sector reads are to be performed, the operation is repeated on the next sector pulse. If only one sector is to be read or the multiple sector read is complete, the controller ready bit is set and the interrupt control circuit sends an interrupt request to the CPU.

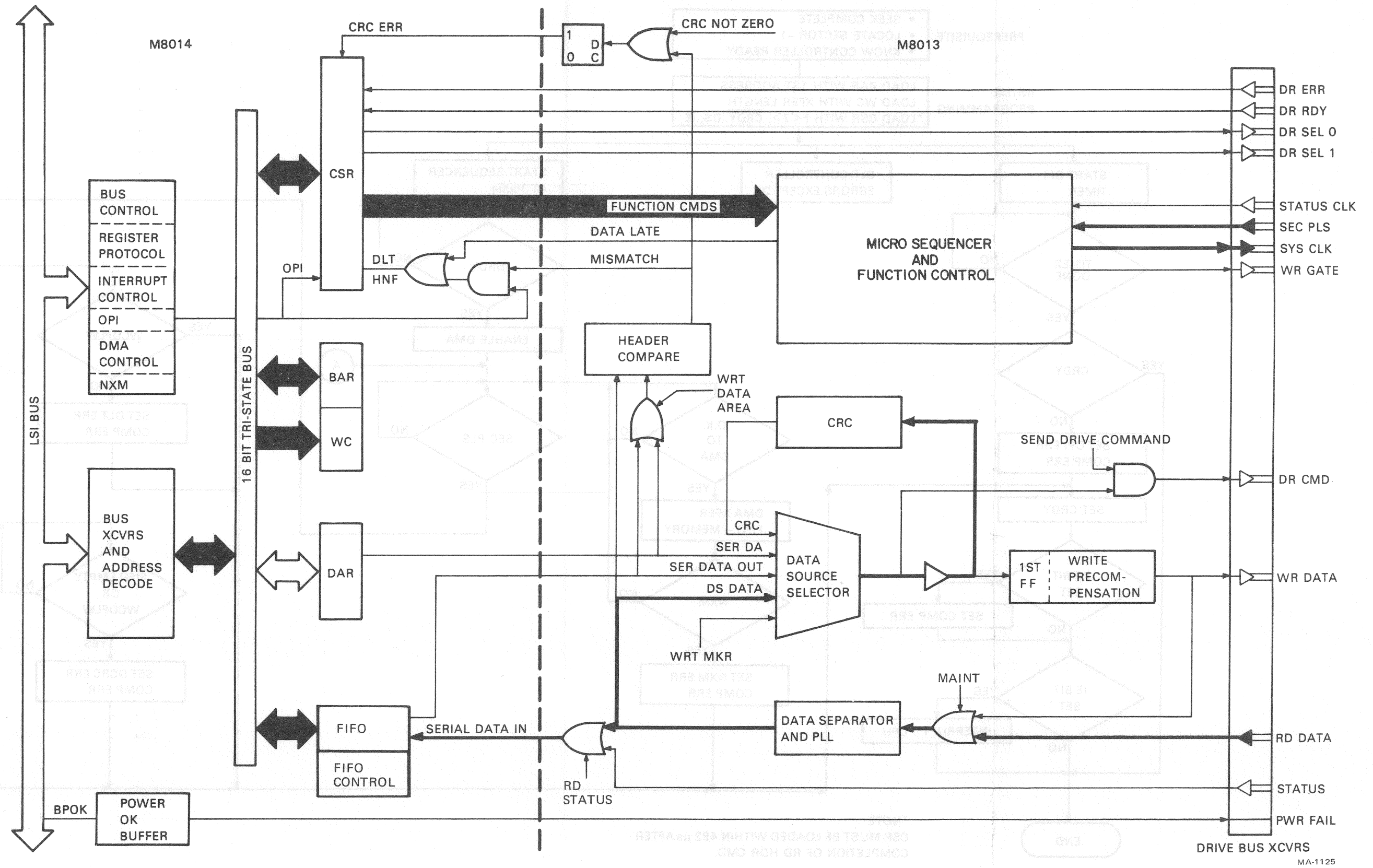


Figure 4-17 Read Data Without Header Check Functional Block Diagram

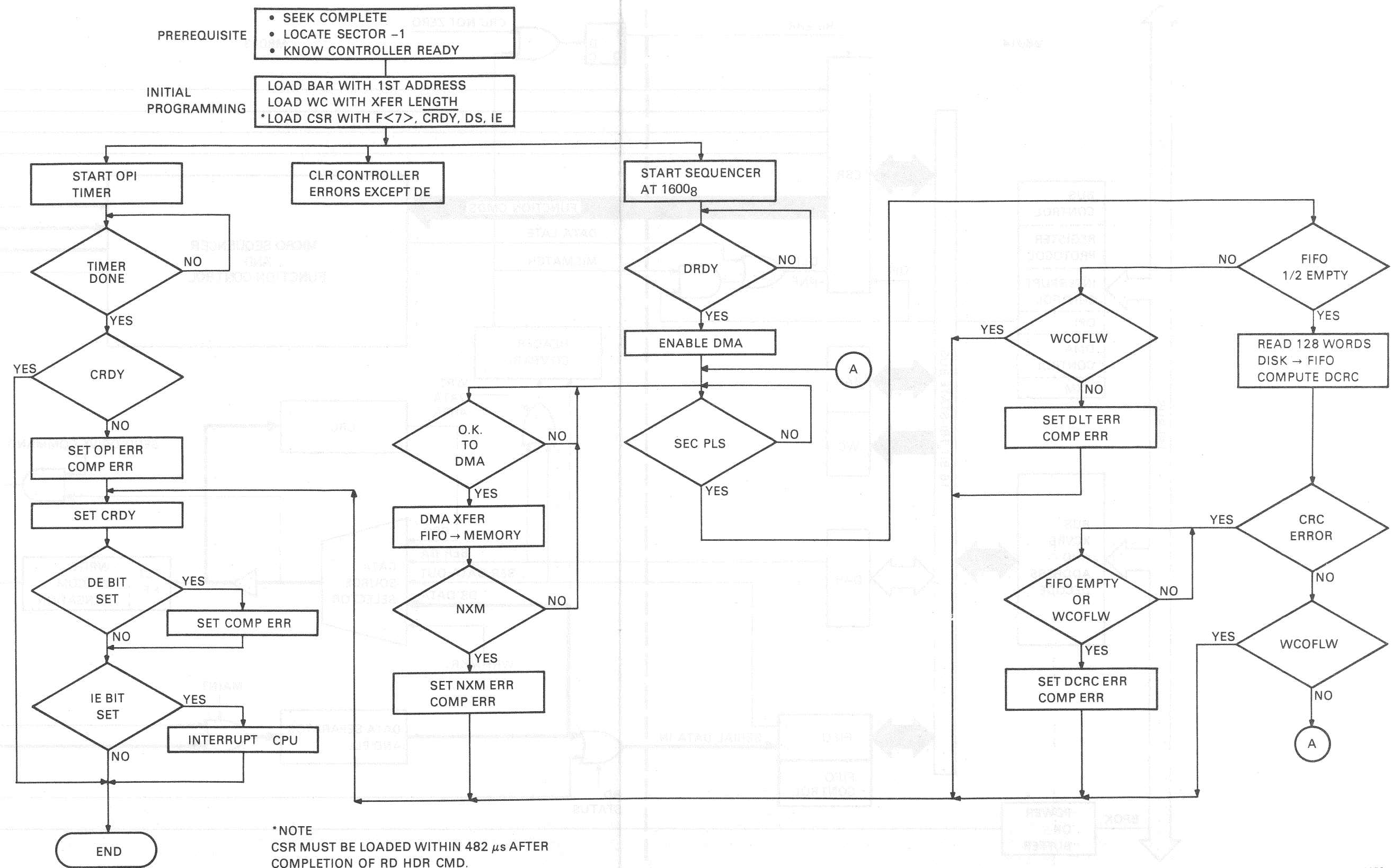


Figure 4-18 Read Data Without Header Check Command Flowchart

CHAPTER 5 UNIT LEVEL DESCRIPTION

5.1 GENERAL

This chapter provides a detailed theory of operation of the various controller functions. Detailed block diagrams, flowcharts, and timing diagrams are used to illustrate the sequential relationships of the controller's internal signals. The functional logic blocks used throughout this chapter are directly related to the RLV11 controller annotated print set.

5.2 LSI-11 BUS CONTROL CIRCUIT

The LSI-11 bus control circuit consists of the following functional elements (Figure 5-1).

- Bus Transceivers
- Register Protocol
- DMA Control
- Interrupt Control
- NXM and OPI Timers
- Addressable Registers

5.2.1 Bus Transceivers

The bus transceivers consist of four DC005 integrated circuits. Each DC005 integrated circuit is a 4-bit transceiver that functions as a bidirectional buffer between the LSI-11 data bus and the controller tri-state disk/address lines. In addition to its isolation function, this device also provides a comparison circuit for address selection and a constant generator, useful for interrupt vector addresses.

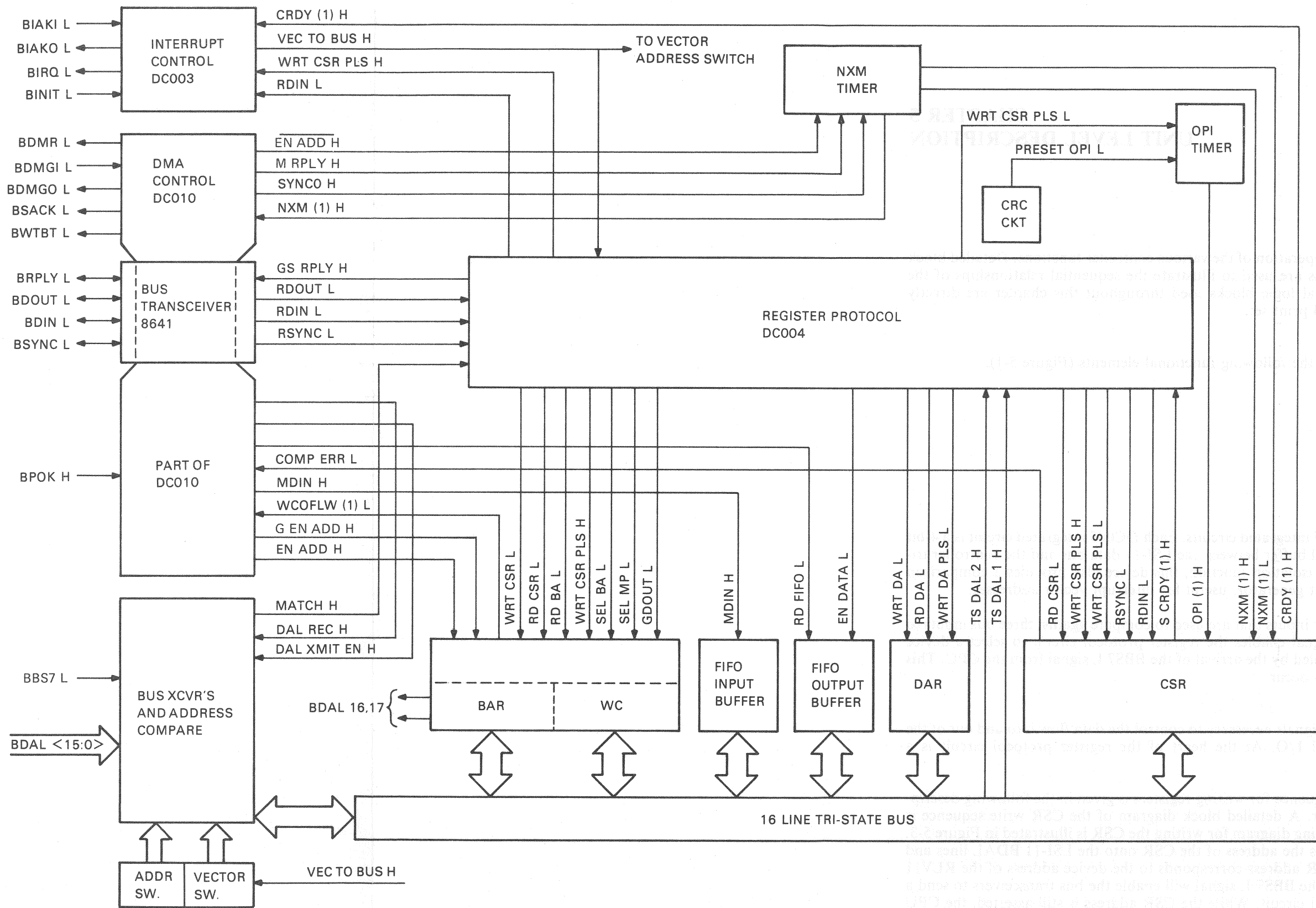
Each DC005 chip has three address jumper inputs that are used to compare against three bus inputs to generate the match signal. This match signal enables the register protocol circuit to select a device register. The bus transceiver circuit is enabled by the arrival of the BBS7 L signal from the CPU. This allows the generation of a match signal to occur.

5.2.2 Register Protocol

The register protocol circuit provides the signals necessary to control the data flow into and out of the addressable registers during programmed I/O. At the heart of the register protocol circuit is a DIGITAL DC004 integrated circuit.

5.2.2.1 Writing a Register – A typical sequence for writing registers is given in the following description for writing the control status register. A detailed block diagram of the CSR write sequence is provided in Figure 5-2. A bus protocol timing diagram for writing the CSR is illustrated in Figure 5-3. The sequence begins when the CPU places the address of the CSR onto the LSI-11 BDAL lines and asserts BWTBT L and BBS7 L. If the CSR address corresponds to the device address of the RLV11 controller set up in the address switches, the BBS7 L signal will enable the bus transceivers to send a MATCH H signal to the register protocol circuit. While the CSR address is still asserted, the CPU issues a B SYNC L signal which arrives at the register protocol chip in the form R SYNC L. The assert edge of R SYNC L latches the MATCH H signal and the address information into the protocol chip.

LSI-11 BUS



MA-1147

Figure 5-1 LSI-11 Bus Control Circuit

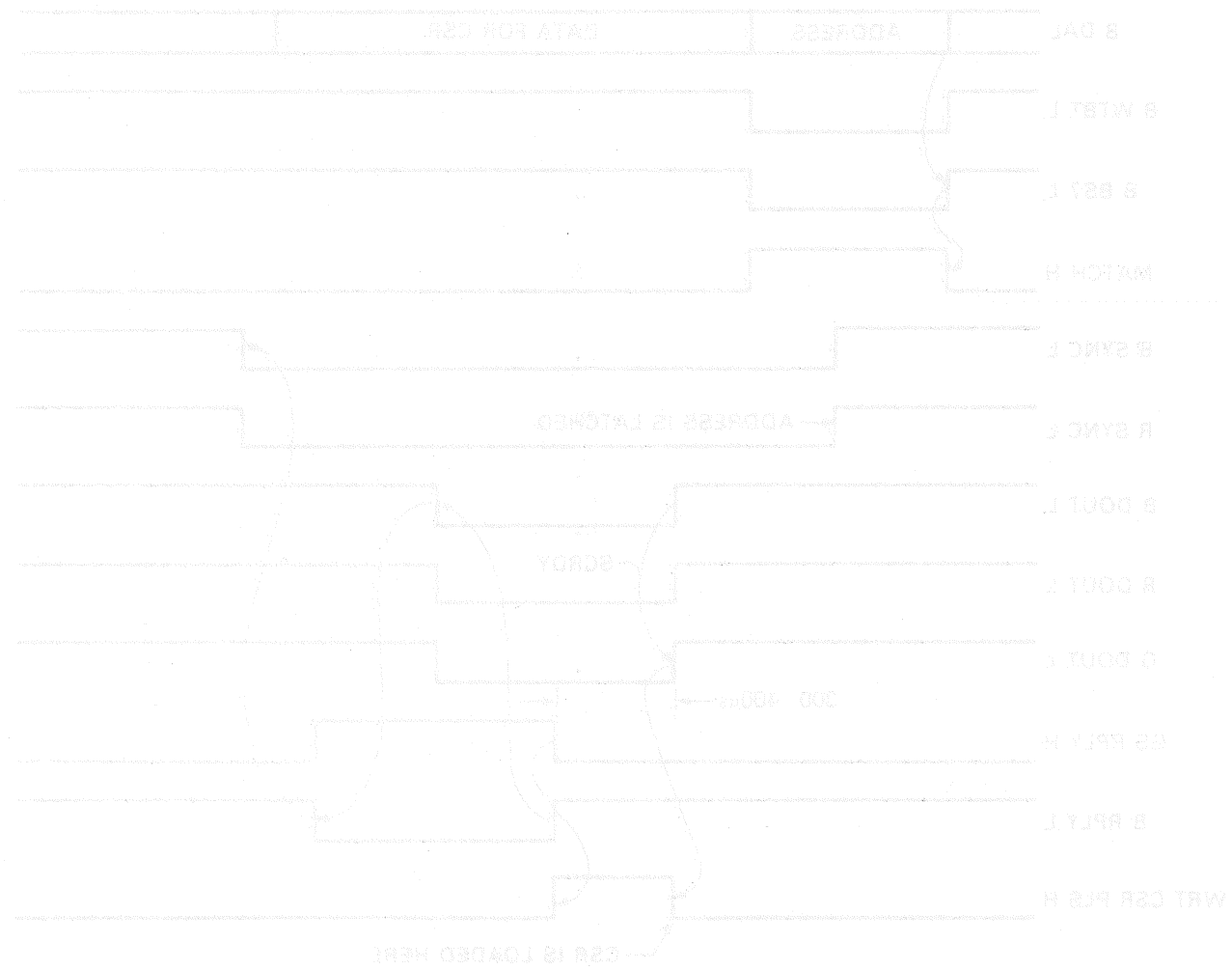
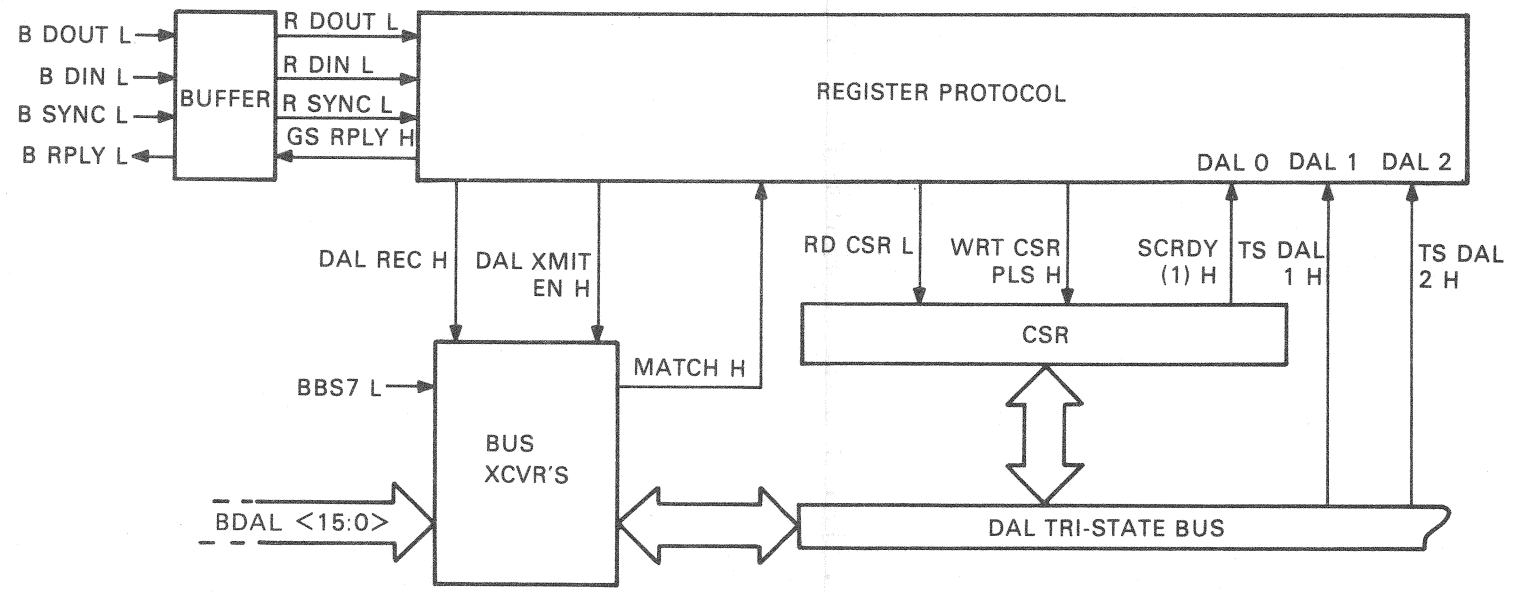
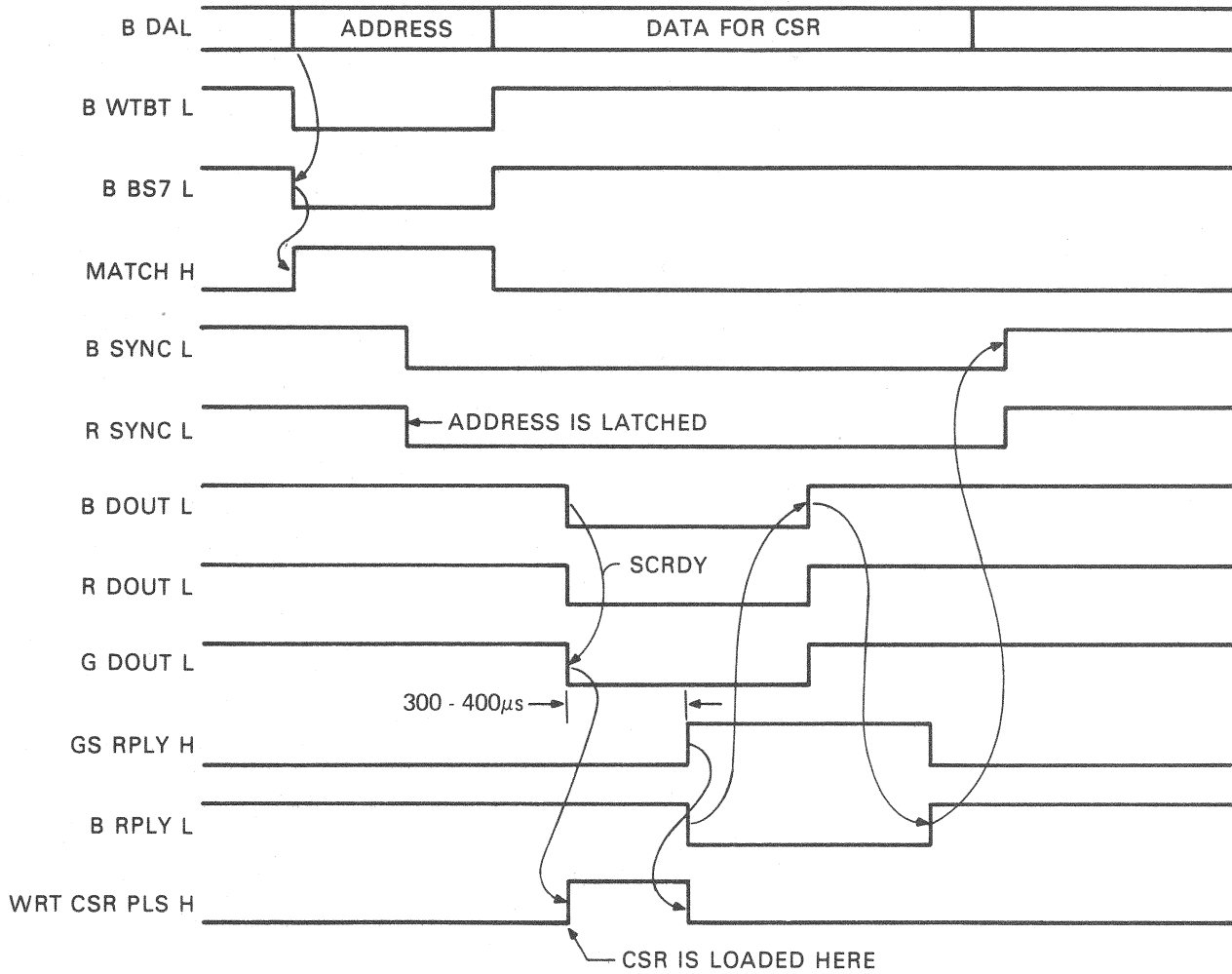


Figure 5-1 Bus Protocol Timing for Writing to CSR



MA-1120

Figure 5-2 CSR Read/Write Diagram



MA-1122

Figure 5-3 Bus Protocol Timing for Writing the CSR

Next the CPU places data onto the BDAL lines which appears at the CSR input via the tri-state DAL bus and asserts the B DOUT L signal. This signal arrives at the protocol chip in the form R DOUT L and is reissued as G DOUT L if the controller is ready. The G DOUT L signal generates the WRT CSR PLS H which loads the CSR with the data on the tri-state bus. When the RLV11 is slave, the BDAL transceivers are always receiving except during DIN of a data in transfer. After a short timeout period, the GS RPLY L signal is generated by the protocol circuit. The GS BPLY L signal terminates the WRT CSR PLS H and the dialogue on the LSI-11 bus. If the controller is not ready when the RDOUT signal arrives, the WRT CSR PLS is not generated but GS RPLY is generated to complete the bus protocol.

5.2.2.2 Reading a Register – The sequence for reading the contents of the CSR is very similar to that for writing the CSR. The MATCH H signal is generated by the bus transceivers in the same manner as before. When the CPU asserts the B SYNC L signal, the register address is latched by the protocol chip. The first major difference between writing and reading the CSR occurs now.

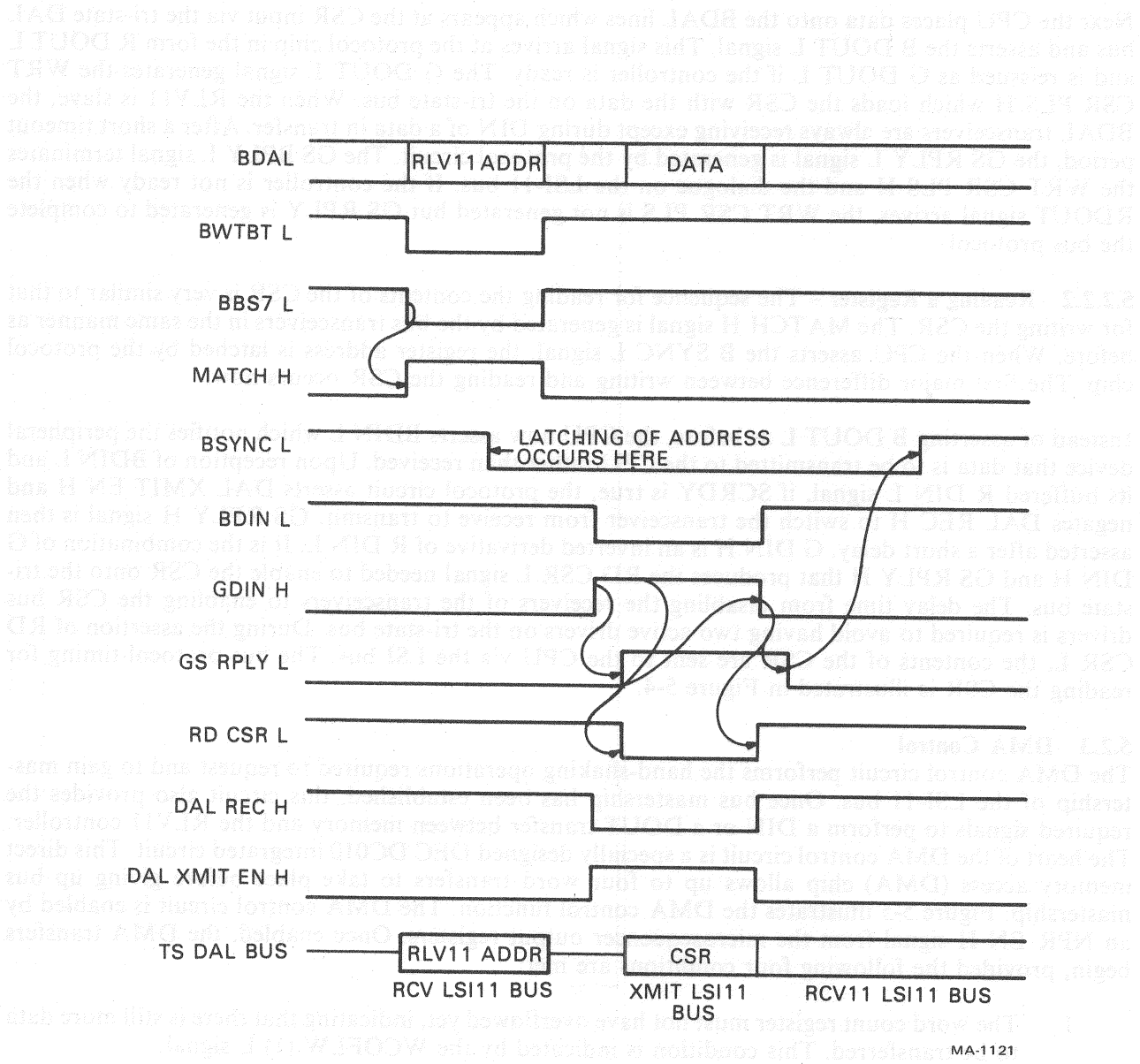
Instead of asserting B DOUT L as before, the CPU now asserts BDIN L which notifies the peripheral device that data is to be transmitted to the CPU rather than received. Upon reception of BDIN L and its buffered R DIN L signal, if SCRDI is true, the protocol circuit asserts DAL XMIT EN H and negates DAL REC H to switch the transceiver from receive to transmit. GS RPLY H signal is then asserted after a short delay. G DIN H is an inverted derivative of R DIN L. It is the combination of G DIN H and GS RPLY H that produces the RD CSR L signal needed to enable the CSR onto the tri-state bus. The delay time from disabling the receivers of the transceivers to enabling the CSR bus drivers is required to avoid having two active drivers on the tri-state bus. During the assertion of RD CSR L, the contents of the CSR are sent to the CPU via the LSI bus. The bus protocol timing for reading the CSR is illustrated in Figure 5-4.

5.2.3 DMA Control

The DMA control circuit performs the hand-shaking operations required to request and to gain mastership of the LSI-11 bus. Once bus mastership has been established, this circuit also provides the required signals to perform a DIN or a DOUT transfer between memory and the RLV11 controller. The heart of the DMA control circuit is a specially designed DEC DC010 integrated circuit. This direct memory access (DMA) chip allows up to four word transfers to take place before giving up bus mastership. Figure 5-5 illustrates the DMA control function. The DMA control circuit is enabled by an NPR EN H signal from the microsequencer output registers. Once enabled, the DMA transfers begin, provided the following four conditions are met.

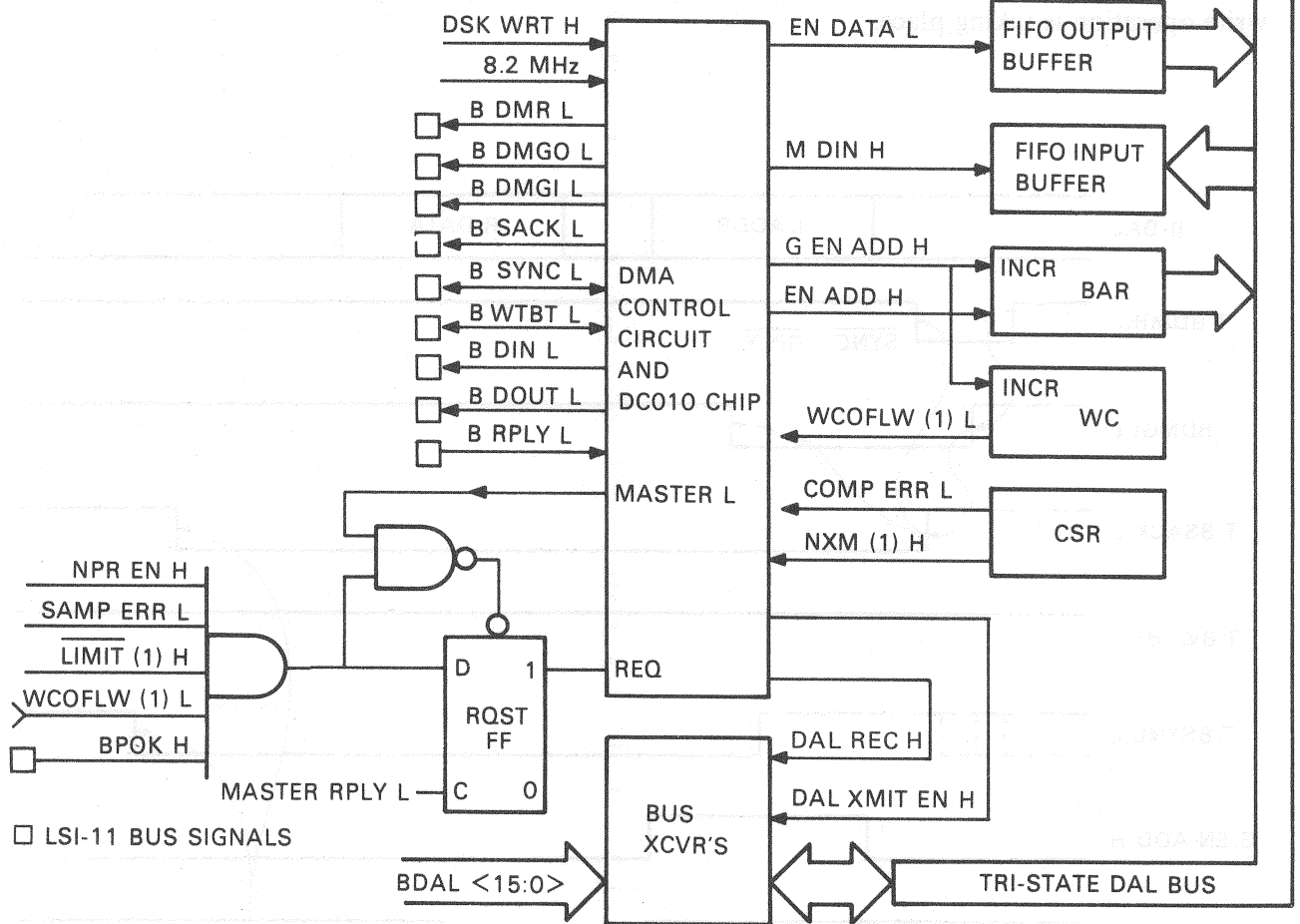
1. The word count register must not have overflowed yet, indicating that there is still more data to be transferred. This condition is indicated by the WCOFLW (1) L signal.
2. The FIFO must not have reached any of its limits. During a disk write operation, the FIFO reaches its limit when it is full and cannot accept any more DMA transfers. During a disk read operation, the FIFO reaches its limit when it is empty with no more words to transfer. The limit condition is indicated by the LIMIT (1) L signal.
3. Any error condition in the CSR causes the composite error bit to set and to disable the DMA transfers. Thus, the COMP ERR H signal indicates proper conditions for DMA transfers to occur.
4. Finally, the BPOK H signal is an indication that the ac power is within tolerances.

If the above four conditions are met and the DMA control circuit receives an NPR EN H, mastership can be acquired and the actual word transfers to or from memory can occur.



MA-1121

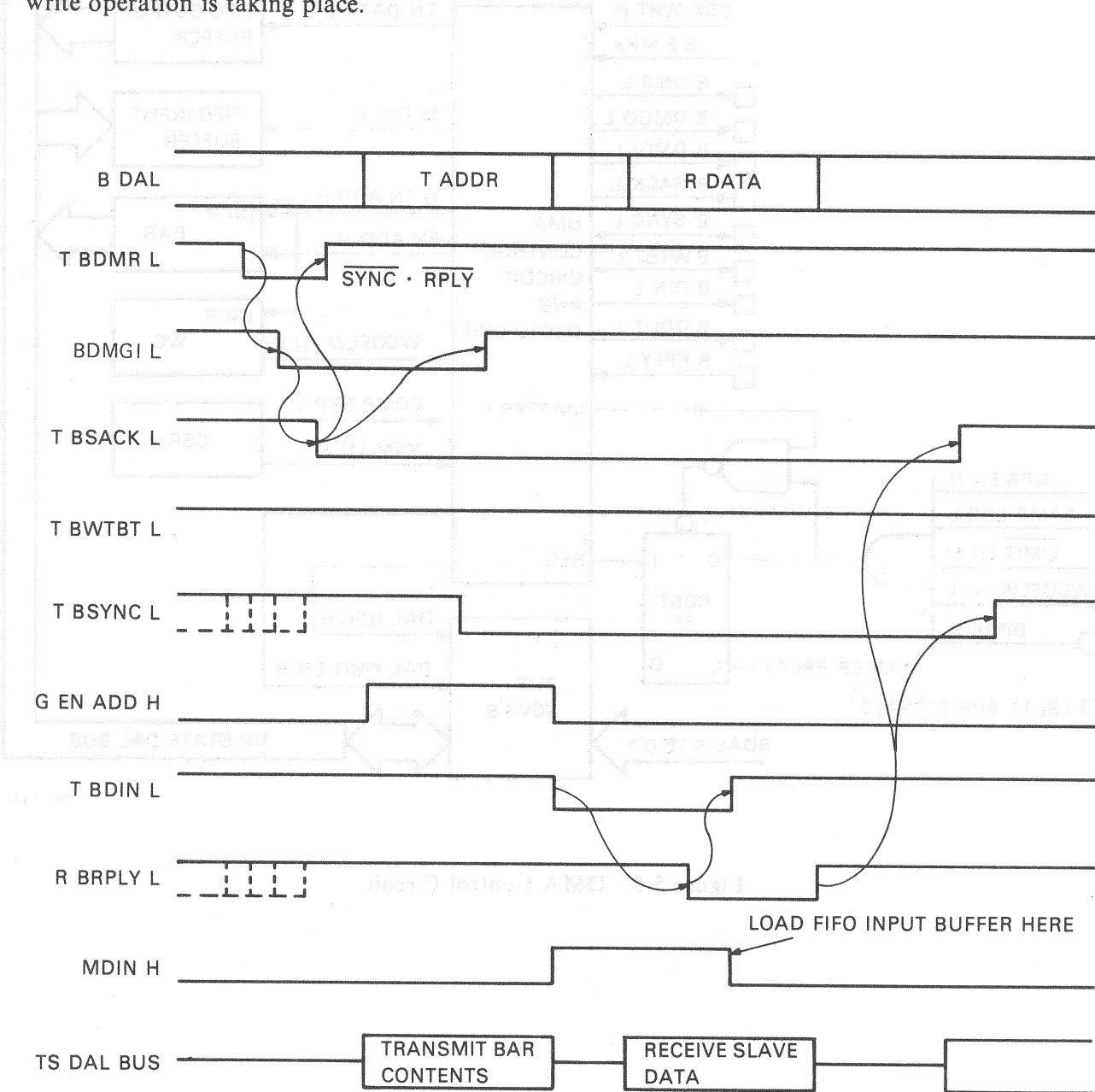
Figure 5-4 Protocol Timing for Reading the CSR



MA-1131

Figure 5-5 DMA Control Circuit

The sequence of events that occur in gaining bus mastership and transferring data is shown in Figure 5-6. To become bus master, the RLV11 controller asserts a direct memory access request (DMR L). The CPU responds by asserting DMA grant-output (DMGO L) which passes through daisy-chained DMA devices closer to the CPU and arrives at BDMGI of the RLV11. The controller answers to DMGI L by waiting for negation of SYNC and RPLY, then asserting BSACK L indicating that it is bus master. From this point on, DATI or DATO cycles may be executed depending on whether a disk read or disk write operation is taking place.



NOTE
RLV11 IS BUS MASTER

MA-1140

Figure 5-6 Single Word DMA Transfer to FIFO

The DMA circuit transmits the address of the memory location (contents of the bus address register) by asserting EN ADD H. If a data word is to be written into the FIFO input buffer, the DMA circuit asserts M DIN H which gates data from the LSI-11 bus onto the tri-state DAL bus. Data is taken at the trailing edge of DIN. If a data word is to be read out of the FIFO output buffer, the DMA circuit asserts EN DATA L and DOUT.

The DC010 integrated circuit maintains a count of how many transfers are made per bus mastership. It allows no more than four words to be transferred at a time. Multiple word transfers occur only if the FIFO has not reached its limit, i.e., if filling the FIFO and it is not yet full or emptying the FIFO and it is not yet empty. After dropping mastership, a timeout of 4 μ s occurs before DMR is again asserted. This allows other DMA devices on the bus to become master between RLV11 bus masterships. The DMA transfer sequences are finally complete when the word counter register overflows and sets the WCOFLW (1) L signal.

5.2.4 Interrupt Control

At the completion of any of the eight functions, the RLV11 controller can notify the CPU that it is in need of service. It does this by means of an interrupt control circuit (Figure 5-7). The interrupt control makes use of a DC003 integrated circuit to generate the interrupt timing sequence (Figure 5-8).

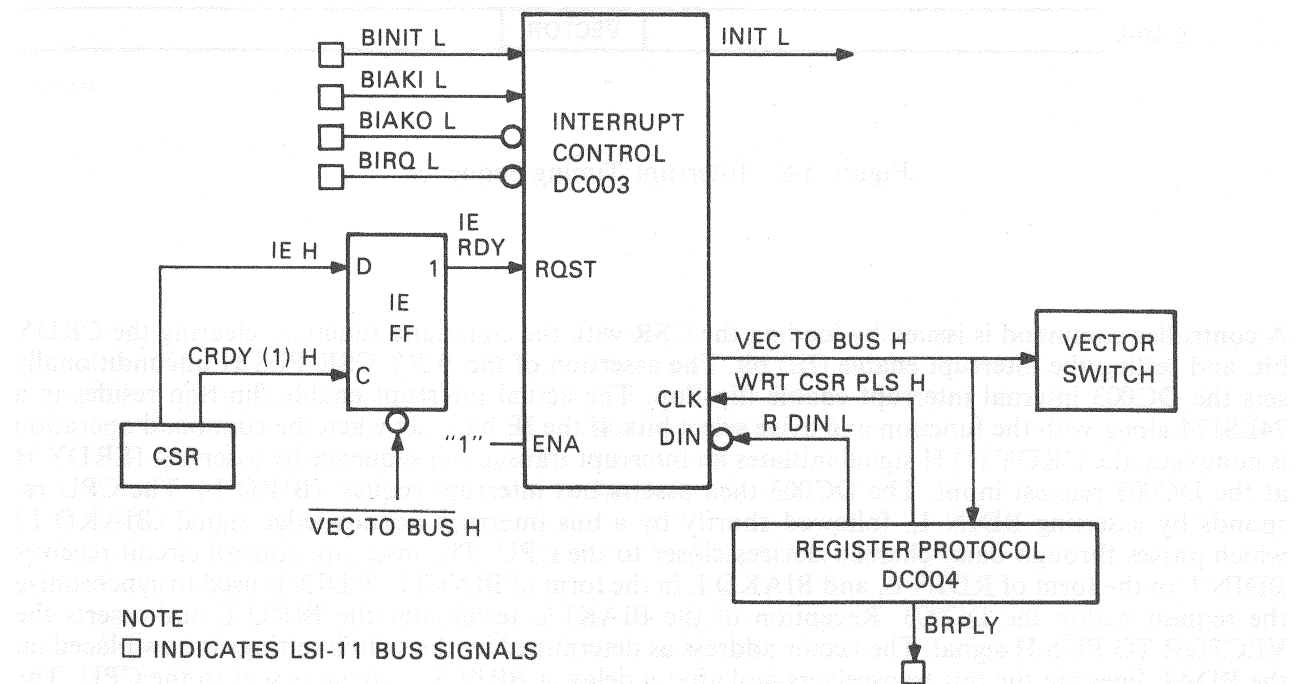
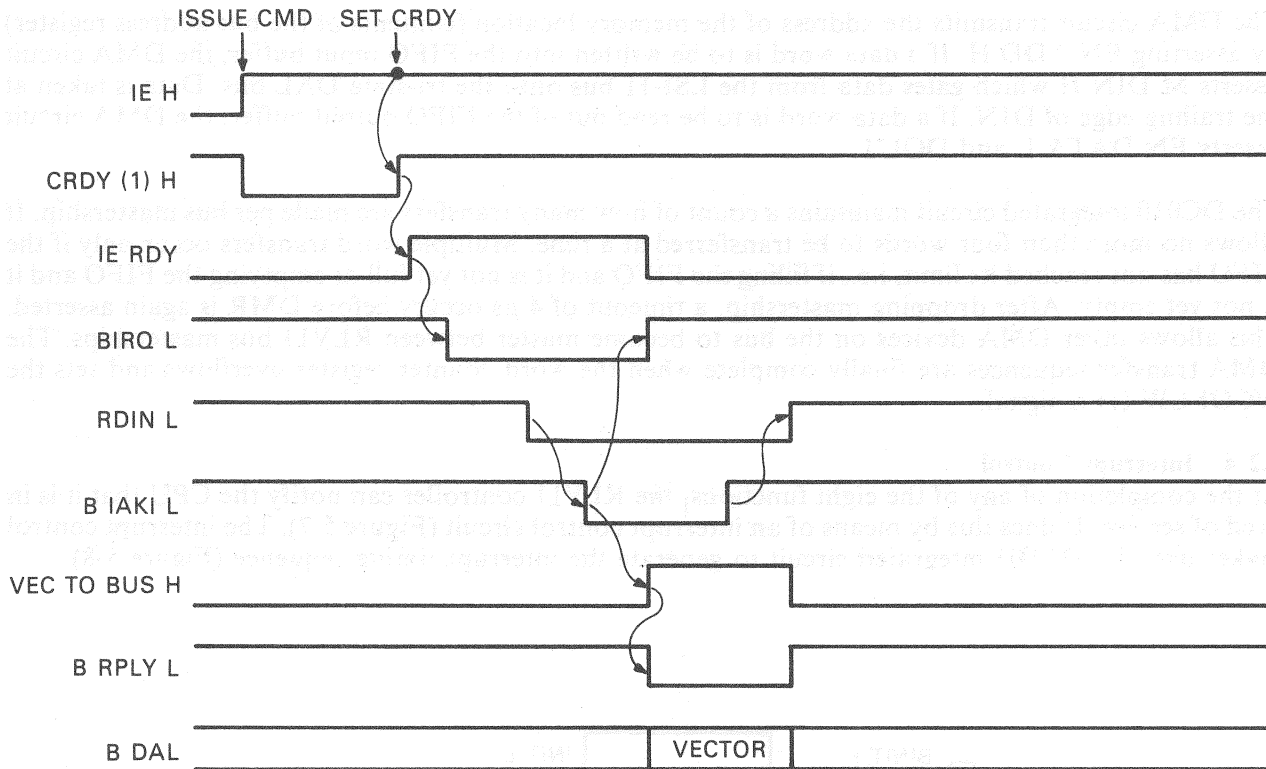


Figure 5-7 Interrupt Control Circuit



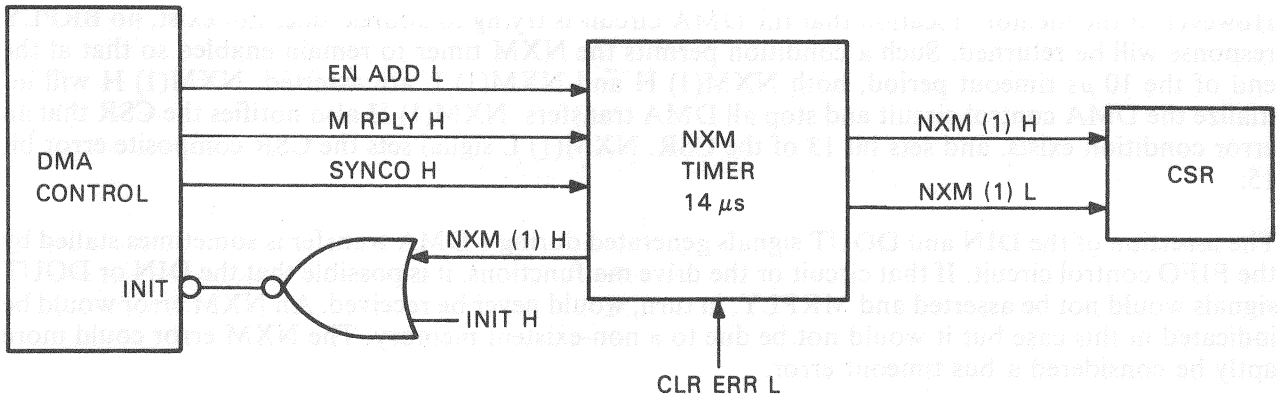
MA-1115

Figure 5-8 Interrupt Timing Sequence

A controller command is issued by loading the CSR with the command function, clearing the CRDY bit, and setting the interrupt enable (IE) bit. The assertion of the WRT CSR PLS H unconditionally sets the DC003 internal interrupt enable flip-flop. The actual interrupt enable flip-flop resides in a 74LS174 along with the function and drive select bits. If the IE bit is set when the command operation is complete, the CRDY (1) H signal initiates an interrupt transaction sequence by asserting IERDY H at the DC003 request input. The DC003 then asserts bus interrupt request (BIRQ L). The CPU responds by asserting BDIN L, followed shortly by a bus interrupt acknowledge signal (BIAKO L) which passes through daisy-chained devices closer to the CPU. The interrupt control circuit receives BDIN L in the form of RDIN L, and BIAKO L in the form of BIAKI L. RDIN is used to synchronize the request within the DC003. Reception of the BIAKI L terminates the BIRQ L and asserts the VECTOR TO BUS H signal. The vector address as determined by the switch settings is now placed on the BDAL lines via the bus transceivers and after a delay, a BRPLY L signal is sent to the CPU. The CPU then negates DIN and IAKO which in turn causes the RLV11 DC003 to negate VEC TO BUS and the DC004 to negate RPLY.

5.2.5 Non-Existent Memory (NXM) Timer Circuit

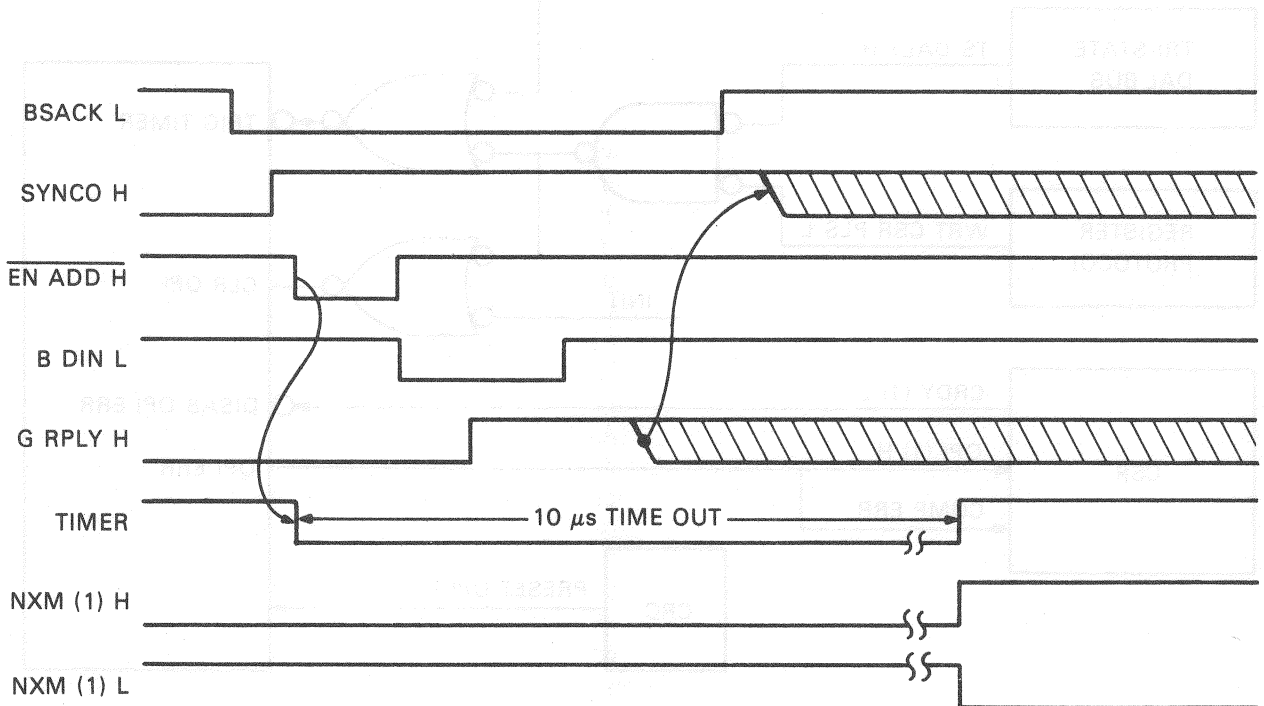
The non-existent memory circuit is used to detect when the controller is attempting to address a memory location that does not exist. Figure 5-9 shows how the NXM timer circuit relates to the CSR and DMA control circuit.



MA-1123

Figure 5-9 NXM Timer Circuit

The NXM timer sequence functions within the DMA transfer cycle. The actual NXM timing sequence is illustrated in Figure 5-10. The sequence begins during the DMA cycle with the reception of the SYNCO H and the EN ADD H signals by the NXM timer. The EN ADD H triggers a 10 μs one-shot timer. Next, the DMA circuit issues a DIN or DOUT signal to try to transfer data from or to a particular memory address. When the memory recognizes its address, it responds by asserting a BRPLY signal. This BRPLY signal is received by the NXM timer circuit in the form of GRPLY H, and will disable the NXM output signal.



MA-1114

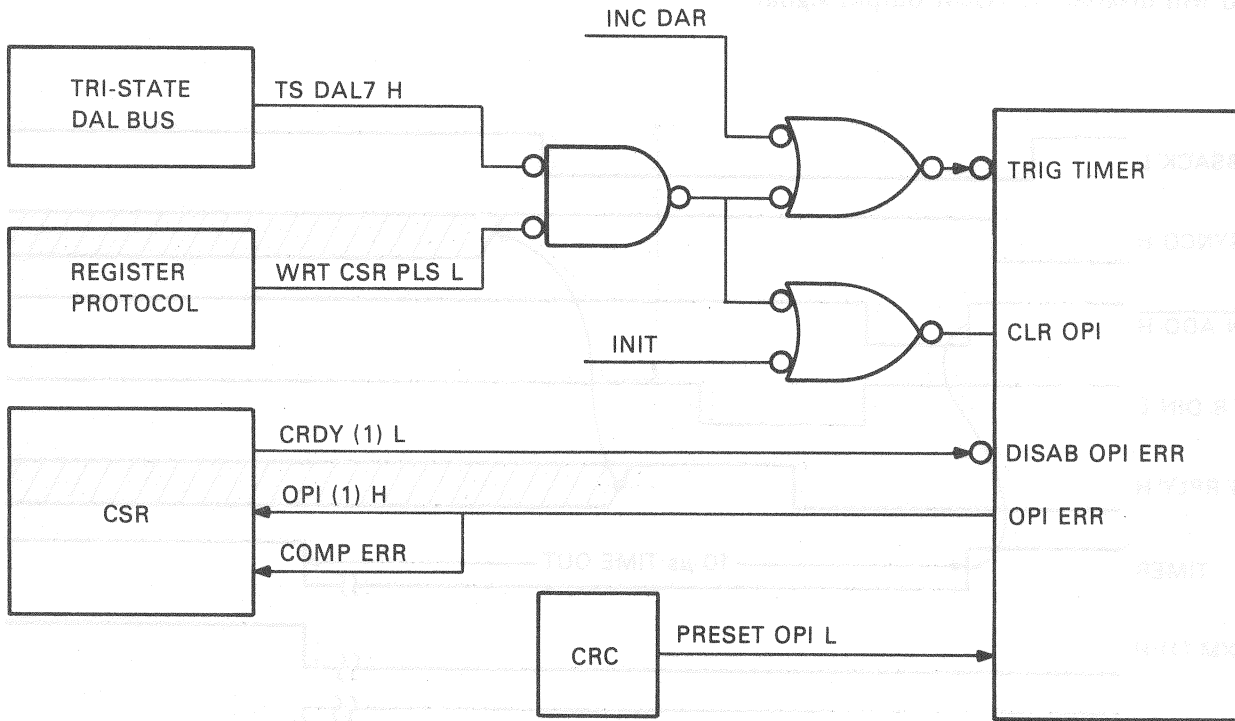
Figure 5-10 NXM Timing Sequence

However, if the memory location that the DMA circuit is trying to address does not exist, no BRPLY response will be returned. Such a condition permits the NXM timer to remain enabled so that at the end of the 10 μ s timeout period, both NXM(1) H and NXM(1) L are asserted. NXM(1) H will initialize the DMA control circuit and stop all DMA transfers. NXM(1) H also notifies the CSR that an error condition exists, and sets bit 13 of the CSR. NXM(1) L signal sets the CSR composite error bit 15.

The assertion of the DIN and DOUT signals generated during a DMA transfer is sometimes stalled by the FIFO control circuit. If that circuit or the drive malfunctions, it is possible that the DIN or DOUT signals would not be asserted and MRPLY, in turn, would never be received. An NXM error would be indicated in this case but it would not be due to a non-existent memory. The NXM error could more aptly be considered a bus timeout error.

5.2.6 Operation Incomplete (OPI) Timer Circuit

The OPI circuit was designed to provide a means of detecting if a controller command has been prevented from completing its operational sequence. The reason may be the lack of a drive response, excessive DMA competition, or a failure mode internal to the controller. Whatever the cause, the OPI timer provides a set time interval (490 ms, nominal) for a command sequence to complete its operation. The relationship of the OPI timer to other system functional blocks is illustrated in Figure 5-11. The timer is triggered by the WRT CSR PLS L signal once the TS DAL 7H is asserted low, indicating that CRDY is being cleared or INC DA PLS L to indicate that progress is being made. If the command sequence completes its operation within the timeout period, the microsequencer issues a SET CRDY L signal and disables the timer output.



MA-1141

Figure 5-11 OPI Timer Circuit

If the timer reaches the end of its timeout interval without receiving the SET CRDY L or COMP ERR signal, it asserts the OPI(1) H signal and sets the OPI error bit (10 in the CSR). The OPI(1) H signal line can be reset or cleared by INIT or CLR CRDY.

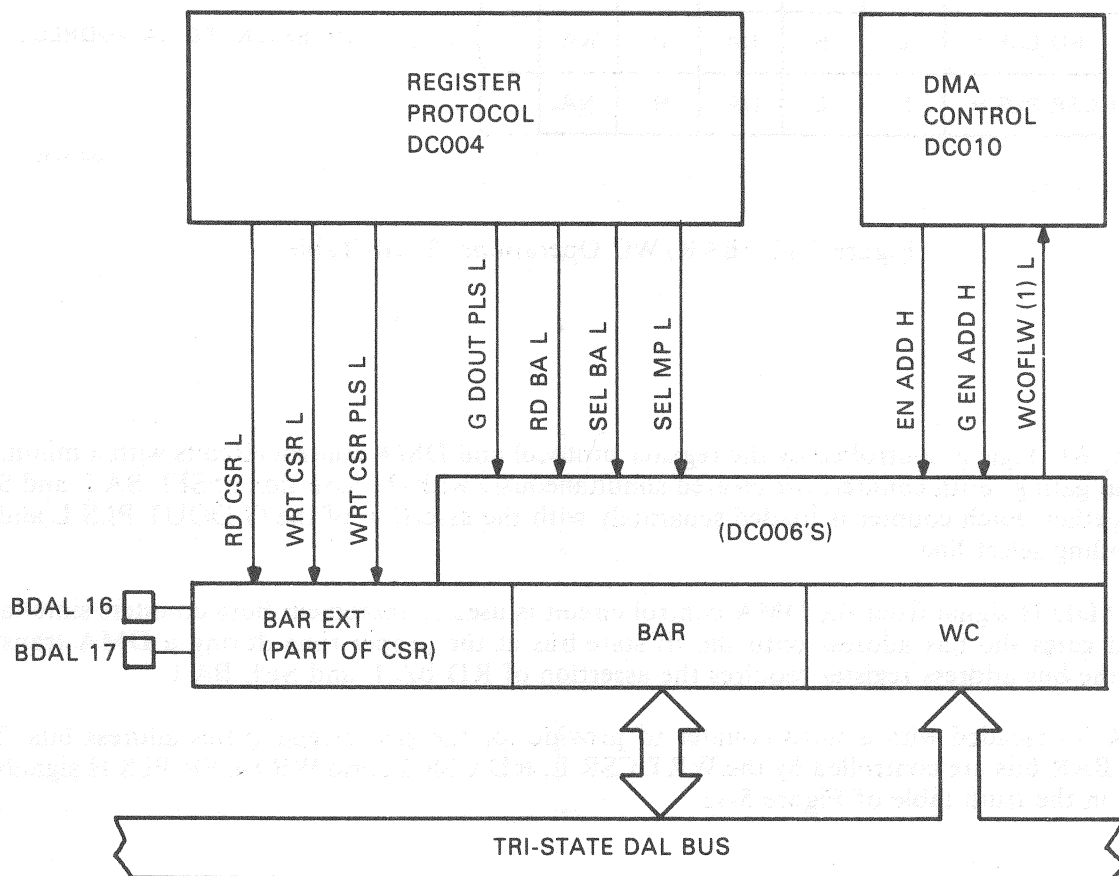
Note that the CRC circuit can also cause an OPI timeout error through the PRESET OPI L signal. This signal can override the normal timeout period and assert OPI(1) H immediately. This occurs when a header CRC error is encountered.

5.2.7 BAR/WC

The word counter (WC) is a 16-bit write-only register that controls the number of transfers to be made at a time. It is loaded with the 2's complement of the number of words to be transferred. The counter is incremented after each transfer and when it reaches 0, the WCOFLW (1) L signal is asserted.

The bus address register is a 16-bit read/write register used to generate the bus address that specifies the location to or from which data is to be transferred. The counter is incremented by 2 after each transfer. Bit 0 of the BAR is always written as a 0 so that the bus address is even, indicating word rather than byte transfers.

The BAR and WC are composed of two DEC-designed DC006 integrated circuits. Each BAR/WC IC has two 8-bit binary up-counters, one for word count and another for bus address. The two ICs are cascaded for full 16-bit implementation. The BAR/WC block diagram is shown in Figure 5-12. An operational truth table for the BAR/WC is provided in Figure 5-13.



MA-1142

Figure 5-12 BAR/WC Block Diagram

INPUT SIGNALS	FUNCTION				
	READ BAR AS REGISTER	LOAD BAR	LOAD WC	INCREMENT BAR/WC AND READ BAR FOR DMA	CLEAR BAR
EN ADD H	L	L	L		L
RD BA L	L	H	H	H	H
SEL BA L	L	L	H	H	
SEL MP L	H	H	L	H	
G DOUT PLS L	H			H	H
G EN ADD H	L	L	L		L
WRT CSR L	H	L	NA	H	NA
RD CSR L	L	H	NA	H	NA
WRT CSR PLS H	L	L	NA	H	NA

} INPUTS FOR EXTENDED BA ADDRESS

MA-1143

Figure 5-13 BAR/WC Operational Truth Table

The BAR/WC logic is controlled by the register protocol and DMA control circuits with a minimum of external gating. Both counters are cleared simultaneously with the assertion of SEL BA L and SEL MP L together. Each counter is loaded separately with the assertion of the G DOUT PLS L and its corresponding select line.

The EN ADD H signal from the DMA control circuit is used to increment both counters simultaneously and gates the bus address onto the tri-state bus at the proper time during a DMA transfer. Reading the bus address register requires the assertion of RD BA L and SEL BA L.

The BAR is cascaded with a third counter to provide for the two extended bus address bits. The extended BAR bits are controlled by the WRT CSR L, RD CSR L, and WRT CSR PLS H signals as indicated in the truth table of Figure 5-13.

5.2.8 DAR and Serializer

The disk address register (DAR) is a 16-bit read/write word-addressable register that is used to perform several functions. While writing or reading the disk, this register stores the sector, cylinder, and head address of the current sector. During a seek operation, the contents of the DAR is a difference address word that informs the drive how far and in which direction to move the read/write heads. When performing a get status command, the contents of the DAR notifies the drive to send back a drive status word.

The relationship of the DAR to other circuit functional blocks is shown in Figure 5-14. The DAR is composed of four programmable binary counters; these counters are loaded with a 16-bit word on the tri-state DAL bus by the assertion of WRT DA PLS L and WRT DA L.

To increment the contents of the DAR, the microsequencer asserts INC DA L. The DAR is incremented after each sector is read or written so that a header compare can be performed on the next sector. Note that the two least significant bits (DA0 H and DA1 H) of the DAR are sent to the microsequencer circuit. These are used by the sequencer circuit to ensure that the marker and get status bits are in the correct state for the function being performed.

The DAR contents must be serialized in order to move its contents to the drive, and also to perform header compares. The serialization is done via a 16:1 multiplexer. A binary count is applied to the four multiplexer selector inputs which enables the state of each of the 16 inputs to appear at the multiplexer output in correct sequence.

5.2.9 CSR and Error Bits

The control status register (CSR) is used primarily to issue commands to the controller and to monitor the status of various error messages. The CSR is a 16-bit register. Nine of the 16 bits can be read and written for control operations. The remaining seven bits are readable only and are used for monitoring errors and status. Refer to Paragraph 3.4 for a more detailed description of the CSR bits.

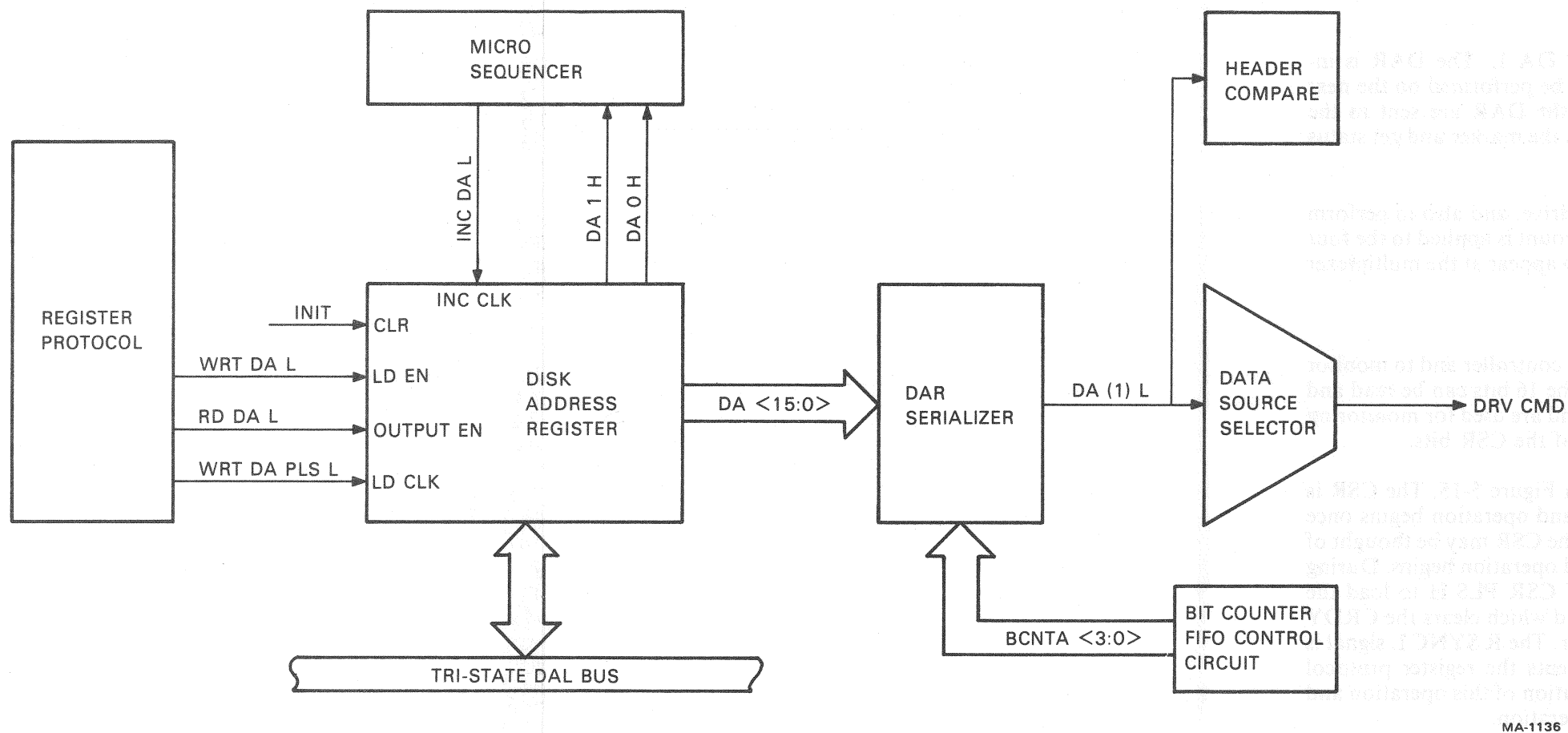
The relationship of the CSR to other circuit functional blocks is shown in Figure 5-15. The CSR is always the last register to be loaded when issuing a command. The command operation begins once the CSR bit 7 CRDY is written to a 0. The controller ready (CRDY) bit in the CSR may be thought of as a GO bit. When the CRDY bit is programmed low (state 0), the command operation begins. During the process of writing the CSR, the register protocol circuit asserts WRT CSR PLS H to load the writeable portion of the CSR. Simultaneously, WRT CSR PLS L is asserted which clears the CRDY flip-flop if TS bit 7 is low and initiates the OPI timer and the microsequencer. The R SYNC L signal is used to clock the SCR DY flip-flop which asserts SCR DY(1)L and prevents the register protocol circuit from recognizing any more DIN or DOUT CPU signals for the duration of this operation and ensures that bit 7 of the CSR is stable for the duration of an RD CSR operation.

The three function lines F<2:0>(1)H are used to address a unique instruction routine within the microsequencer ROM. The assertion of the CRDY(1)L signal enables the clocking for the sequencer routine to begin. The sequencer routine ends with the assertion of SET CRDY L which disables the OPI timer and sets the CRDY flip-flop. The setting of CRDY then may cause the interrupt circuit to interrupt the CPU.

The CSR monitors some drive signals directly from the drive bus transceivers. DRV RDY H, when asserted, indicates that the selected drive is locked on track. DRV ERR H, when asserted, indicates that the selected drive has flagged an internal error. The two drive select lines DRV SEL <1:0>(1)H are buffered directly to the drive bus and are used to select one of four drives that may be used.

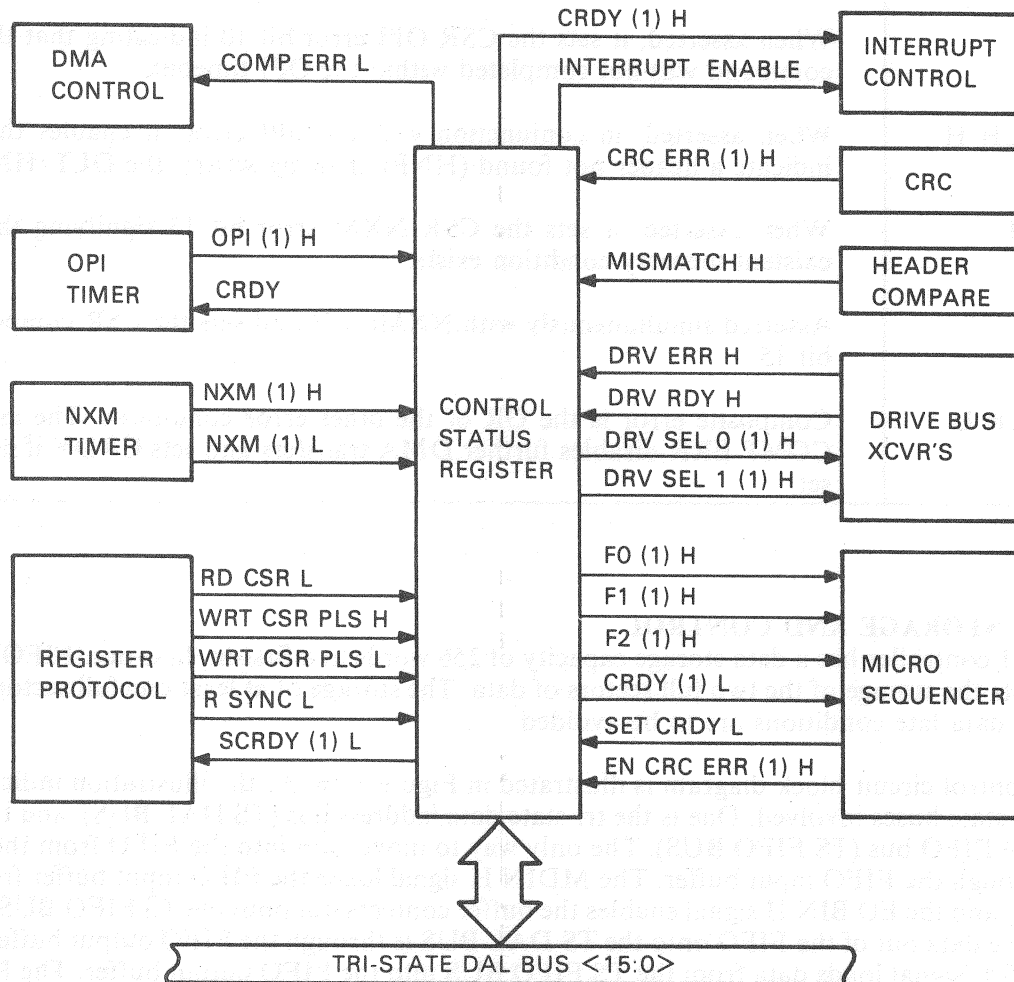
When interrupt enable bit 6 is set high, the CPU will be interrupted at the normal or error termination of a function. This bit will be set and cleared by the software writing the register.

The remaining signal lines to the CSR are all either directly or indirectly related to the error bit status. Their functions are described in Table 5-1.



MA-1136

Figure 5-14 DAR Block Diagram



MA-1133

Figure 5-15 CSR Block Diagram

Table 5-1 Error Status Related Signals

Signal	Function Description
CRC ERR(1) H	When asserted, it sets the CSR CRC error bit 11.
OPI(1) H	When asserted, it sets the CSR OPI error bit 10 indicating that the current command was not completed within the OPI timeout.
MISMATCH H	When asserted, in conjunction with an OPI error, it enables the CSR to indicate a header not found (HNF) error by setting the DLT/HNF bit 12.
NXM(1) H	When asserted, it sets the CSR NXM error bit 13 signifying that a non-existent memory condition exists.
NXM(1) L	Asserted simultaneously with NXM(1) H and sets the CSR composite error bit 15.
COMP ERR L	Composite error is the OR of the other error conditions. The assertion of COMP ERR disables further DMA transfers and sets CRDY if not already set.

5.3 FIFO STORAGE AND CONTROL

The RLV11 controller has a data storage capacity of 256 words in a first-in/first-out (FIFO) memory. This permits the storage of the two full sectors of data. The storage of at least one full sector of data is essential if data late conditions are to be avoided.

A FIFO control circuit block diagram is illustrated in Figure 5-16. As the illustration indicates, there are two tri-state buses involved. One is the tri-state data/address bus (TS DAL BUS), and the other is the tri-state FIFO bus (TS FIFO BUS). The only way to move data into the FIFO from the TS DAL BUS is through the FIFO input buffer. The MDIN H signal loads the FIFO input buffer from the TS DAL BUS, and the EO BIN H signal enables the buffer contents out onto the TS FIFO BUS. The only way to move data out of the FIFO onto the TS DAL BUS is through the FIFO output buffer. The LD BOUT PLS L signal loads data from the TS FIFO BUS into the FIFO output buffer. The EN DATA L and RD FIFO L signals are used to enable and read the data out of the buffer and onto the TS DAL BUS.

Another method of reading and writing the FIFO is through the FIFO serializer circuit. This is the data path that is used to read and write the disk. In the process of writing data onto the disk, parallel data must first be obtained from the FIFO and sent to the FIFO serializer via the TS FIFO BUS. The SR LD EN H and the BCNTA clock signals are used to load this data from the TS FIFO BUS into the serializer. Once in the serializer, the data is shifted out serially over the serial data out line.

When reading data from the disk, the data is first detected and separated by other circuit functions to be described later. This disk data is then shifted into the FIFO serializer over the serial data in line. Once inside the serializer, it can be read out onto the TS FIFO BUS by asserting the WRT RAM and DSK WRT signals. In this process the serial data is converted back into the parallel form and it is stored in the FIFO.

The register contents must be transferred to the RAM before the first bit of the next word arrives at the serial data input. The RAM has an address access delay that must be accommodated in the timing of the incoming word is held in the RAM for a 1-bit period. This extra time between words is used for the write-to-RAM transfer. From the second bit arrives and is clocked directly into the shift register while the first bit is simultaneously clocked from the flip-flop to the register. This alternating action continues for the rest of the bits in the word because of the constant design delay.

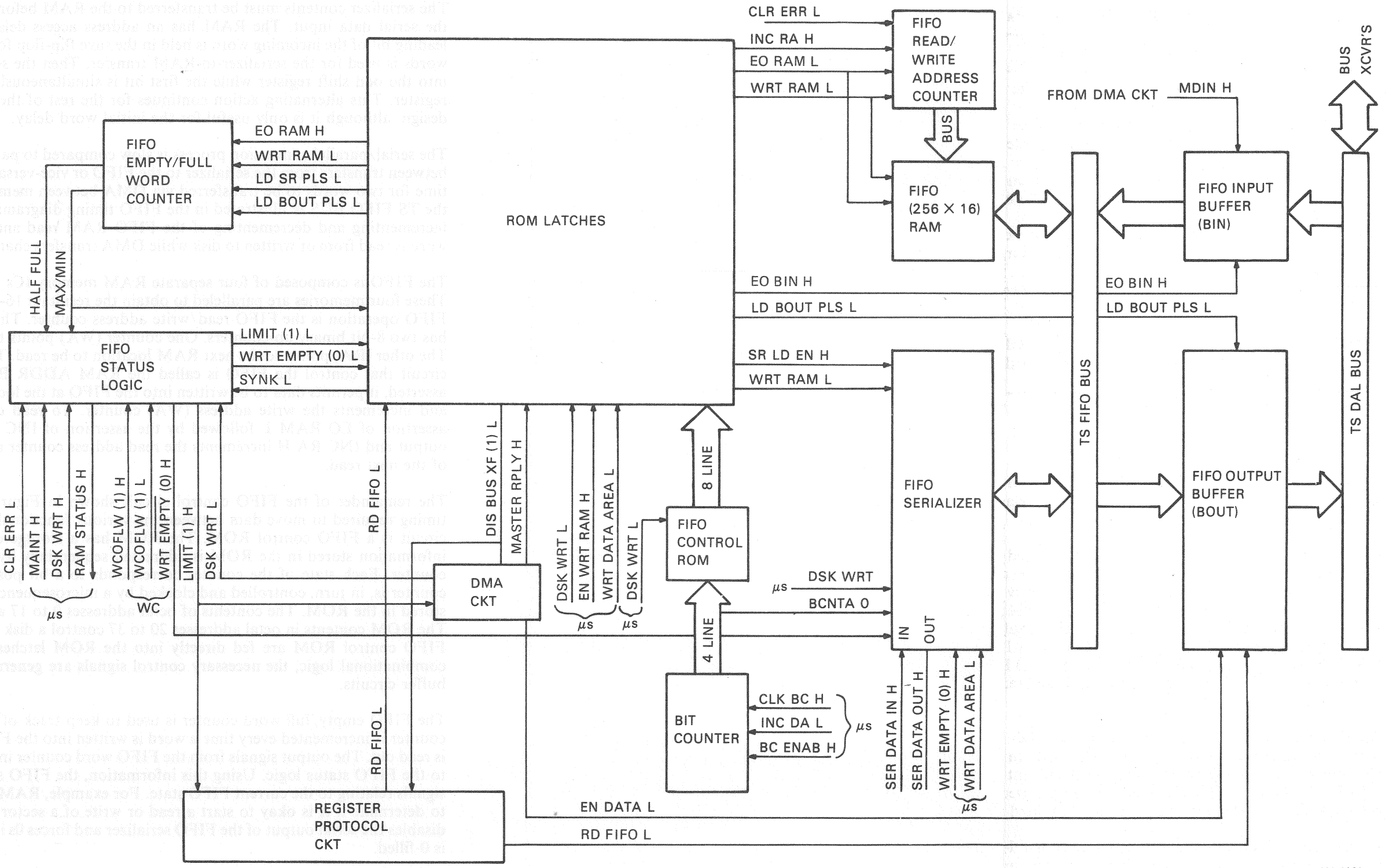
The shift register is compared to parallel transfer. Therefore, in the interval between the two words over the TS FIFO BUS, there is enough time for the RAM to transfer data to the RAM. This time-sharing of the TS FIFO BUS between the RAM and the FIFO. This time-sharing of the TS FIFO BUS between the RAM and the FIFO is shown in Figure 5-17 and 5-18. They show the timing and decoupling of the RAM and the FIFO. The timing of the RAM and the FIFO is shown in Figure 5-17 and 5-18. They show the timing and decoupling of the RAM and the FIFO.

The FIFO is composed of four separate RAMs. Each RAM is 256 words deep and 16 bits wide. The RAMs are used to store the data that is transferred from the RAM to the FIFO. The RAMs are used to store the data that is transferred from the RAM to the FIFO. The RAMs are used to store the data that is transferred from the RAM to the FIFO.

The timing of the RAM and the FIFO is shown in Figure 5-17 and 5-18. They show the timing and decoupling of the RAM and the FIFO. The timing of the RAM and the FIFO is shown in Figure 5-17 and 5-18. They show the timing and decoupling of the RAM and the FIFO.

The timing of the RAM and the FIFO is shown in Figure 5-17 and 5-18. They show the timing and decoupling of the RAM and the FIFO. The timing of the RAM and the FIFO is shown in Figure 5-17 and 5-18. They show the timing and decoupling of the RAM and the FIFO.

A more comprehensive understanding of how the FIFO and its control circuit operate during a data read cycle can be gained from the read FIFO timing diagram shown in Figure 5-17. For the data write FIFO timing, refer to Figure 5-18.



MA-1161

Figure 5-16 RLV11 FIFO Control Circuit

The serializer contents must be transferred to the RAM before the first bit of the next word arrives at the serial data input. The RAM has an address access delay that must be accommodated, so the leading bit of the incoming word is held in the save flip-flop for a 1-bit period. This extra time between words is used for the serializer-to-RAM transfer. Then the second bit arrives and is clocked directly into the odd shift register while the first bit is simultaneously clocked from the flip-flop to the even register. This alternating action continues for the rest of the bits in the word because of the circuit design, although it is only useful for the initial word delay.

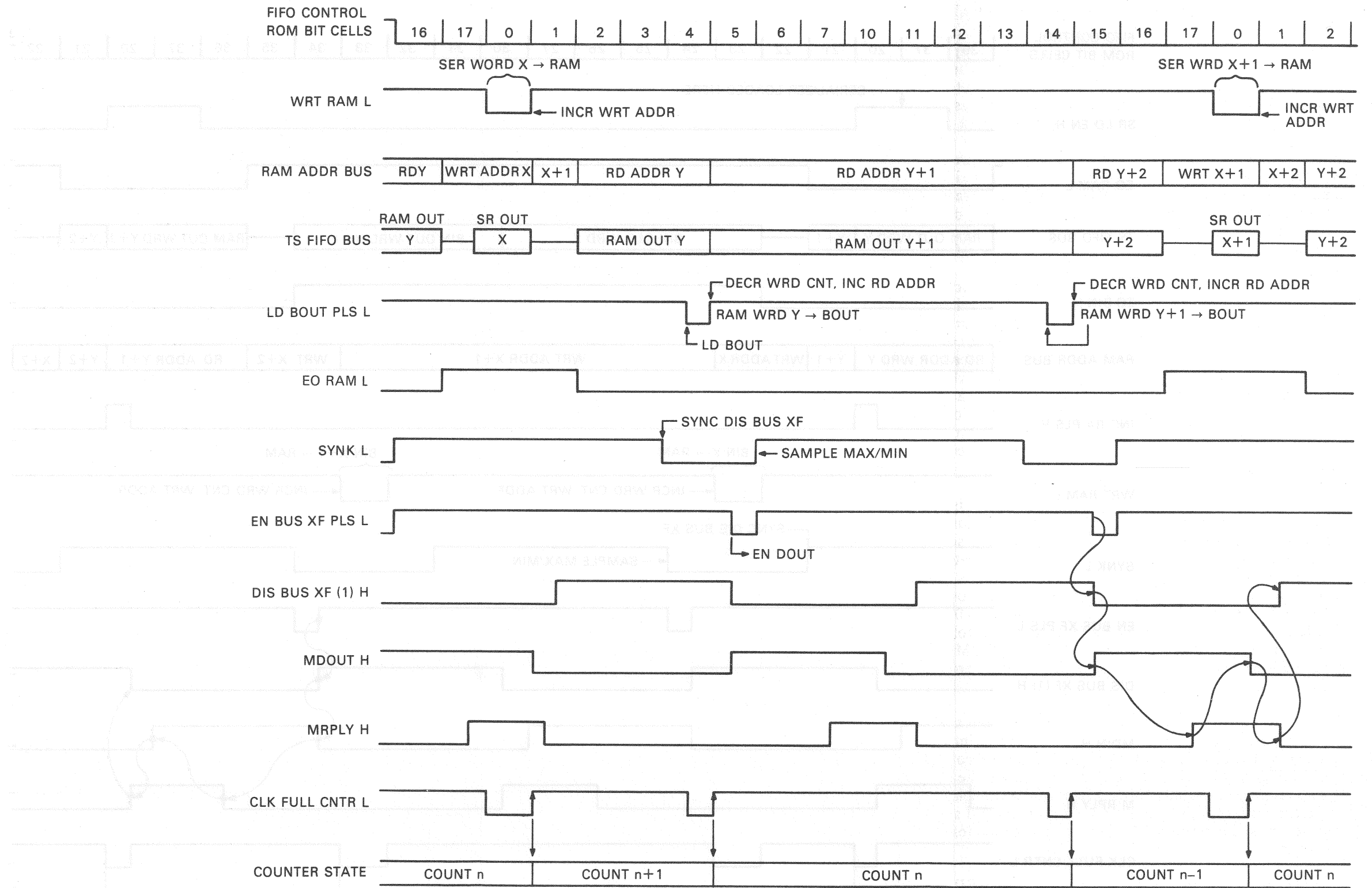
The serial/parallel conversion process is slow compared to parallel transfers. Therefore, in the interval between transfers from the serializer to the FIFO or vice-versa over the TS FIFO BUS, there is enough time for two words to be transferred via DMA between memory and the FIFO. This time-sharing of the TS FIFO BUS is illustrated in the FIFO timing diagrams, Figures 5-17 and 5-18. They show the incrementing and decrementing of the FIFO RAM read and write address and word counters as a word is read from or written to disk while DMA transfers change the count between serial transfers.

The FIFO is composed of four separate RAM memory ICs that have 256 locations for 4-bit words. These four memories are paralleled to obtain the resultant 16-bit by 256-word capacity. Integral to the FIFO operation is the FIFO read/write address counter. This circuit makes use of a DC006 IC that has two 8-bit binary up-counters. One counter (WA) points to the next RAM location to be written. The other (RA) points to the next RAM location to be read. The group of eight address lines from this circuit that control the FIFO is called the RAM ADDR BUS. When the WRT RAM L signal is asserted, it permits data to be written into the FIFO at the location specified by the RAM address bus and increments the write address (WA) counter. To read data out of the FIFO requires first the assertion of EO RAM L followed by the assertion of INC RA H. EO RAM L enables the RAM output and INC RA H increments the read address counter after the read is complete in anticipation of the next read.

The remainder of the FIFO control circuit shown in Figure 5-16 provides the control signals and timing required to move data between the various functional blocks. The heart of the FIFO control circuit is a FIFO control ROM. The ROM has a storage capacity of 32 8-bit words. The control information stored in the ROM is addressed sequentially by means of four address lines from a bit counter. Each state of the counter corresponds to a bit position in the serial data stream. The bit counter is, in turn, controlled and clocked by a microsequencer (μs). There are two control sequences stored in the ROM. The contents of octal addresses 0 to 17 are used to control a disk read sequence. The ROM contents in octal addresses 20 to 37 control a disk write sequence. The eight outputs of the FIFO control ROM are fed directly into the ROM latches. Here, through the use of some extra combinational logic, the necessary control signals are generated for manipulating the FIFO and its buffer circuits.

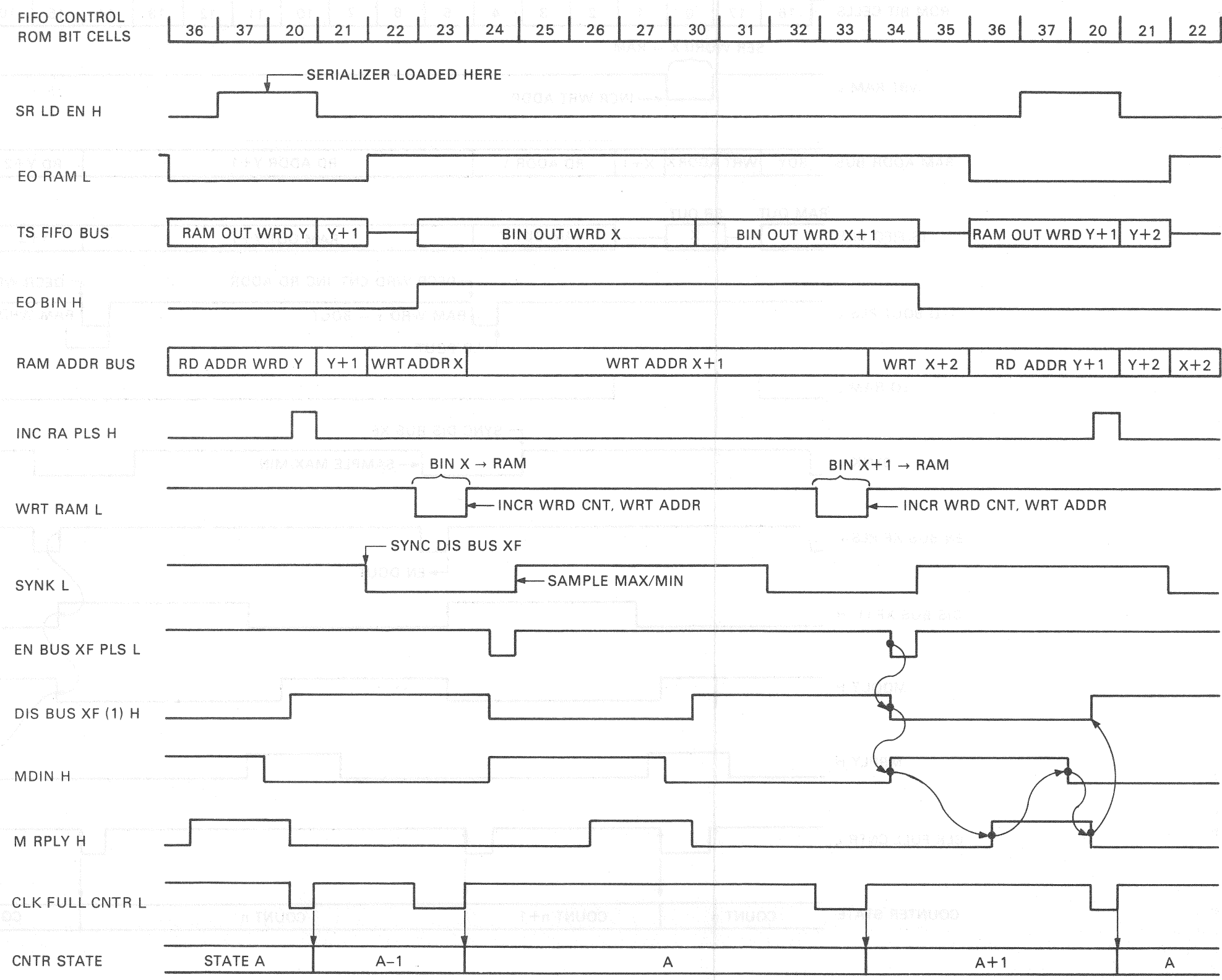
The FIFO empty/full word counter is used to keep track of how many words are in the FIFO. The counter is incremented every time a word is written into the FIFO and decremented every time a word is read out. The output signals from the FIFO word counter indicating full, half-full, and empty are fed to the FIFO status logic. Using this information, the FIFO status logic can generate various control signals relating to the current FIFO state. For example, RAM STATUS is used by the microsequencer to determine if it is okay to start a read or write of a sector. WRT EMPTY (0) H is the signal that disables the serial output of the FIFO serializer and forces 0s into the data stream when a partial sector is 0-filled.

A more comprehensive understanding of how the FIFO and its control circuit operate during a disk read cycle can be gained from the disk read FIFO timing diagram shown in Figure 5-17. For the disk write FIFO timing, refer to Figure 5-18.



MA-1148

Figure 5-17 Disk Read FIFO Timing



MA-1151

Figure 5-18 Disk Write FIFO Timing

5.4 MICROSEQUENCER

The internal operations of the RLV11 controller are under the control of the microsequencer. At the heart of the microsequencer is the sequencer ROM. The ROM contains all the preprogrammed instruction routines to control the eight different command operations. Figure 5-19 shows all the functional elements that are incorporated in the microsequencer circuit.

5.4.1 Clock Select 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 0 H), and the drive status clock (STATUS CLK H). The data separator clock is used whenever data is being read off the disk to keep sequencer timing in sync with clock data. The system clock is used by the microsequencer to generate timing when the data separator clock and the status clock are not in use. It provides the accurate frequency source required for writing data on the disk. The drive status clock is sent over from the drive during the transmission of a drive status word. The outputs of the clock select circuit are the CLK BC L and CLK BC H signals. CLK BC L is used to clock the bit counter, the CRDY synchronizer, the sequencer word counter, the condition synchronizer, and the wait and branch control circuits.

5.4.2 Branch Condition Selector

The branch condition selector is used to select one of eight different branch conditions. Which of these eight conditions is to be monitored is selected under the control of the COND <2:0> H lines, which are sequencer outputs. The sequencer has the ability to branch if the selected condition is true or to wait for the condition to become true. An example of this is the sector pulse condition (SEC PLS 0 H). During a read header command, the controller must wait for the arrival of the sector pulse before it can start reading the header. The eight branch conditions that are monitored are:

1. CRC ERR (0) H
2. DRIVE READY H
3. SEC PULSE (0) H
4. MISMATCH H
5. RAM STATUS H
6. WCOFLW (1) H
7. DA 0 H (disk address register)
8. DA 1 H

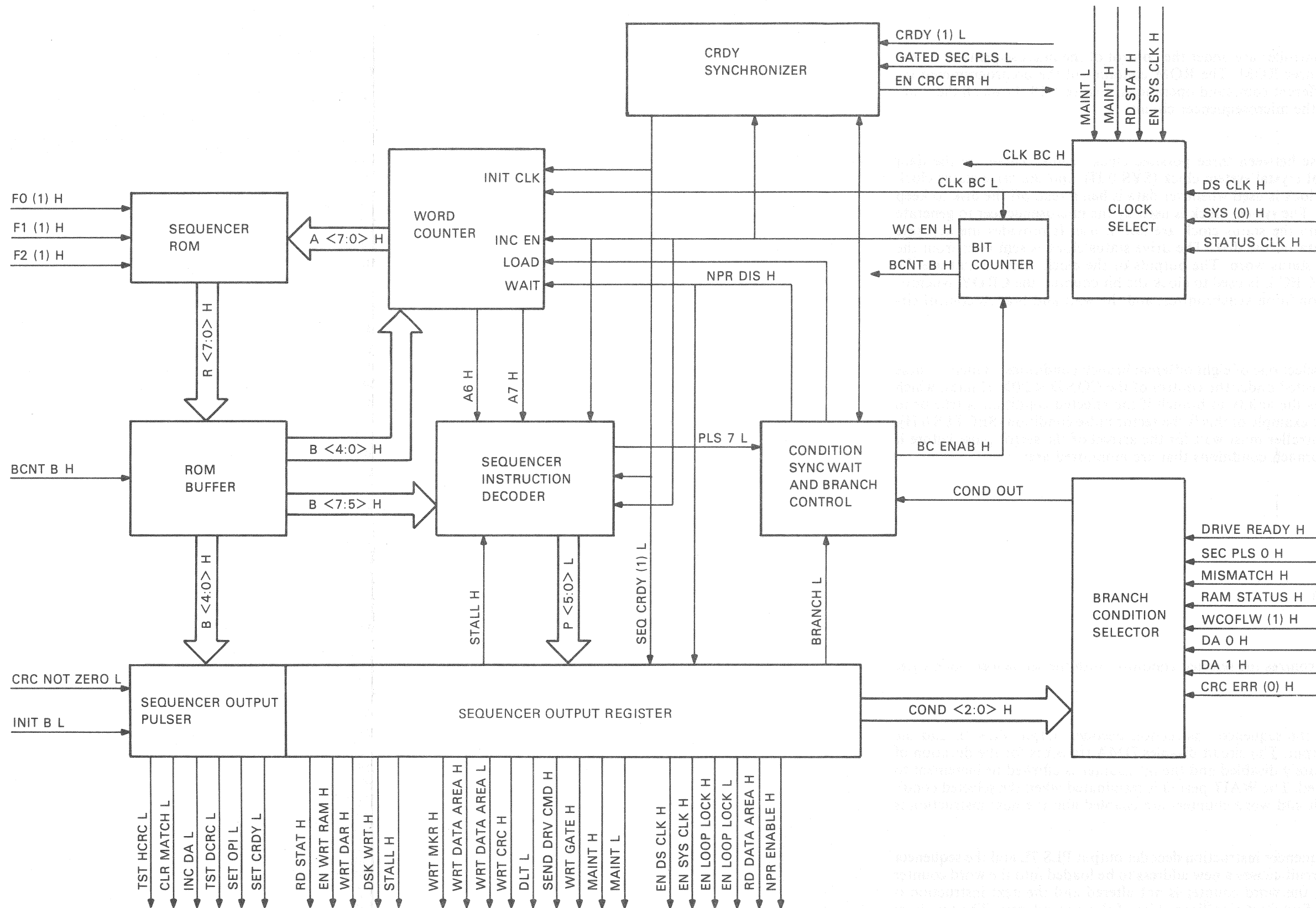
5.4.3 Condition Synchronizer

The condition synchronizer circuit synchronizes the selected condition with the sequencer clock CLK BC L.

5.4.4 Wait and Branch Control Circuit

The wait control circuit is activated by the sequencer instruction decoder output PLS 7L and the negation of the sequencer BRANCH output. The circuit disables DMA transfers for the duration of the WAIT. The word counter is immediately disabled and the bit counter is allowed to increment to the overflow condition and is then disabled. The WAIT period is terminated when the selected condition becomes true. At that time, both bit and word counters are enabled and the next instruction is immediately executed.

The branch control is activated by the sequencer instruction decoder output PLS 7L and the sequencer output BRANCH being asserted. This circuit causes a new address to be loaded into the word counter if the selected condition is true. If false, the word counter is not altered and the next instruction is executed. The ROM buffer supplies the five most significant bits of the new address. The two least significant bits are 0. Thus, the sequencer is capable of branching conditionally to any address that is a multiple of 4 within the 200 octal locations of the current function.



MA-1149

Figure 5-19 Microsequencer Detailed Block Diagram

5.4.5 CRDY Synchronizer

This circuit monitors the controller ready state. If the controller is in the ready state, it clears and disables the word counter. When the controller changes from ready to unready, this circuit ensures that the word count is 0 and that the ROM outputs are settled before a micro-instruction is executed.

5.4.6 Word Counter

The word counter sequences the ROM through its routines by providing successive address locations. Each command routine is allowed 200 octal locations.

5.4.7 Sequencer ROM

The sequencer ROM is a read-only memory used to store the controller micro-instructions. The micro-programs for all eight controller commands were designed to fit within a 1K by 8 memory space. Which 200 word block is being addressed at a given moment is determined by the three function signals (F<2:0>(1)H). Originating in the control status register, they select the starting address in multiples of 200 octal.

5.4.8 ROM Buffer

The output instructions from the sequencer ROM are fed into the ROM buffer. The buffer latches the instructions upon command from the bit counter. Of the eight output lines from the buffer, the three most significant bits are sent to the instruction decoder circuit. The remaining instruction lines are used to set up data at a group of five output registers and one pulse generator. Their outputs provide the majority of the control signals for controller operations.

5.4.9 Instruction Decoder

The major function of the instruction decoder is to generate eight instruction pulses. One of the pulses causes the conditional BRANCH or WAIT instruction to be executed. Five of the pulses load five separate output registers. One pulse activates a pulser that generates one or eight pulses (two unused), depending on the ROM buffer outputs B<3:1> H.

5.4.10 Sequencer Stall Instruction

The sequencer has the ability to stall the execution of micro-instructions for a specific number of word times. This is useful for counting the number of words being read or written in a sector without executing micro-instructions. By asserting the STALL output of the sequencer, the execution of micro-instructions is inhibited until the word counter is incremented to octal 300. Since the ROM does not see the most significant bit, it starts executing instructions beginning at octal 100 for that function.

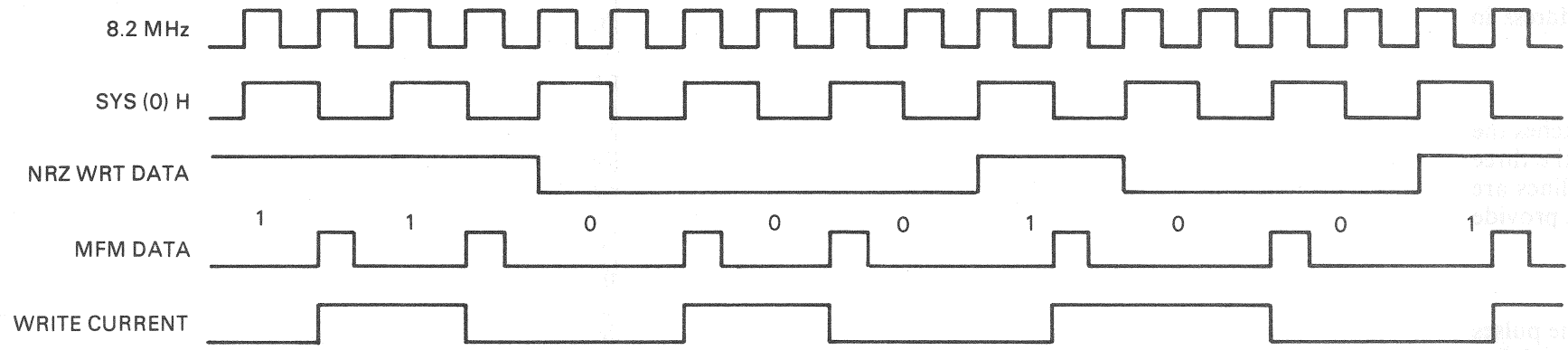
As an example, presume the STALL was asserted at word count 70. The word counter continues to increment from octal 71 to octal 277. However, the micro-instructions stored in these corresponding ROM locations (octal 70-177, 0-77) are not executed. When the word count reaches octal 300, the stall ends and the micro-instructions located at ROM location 100 within the function is executed. Thus, execution has been stalled for octal 207 word times.

5.5 WRITE ENCODE AND PRECOMPENSATION CIRCUIT

The write encode and precompensation circuit performs two major functions in the RLV11 Controller. First, it converts serial digital data into a modified frequency modulation (MFM) format. Second, it time shifts the MFM data to precompensate for magnetic peak shift phenomena on the disk.

5.5.1 MFM Encoding

With the MFM encoding technique, each logical 1 produces a flux reversal in the center of its bit cell. Two successive logical 0s produce a flux reversal at the bit cell boundary between them. This recording technique guarantees that there will be at least one flux reversal on the disk for every two bit cells. Figure 5-20 illustrates the MFM encoding scheme.



MA-1159

Figure 5-20 MFM Encoding

5.5.2 Peak Shift Phenomenon

The RL01 disk drive, like all high-density recording systems, experiences the phenomenon of peak shifting. This arises from the fact that in reading data off the disk, the voltage peaks resulting from detection of the recorded flux reversals appear to be shifted relative to their theoretical placement on the track. The practical effect of this is a reduced margin of data recovery reliability. Figure 5-21 illustrates the cause of peak shift. The illustration shows the MFM data and the flux reversal pattern that it generates on the disk track. Part d of Figure 5-21 shows the idealized read signal voltage pulses that might occur if this were a perfect system where the flux field could change instantaneously. In practical situations, flux fields, air gaps, magnetic heads, coil windings, and amplifiers all contribute to a distorting effect on these ideal pulses. Part e of Figure 5-21 shows a more realistic shape for individual read pulse waveforms. When the contributions of each individual pulse are summed into one composite waveform, the result is as shown in Part f. The peaks of the composite read voltage waveform have shifted. Notice that the direction of peak shift can be correlated with the flux reversal pulses above in part c of Figure 5-21. As a general rule, it may be stated that two adjacent flux reversals appear to repel each other. The degree of shift is a function of the proximity of the two reversals and of their environment, that is, the location of the pair relative to other flux reversals in the sequence.

5.5.3 Write Precompensation Circuit

As was mentioned earlier, the write precompensation circuit has the function of time shifting the data pulses to precompensate for the effect of peak shift. Figure 5-22 shows the precompensation circuit detailed block diagram.

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 serial data into MFM data. The bit pattern from the shift register that addresses the ROM will determine the ROM output. This output does two things. Two bits 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 0s. A center pulse is generated to encode a logical 1.) 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 20 ns. This is the time shifting that accomplishes the precompensation for peak shift phenomenon.

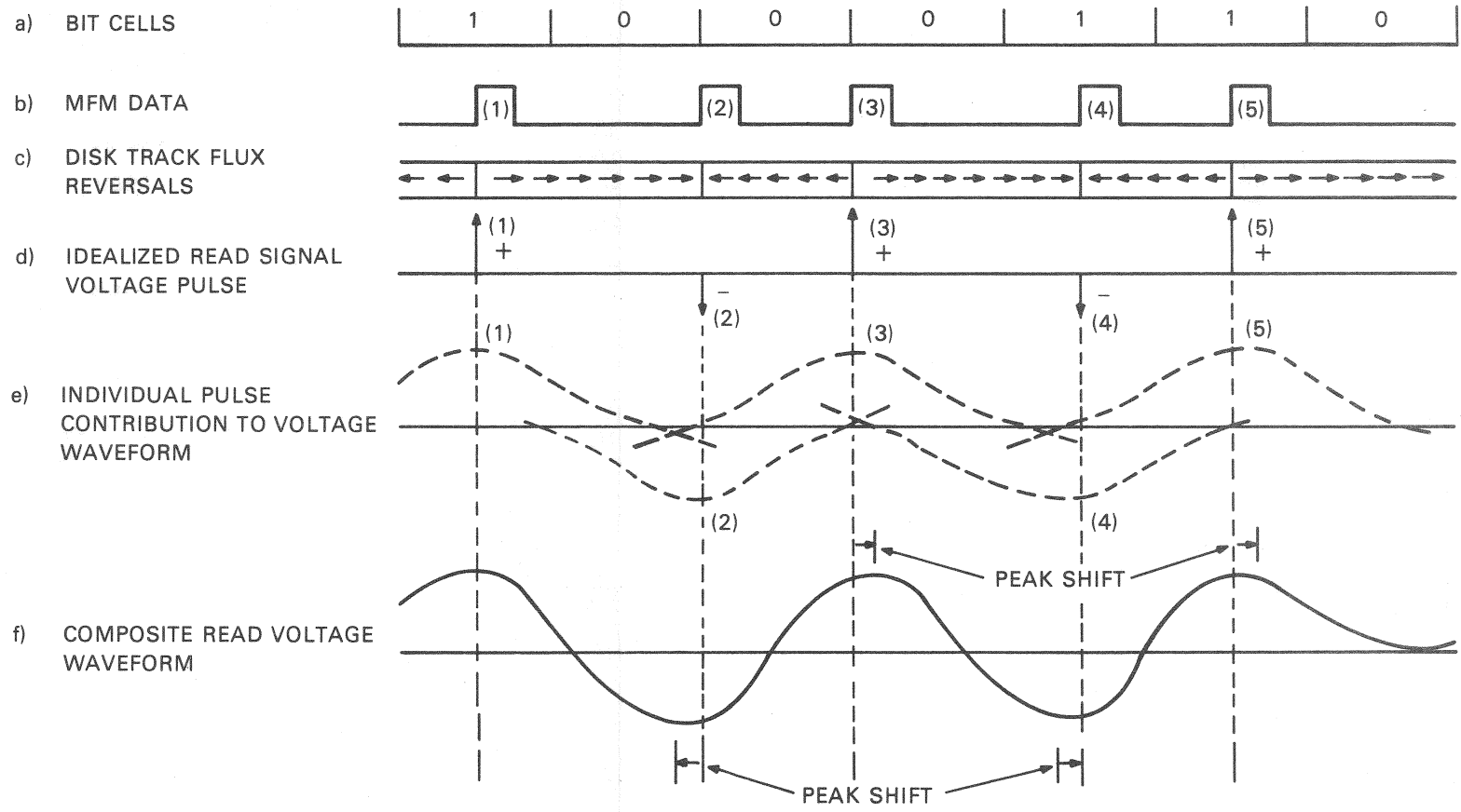
5.6 DATA SEPARATOR AND PLL

The basic problem with the recovery of MFM encoded data is the absence of a regularly recurring reference. Since the data pulse may occur at either 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 that there will be at least one flux reversal on the disk for every two bit cells. This fundamental frequency makes it feasible to use phase-locked loop techniques to form a self-clocking data recovery system.

5.6.1 Phase-Locked Loop (PLL)

The phase-locked loop 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. A simplified block diagram is shown in Figure 5-23.

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 sending write data back into the read circuits.



MA-1160

Figure 5-21 Peak Shift Waveform

The KRM disk drive has a high-density recording system, employing the precision of peak shifting. This system has the effect of reducing peak-to-peak voltage from the detection of the recorded flux reversals. This system is shown in Figure 5-21. The practical effect of this is a reduced margin of data recovery. Figure 5-21 illustrates the cause of peak shift. The illustration shows the MFM data and the flux reversal pattern. Part (b) of Figure 5-21 shows the idealized read signal voltage pulses that it generates on the disk track. Part (c) shows the idealized read signal voltage pulses that it generates on the disk track when the flux field could change instantaneously. In a real system, flux fields are generated by magnetic heads, and windings and amplifiers all contribute a distorting effect on these ideal pulses. Part (d) of Figure 5-21 shows a more realistic shape for individual read pulse waveforms. When the contributions of each individual pulse are summed into one composite waveform, the result is as shown in Part (e). The peaks of the composite read voltage waveform have shifted. Notice that the direction of peak shift can be correlated with the flux reversal pattern. As a general rule, it may be stated that two adjacent flux reversals appear to repel each other. The degree of shift is a function of the proximity of the reversals and of the environment that is the location of the pair relative to other flux reversals in the sequence.

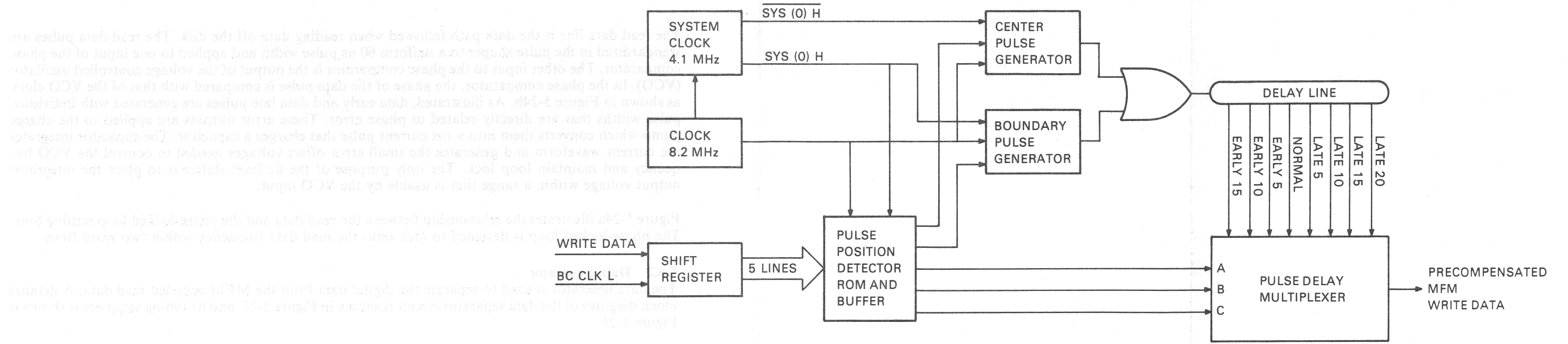
As was mentioned earlier, the write precompensation system has the function of time shifting the data pulses to compensate for the effect of peak shift. Figure 5-22 shows the precompensation circuit-related block diagram.

The write data is initially shifted into a 5-bit shift register. The shift register output is then fed into a write precompensation circuit. This ROM can receive five serial data bits at a time. Initially, the ROM is programmed to encode read data into MFM data. The bit pattern from the shift register that addresses the ROM will determine the ROM output. This output does two things. Two bits are generated either a bit counter pulse or a bit counter pulse. A boundary pulse is generated when encoding two successive logical '1's. A counter pulse is generated to encode a logical '1' when three bits of the instruction shift the data. This pulse runs to the data multiplexer. Various bit patterns can advance or retard the leading edge of the MFM data pulse by as much as 30ns. This is the time shifting that accomplishes the precompensation for peak shift phenomenon.

The DATA SEPARATOR AND PLL is the basic problem with the recovery of MFM recorded data is the absence of a regularly recurring reference. Since the data rate may occur at either 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 that there will be at least one flux reversal on the disk for every two bit cells. This fundamental frequency makes it possible to use phase-locked loop techniques to form a self-clocking data recovery system.

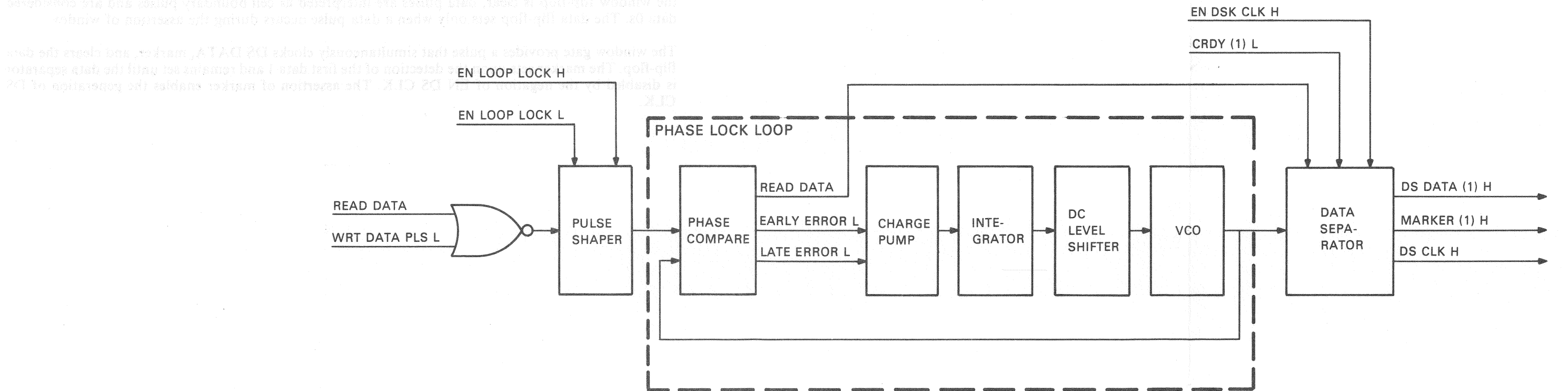
The phase-locked loop is a closed-loop circuit that provides an output that is in phase and frequency with the input signal. Its output frequency is twice that of the incoming read data bit rate. A simplified block diagram is shown in Figure 5-23.

The input data into the pulse shaper can come from two different sources. The WRITE DATA PLL 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 sending write data back into the read circuit.



MA-1150

Figure 5-22 Write Encode and Precompensation Circuit



MA-1157

Figure 5-23 Data Separator and PLL Block Diagram

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 5-24b. As illustrated, data early and data late pulses are generated with individual pulse widths that 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 5-24a illustrates the relationship between the read data and the phase-locked loop settling time. The phase-locked loop is designed to lock onto the read data frequency within two word times.

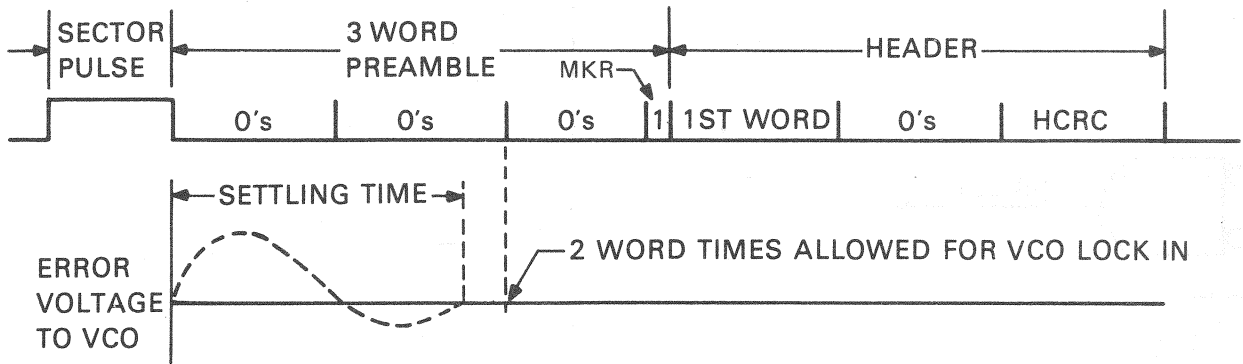
5.6.2 Data Separator

The data separator is used to separate the digital data from the MFM encoded read data. A detailed block diagram of the data separator circuit is shown in Figure 5-25, and its timing sequence is shown in Figure 5-26.

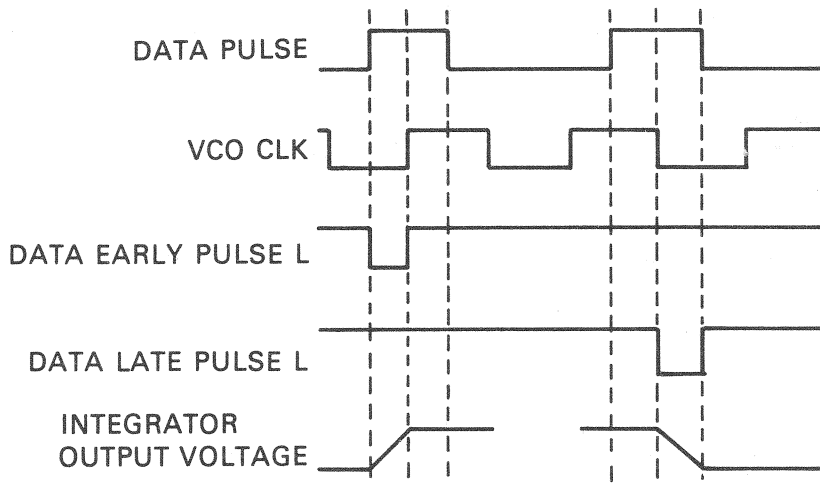
When a read header command is decoded and the first sector pulse is detected, header preamble PR1 appears on the read data line from the selected drive. LOOP LOCK is asserted and enables the preamble 0s to enter the phase-locked loop. The PLL is given two word times to allow it enough time to lock onto the bit rate frequency. Then EN DSK CLK H is asserted to enable the data separator circuit and switch the clock input of the microsequencer from SYS CLK to DS CLK.

The window flip-flop is now allowed to toggle under control of the VCO OUT. When set, this flip-flop indicates a window time during which data pulses are interpreted as cell center pulses (data 1s). When the window flip-flop is clear, data pulses are interpreted as cell boundary pulses and are considered data 0s. The data flip-flop sets only when a data pulse occurs during the assertion of window.

The window gate provides a pulse that simultaneously clocks DS DATA, marker, and clears the data flip-flop. The marker sets upon the detection of the first data 1 and remains set until the data separator is disabled by the negation of EN DS CLK. The assertion of marker enables the generation of DS CLK.



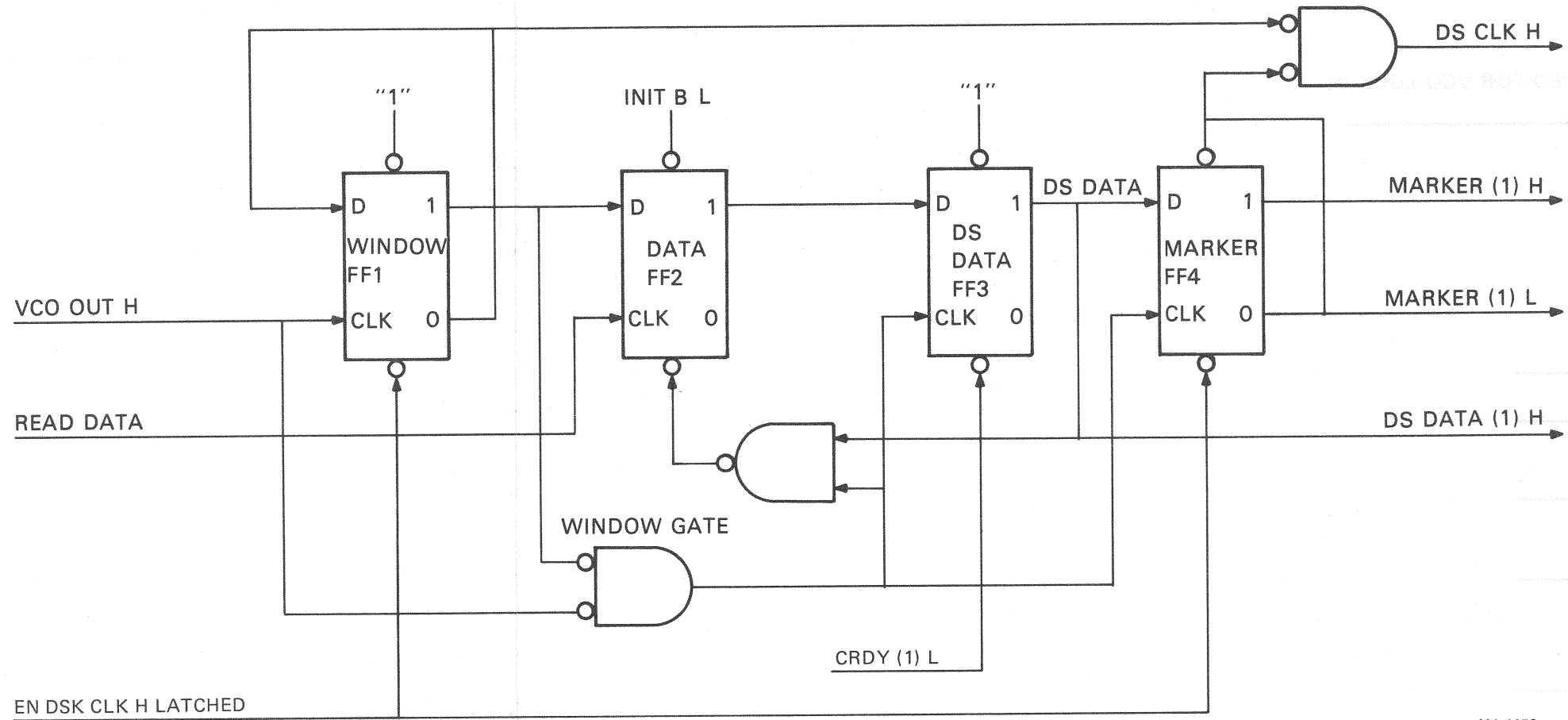
a. Loop Lock Settling Time



b. Data Error Pulse Voltages

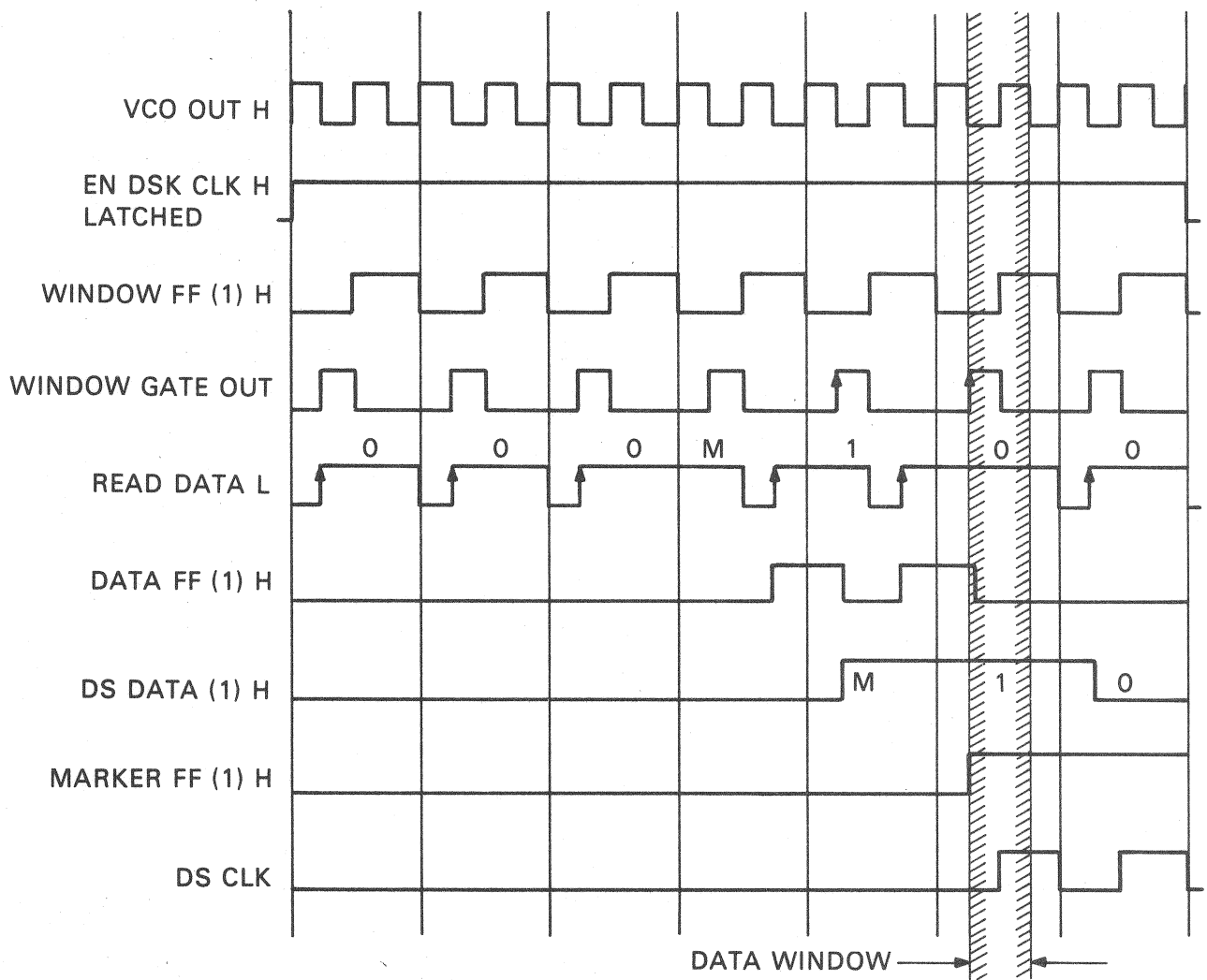
Figure 5-24 PLL Timing Relationships

MA-1171



MA-1170

Figure 5-25 Data Separator Circuit



MA-1158

Figure 5-26 Data Separator Waveforms



Figure 26 Disk Register Window

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-RLV11-TD-001

Reader's Comments

REV. 11-73-001
TECHNICAL DESCRIPTION MANUAL
REV. 11-73-001

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?

Fold Here

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 the return copy of the Technical Documentation Catalog, which contains information on the remainder of DIGITAL's technical documentation.

Do Not Tear - Fold Here and Staple

Name _____
Street _____
City _____
State/County _____
Zip _____

**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

