

EK-RLV12-TD-001

# RLV12 Technical Description

**digital**

EK-RLV12-TD-001

# **RLV12 Technical Description**

Prepared by Educational Services  
Digital Equipment Corporation

Copyright © 1981 by Digital Equipment Corporation

All Rights Reserved

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.**

- **Class A Computing Devices:**

**Notice:** This equipment generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such radio frequency interference when operated in a commercial environment. Operation of this equipment in a residential area may cause interference in which case the user at his own expense may be required to take measures to correct the interference.

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

DEC	DECnet	OMNIBUS
DECUS	DECsystem-10	OS/8
DIGITAL	DECSYSTEM-20	PDT
Digital Logo	DECwriter	RSTS
PDP	DIBOL	RSX
UNIBUS	EduSystem	VMS
VAX	IAS	VT
	MASSBUS	

# CONTENTS

	Page
<b>CHAPTER 1 INTRODUCTION</b>	
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
1.5 ADDRESSABLE REGISTERS .....	1-1
1.6 CONTROLLER COMMANDS .....	1-4
1.7 REFERENCE DOCUMENTS .....	1-5
1.8 CONTROLLER SPECIFICATIONS .....	1-5
1.9 RL01/RL02 DISK DRIVE SPECIFICATION .....	1-7
<b>CHAPTER 2 SYSTEM LEVEL DESCRIPTION</b>	
2.1 GENERAL .....	2-1
2.2 SYSTEM BLOCK DIAGRAM .....	2-1
2.2.1 KDF11-B CPU .....	2-1
2.2.2 RLV12 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 .....	2-4
2.2.3 RL01/RL02 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 .....	2-8
2.3.2.1 Seek Command .....	2-8
2.3.2.2 Write Data Command .....	2-8
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

	<b>Page</b>
<b>CHAPTER 2</b>	<b>SYSTEM LEVEL DESCRIPTION (Cont)</b>
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..... 2-13
2.10	READ DATA FUNCTION..... 2-13
2.11	READ DATA WITHOUT HEADER CHECK FUNCTION ..... 2-16
<b>CHAPTER 3</b>	<b>INTERFACE LEVEL DESCRIPTION</b>
3.1	GENERAL..... 3-1
3.2	LSI-11 BUS ..... 3-2
3.3	DRIVE BUS INTERFACE SIGNALS ..... 3-2
3.3.1	General ..... 3-2
3.3.2	Drive Bus Signal Descriptions ..... 3-2
3.3.2.1	Drive Select (DRV SEL 0,1) ..... 3-2
3.3.2.2	System Clock (SYS CLK) ..... 3-2
3.3.2.3	Drive Command (DR CMD) ..... 3-2
3.3.2.4	Write Gate (WR GATE) ..... 3-2
3.3.2.5	Write Data (WR DATA) ..... 3-2
3.3.2.6	Power Fail (PWR FAIL)..... 3-2
3.3.2.7	Drive Read (DR RDY) ..... 3-7
3.3.2.8	Drive Error (DR ERR) ..... 3-7
3.3.2.9	Status Clock (STATUS CLK)..... 3-8
3.3.2.10	Status (STATUS)..... 3-8
3.3.2.11	Sector Pulse (SEC PLS) ..... 3-8
3.3.2.12	Read Data (RD DATA)..... 3-8
3.3.3	Drive Bus Dialogue..... 3-8
3.4	ADDRESSABLE REGISTERS..... 3-16
3.4.1	Control Status Register ..... 3-16
3.4.2	Bus Address Register..... 3-18
3.4.3	Disk Address Register ..... 3-19
3.4.3.1	DA Register During a Seek Command ..... 3-29
3.4.3.2	DA Register During Read or Write Data Command..... 3-20
3.4.3.3	DA Register During a Get Status Command..... 3-21
3.4.4	Multipurpose Register ..... 3-22
3.4.4.1	MP Register During a Get Status Command..... 3-22
3.4.4.2	MP Register During a Read Header Command ..... 3-24
3.4.4.3	MP Register During Read/Write Data Commands ..... 3-25
3.4.4.4	Bus Address Extension Register..... 3-26
3.4.4.5	RLV12 Subsystem Register Summary ..... 3-27
3.5	I/O TRANSFER OPERATIONS..... 3-29
3.5.1	Programmed I/O Transfers..... 3-29
3.5.1.1	Writing Controller Registers ..... 3-29
3.5.1.2	Reading Controller Registers ..... 3-30
3.5.2	DMA I/O Transfers ..... 3-31
3.5.3	Interrupt-Driven I/O Transfers..... 3-32
3.6	BUS SIGNAL TIMING ..... 3-32

**CHAPTER 4 FUNCTIONAL LEVEL DESCRIPTION**

4.1	GENERAL.....	4-1
4.2	CONTROLLER SIMPLIFIED BLOCK DIAGRAM.....	4-1
4.2.1	Bus Control Functions .....	4-1
4.2.2	Bus Transceivers.....	4-1
4.2.3	Programmable Registers .....	4-3
4.2.4	FIFO.....	4-3
4.2.5	Microsequencer Logic .....	4-3
4.2.6	Write Precompensation .....	4-3
4.2.7	Data Separator.....	4-4
4.2.8	CRC Circuit .....	4-4
4.2.9	Data Source Multiplexer .....	4-4
4.2.10	Compare Logic .....	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.....	4-24
4.10	READ DATA WITHOUT HEADER CHECK FUNCTIONAL FLOW DIAGRAM .....	4-27

**CHAPTER 5 UNIT LEVEL DESCRIPTION**

5.1	INTRODUCTION.....	5-1
5.2	BUS PROTOCOL.....	5-3
5.3	BUS TRANSCEIVERS.....	5-3
5.4	PROGRAMMABLE REGISTERS .....	5-3
5.4.1	Bus Address Register (BAR).....	5-3
5.4.2	Bus Address Extension Register (BAE) .....	5-7
5.4.3	Disk Address Register (DAR).....	5-8
5.4.4	Control/Status Register (CSR).....	5-8
5.4.5	Multipurpose Register (MPR) .....	5-10
5.5	DATA SOURCE MULTIPLELXER AND CRC GENERATOR.....	5-12
5.6	MICROSEQUENCER.....	5-12
5.7	CONTROL REGISTERS AND PULSE GENERATORS.....	5-14
5.8	WRITE ENCODER AND PRECOMPENSATION LOGIC .....	5-14
5.9	DATA SEPARATOR READ CIRCUIT .....	5-17

**FIGURES**

1-1	RLV12 Subsystem Configuration.....	1-2
1-2	11/23B Backplane Priority Structure.....	1-3
2-1	System Block Diagram .....	2-1
2-2	RLV12 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)

2-6	Simplified Get Status Operation .....	2-10
2-7	Simplified Read Header Operation .....	2-11
2-9	Simplified Write Data Operation .....	2-14
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.....	3-3
3-3	Drive Bus Signals.....	3-7
3-4	Get Status Drive Bus Sequence .....	3-9
3-5	Read Header Drive Bus Sequence.....	3-10
3-6	Seek Drive Bus Sequence .....	3-11
3-7	Write Data Drive Bus Sequence.....	3-12
3-8	Write Check Drive Bus Sequence.....	3-13
3-9	Read Data Drive Bus Sequence.....	3-14
3-10	Read Data without Head Check Drive Bus Sequence.....	3-15
3-11	Control Status Register.....	3-16
3-12	BA Register.....	3-19
3-13	DAR Contents to Execute a Seek Command .....	3-20
3-14	DAR Contents During a Read/Write Data Command .....	3-21
3-15	DAR Contents to Execute a Get Status Command.....	3-21
3-16	MPR – Following a Get Status Command.....	3-22
3-17	MPR – Following a Read Header Command .....	3-25
3-18	MPR – Used as a Word Counter .....	3-26
3-19	BAE Register .....	3-27
3-20	Register Summary .....	3-27
3-21	DATO Bus Cycle .....	3-30
3-22	DATI Bus Cycle .....	3-31
3-23	DMA Request/Grant Sequence .....	3-33
3-24	Interrupt Request/Acknowledge Sequence.....	3-34
4-1	RLV12 Controller Simplified Block Diagram.....	4-2
4-2	Maintenance Command Functional Block Diagram.....	4-5
4-3	Maintenance Command Flowchart.....	4-7
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
4-15	Read Data Functional Block Diagram.....	4-25
4-16	Read Data Command Flowchart .....	4-26
4-17	Read Data without Header Check Functional Block Diagram .....	4-28
4-18	Read Data without Header Check Command Flowchart .....	4-29
5-1	RLV12 Block Diagram .....	5-2
5-2	Bus Protocol Logic .....	5-4
5-3	Bus Transceivers .....	5-5

## FIGURES (Cont)

5-4	Bus Address Register (BAR) Circuit.....	5-6
5-5	Bus Address Extension Register (BAE) Circuit .....	5-7
5-6	Disk Address Register (DAR) Circuit.....	5-8
5-7	Control/Status Register (CSR) Circuit.....	5-9
5-8	FIFO Buffer .....	5-11
5-9	Microsequencer Logic.....	5-13
5-10	MFM Encoding.....	5-15
5-11	Peak Shift Waveforms .....	5-15
5-12	Write Encoder and Precompensation Circuit .....	5-16
5-13	Data Separator Read Circuit.....	5-17

## TABLES

1-1	Controller Addressable Registers .....	1-4
1-2	Reference Documents .....	1-5
1-3	RLV12 Controller Specifications .....	1-5
1-4	RL01/RL02 Drive Specification.....	1-8
3-1	LSI-11 Bus Signals.....	3-4
3-2	Register Address .....	3-16
3-3	Control Status Register Bit Description .....	3-17
3-4	Bus Address Register Bit Description.....	3-19
3-5	Disk Address Register Bit Description for Seek Commands .....	3-20
3-6	Disk Address Register Bit Description for Data Transfer Commands .....	3-21
3-7	Disk Address Register Bit Description for Get Status Commands .....	3-22
3-8	MP Register Bit Description for Get Status Commands.....	3-23
3-9	MP Register Bit Description for Read Header Commands.....	3-25
3-10	MP Register Bit Description for Data Transfer Commands .....	3-26
5-1	Control/Status Register Bit Summary .....	5-10



# CHAPTER 1 INTRODUCTION

## 1.1 PURPOSE AND SCOPE OF MANUAL

This manual contains a physical and technical description of the RLV12 Disk Controller. It is intended to aid Field Service and user maintenance personnel in servicing the RLV12. 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 RLV12 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 RLV12 controller and from one to four RL01 or RL02 disk drives. Each RL01 drive can store 5.24 million bytes. In the maximum configuration, the RL01 subsystem storage capability approaches nearly 21 million bytes. This capacity is doubled using RL02 drives.

The RLV12 controller provides the control functions for the disk subsystem. It consists of one quad module that mounts on the backplane inside the LSI-11 CPU cabinet or any backplane adhering to the LSI-11 bus specification (DEC STD 160). The controller is the interface between the LSI-11 bus and the disk drive.

## 1.3 SUBSYSTEM CONFIGURATION

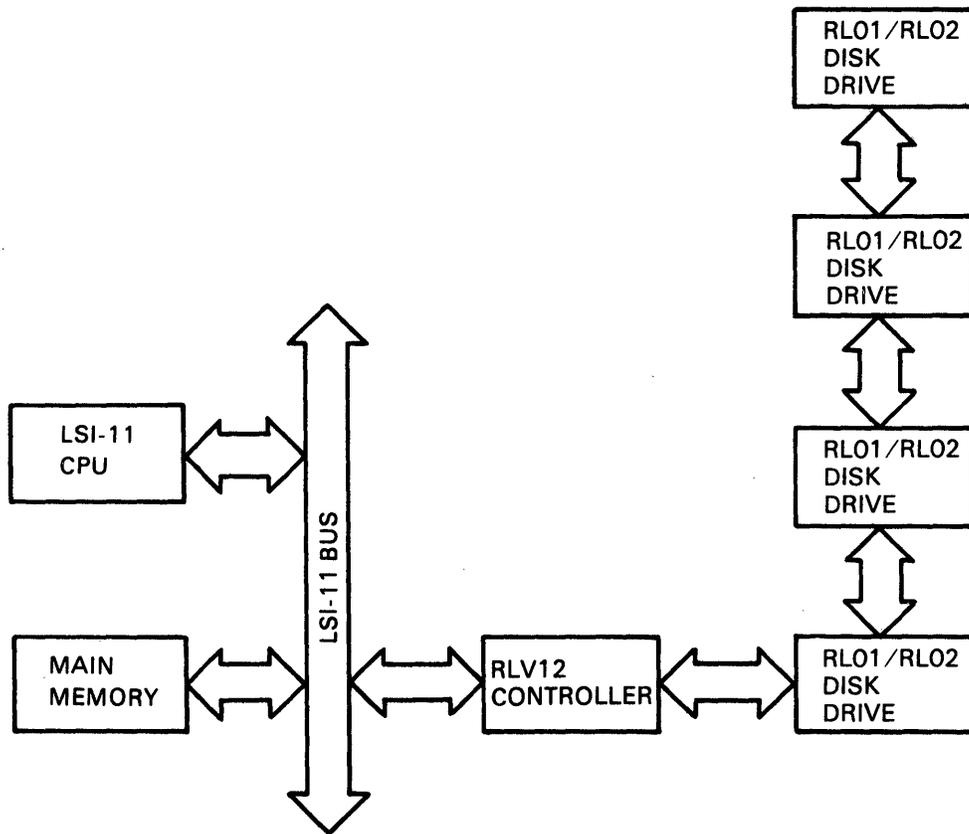
In the minimum subsystem configuration, there is only one disk drive attached to the drive bus. In the maximum configuration, the RLV12 controller can support up to four RL01 or RL02 disk drives (Figure 1-1). The design is such that the controller can communicate with two or more drives in a time-shared fashion. Connections between controller and drives are made in daisy-chain arrangement.

## 1.4 PHYSICAL DESCRIPTION OF CONTROLLER

The RLV12 controller consists of one quad-height module (M8061) that inserts into the LSI-11 micro-computer system backplane. Figure 1-2 shows the placement of the RLV12 in the system.

## 1.5 ADDRESSABLE REGISTERS

The controller contains five 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.



CZ-0505

Figure 1-1 RLV12 Subsystem Configuration

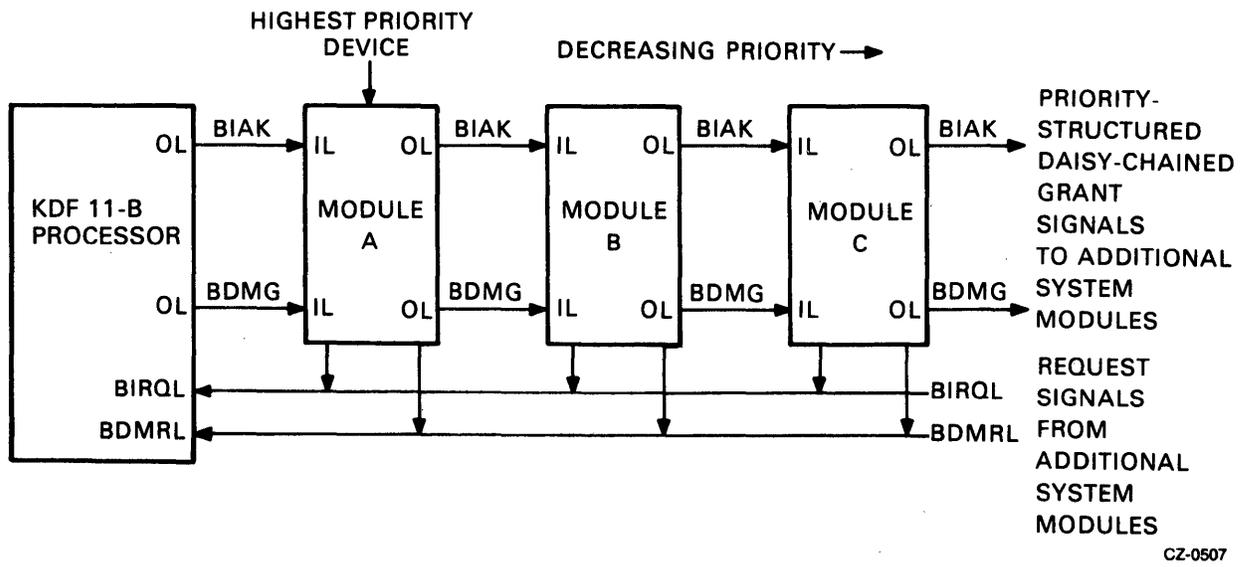
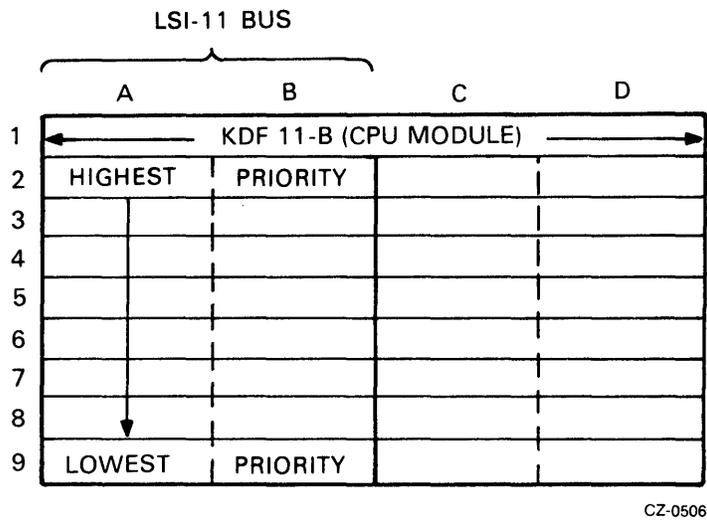


Figure 1-2 11/23B Backplane Priority Structure

**Table 1-1 Controller Addressable Registers**

Standard Address (Octal)	Description
174400	<b>Control Status (CS)</b> – Indicates subsystem ready condition, holds drive commands, and provides overall control functions and error indications.
174402	<b>Bus Address (BAR)</b> – Indicates memory location involved in a DMA data transfer during a read or write operation.
174404	<b>Disk Address (DAR)</b> – Stores information for: <ol style="list-style-type: none"> <li>1. Seeking to a desired track</li> <li>2. Selecting sectors to be transferred during read/write operations.</li> <li>3. Used to request the drive status.</li> </ol>
174406	<b>Multipurpose (MPR)</b> – 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.
17774410	<b>Bus Address Extension (BAE)</b> – Contains the upper six bits of the extended bus address when using 22-bit addressing mode.
17774412	<b>Not Used</b> – Reserved (read only).
17774414	<b>Not Used</b> – Reserved (read only).
17774416	<b>Not Used</b> – Reserved (read only).

### 1.6 CONTROLLER COMMANDS

The controller has a repertoire of eight commands. The following paragraphs list the eight commands and include a brief description. A more complete technical description of each command is in Chapter 4 of this manual.

- **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 1-2 contain information necessary to understand the function, installation, operation, programming, and maintenance of the RLV12 controller and RL01/RL02 disk drives.

**Table 1-2 Reference Documents**

Title	Document No.
RL01/RL02 Disk Subsystem User's Manual RL01/RL02 Disk Subsystem Preventive Maintenance RL01 Disk Drive Illustrated Parts Breakdown RL01/RL02 Disk Drive Technical Manual RL02 Disk Drive Illustrated Parts Breakdown RL01/RL02 Disk Drive Pocket Service Guide RL01 Field Maintenance Print Set RL02 Field Maintenance Print Set RLV12 Field Maintenance Print Set RLV22 Field Maintenance Print Set	EK-RL012-OP EP-00008-PM EK-RL01-IP EK-RL012-TM EP-RL02-IP EK-RL012-PG MP-00347 MP-00553 MP-01282 MP-01283

### 1.8 CONTROLLER SPECIFICATIONS

The performance, power, and physical specifications for the RLV12 controller are listed in Table 1-3.

**Table 1-3 RLV12 Controller Specifications**

Characteristic	Specification
Module	1 quad-size module, M8061
Size	Height (10.457 in) Width (0.5 in) Length (8.94 in)
Power Requirements	+5 Vdc $\pm$ 5% at 5.0 A +12 Vdc $\pm$ 5% at 0.1 A

**Table 1-3 RLV12 Controller Specifications  
(Cont)**

Characteristic	Specification								
Bus Loads AC bus loads DC bus loads	3 1								
Addressing Modes	16-, 18-, and 22-bit; (determined by user)								
Requirement to use 22-bit Address Mode	H9276-A or similar backplane with memory capable of 22-bit addresses, such as the MSV11-L or the MSV11-P.								
Limitations LSI-11 mini-series backplane H9281.	The RLV12 will not fit in the dual-height								
No. Drives per Controller	Up to 4 RL01 and RL02 drives in any combination.								
No. of LSI-11 bus-addressable registers	8 (5 are used; 3 are not used)								
Base Device Address	Selected by jumpers as follows. <table border="0" data-bbox="688 1094 1110 1283"> <thead> <tr> <th data-bbox="688 1094 829 1150">Addressing Mode</th> <th data-bbox="959 1094 1110 1150">Base Device Address</th> </tr> </thead> <tbody> <tr> <td data-bbox="688 1188 764 1220">16-bit</td> <td data-bbox="959 1188 1052 1220">174400</td> </tr> <tr> <td data-bbox="688 1220 764 1251">18-bit</td> <td data-bbox="959 1220 1052 1251">774400</td> </tr> <tr> <td data-bbox="688 1251 764 1283">22-bit</td> <td data-bbox="959 1251 1081 1283">17774400</td> </tr> </tbody> </table>	Addressing Mode	Base Device Address	16-bit	174400	18-bit	774400	22-bit	17774400
Addressing Mode	Base Device Address								
16-bit	174400								
18-bit	774400								
22-bit	17774400								
Device Interrupt Vector	000160, jumper selectable								
Data Transfer Rates	4.9 microseconds/word (average) drive to controller, controller to memory  3.9 microseconds/word (peak) drive to controller  2.0 microseconds/word (peak) controller to memory								
Error Detection Capability	Cyclic redundancy check (CRC) on data and headers  Memory parity error abort for use with memories that have parity checking								

**Table 1-3 RLV12 Controller Specifications  
(Cont)**

Characteristic	Specification
Maximum Cable Length between Controller and Last Drive	30 m (100 ft)
Environment Specifications:	
Temperature	
Storage	-40° C to 66° C (-40° F to 150° F)
Operating*	10° C to 60° C (50° F to 110° F)
Relative Humidity	
Storage	10% to 90%, noncondensing
Operating	10% to 90%, noncondensing
Altitude	
Not operating	9 Km (5.6 mi) maximum
Operating*	2.4 Km (1.5 mi) maximum
Airflow	
Operating	Maximum temperature rise across module must not exceed 10° C.

\* Reduce the maximum operating temperature by 1.8° C for each 1000 meters altitude above sea level or 1° F for each 1000 ft above sea level.

**1.9 RL01/RL02 DISK DRIVE SPECIFICATION**

The performance, power, and physical specification for the RL drives are listed in Table 1-4.

**Table 1-4 \ RL01/RL02 Drive Specification**

Characteristic	Specification																									
Storage Type	Magnetic disk cartridge																									
Medium	2 data surfaces																									
No. of Recording Surfaces	2 read/write heads																									
Magnetic heads	RL01	RL02																								
Recording Capacity (formatted)	<table border="0"> <tr> <td data-bbox="204 800 493 831">Cylinders per cartridge</td> <td data-bbox="678 800 727 831">256</td> <td data-bbox="867 800 915 831">512</td> </tr> <tr> <td data-bbox="204 831 444 863">Tracks per cylinder</td> <td data-bbox="678 831 695 863">2</td> <td data-bbox="867 831 883 863">2</td> </tr> <tr> <td data-bbox="204 863 456 894">Tracks per cartridge</td> <td data-bbox="678 863 727 894">512</td> <td data-bbox="867 863 932 894">1024</td> </tr> <tr> <td data-bbox="204 894 418 926">Sectors per track</td> <td data-bbox="678 894 711 926">40</td> <td data-bbox="867 894 899 926">40</td> </tr> <tr> <td data-bbox="204 926 404 957">Bytes per sector</td> <td data-bbox="678 926 727 957">256</td> <td data-bbox="867 926 915 957">256</td> </tr> <tr> <td data-bbox="204 957 391 989">Bytes per track</td> <td data-bbox="678 957 769 989">10,240</td> <td data-bbox="867 957 958 989">10,240</td> </tr> <tr> <td data-bbox="204 989 428 1020">Bytes per cylinder</td> <td data-bbox="678 989 769 1020">20,480</td> <td data-bbox="867 989 958 1020">20,480</td> </tr> <tr> <td data-bbox="204 1020 440 1052">Bytes per cartridge</td> <td data-bbox="678 1020 769 1052">5.24 M</td> <td data-bbox="867 1020 976 1052">10.48 M</td> </tr> </table>		Cylinders per cartridge	256	512	Tracks per cylinder	2	2	Tracks per cartridge	512	1024	Sectors per track	40	40	Bytes per sector	256	256	Bytes per track	10,240	10,240	Bytes per cylinder	20,480	20,480	Bytes per cartridge	5.24 M	10.48 M
Cylinders per cartridge	256	512																								
Tracks per cylinder	2	2																								
Tracks per cartridge	512	1024																								
Sectors per track	40	40																								
Bytes per sector	256	256																								
Bytes per track	10,240	10,240																								
Bytes per cylinder	20,480	20,480																								
Bytes per cartridge	5.24 M	10.48 M																								
Recording Method	Modified Frequency Modulation (MFM)																									
Performance	<p data-bbox="678 1209 1049 1241">40-Sector (16-bit data words):</p> <p data-bbox="678 1272 1193 1335">4.9 microseconds/word (average) drive to controller, controller to memory</p> <p data-bbox="678 1367 1154 1430">3.9 microseconds/word (peak) drive to controller</p> <p data-bbox="204 1461 488 1493">Head Positioning Time</p> <p data-bbox="678 1461 924 1556">55 ms (average) 17 ms (one track) 100 ms (maximum)</p> <p data-bbox="204 1587 448 1619">Revolution Latency</p> <p data-bbox="678 1587 899 1619">12.5 ms (average)</p>																									
Operating Environment	<p data-bbox="204 1713 451 1745">Temperature Range</p> <p data-bbox="678 1713 1182 1745">10° to 40° C (50° to 104° F) at sea level</p> <p data-bbox="204 1776 436 1808">Relative Humidity</p> <p data-bbox="678 1776 1024 1808">10% to 90%, noncondensing</p>																									

**Table 1-4 RL01/RL02 Drive Specification (Cont)**

<b>Characteristic</b>	<b>Specification</b>
Wet Bulb Temperature	28° C (82° F) maximum
Altitude	Up to 2400 m (8000 ft) at maximum temperature of 36° C (96° F)
Heat Dissipation	150W (546 Btu/hr)
Operation	
Start Time	50 s
Stop Time	30 s
Revolutions per Minute	2400
Power	
Drive	Single-phase
Starting Current	5 A (rms) max, 120 V, 47/63 Hz 2.5 A (rms) max, 240 V, 47/63 Hz
Mechanical Drive	
Size	48 cm wide X 63.4 cm deep X 27 cm high (19 in wide X 25 in deep X 10.5 in high)
Weight	33.75 kg (75 lb.)
Mounting	RETMA standard 48.26 cm (19 in) rack-mounted on slides (provided). Recommended maximum height from floor is 18.9 cm (48 in).
Cartridge	Embedded servo Top loading cartridge with 2 data surfaces.

**Table 1-4 RL01/RL02 Drive Specification (Cont)**

Characteristic	Specification		
Standard Cable Lengths			
Power cord	2.74 m (9 ft)		
Controller to Drive	1.83 m (6 ft) of BC80M-06 cables		
Drive to Drive	3.05 m (10 ft) of 70-12122-10 cables		
Optional Drive Cables			
Cable	Part No.	Length	
BC20J-20	7012122-20	6 m (20 ft)	
BC20J-40	7012122-40	12 m (40 ft)	
BC20J-60	7012122-60	18 m (60 ft)	

**NOTE**

**Total length of cable(s) from controller to the last drive must not exceed 30 m (100 ft).**

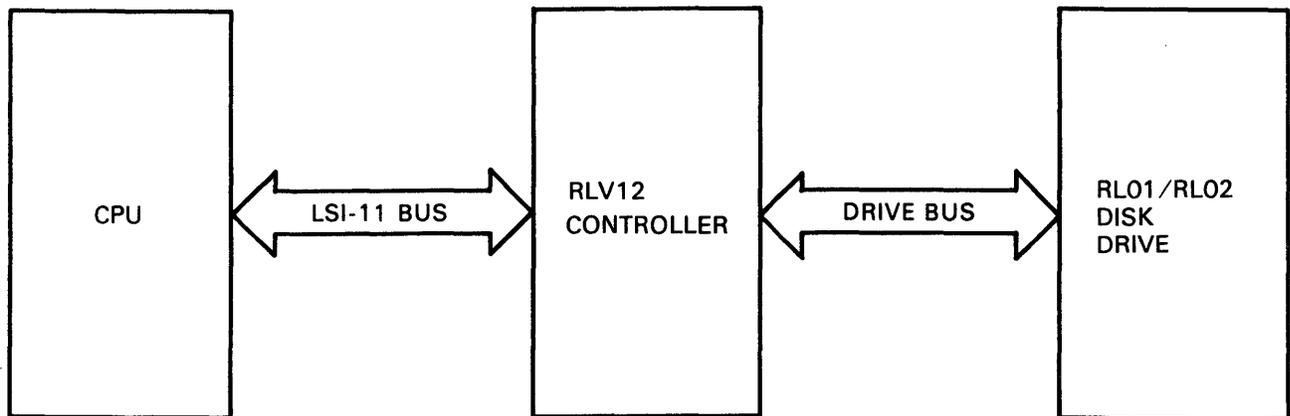
## CHAPTER 2 SYSTEM LEVEL DESCRIPTION

### 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 or RL02 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.



CZ-0508

Figure 2-1 System Block Diagram

#### 2.2.1 KDF11-B CPU

The KDF11-B is a quad-height PDP-11 processor. This processor consists of a single KDF11-B module (M8189). The M8189 module contains the following elements.

- CPU
- Memory management
- Line frequency clock
- BDV11 compatible boot diagnostic ROM
- Two serial line units

Three extra forty-pin sockets are provided for the optional floating point and commercial instruction sets. The CPU and memory management units are functionally compatible with the KDF11-AA CPU and memory.

The KDF11-B CPU supports up to 256K bytes of memory on a traditional LSI-11 bus backplane (18 address bits). When the KDF11-B is inserted into an extended LSI-11 bus backplane (H9276 or H9275), the memory support is expanded to 4M bytes.

The extended LSI-11 bus backplane adds four address lines to the LSI-11 bus providing up to 22 bits of addressing when the KDF11-B is used with the MSV11-P (M8067) memory module.

In the RL02 disk-based microcomputer system, the data storage capability of the system can be extended to over 80 megabytes. To use this data base effectively, the microcomputer must be able to read or write data on the disks easily.

## **2.2.2 RLV12 Controller**

The RLV12 controller is the interface between the RL01/RL02 disk drives and the LSI-11 microcomputer. Once activated by the microcomputer, the RLV12 transfers data between the disk and the main memory. The major functional sections of the RLV12 controller are shown in Figure 2-2.

**2.2.2.1 LSI-11 Bus Interface** – The RLV12 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 a 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 five registers: bus address register (BAR), bus address extension (BAE), disk address register (DAR), control status register (CSR), and multipurpose register (MPR). These registers can be addressed like any other word-addressable memory locations. Of the five 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 256 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 precompensation 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.

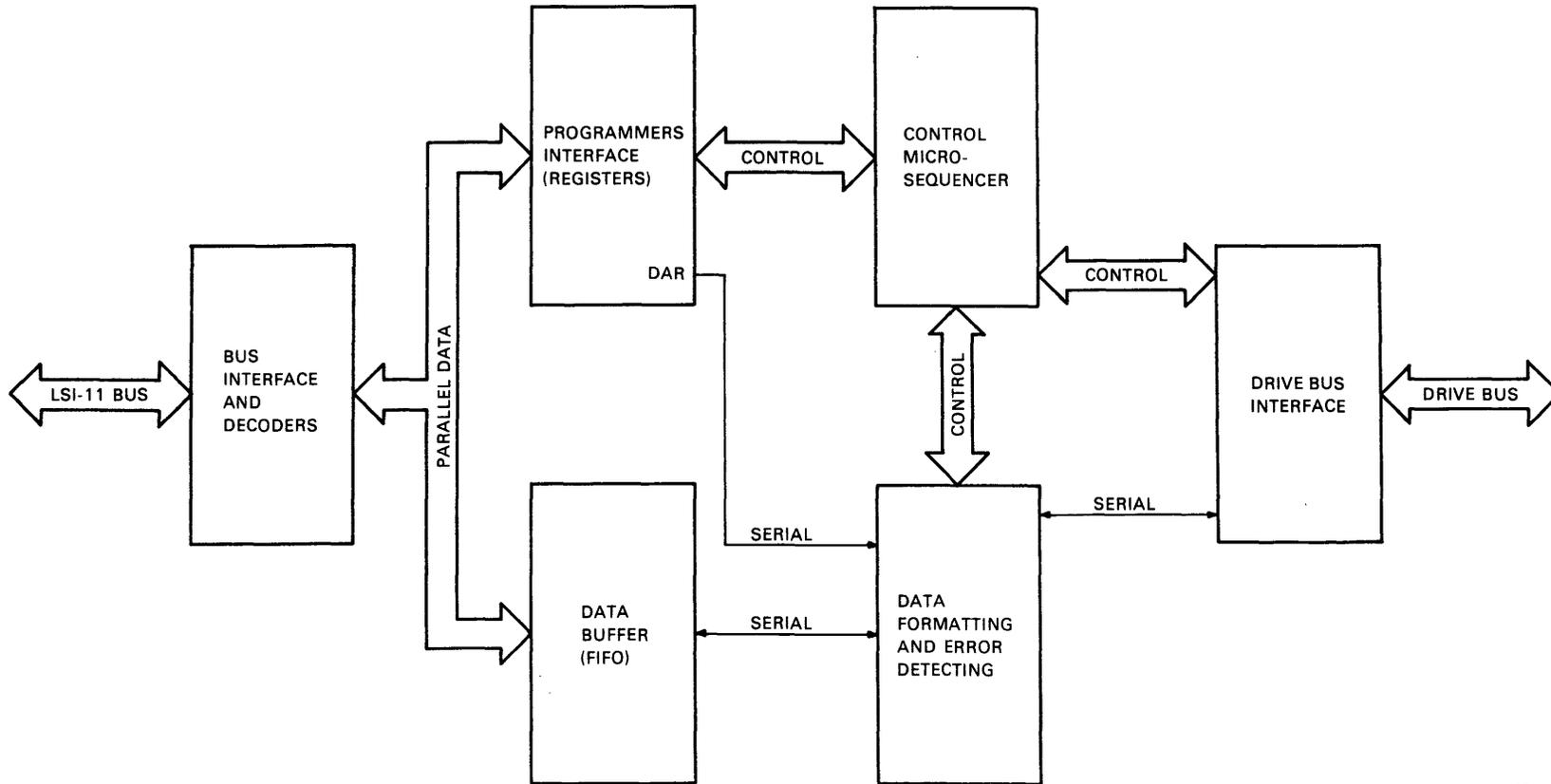


Figure 2-2 RLV12 Controller Major Functional Blocks

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/RL02 disk drive and the RLV12 controller.

### **2.2.3 RL01/RL02 Disk Drive**

**2.2.3.1 Drive Features** – The RL01 or RL02 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.

**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. The RL02K data cartridge also contains a single disk with two recording surfaces. Each surface has a total of 512 tracks yielding 512 cylinders.

Every track on a recording surface is subdivided into 40 equal-length sectors which are further subdivided into fields. The eight 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.

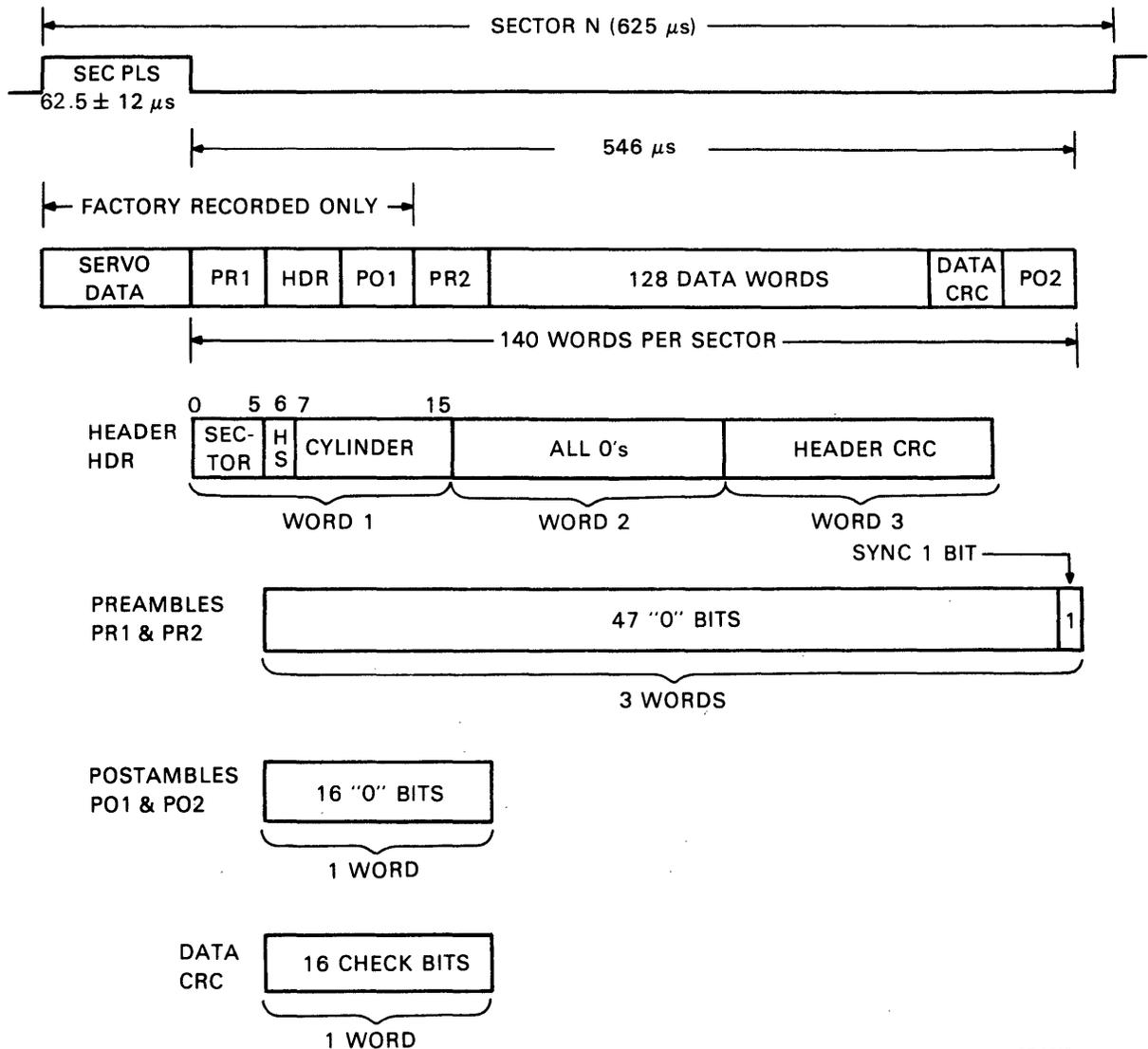
When the disk cartridge is formatted at the factory, both servo and header and related preamble and postamble information are prerecorded in each sector. The servo information is contained in two pulse bursts which occur during the sector pulse. This information identifies the radial position of the 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 identified the drive head (upper or lower), the cylinder address (1 of 512), 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.



ICZ-0510

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) for RL01 disk drives. RL02 drives have the bad sector file at track 511. 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 RLV12 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 locations of information. Figure 2-4 shows command sequential relationships.

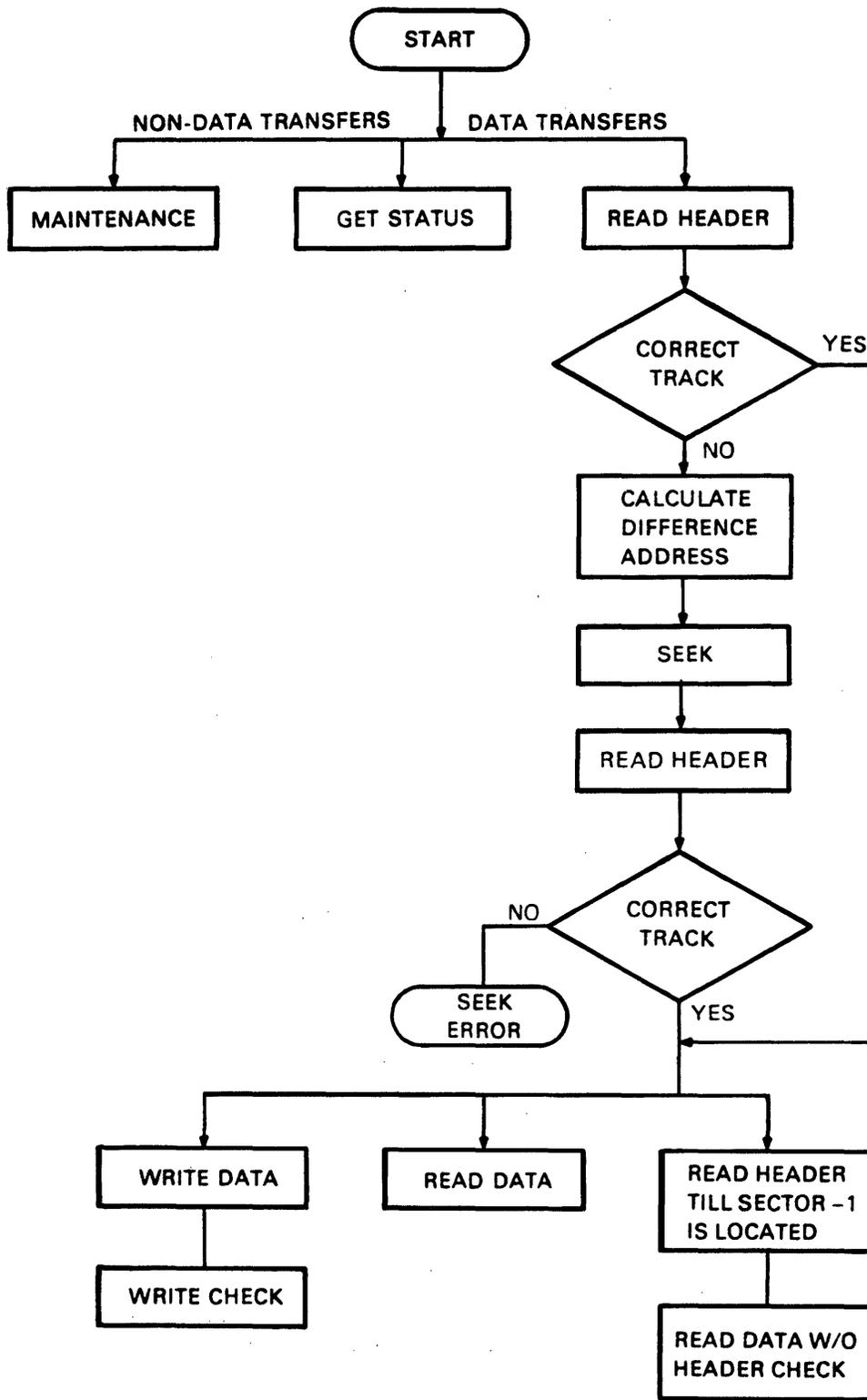
### **2.3.1 Commands Independent of Others**

**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.



MA-1169

Figure 2-4 Sequential Relationship of Controller Commands

## 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.

**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. From 1 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. From 1 to 5120 words can be read off the disk in partial sectors with one read data command.

**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 microseconds after the completion of the read header command.

## 2.4 MAINTENANCE FUNCTION

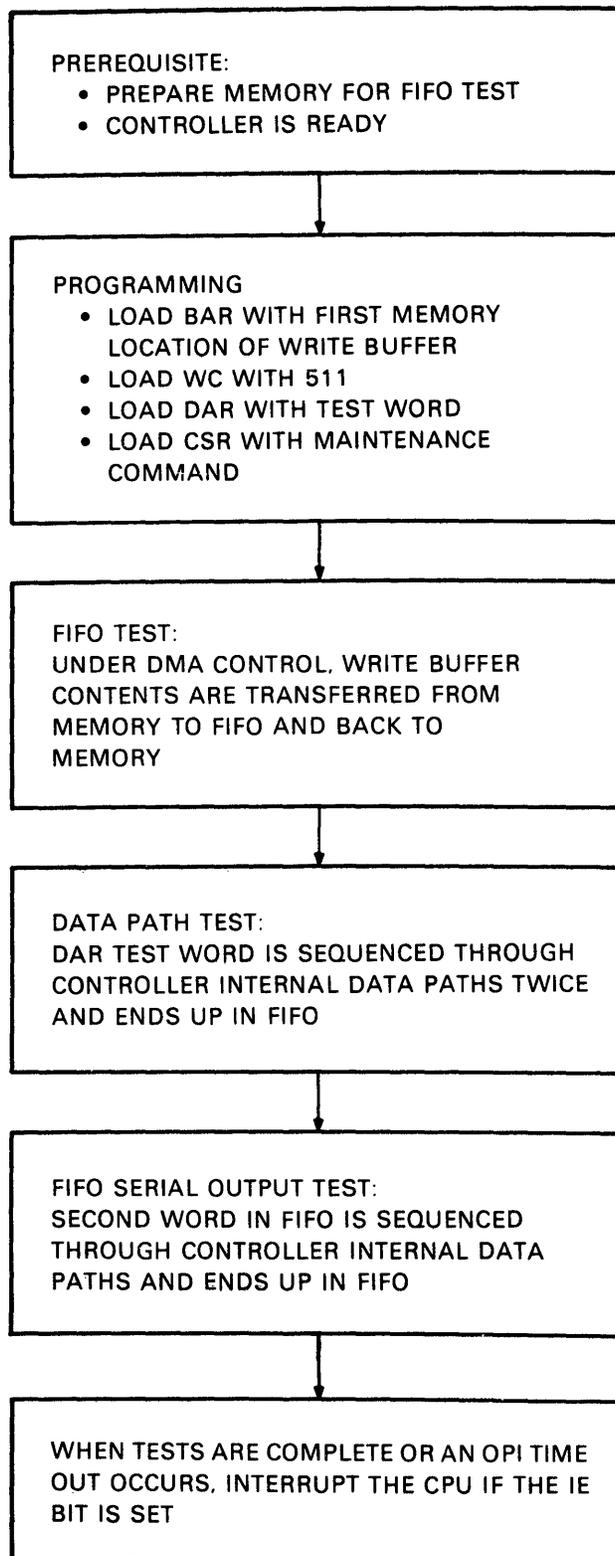
The maintenance command provides a means of exercising the controller logic circuits to test whether the major data paths and data storage functions are operating. This command is used during the diskless diagnostic routine to detect controller malfunctions or to establish a level of confidence in controller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



MA-1110

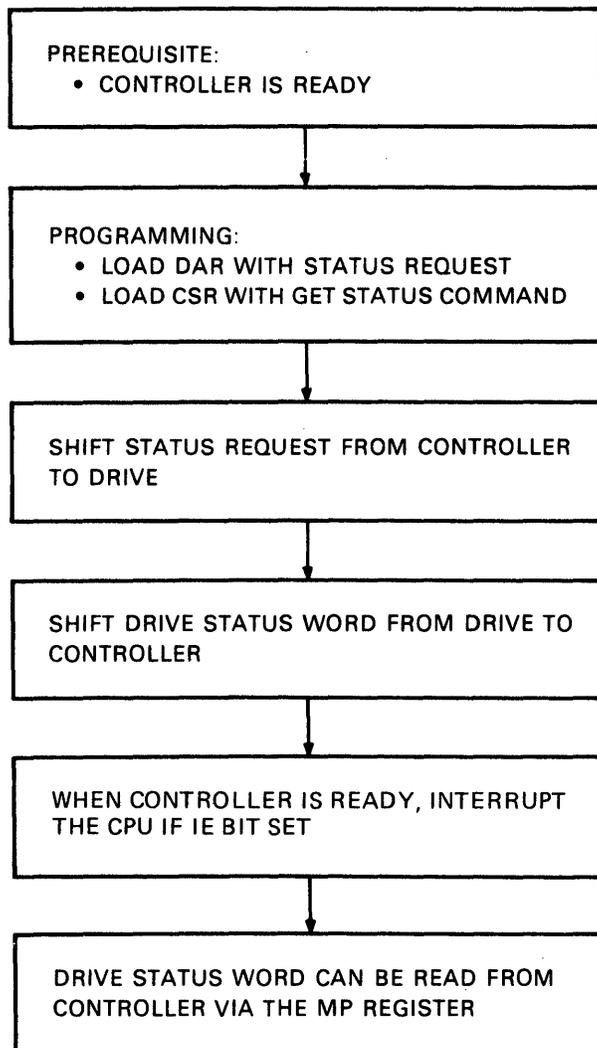
Figure 2-5 Simplified Maintenance Operation

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 the software checked for malfunctions.

### 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

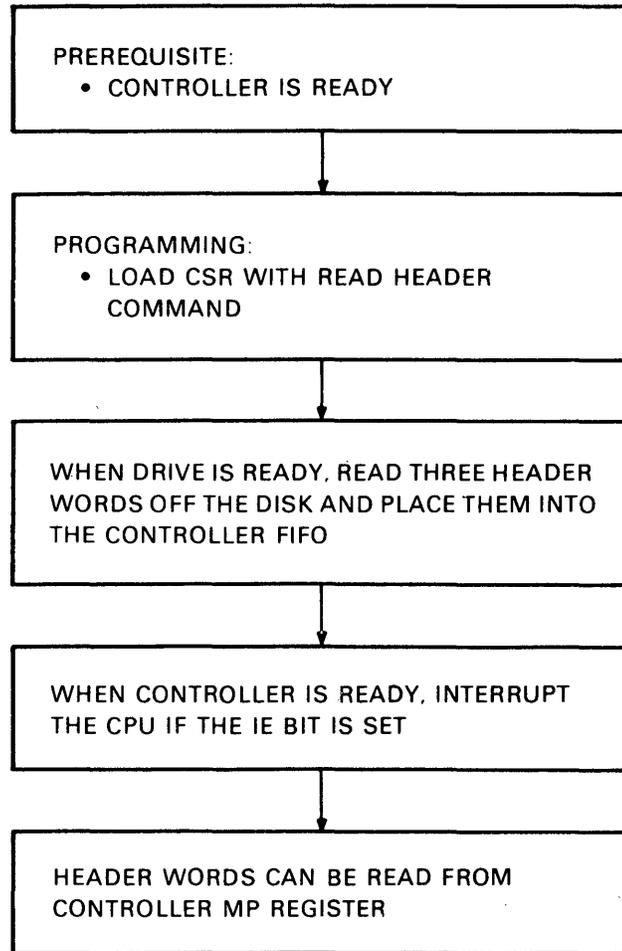
One prerequisite for issuing the get status command is a knowledge that the controller is in the ready state. It is important to note that the drive does not have to be ready (for example, during a seek or when in the load state) to issue a get status command.

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

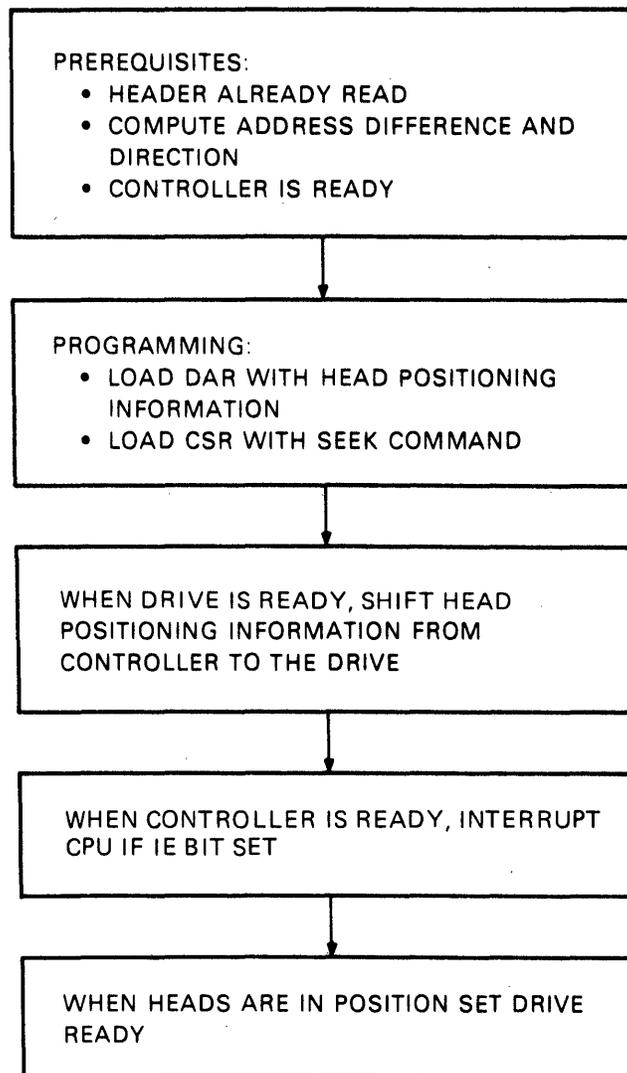
Figure 2-7 Simplified Read Header Operation

## 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. The 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.

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.

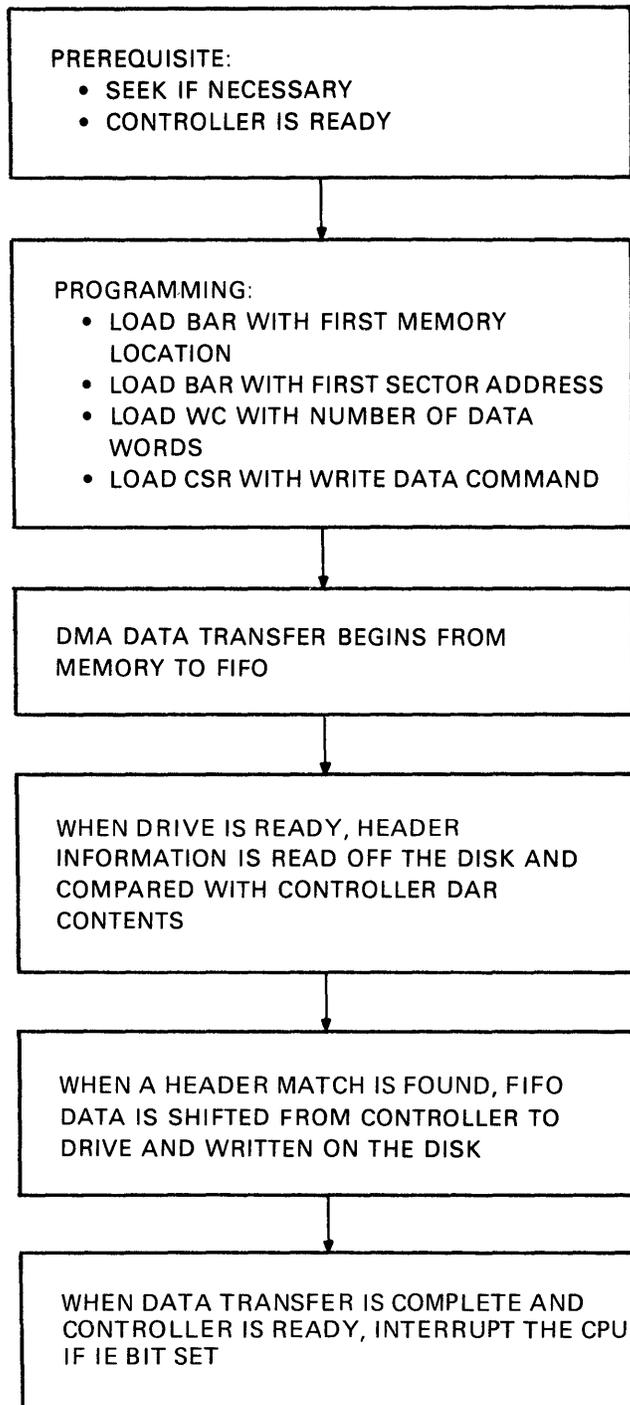
## **2.10 READ DATA FUNCTION**

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.

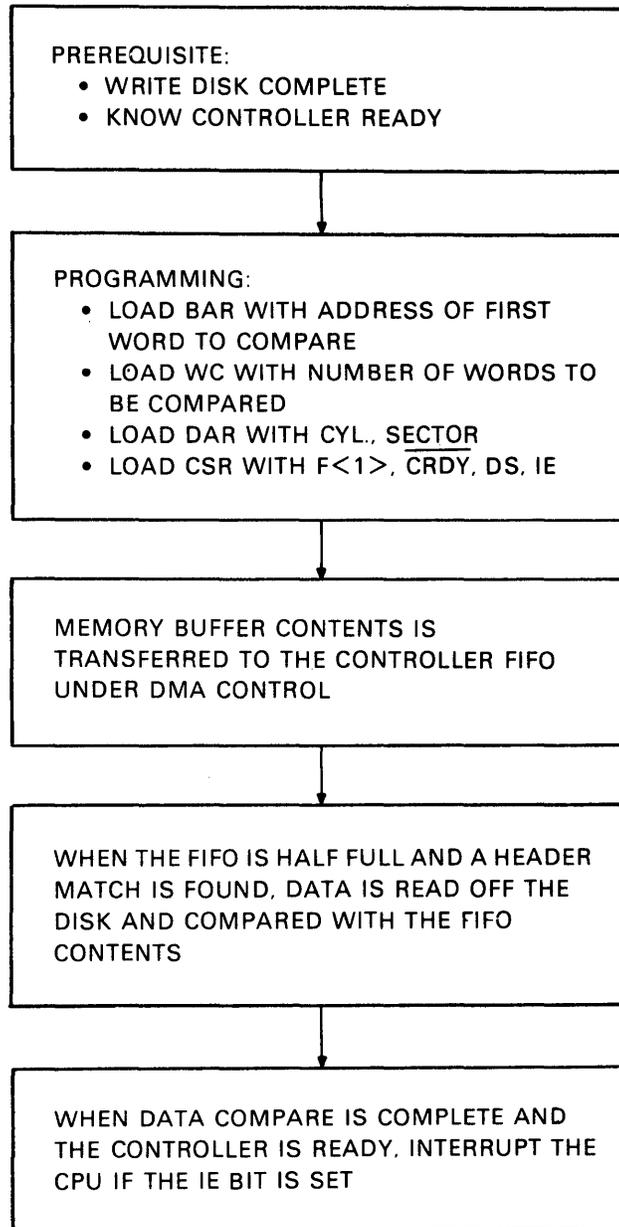
The programming prerequisites are as follows.

- 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.



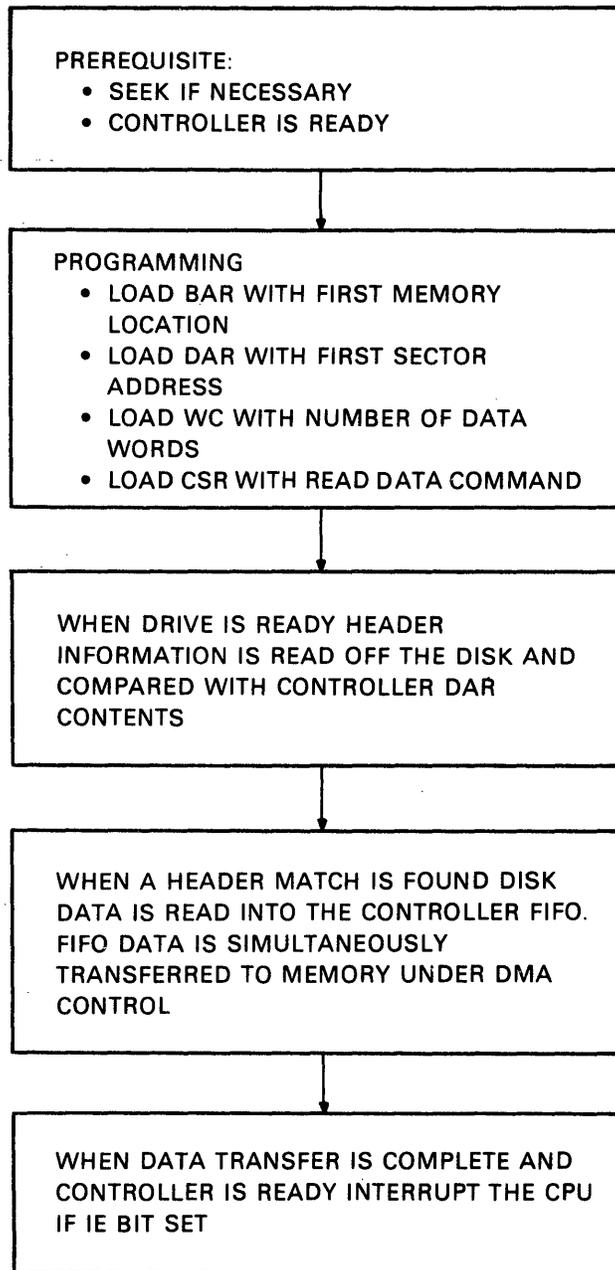
MA-1106

Figure 2-9 Simplified Write Data Operation



MA-1103

Figure 2-10 Simplified Write Check Operation

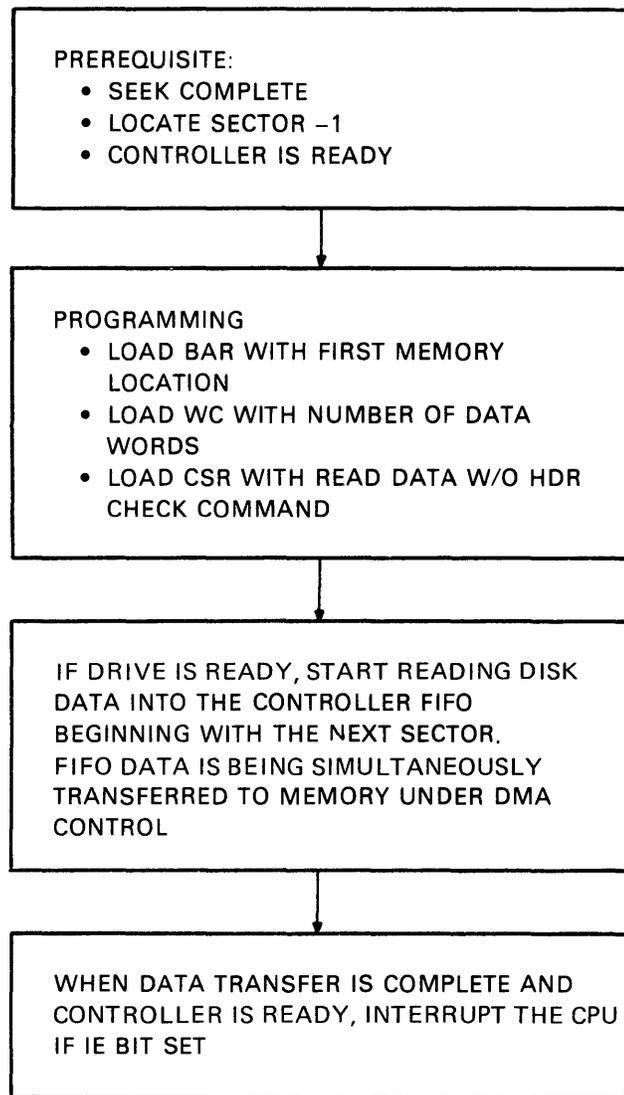


MA-1104

Figure 2-11 Simplified Read Data Operation

### 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.



MA-1105

Figure 2-12 Simplified Read Data Without Header Check Operation

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.

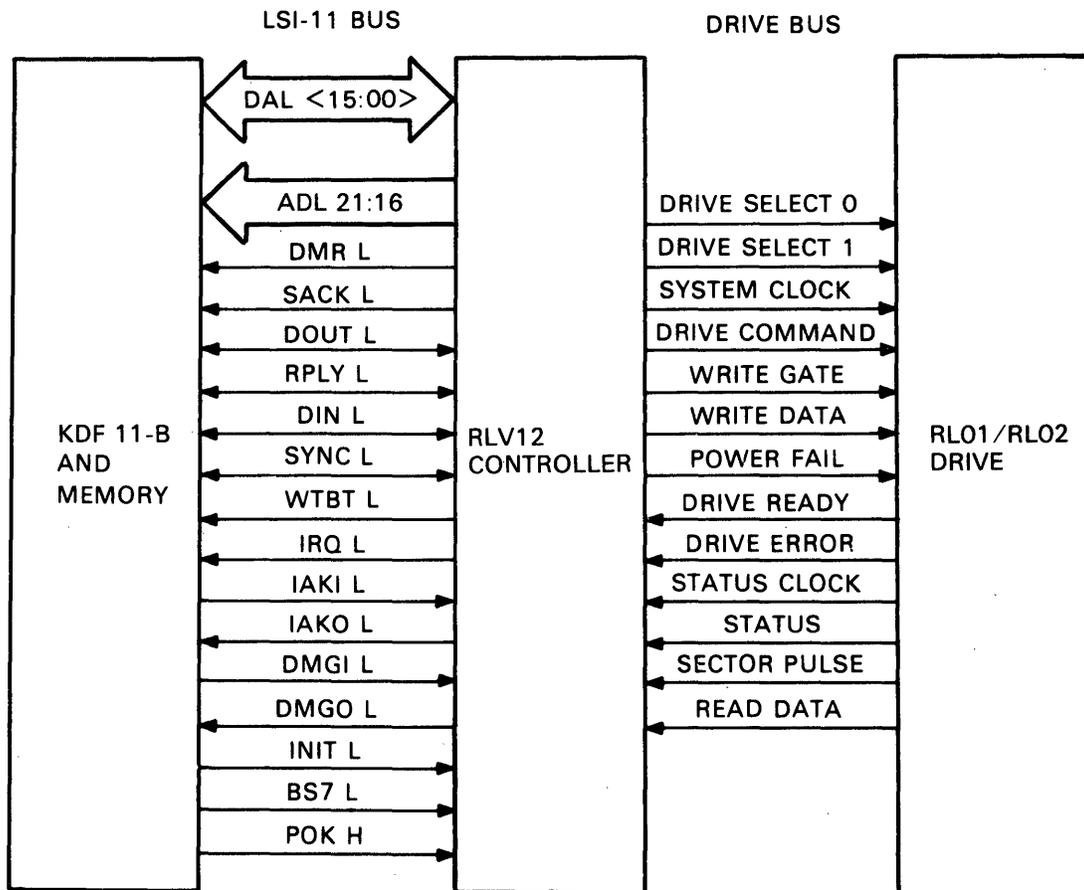


## CHAPTER 3 INTERFACE LEVEL DESCRIPTION

### 3.1 GENERAL

The RLV12 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.



CZ-0511

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 communicate with the RLV12 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.

## **3.3 DRIVE BUS INTERFACE SIGNALS**

### **3.3.1 General**

The drive bus is the avenue of communication between the RLV12 controller and the RL01 or RL02 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.

### **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 RLV12 and reflects the state of power in the logic box holding the RLV12 modules. This signal is received by all drives at all times, regardless of which drive is selected. While the RLV12 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.

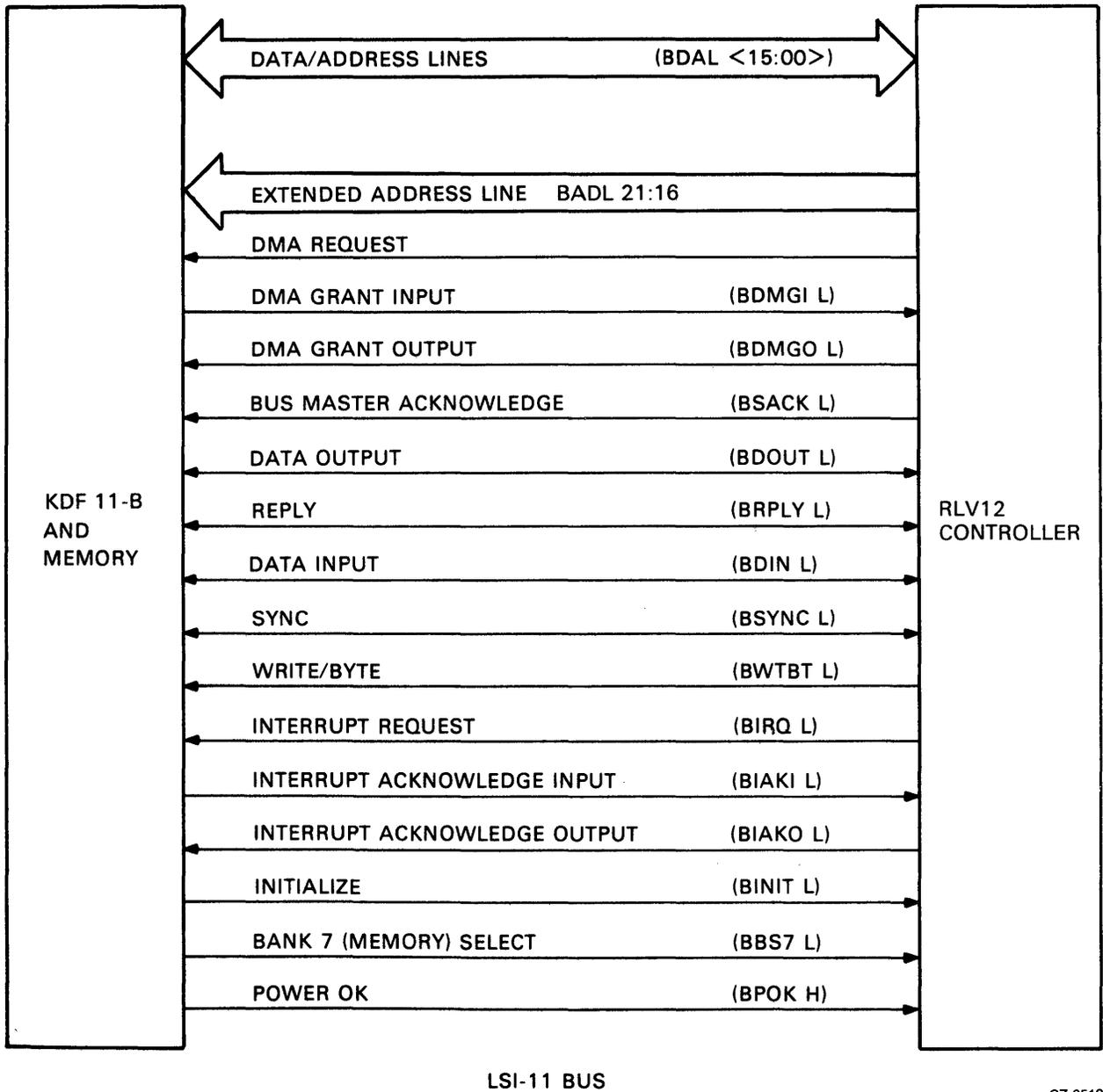


Figure 3-2 LSI-11 Bus Signals

**Table 3-1 LSI-11 Bus Signals**

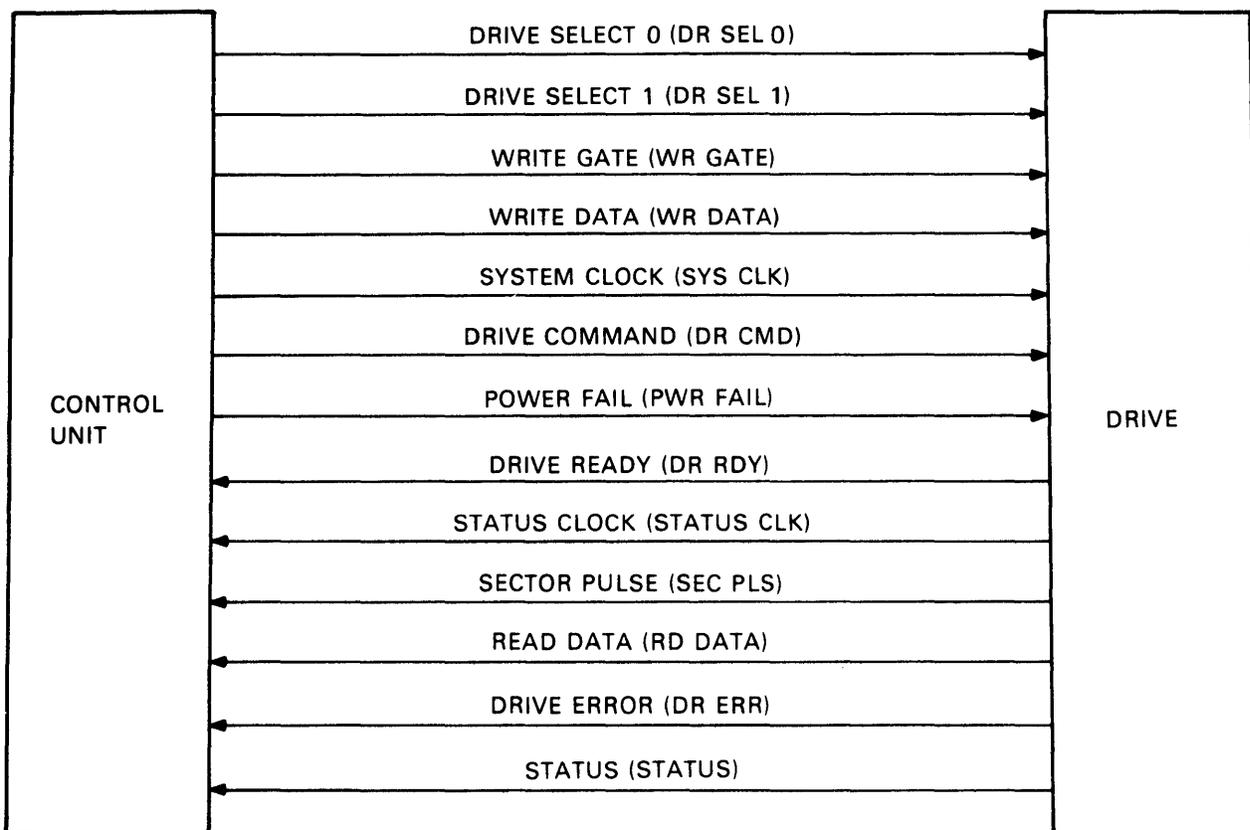
Mnemonic	Description
BAD16-17	Extended address bits.
GND	<b>Ground</b> – System signal ground and dc return.
GND	<b>Ground</b> – Signal ground and dc return.
BDMR L	<b>Direct Memory Access (DMA) Request</b> – The RLV12 asserts this signal to request bus mastership. The processor arbitrates bus mastership 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 RLV12 responds by negating BDMR L and asserting BSACK L. DMA is used for transferring words in and out of the FIFO.
GND	<b>Ground</b> – System signal ground and dc return.
BDCOK H	<b>DC Power OK</b> – Power supply-generated signal that is asserted when there is sufficient dc voltage available to sustain reliable system operation.
BPOK H	<b>Power OK</b> – Asserted by the power supply when primary power is normal. RLV12 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 RLV12 DMA transfers.
BDAL18-21	Extended address bits for 22-bit addressing modes. The extended LSI-11 backplane must be used (M9275 or H9276) in conjunction with the MSV-11 (M8067) memory module.
GND	<b>Ground</b> – System signal ground and dc return.
GND	<b>Ground</b> – System signal ground and dc return.
BSACK L	This signal is asserted by the RLV12 in response to the processor's BDMGO L signal, acknowledging that the RLV12 is bus master.
GND	<b>Ground</b> – System signal ground and dc return.
+5	<b>+5 V Power</b> – +5 Vdc system power.
+5	<b>+5 V Power</b> – Normal +5 Vdc system power.
GND	<b>Ground</b> – System signal ground and dc return.
+12	<b>+12 V Power</b> – +12 Vdc system power.

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

Mnemonic	Description
BDOUT L	<b>Data Output</b> – BDOUT, when asserted, implies that valid data is available on BDALO – 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.
BRPLY L	<b>Reply</b> – BRPLY L is asserted in response to BDIN L or BDOUT L and during IAK transactions. It is generated by RLV12 to indicate that it has input data available on the BDAL bus or that it has accepted output data from the bus.
BDIN L	<p><b>Data Input</b> – BDIN L is used for two types of bus operations:</p> <ol style="list-style-type: none"> <li>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 RLV12 is ready to accept data as bus master.</li> <li>2. When asserted without BSYNC L, it indicates that an interrupt operation is occurring.</li> </ol>
BSYNC L	<b>Synchronize</b> – BSYNC L is asserted by the bus master device to indicate that it has placed an address on BDALO – 15 L. The transfer is in process until BSYNC L is negated.
BWTBT L	<b>Write/Byte</b> – 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.
BIRQ L	<b>Interrupt Request</b> – RLV12 asserts this signal when its interrupt enable and interrupt request flip-flops are set. This signal informs the processor that the RLV12 has completed a function or it is ready to accept a new function. If the processor PS word bit 7 is 0, the processor responds by acknowledging the request by asserting BDIN L and BIAKO L.
BIAKI-O L	<b>Interrupt Acknowledge Input and Interrupt Acknowledge Output</b> – 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 RLV12. If it is requesting an interrupt, it inhibits passing BIAKO L. If it is not asserting BIRQ L, RLV12 passes BIAKI L to the next (lower priority) device via its BIAKO L pin and the lower priority device's BIAKI L pin.
BBS7 L	<b>Bank 7 Select</b> – 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 RLV12 is in this bank.

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

Mnemonic	Description
BDMGI-O L	<p><b>DMA Grant-Input and DMA Grant-Output</b> – 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 RLV12. If it is requesting the bus, it inhibits passing BDMGO L. If it is not requesting the bus, RLV12 passes the BDMGO 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.</p> <p style="text-align: center;"><b>NOTE</b></p> <p style="text-align: center;"><b>The RLV12 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.</b></p>
BINIT L	<p><b>Initialize</b> – BINIT is asserted by the processor to initialize or clear all registers and errors in the RLV12 except drive error (and composite error if drive 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.</p>
BDAL0-1 L	<p><b>Data/Address Lines</b> – 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.</p>
+5	<p><b>+5 V Power</b> – Normal +5 Vdc system power.</p>
GND	<p><b>Ground</b> – System signal ground and dc return.</p>
+12	<p><b>+12 V Power</b> – +12 V system power.</p>
BDAL2-15 L	<p><b>Data/Address Lines</b> – These 14 lines are part of the 22-line data/address bus previously described.</p>



MA-0580

Figure 3-3 Drive Bus Signals

**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 62.5 microsecond pulse is asserted high and occurs every 625 microseconds 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.

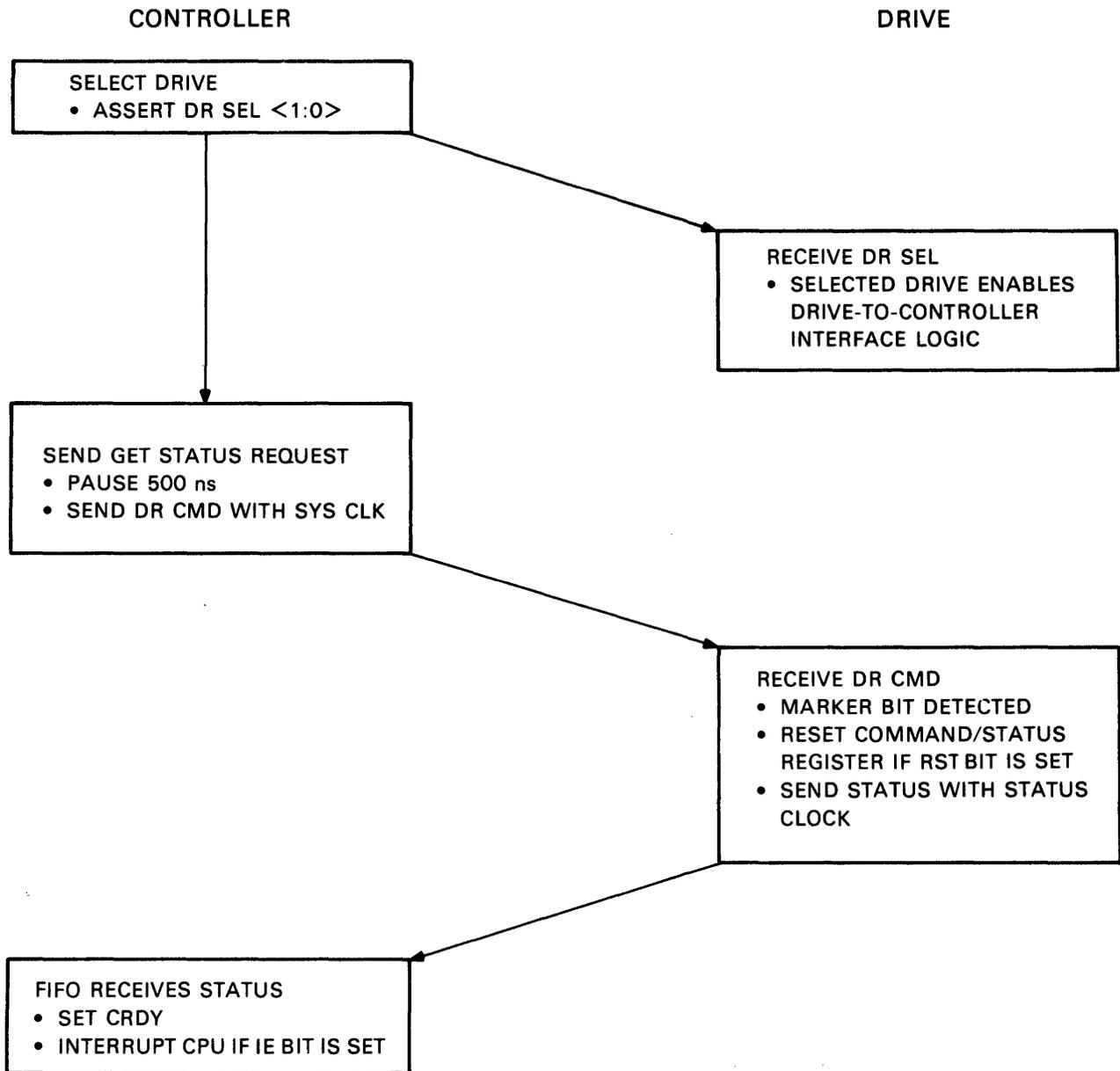
**3.3.2.12 Read Data (RD DATA)** – This line transfers MFM encoded data from the drive read circuits to the controller. Whenever a drive is selected and DR RDY is asserted, RD DATA appears on this line, except when WR GATE is asserted.

The drive senses the amplitude of the header preamble and sends RD DATA over the RD DATA line  $2.5 \pm 0.5$  microseconds 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.

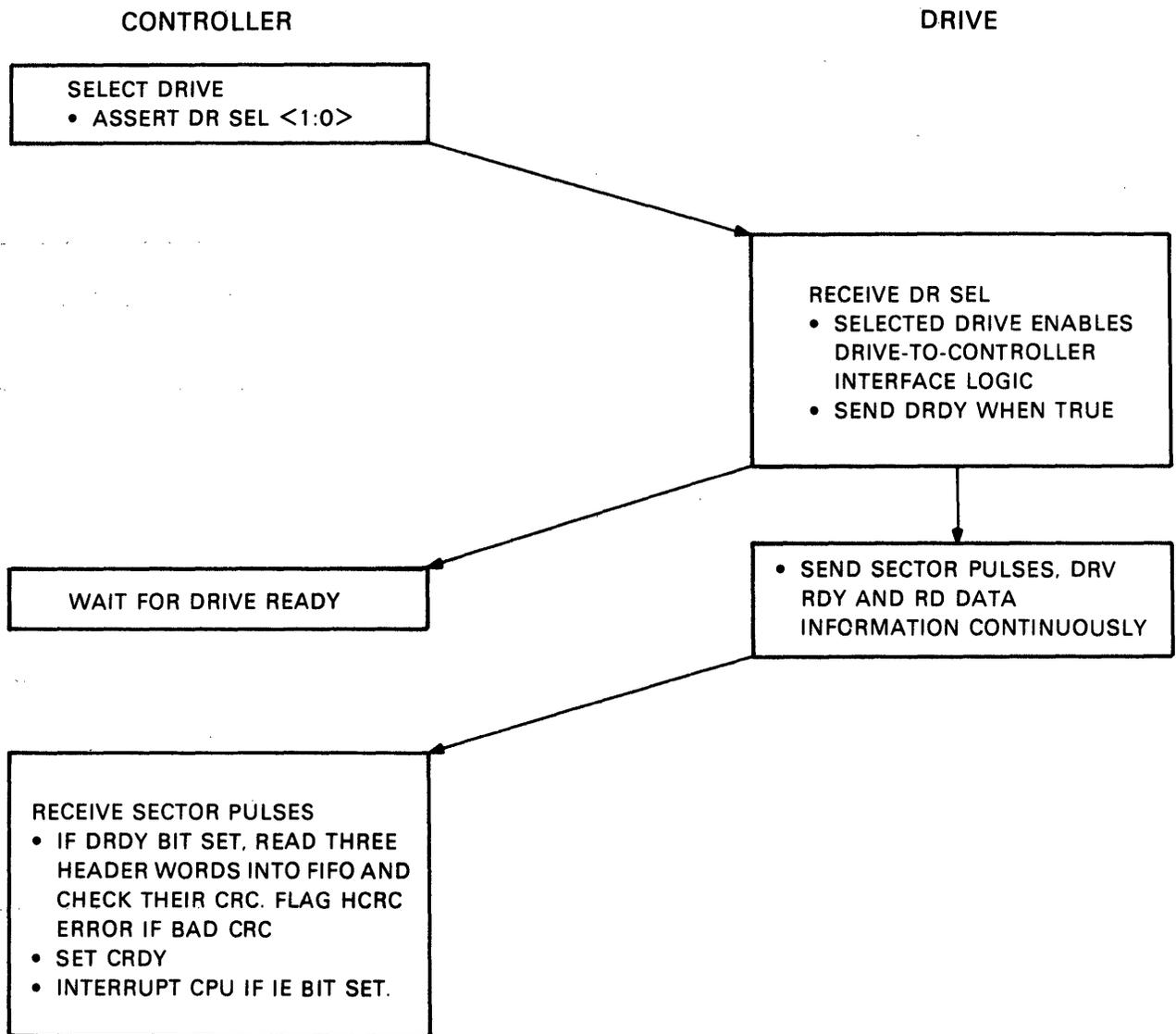
### **3.3.3 Drive Bus Dialogue**

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.



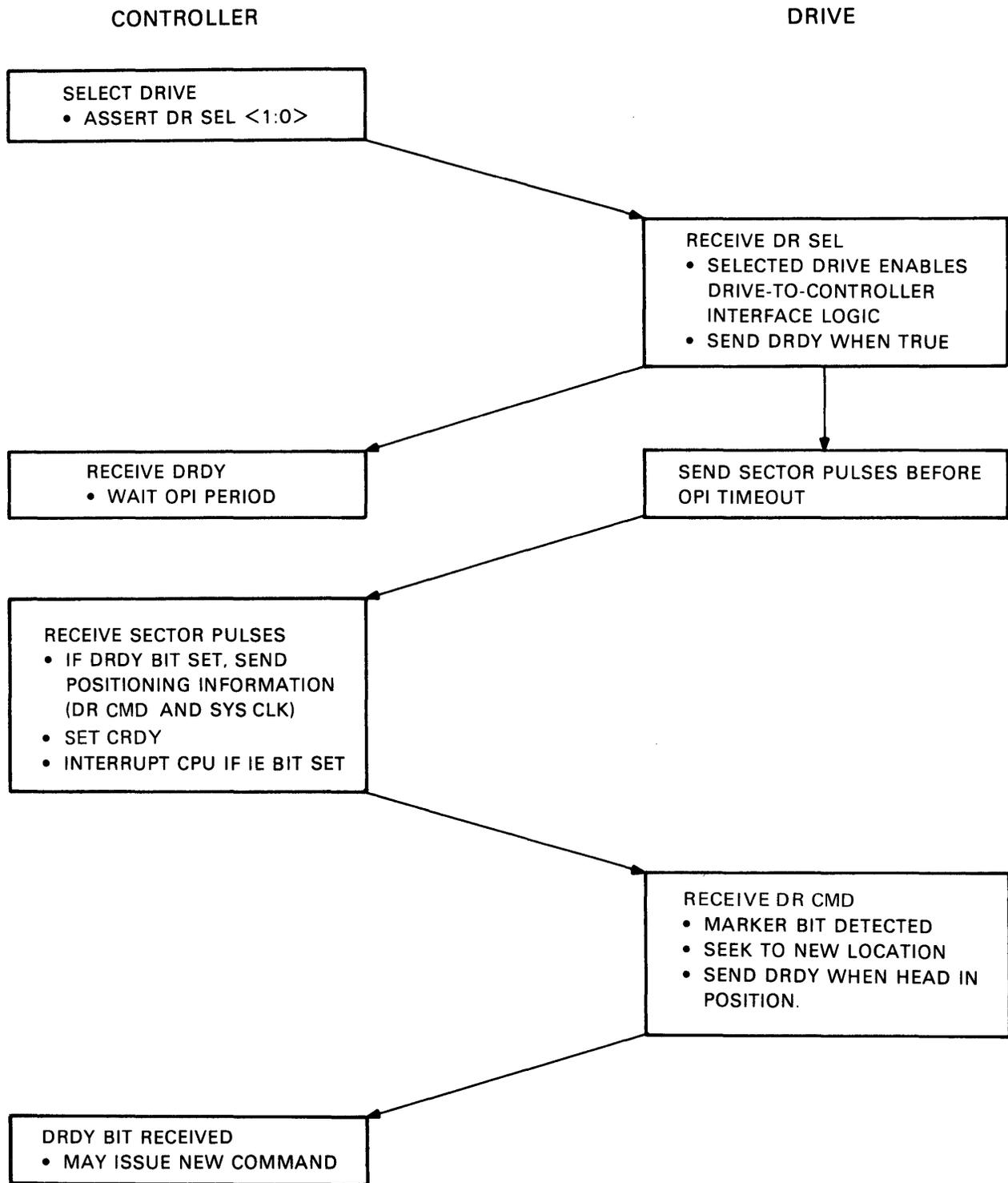
MA-1117

Figure 3-4 Get Status Drive Bus Sequence



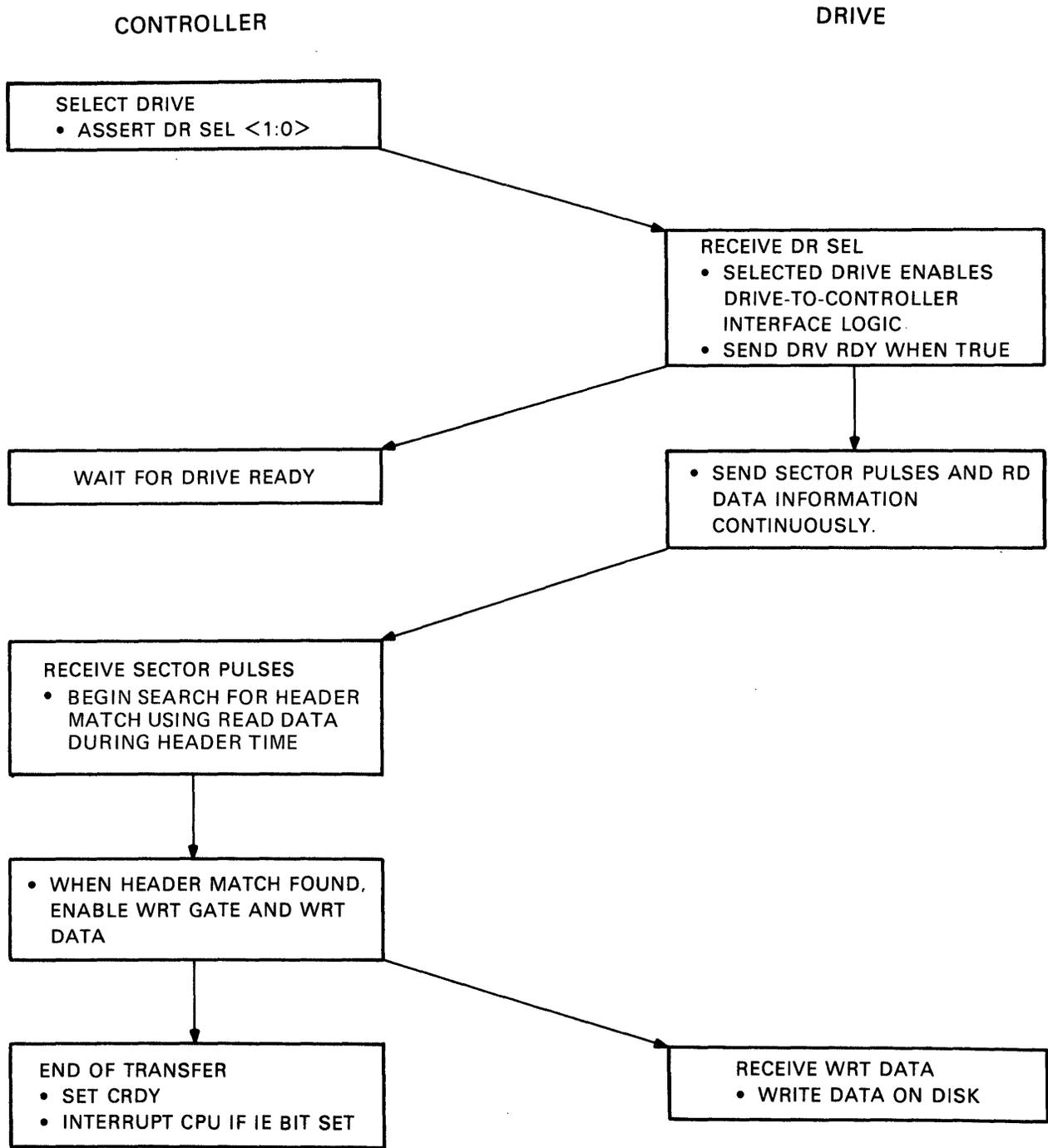
MA-1118

Figure 3-5 Read Header Drive Bus Sequence



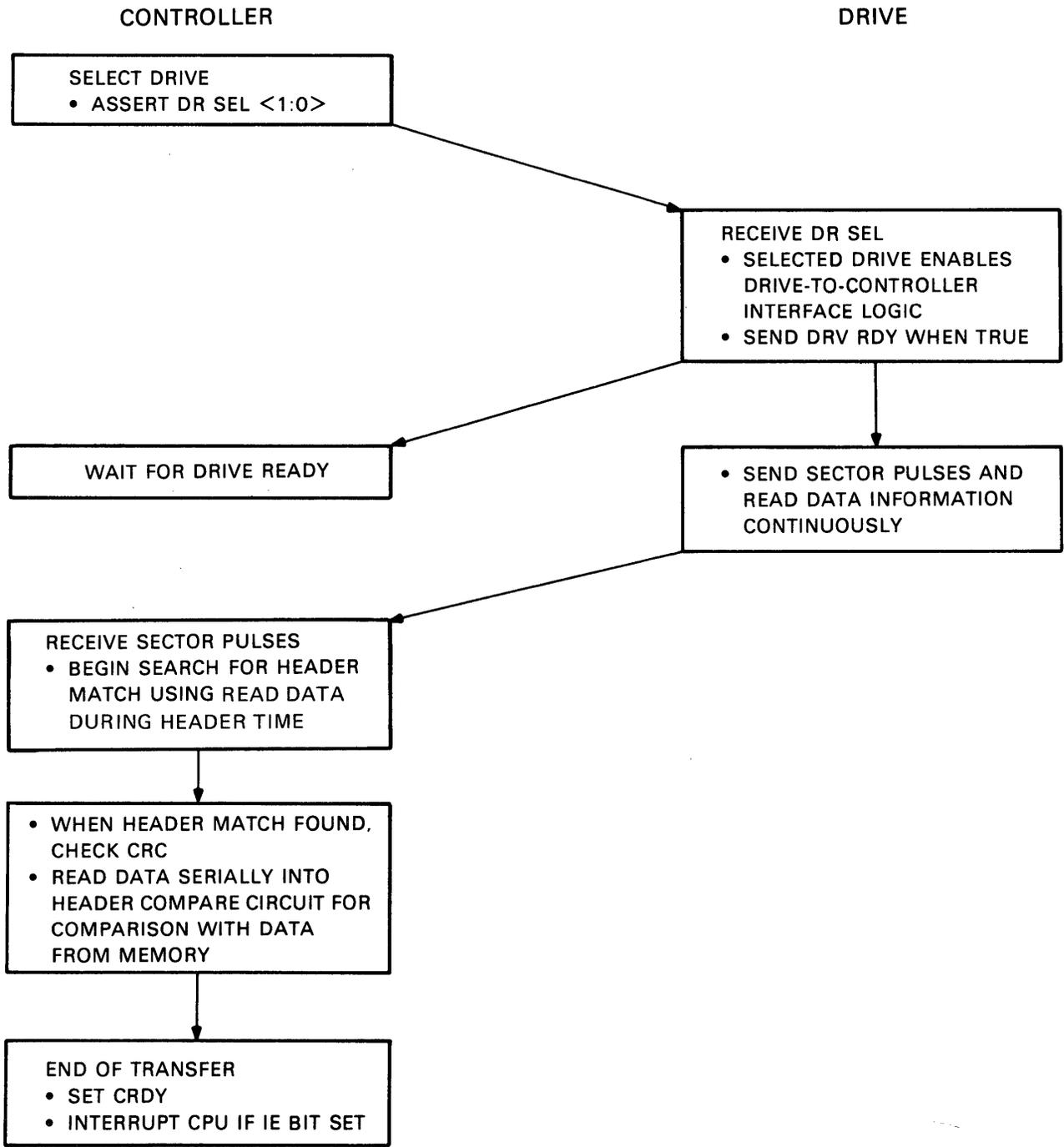
MA-1146

Figure 3-6 Seek Drive Bus Sequence



MA-1145

Figure 3-7 Write Data Drive Bus Sequence

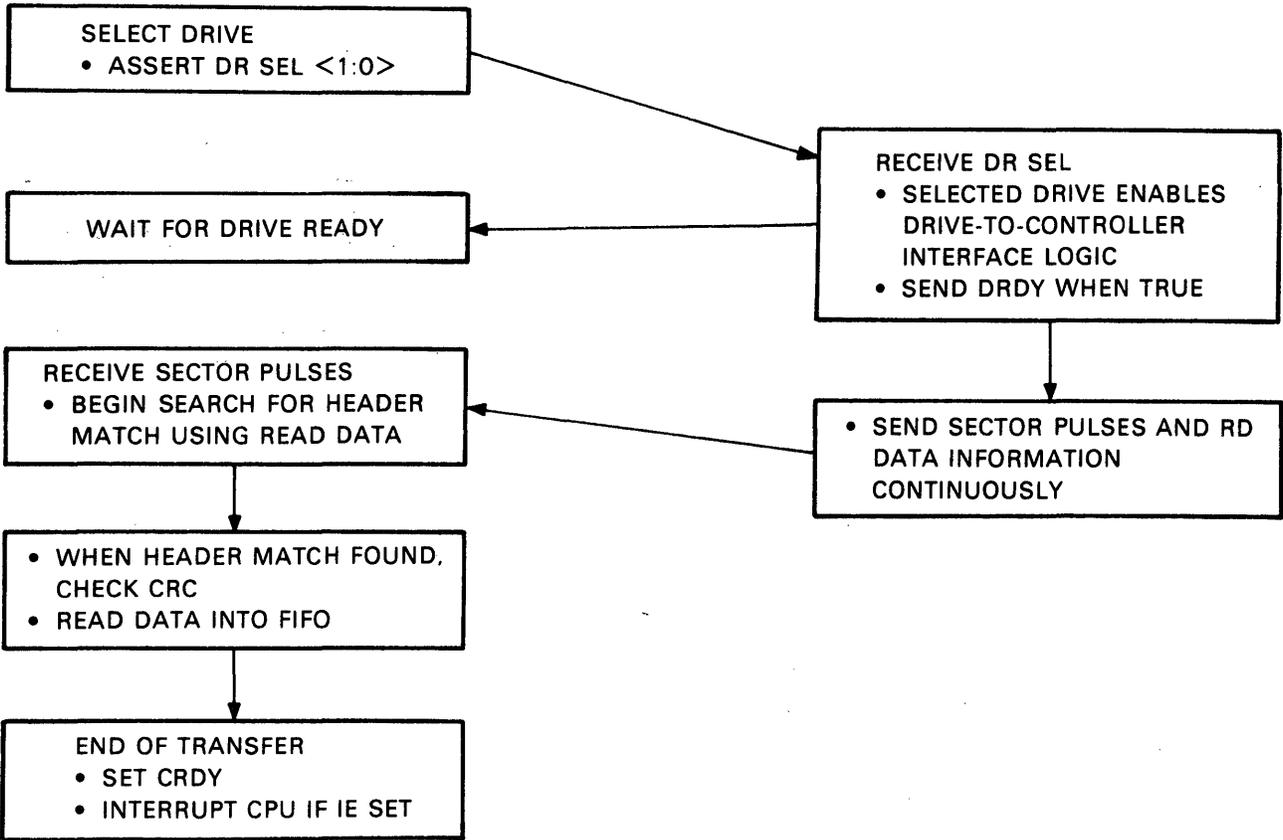


MA-1144

Figure 3-8 Write Check Drive Bus Sequence

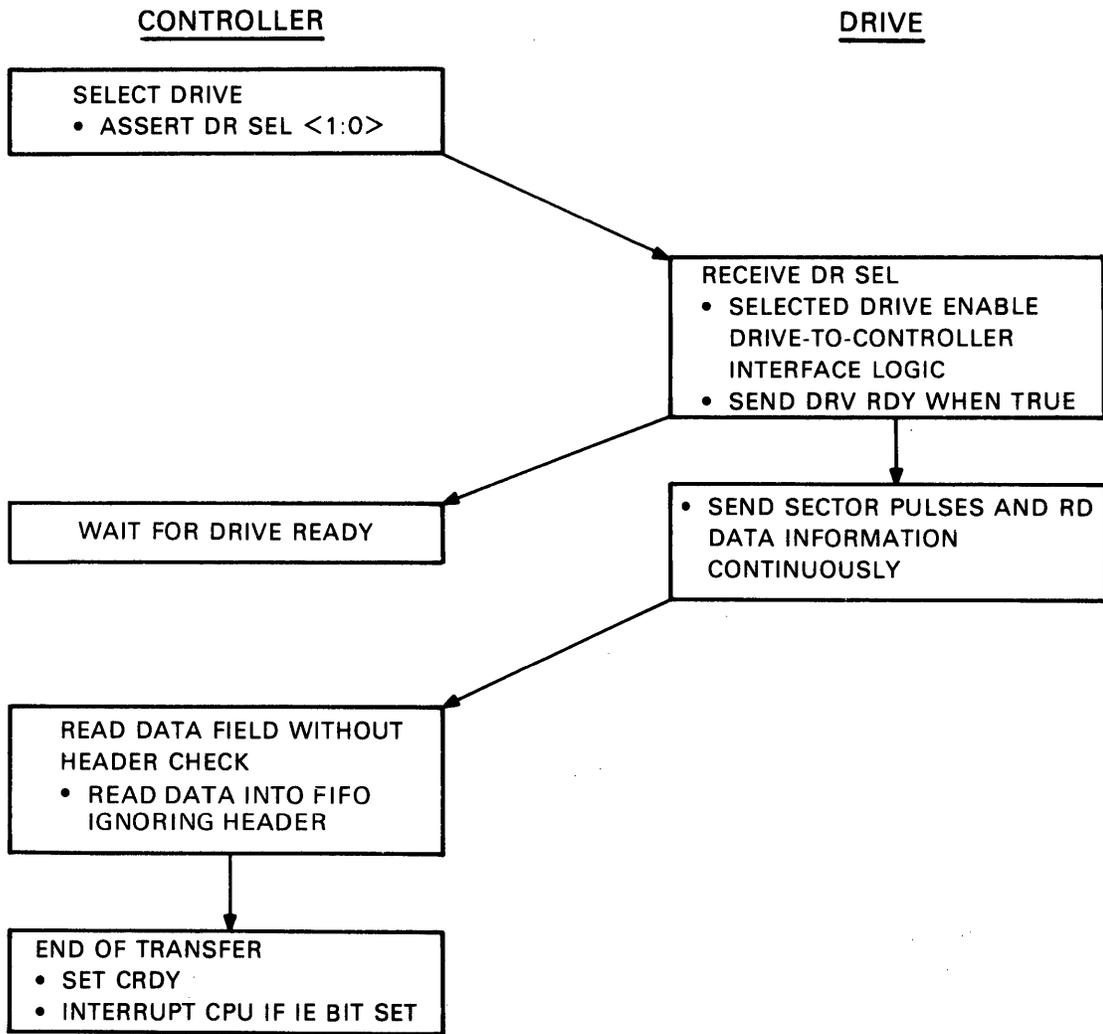
CONTROLLER

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 RLV12 controller has five 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 is reserved for I/O device addresses. Within this range, each RLV12 register has a unique address assigned as shown in Table 3-2.

**Table 3-2 Register Addresses**

LSI-11 Bus Address	Register Name
174400	Control status register
174402	Bus address register
174404	Disk address register
174406	Multipurpose register
1774410	Bus address extension

**NOTE**

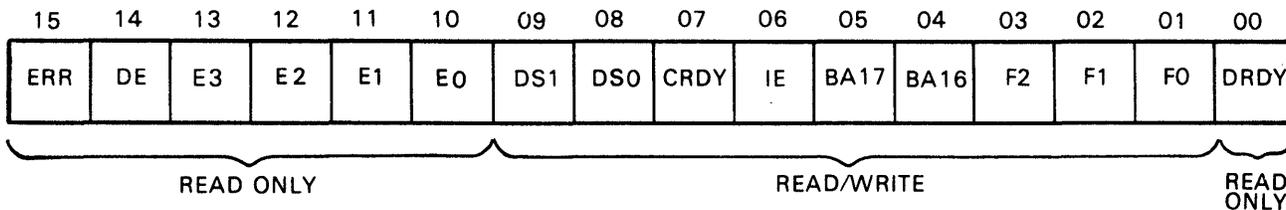
**Only the lower 12 bits of the addresses are decoded. The upper 10 bits are asserted with the signal BANK SEL 7.**

#### 3.4.1 Control Status Register

The Control Status (CS) register (Figure 3-11) is a 16-bit word-addressable register with a base address of 774400. Bits 1 through 9 can be read or written; the other bits can only be read. Table 3-3 describes the bit format of the Control Status register.

When the controller is initialized, 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 set whenever there is a drive error; it is cleared when 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 (indicated in bits 10-14).

**CONTROL STATUS REGISTER (CSR)**



CZ-2009

**Figure 3-11 Control Status Register**

**Table 3-3 Control Status Register  
Bit Description**

Bit(s)	Description																		
0	<p><b>Drive Ready (DRDY)</b> – When set, this bit indicates that the selected drive is ready to receive a command. The bit is cleared when a seek or head-select operation is initiated and set when the operation is completed.</p>																		
1-3	<p><b>Function Code</b> – These bits are set by software to indicate the command to be executed.</p> <table border="0" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: left;">Command</th> <th style="text-align: left;">F2-F0</th> </tr> </thead> <tbody> <tr> <td>No Op (RL11) or Maintenance Mode (RLV11/RLV12)</td> <td>000</td> </tr> <tr> <td>Write Check</td> <td>001</td> </tr> <tr> <td>Get Status</td> <td>010</td> </tr> <tr> <td>Seek</td> <td>011</td> </tr> <tr> <td>Read Header</td> <td>100</td> </tr> <tr> <td>Write Data</td> <td>101</td> </tr> <tr> <td>Read Data</td> <td>110</td> </tr> <tr> <td>Read Data Without Header Check</td> <td>111</td> </tr> </tbody> </table>	Command	F2-F0	No Op (RL11) or Maintenance Mode (RLV11/RLV12)	000	Write Check	001	Get Status	010	Seek	011	Read Header	100	Write Data	101	Read Data	110	Read Data Without Header Check	111
Command	F2-F0																		
No Op (RL11) or Maintenance Mode (RLV11/RLV12)	000																		
Write Check	001																		
Get Status	010																		
Seek	011																		
Read Header	100																		
Write Data	101																		
Read Data	110																		
Read Data Without Header Check	111																		
4-5	<p><b>Bus Address Extension Bits (BA16, BA17)</b> – The two most significant bus address bits when operating in 18-bit addressing modes. Read and written as data bits 4 and 5 of the CS register but considered as address bits 16 and 17 of the bus address register (see Paragraph 3.4.2).</p>																		
6	<p><b>Interrupt Enable (IE)</b> – When this bit is set by software, the controller is allowed to interrupt the processor at the normal command or error termination.</p>																		
7	<p><b>Controller Ready (CRDY)</b> – When cleared by software, this bit indicates that the command code in bits 1-3 is to be executed (negative GO bit). When set, this bit indicates the controller is ready to accept another command.</p>																		
8-9	<p><b>Drive Select (DS0, DS1)</b> – These bits determine which drive will communicate with the controller via the drive bus.</p>																		

**Table 3-3 Control Status Register Bit Description (Cont)**

Bit(s)	Description																
10-13	<p><b>Error Code</b></p> <table border="0"> <thead> <tr> <th data-bbox="334 527 410 583">Error Name</th> <th data-bbox="837 558 914 583">E3-E0</th> </tr> </thead> <tbody> <tr> <td data-bbox="334 621 610 678">Operation Incomplete (OPI)</td> <td data-bbox="837 653 898 678">0001</td> </tr> <tr> <td data-bbox="334 716 634 810">Read Data CRC (DCRC or Write Check Error (WCE))</td> <td data-bbox="837 779 898 804">0010</td> </tr> <tr> <td data-bbox="334 842 618 867">Header CRC (HCRC)</td> <td data-bbox="837 842 898 867">0011</td> </tr> <tr> <td data-bbox="334 905 553 930">Data Late (DLT)</td> <td data-bbox="837 905 898 930">0100</td> </tr> <tr> <td data-bbox="334 968 570 1024">Header Not Found (HNF)</td> <td data-bbox="837 999 898 1024">0101</td> </tr> <tr> <td data-bbox="334 1062 618 1119">Non-Existent Memory (NXM)</td> <td data-bbox="837 1094 898 1119">1000</td> </tr> <tr> <td data-bbox="334 1157 602 1213">Memory Parity Error (MPE) RLV12 only</td> <td data-bbox="837 1188 898 1213">1001</td> </tr> </tbody> </table>	Error Name	E3-E0	Operation Incomplete (OPI)	0001	Read Data CRC (DCRC or Write Check Error (WCE))	0010	Header CRC (HCRC)	0011	Data Late (DLT)	0100	Header Not Found (HNF)	0101	Non-Existent Memory (NXM)	1000	Memory Parity Error (MPE) RLV12 only	1001
Error Name	E3-E0																
Operation Incomplete (OPI)	0001																
Read Data CRC (DCRC or Write Check Error (WCE))	0010																
Header CRC (HCRC)	0011																
Data Late (DLT)	0100																
Header Not Found (HNF)	0101																
Non-Existent Memory (NXM)	1000																
Memory Parity Error (MPE) RLV12 only	1001																
14	<p><b>Drive Error (DE)</b> – This bit is tied directly to the DE interface line. When set, it indicates that the selected drive has flagged an error. (The source of the error can be determined by executing a get status command and then executing an MPR read.)</p> <p>DE can be cleared by executing a get status command with bit 3 of the DA register set.</p>																
15	<p><b>Composite Error</b> – When set, this bit indicates that one or more of the error bits (bits 10-14) is set. If the IE bit (bit 6 of CS) is set and an error occurs (which sets bit 7), an interrupt will be initiated.</p>																

**3.4.2 Bus Address Register**

The Bus Address (BA) register (Figure 3-12) is a 16-bit word-addressable register with an address of 774402. Bits 1 through 15 can be read or written; bit 0 is always zero. Bus address bits 16 and 17 are contained in bits 4 and 5 of the CS register.

**BUS ADDRESS REGISTER (BAR)**

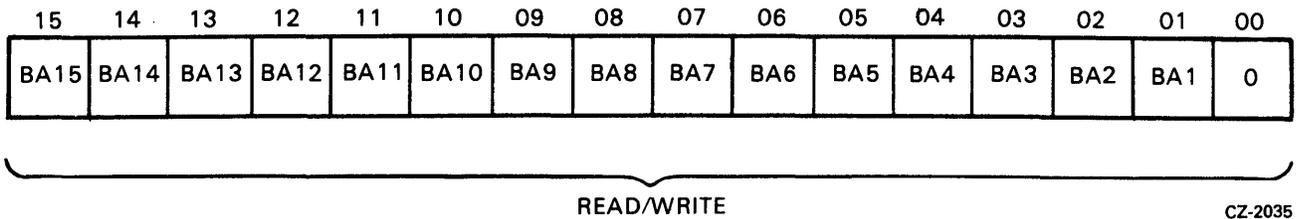


Figure 3-12 BA Register

The BA register indicates the memory location involved in the data transfer during a normal read or write operation. The contents of the BA register are automatically incremented by two as each word is transferred between the bus and the I/O buffer. This register overflows into CS register bits 4 and 5.

Table 3-4 describes the bit format of the Bus Address register.

The BA register is cleared by initializing the drive or by loading the register with zeros.

**Table 3-4 Bus Address Register Bit Description**

Bit(s)	Description
0-15	<b>BA0 thru BA15</b> – These bits point to the memory address that data is to be transferred to/from. BA16 and BA17 are in CSR bits 4 and 5. If the controller is an RLV12 and 22-bit addressing mode is enabled, then bits BA16 through 21 can be found in the BAE register.

**3.4.3 Disk Address Register**

The Disk Address (DA) register is a 16-bit register with an address of 774404. Its contents can have one of three meanings depending on the function being performed. This register is cleared by initializing the device or loading the register with zeros. All 16 bits can be read or written by the processor.

**3.4.3.1 DA Register During a Seek Command** – To perform a seek function, it is necessary to provide cylinder address difference, head select, and head directional information to the selected drive. Figure 3-13 shows the bit layout of the Disk Address register during seek commands, while Table 3-5 describes the bit format.

DAR DURING SEEK COMMAND

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
DF8	DF7	DF6	DF5	DF4	DF3	DF2	DF1	DF0	0	0	HS	0	DIR	0	1

CZ-2010

Figure 3-13 DAR Contents to Execute a Seek Command

Table 3-5 Disk Address Register Bit Description for Seek Commands

Bit(s)	Description
0	<b>Marker Bit</b> – Must be a 1.
1	<b>Seek</b> – Must be a 0, indicating to the drive that a seek is being requested. With this bit cleared, the drive uses the remaining contents of the register as seek parameters.
2	<b>Direction (DIR)</b> – 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 7-15).
3	Must be a 0.
4	<b>Head Select (HS)</b> – Indicates which head (disk surface) is to be selected. A one selects the lower head; a zero, the upper head.
5-6	Reserved.
7-15	<b>Cylinder Address Difference DF 08:00</b> – Indicates the number of cylinders the heads are to move on a seek.

**3.4.3.2 DA Register During Read or Write Data Command** – For a read or write operation, the DA register is loaded with the address of the first sector to be transferred. As each successive sector is transferred, the DA register is automatically incremented. The contents of this register are used by the header comparison logic to locate the desired sector. The header read from the disk is compared against the contents of this register.

Figure 3-14 shows the bit layout of the Disk Address register during data transfer commands, while Table 3-6 describes the bit format.

DAR DURING READING OR WRITING DATA COMMANDS

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CA8	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0	HS	SA5	SA4	SA3	SA2	SA1	SA0

CZ-2011

Figure 3-14 DAR Contents During a Read/Write Data Command

Table 3-6 Disk Address Register Bit Description for Data Transfer Commands

Bit(s)	Description
0-5	<b>Sector Address SA 05:00</b> – Desired address of one of the 40 sectors on a track as supplied by the software (range is 0 through 47, octal).
6	<b>Head Select (HS)</b> – Desired head address of one of the two drive heads. A one indicates the lower head; a zero, the upper head.
7-15	<b>Cylinder Address CA 08:00</b> – Desired address of one of the cylinders on the disk (range is 0 through 777, octal). The RL01 has 256 cylinders and the RLV12 has 512 cylinders.

3.4.3.3 DA Register During a Get Status Command – For a get status command, the DA register bits must be programmed as shown by Figure 3-15 and described in Table 3-7.

DAR DURING GET STATUS COMMAND

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
X	X	X	X	X	X	X	X	0	0	0	0	RST	0	1	1

CZ-2037

Figure 3-15 DAR Contents to Execute a Get Status Command

**Table 3-7 Disk Address Register Bit Description for Get Status Commands**

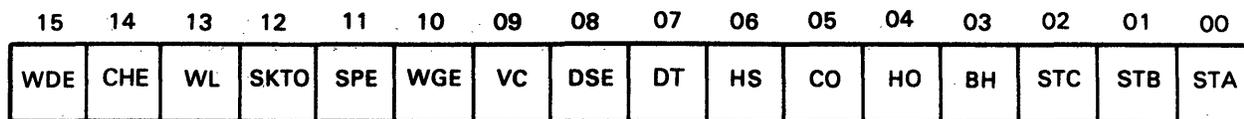
Bit(s)	Description
0	<b>Marker Bit</b> – Must be a 1.
1	<b>Get Status (GS)</b> – Must be a 1, indicating to the drive that the status word is being requested. At the completion of the get status command, the drive status word is read into the controller Multipurpose (MP) register (see Paragraph 3.2.4). With this bit set, the drive ignores bits 8-15.
2	Must be a 0.
3	<b>Reset (RST)</b> – When this bit is set, the drive clears its error register (resets all drive faults) before sending the status word to the controller.
4-7	Must be a 0.
8-15	Not used during a get status command.

**3.4.4 Multipurpose Register**

The multipurpose (MP) register is a 16 bit register with an address of 774406. Its contents can have one of three meanings, depending on the function being performed.

**3.4.4.1 MP Register During a Get Status Command** – When a get status command is executed, the status word is returned to the controller and transferred to the MP register. Figure 3-16 shows the bit layout while Table 3-8 describes the bit format.

**MPR AFTER GET STATUS COMMAND**



CZ-2012

**Figure 3-16 MPR – Following a Get Status Command**

**Table 3-8 MP Register Bit Description for Get Status Commands**

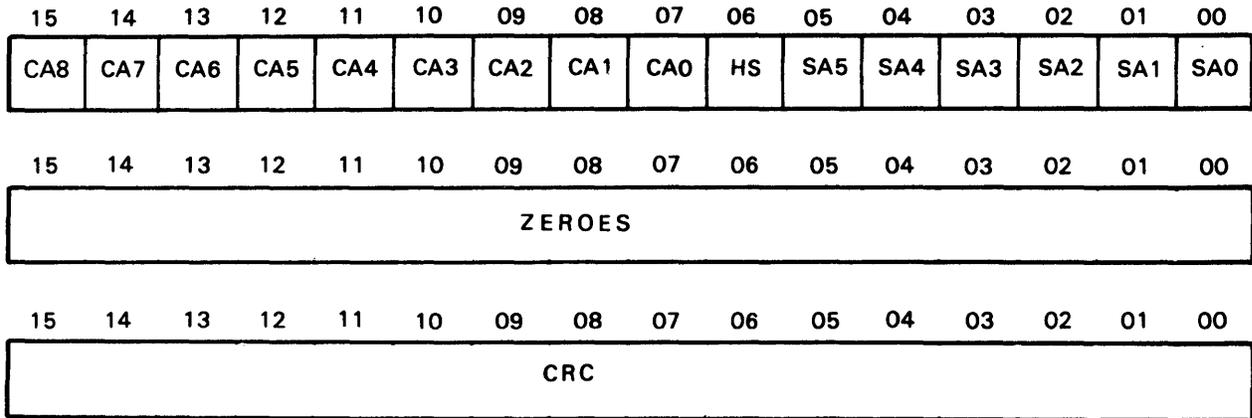
Bit(s)	Description																																				
0-2	<p><b>State C:A ST C:A</b> – These bits define the state of the drive.</p> <table border="0" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: center;">STC</th> <th style="text-align: center;">STB</th> <th style="text-align: center;">STA</th> <th></th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Load cartridge</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>Spin-up</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>Brush cycle</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>Load heads</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Seek</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>Lock on</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>Unload heads</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>Spin down</td> </tr> </tbody> </table>	STC	STB	STA		0	0	0	Load cartridge	0	0	1	Spin-up	0	1	0	Brush cycle	0	1	1	Load heads	1	0	0	Seek	1	0	1	Lock on	1	1	0	Unload heads	1	1	1	Spin down
STC	STB	STA																																			
0	0	0	Load cartridge																																		
0	0	1	Spin-up																																		
0	1	0	Brush cycle																																		
0	1	1	Load heads																																		
1	0	0	Seek																																		
1	0	1	Lock on																																		
1	1	0	Unload heads																																		
1	1	1	Spin down																																		
3	<b>Brush Home (BH)</b> – Set when the brushes are home on those RL drives that have brushes.																																				
4	<b>Heads Out (HO)</b> – Set when the heads are over the disk.																																				
5	<b>Cover Open (CO)</b> – Set when the drive access cover is open or the dust cover is not in place.																																				
6	<b>Head Select (HS)</b> – Indicates the currently selected head. A zero indicates the upper head; a one, the lower head.																																				
7	<b>Drive Type (DT)</b> – A zero indicates an RL01; a one, an RL02.																																				
8	<b>Drive-Select Error (DSE)</b> – Set when a multiple drive selection is detected.																																				
9	<b>Volume Check (VC)</b> – Set during transition from a head-load state to a head-on-track state. Cleared by execution of a Get Status command with Bit 3 asserted.																																				

**Table 3-8 MP Register Bit Description  
for Get Status Commands (Cont)**

Bit(s)	Description
10	<p><b>Write Gate Error (WGE)</b> – Sets when write gate is asserted and if one or more of the following conditions exist.</p> <ul style="list-style-type: none"> <li>• Drive is not “ready to read/write”</li> <li>• Drive is write protected</li> <li>• Sector pulse is occurring</li> <li>• Drive has another error</li> </ul>
11	<p><b>Spin Error (SPE)</b> – Set when spindle has not reached speed in the required time during spin-up or when spindle speed is too high.</p>
12	<p><b>Seek Time Out (SKTO)</b> – Set when the heads do not come on-track in the required time during a seek command or when “ready to read/write” is lost while the drive is in position (lock-on) mode.</p>
13	<p><b>Write Lock (WL)</b> – Set when the drive is write protected.</p>
14	<p><b>Current Head Error (CHE)</b> – Set if write current is detected in the heads when write gate is not asserted.</p>
15	<p><b>Write Data Error (WDE)</b> – Set if write gate is asserted but no transitions are being detected on the write data line.</p>

**3.4.4.2 MP Register During a Read Header Command** – When a read header command is executed, the next header is read and its three words are transferred to the MP register. The first word contains sector address, head select, and cylinder address information. The second word contains zeros. The third word contains header CRC information. All three words can be read sequentially by the program by reading the contents of the MPR. Figure 3-17 shows the bit layout of the MP register for read header Commands, while Table 3-9 describes the bit format.

**MPR AFTER READ HEADER COMMAND**



CZ-2013

Figure 3-17 MPR – Following a Read Header Command

Table 3-9 MP Register Bit Description for Read Header Commands

Bit(s)	Description
0-5	SA 0:5 – Sector Address
6	HS – Head Select
7-15	CA 0:8 – Cylinder Address

**3.4.4.3 MP Register During Read/Write Data Commands** – Before the reading or writing data, the program loads the word count into the MP register in two’s complement form. The counter is incremented as each word is transferred. Usually, the reading or writing operation is terminated when the word counter reaches zero (overflows). The word counter can keep track of from 1 to the full 40-sector count of 5120 data words (decimal).

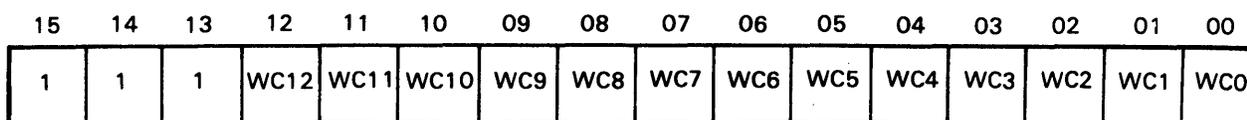
Figure 3-18 shows the bit format of the MP register for data transfer commands, while Table 3-10 describes the bit format.

**NOTE**

The RL01/RL02 disk drive will not do spiral read/writes. If data is to be transferred past the end of the last sector of a track, it is necessary to break up the operation into the following steps.

1. Program the data transfer to terminate at the end of the last sector of the track.
2. Program a seek to the next track. This can be either a head switch to the other surface but same cylinder or a head switch and move to the next cylinder.
3. Program the data transfer to continue at the start of the first sector at the next track.

**MPR DURING READ/WRITE COMMANDS FOR WORD COUNT**



CZ-2036

Figure 3-18 MPR – Used as a Word Counter

**Table 3-10 MP Register Bit Description for Data Transfer Commands**

<b>Bit(s)</b>	<b>Description</b>
0-12	<b>Word Count WC 12:00</b> – Contains the two's complement of total number of words to be transferred.
13-15	Must be ones.

**3.4.4.4 Bus Address Extension Register** – The Bus Address Extension (BAE) register is a six-bit register with an address of 17774410. It is used *only* with the RLV12 controller, and then only when 22-bit addressing mode is enabled. Bits 0 through 5 can be read or written. Bits 0 and 1 contain the same information found in the CSR bits 4 and 5.



**DAR DURING READING OR WRITING DATA COMMANDS**

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CA8	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0	HS	SA5	SA4	SA3	SA2	SA1	SA0

CZ-2011

**DAR DURING GET STATUS COMMAND**

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
X	X	X	X	X	X	X	X	0	0	0	0	RST	0	1	1

CZ-2037

**MPR AFTER GET STATUS COMMAND**

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
WDE	CHE	WL	SKTO	SPE	WGE	VC	DSE	DT	HS	CO	HO	BH	STC	STB	STA

CZ-2012

**MPR AFTER READ HEADER COMMAND**

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CA8	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0	HS	SA5	SA4	SA3	SA2	SA1	SA0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ZEROES															

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CRC															

CZ-2013

Figure 3-20 Register Summary (Sheet 2 of 3)

**MPR DURING READ/WRITE COMMANDS FOR WORD COUNT**

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	1	1	WC12	WC11	WC10	WC9	WC8	WC7	WC6	WC5	WC4	WC3	WC2	WC1	WC0

CZ-2036

15	14	13	12	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	0	BA21	BA20	BA19	BA18	BA17	BA16

READ/WRITE

CZ-0488

Figure 3-20 Register Summary (Sheet 3 of 3)

**3.5 I/O TRANSFER OPERATIONS**

There are three kinds of I/O transfers that are used to interface the processor with the RLV12 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 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 RLV12 addressable registers and LSI-11 Q-bus 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.

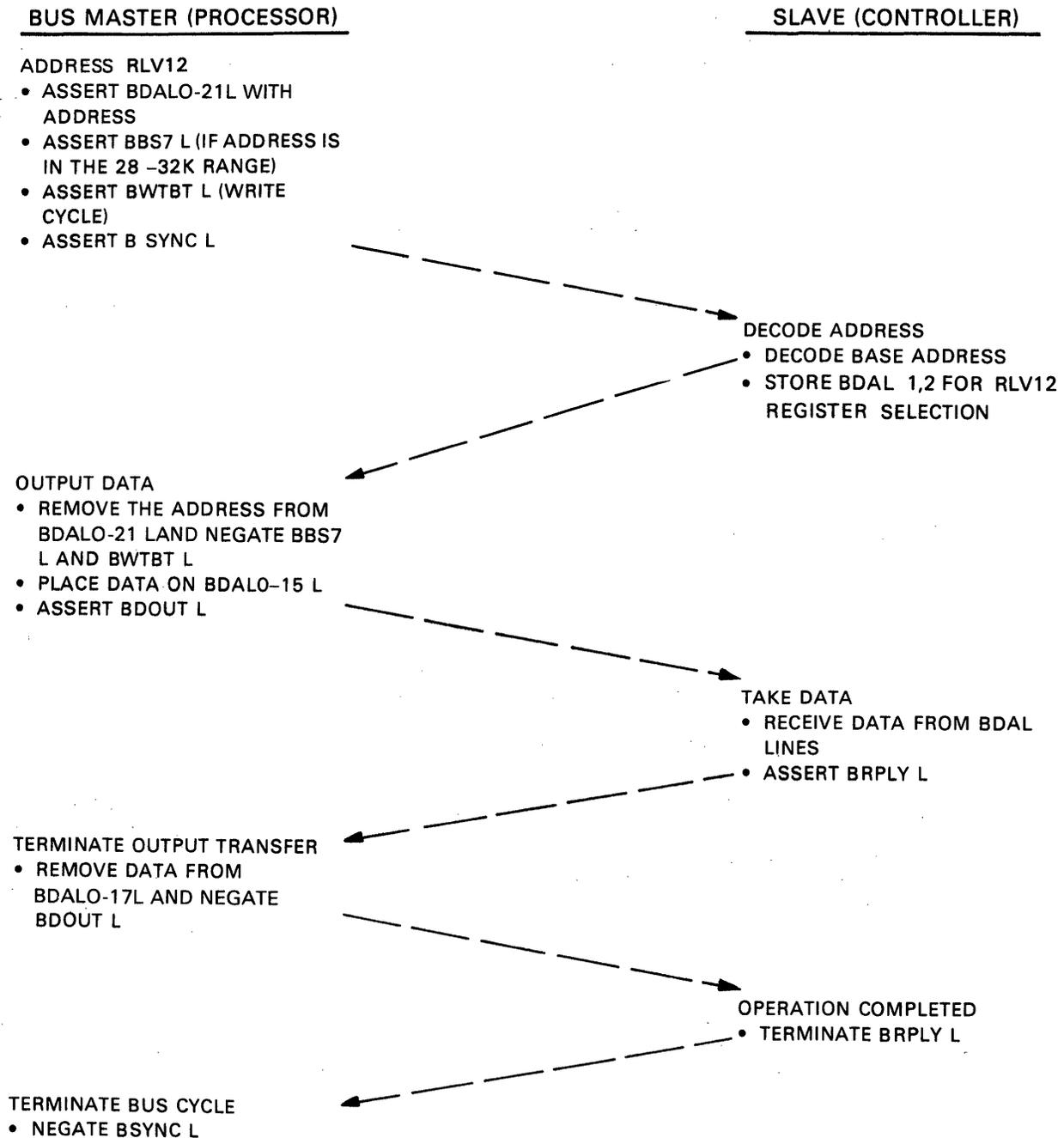
Interrupt-driven transfers allow the processor to continue a programmed operation without waiting for the controller to become ready. When the controller becomes ready, it interrupts the processor's background program sequence and causes execution of the controller's service routine. After the controller's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

**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.

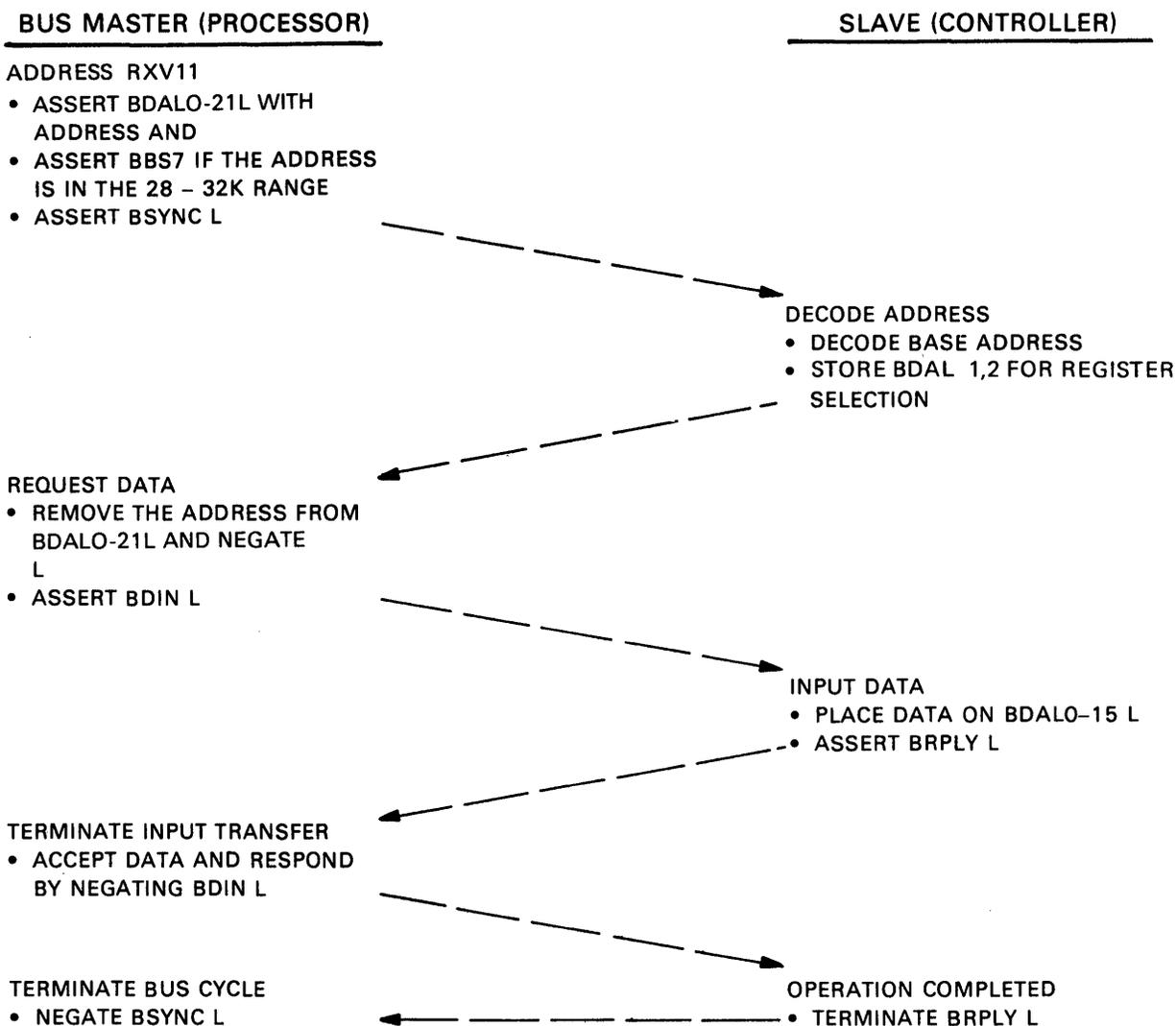
The DATO bus cycle is illustrated in Figure 3-21.



CZ-0513

Figure 3-21 DATO Bus Cycle

**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 3-22) to obtain the data from the RLV12 registers. The DATI cycle is a result of a CPU-programmed instruction which addresses the controller registers.



CZ-0514

Figure 3-22 DATI Bus Cycle

### 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 grant, regardless of other lower priority requests. If it is not requesting bus mastership, it will pass the processor BDMGO through another non-requesting 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 microseconds, 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 3-23.

### **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 RLV12 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.

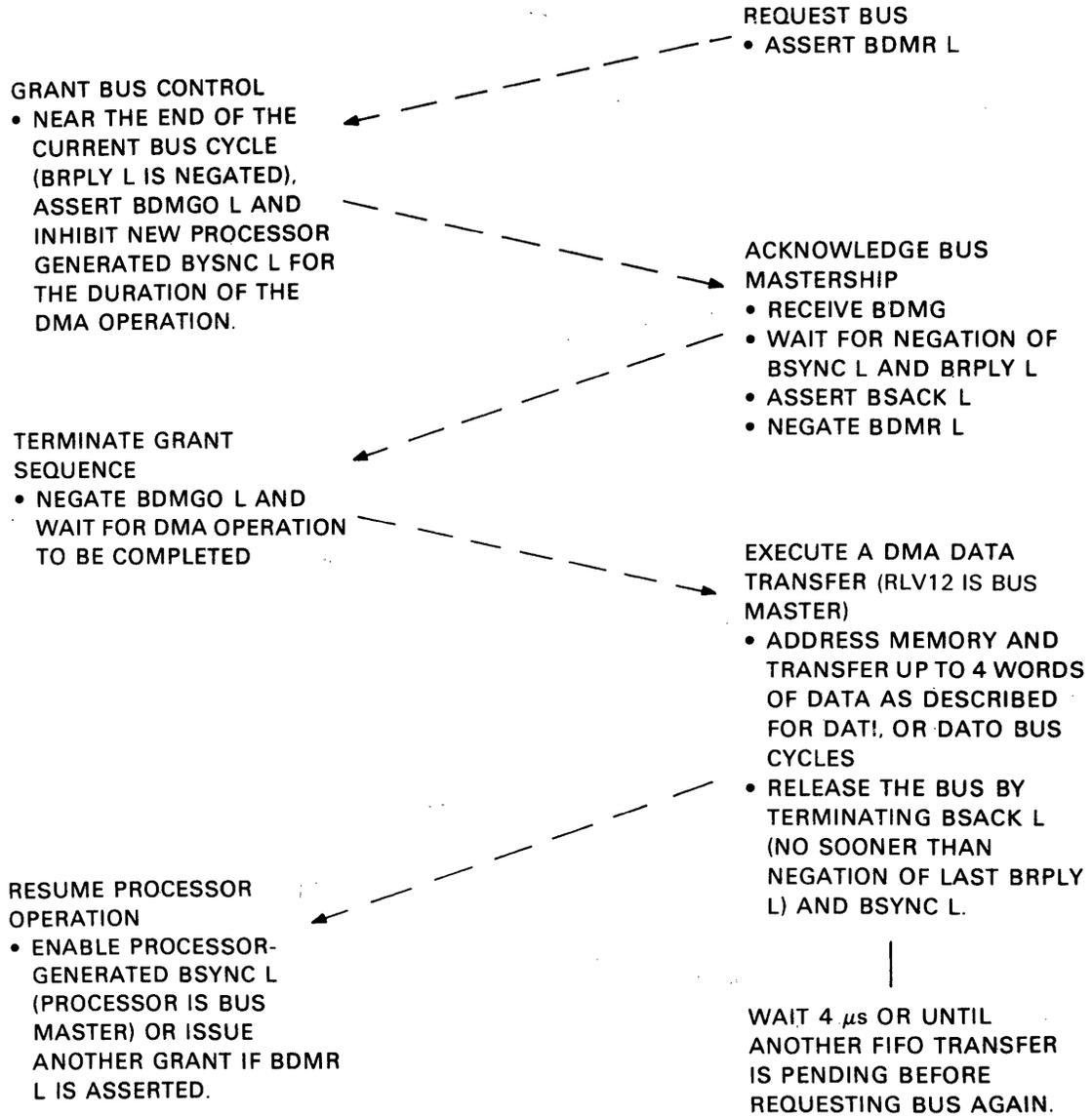
The interrupt request sequence is illustrated in Figure 3-24. The controller requests interrupt service by asserting BIRQ L. The processor acknowledges the interrupt request by asserting BDIN L followed by BIAKO L. The first device on the bus receives this daisy-chained signal at its BIAKI L input. If 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 RLV12 controller, given in general master/slave device terms, may be found in the Microcomputer Handbook published by Digital Equipment Corporation.

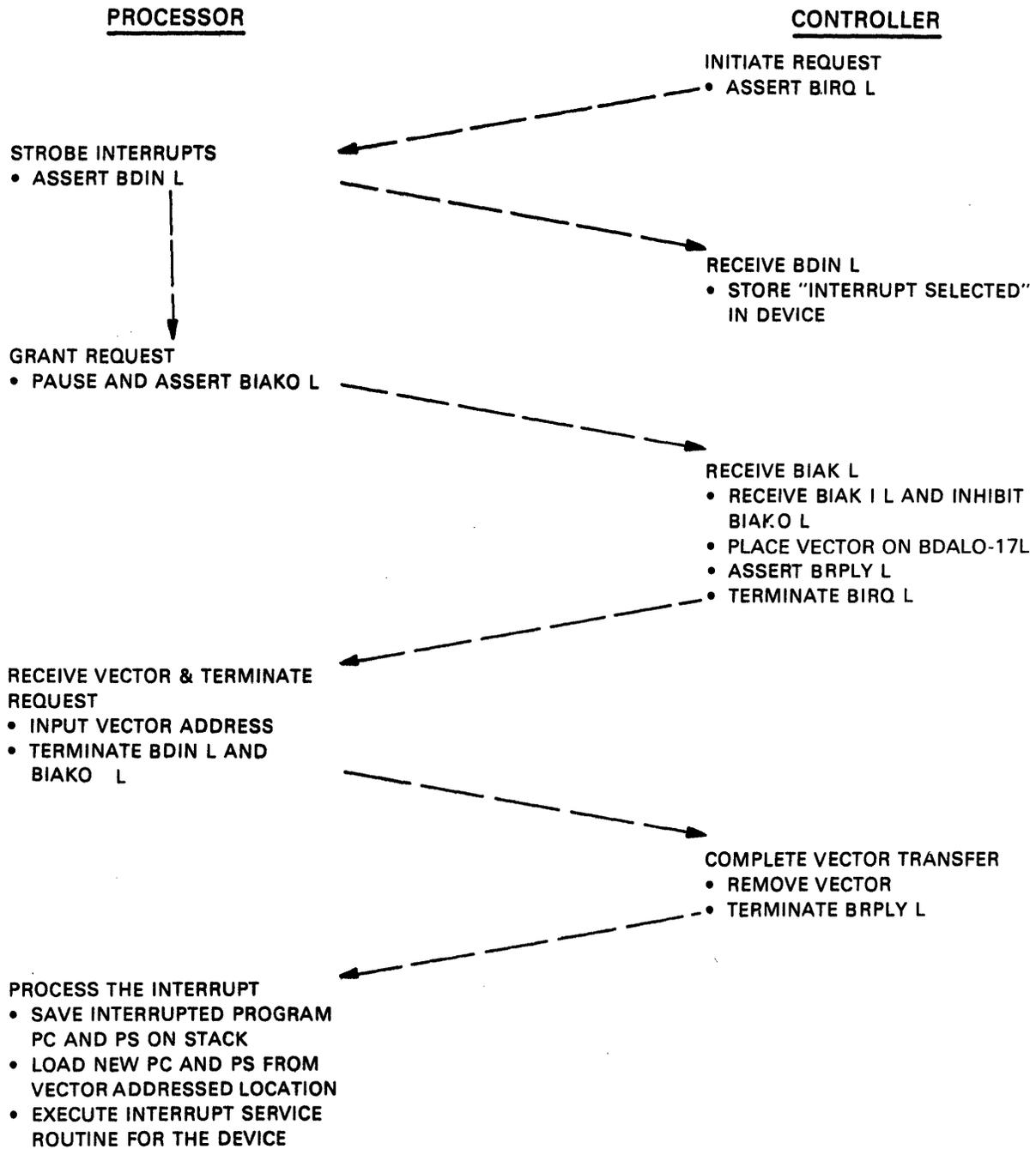
**LSI-11 PROCESSOR  
(MEMORY IS SLAVE)**

**BUS MASTER  
(CONTROLLER)**



CZ-0515

Figure 3-23 DMA Request/Grant Sequence



CZ-0516

Figure 3-24 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.

##### **4.2.1 Bus Control Functions**

The bus control block consists of five separate functional units.

- Register Protocol Circuit – The circuit selects which control 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 550 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. A memory parity error also sets this bit.
- 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 microseconds to reply to a controller Data In (DIN) or Data Out (DOUT) signal. If the reply (BRPLY) is not received within 10 microseconds, the NXM error bit is set in the CSR. NXM timeout can also occur when a memory parity error is flagged (provided the memory being used has the memory parity option).

##### **4.2.2 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.

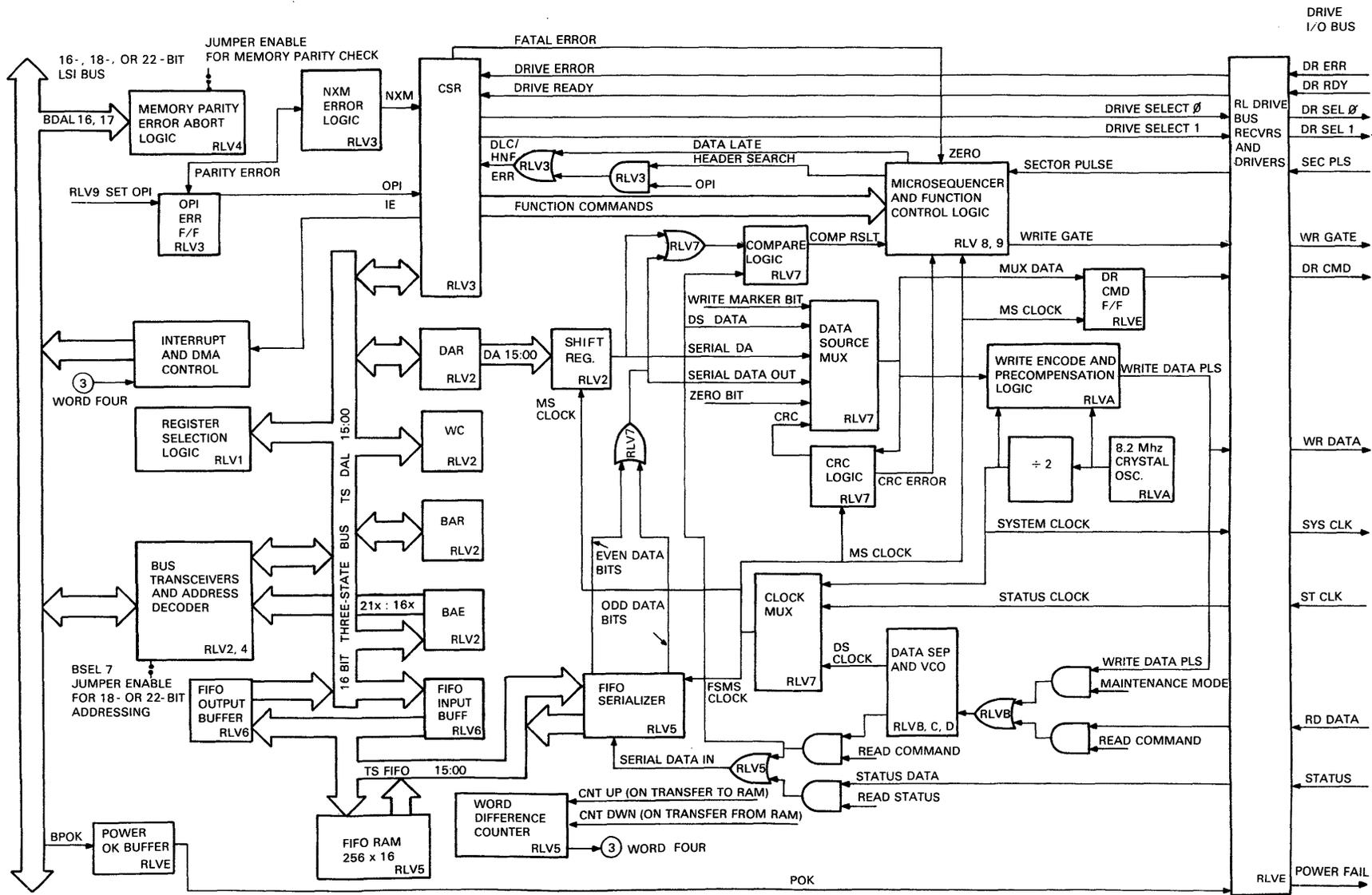


Figure 4-1 RLV12 Controller Simplified Block Diagram

### 4.2.3 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 extend address bits (16 and 17). 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.

**Bus Address Extension Register (BAE)** – The BAE contains the upper six bits of the extended 22-bit addressing. Bits 16 and 17 from the CSR are duplicated in this register.

**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. The word counter is a write-only register, while the output buffer is a read-only register.

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 one 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. The FIFO output buffer is loaded from the FIFO.

### 4.2.4 FIFO

The FIFO is a first-in/first-out silo-type memory element that can store up to 255 data words. When full, it holds two sectors of data. A FIFO serializer/deserializer (SERDES) 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 using the same serializer circuit.

The FIFO contents can be recovered by reading the current data word in the MPR. During disk read and write operations, the FIFO is emptied and filled under control of the microsequencer.

### 4.2.5 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.

### 4.2.6 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 RL drives. 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.

#### **4.2.7 Data Separator**

The data separator circuit makes use of a phase-locked loop oscillator to detect and decode incoming MFM disk data into a binary NRZ bit stream and a clock.

#### **4.2.8 CRC Circuit**

The cyclic redundancy check (CRC) logic is an error-detection circuit. For user's data written on the disk, a code is generated in the CRC circuit by an internal algorithm. This code is then appended onto the end of each sector's data field in the form of a CRC word. When the data field is subsequently read from the disk, the data is channeled through the CRC circuit. Any errors introduced into the data or its CRC word are detected as a CRC error.

Header data is written at the factory only and a CRC word is appended onto the end of the header. When reading header data, it is channeled through the CRC circuit for detection of possible header/CRC errors.

#### **4.2.9 Data Source Multiplexer**

This circuit allows the multiplexing of different data sources under the control of the microsequencer. There are six different sources of data: CRC data, serial disk address data, serial FIFO output data, data separator (DS) data, write marker pulse, and zero. The zero input is responsible for writing the preambles or postambles of the sector. It also is used to zero-fill the sector when writing partial sectors.

#### **4.2.10 Compare Logic**

The function of the compare logic is twofold. First, the header word from the data separator is compared with the serial disk address word stored in the DAR. Secondly, this circuit compares the contents of the FIFO with the data read from the disk (WRITE CHECK commands). These comparisons are done serially on a bit-by-bit basis. If any pair of bits is not identical, a mismatch signal is generated (COMP RSLT).

### **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. The 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 are loaded by software with the following contents.

- BAR and BAE registers with address of first memory buffer location
- WC register with 2's complement of  $511_{10}$
- DAR with test word
- CSR with F<0>, CRDY, IE

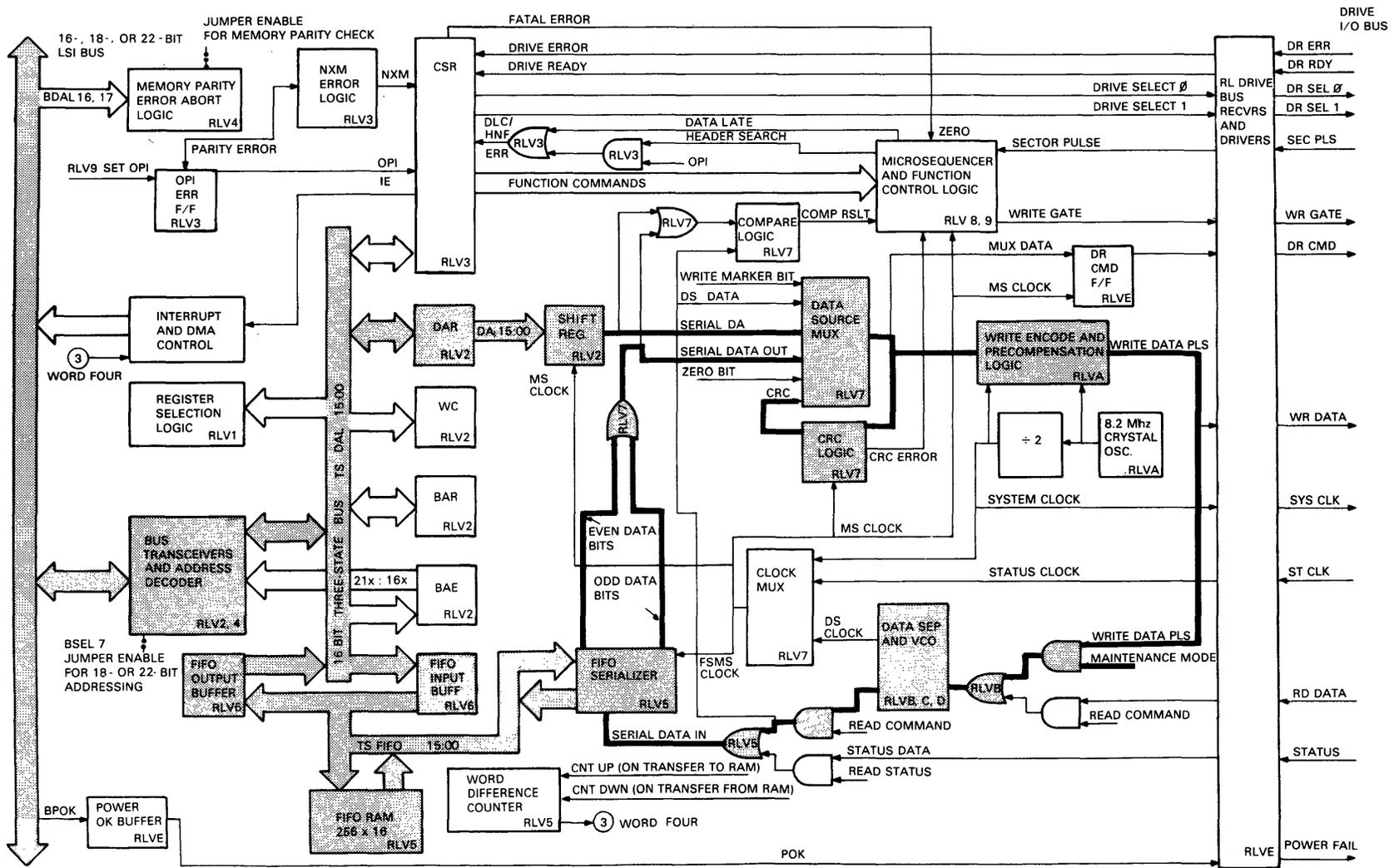
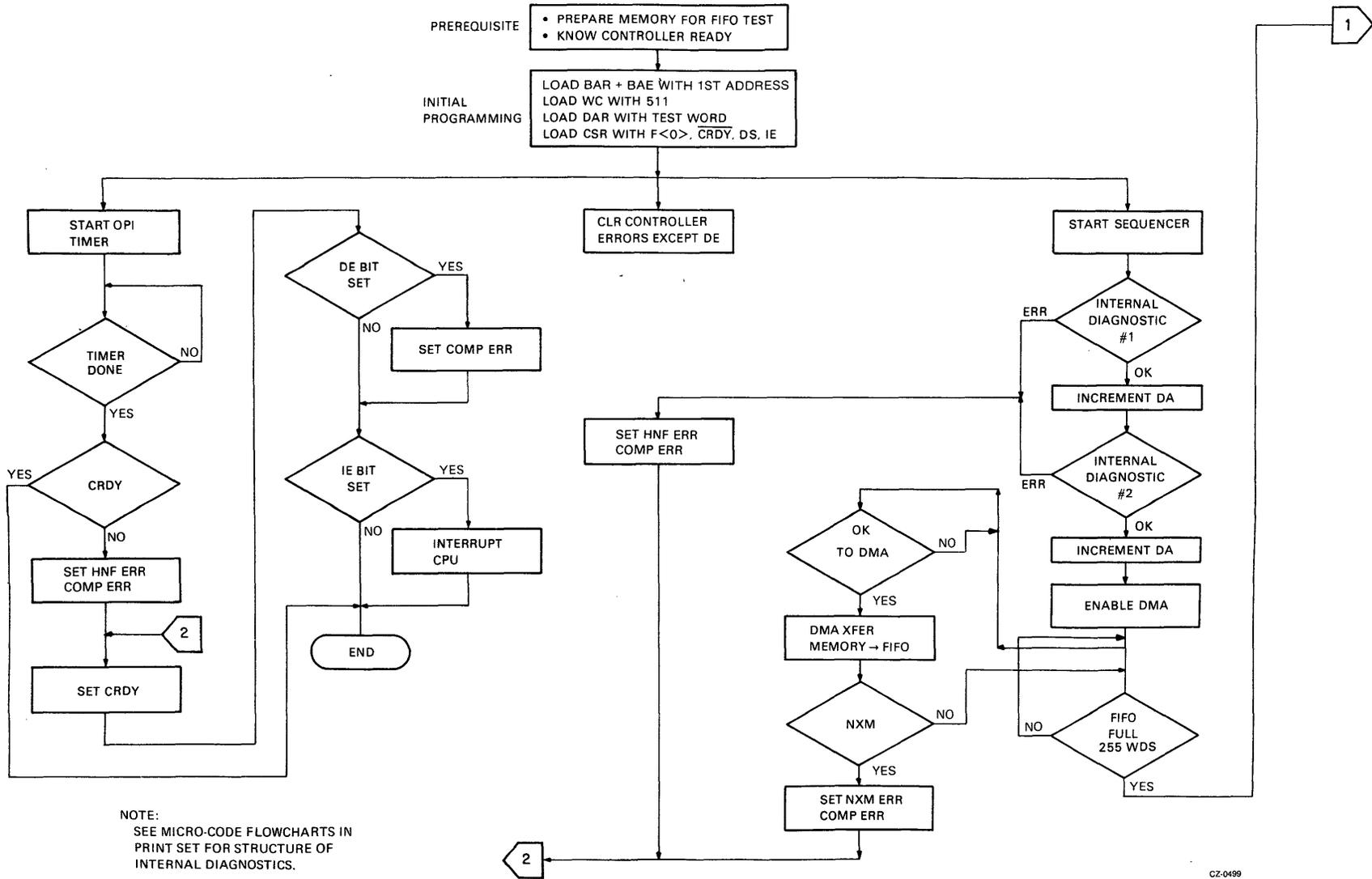


Figure 4-2 Maintenance Command Functional Block Diagram



CZ-0499

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

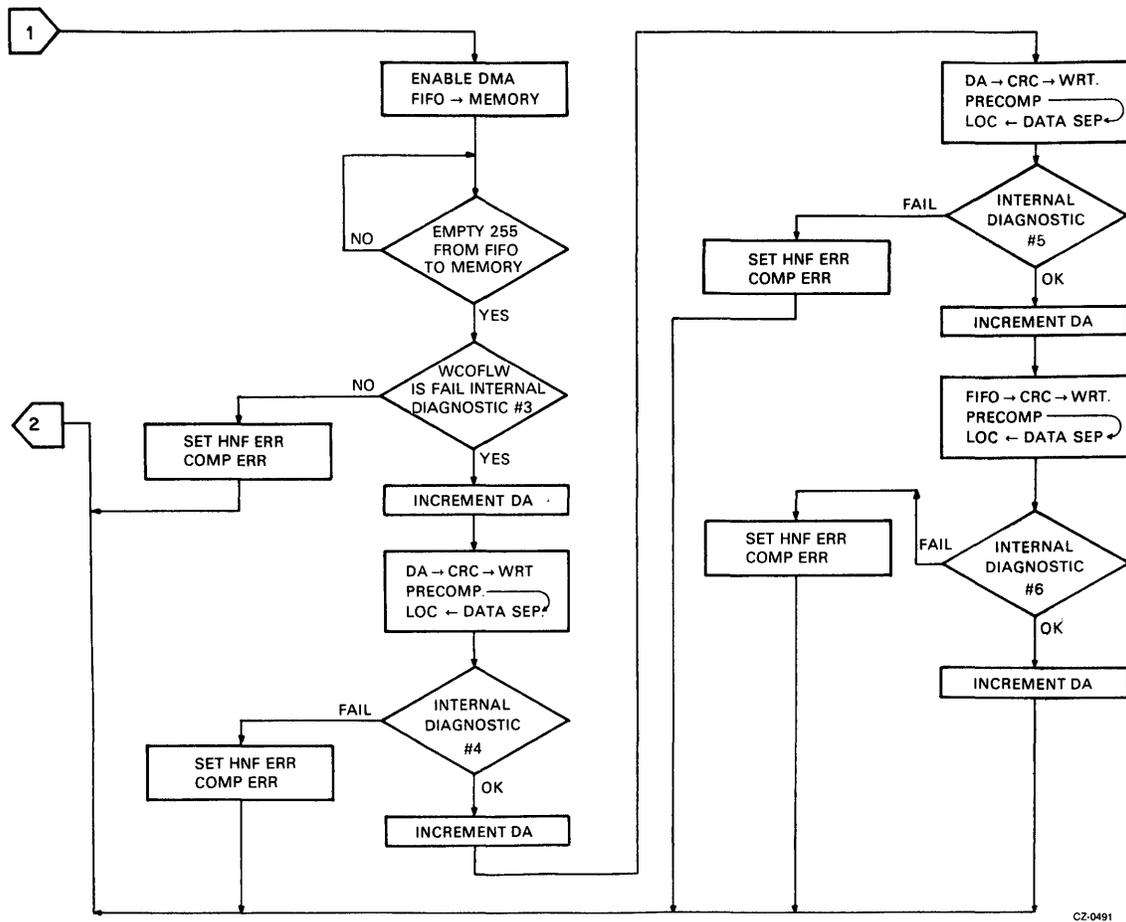


Figure 4-3 Maintenance Command Flowchart  
(Sheet 2 of 2)

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, error checks are made and, if an error occurs, the function stops with HNF 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 MUX and into the CRC circuit. A CRC word is generated from this test word and sent through the data MUX again. This test word CRC then passes through the write precompensation and data separator circuits to eventually end up in the FIFO.

The contents of the DAR are then incremented by one and become 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.

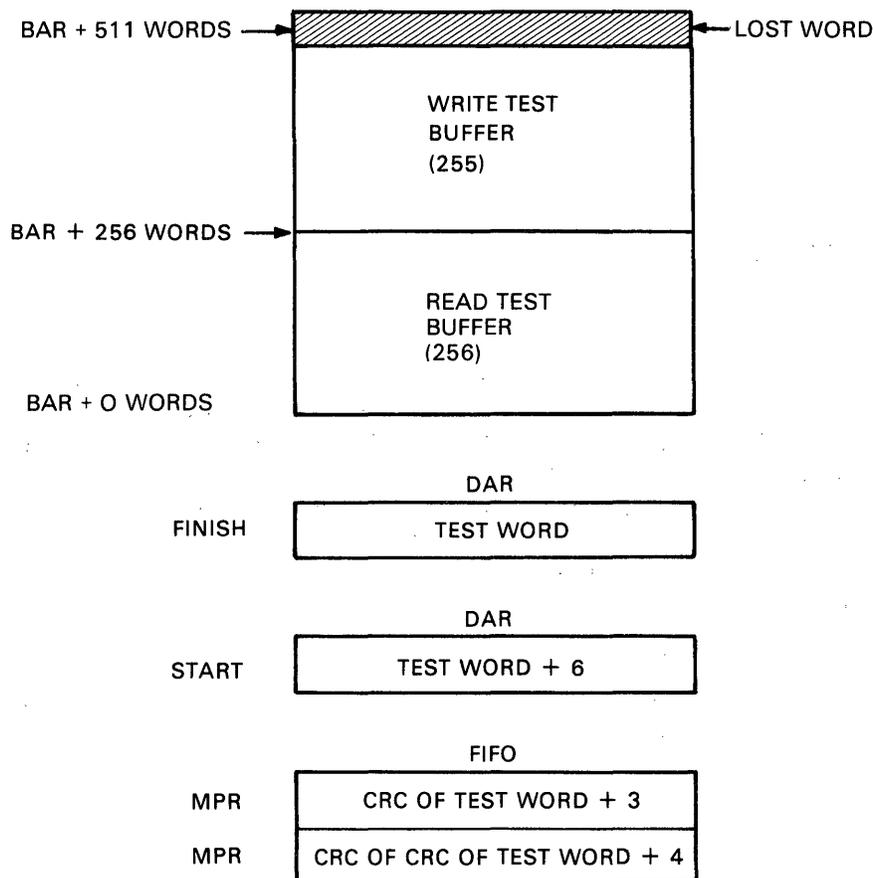
Word	FIFO
1st	CRC of test word +3
2nd	CRC of test word +4

The contents of the DAR are now incremented once again and become test word +5.

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 MUX, the CRC, and data MUX 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 test word +4

The contents of the DAR are 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.



CZ-0517

Figure 4-4 Results of Maintenance Command

As a result of this maintenance test, the following circuits are tested: the FIFO, the registers, the data source selector, the CRC 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 and get status bits 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 MUX. A send drive command signal is issued that allows the status request word to be gated to 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 is 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.

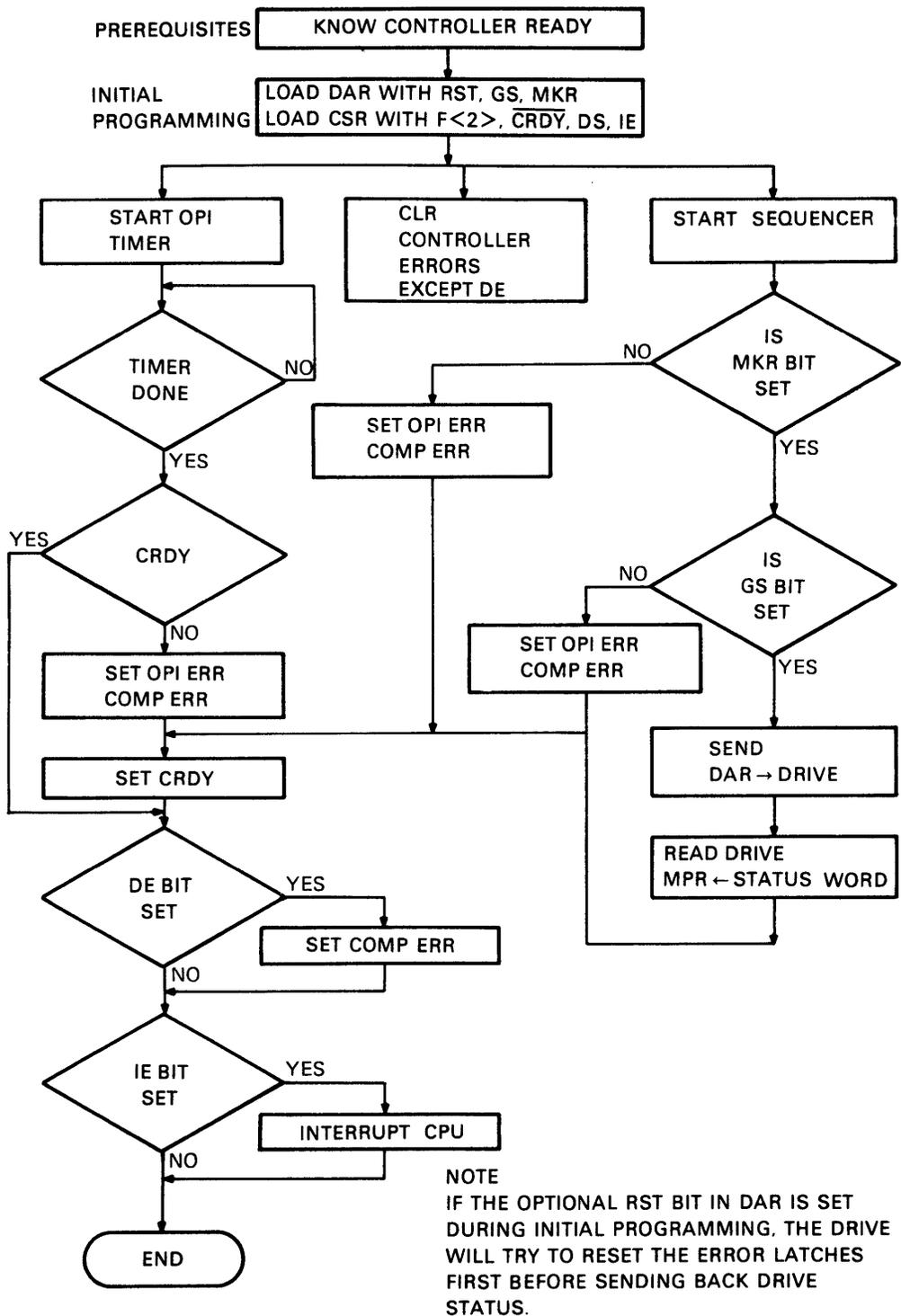
#### **4.5 READ HEADER FUNCTION**

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 available sector pulse initiates the read header sequence. Three header words are read off the disk and enter the controller via 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 MUX 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 is set. The two header words and the header CRC are then available to be read from the multipurpose register.

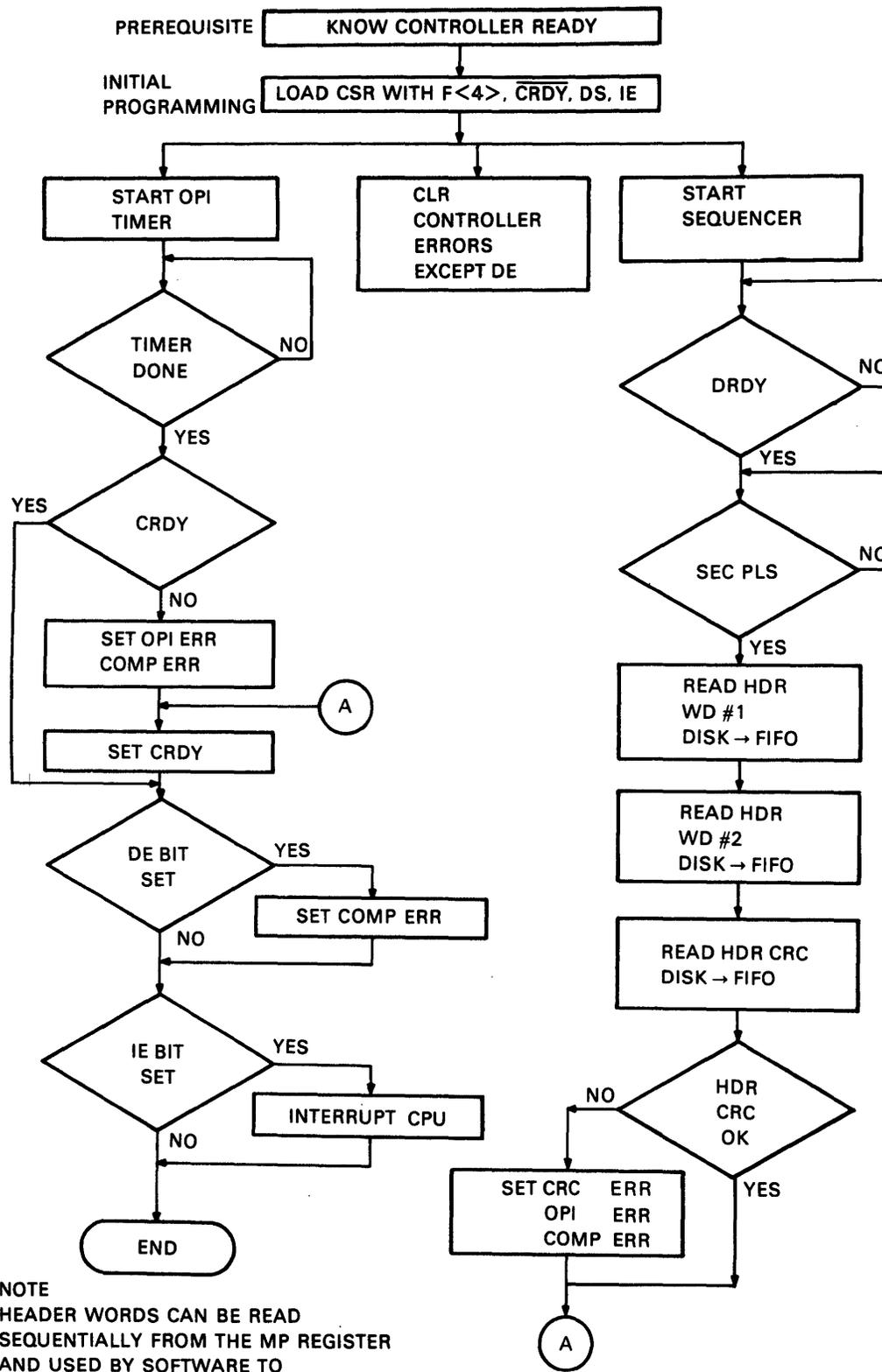




CZ-0496

Figure 4-6 Get Status Command Flowchart





CZ-0494

Figure 4-8 Read Header Command Flowchart

#### **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 are indeed for a seek command and not a get status request. If either of these two bits differ, 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 available sector pulse begins the transfer of the DAR contents to the drive after 40 microseconds delay. The difference address word in the DAR is serialized and passes through the data MUX. The serial data is synchronized with system clock, and a send drive command signal channels the difference address word out of the controller to the drive command line. Once the 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 and BAE registers are 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 are 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 occurs, the controller begins reading each header off the disk. As each header enters the controller through the data separator circuit, it is then channelled through the data MUX and the compare logic. The header that enters the compare logic 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 enters the data MUX is channelled into the CRC circuit and a CRC word is computed.

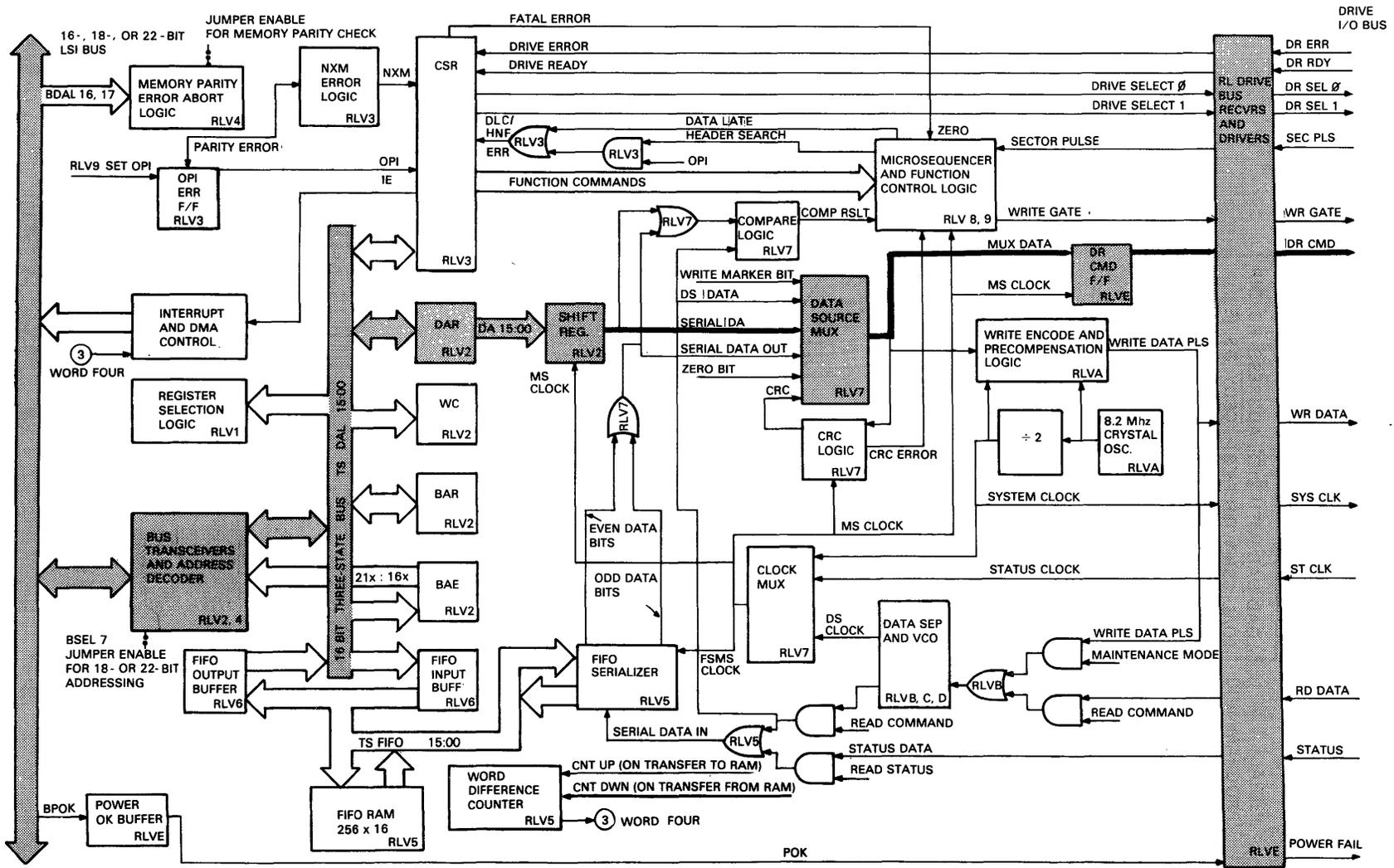
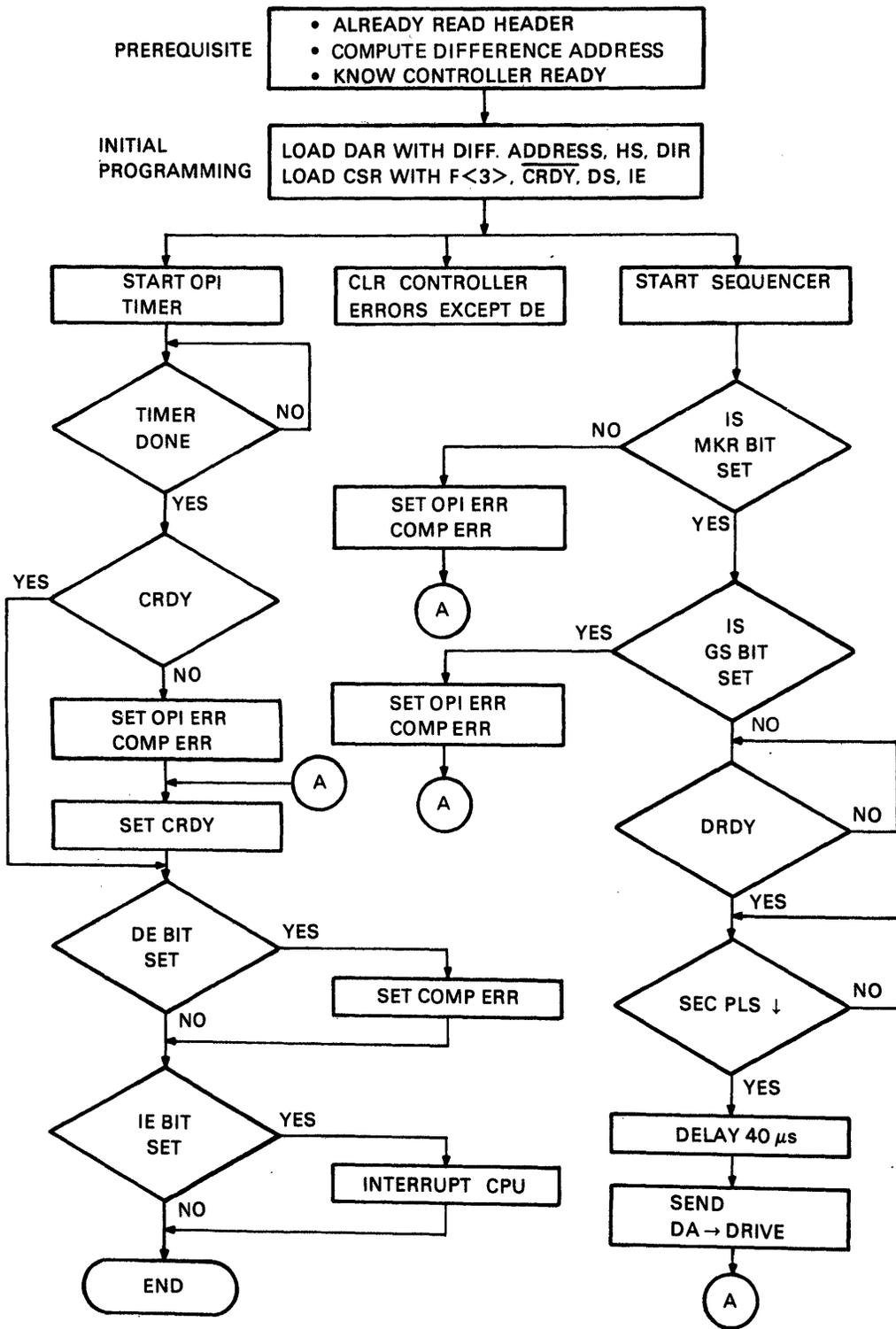


Figure 4-9 Seek Functional Block Diagram



CZ-0492

Figure 4-10 Seek Command Flowchart

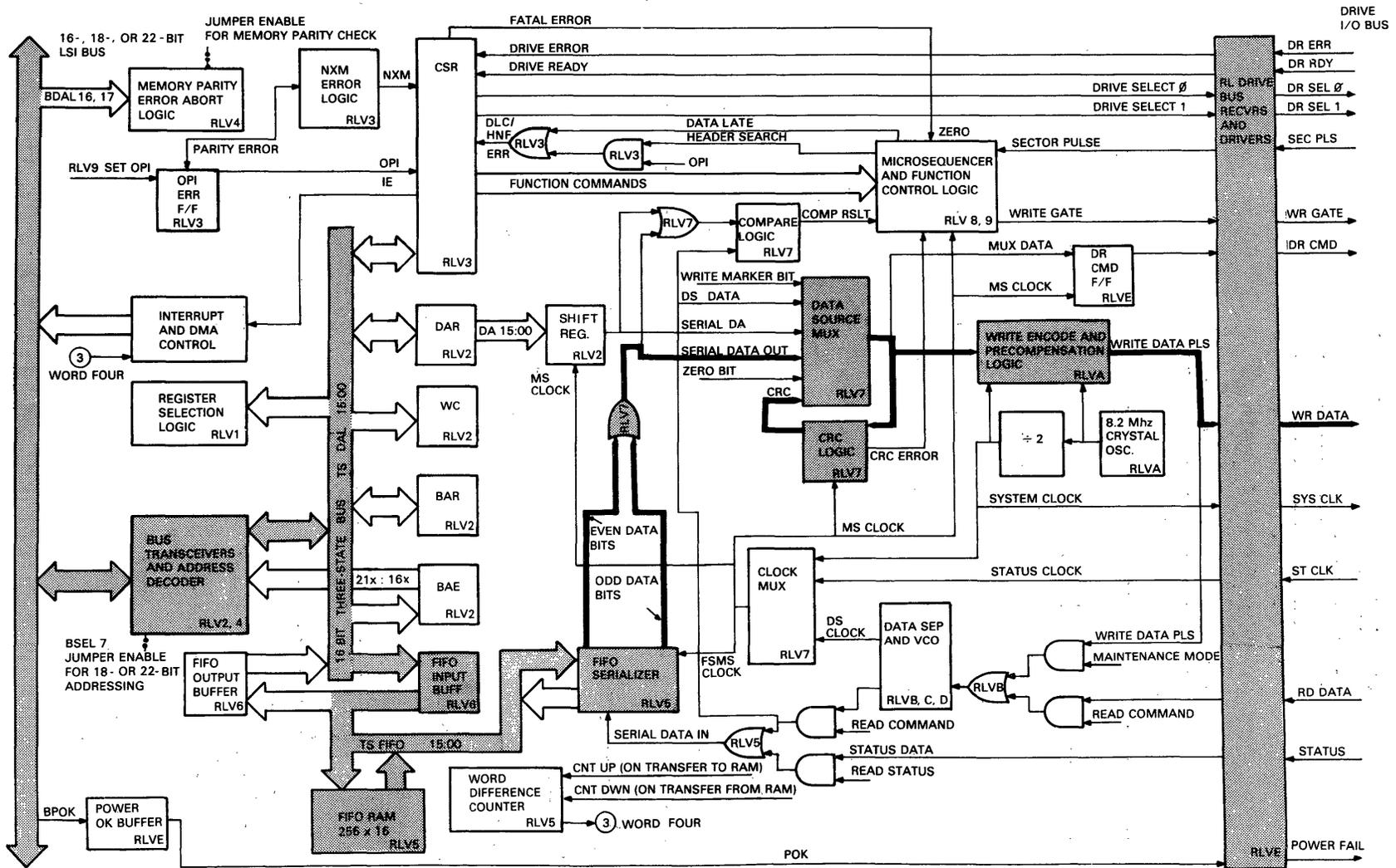
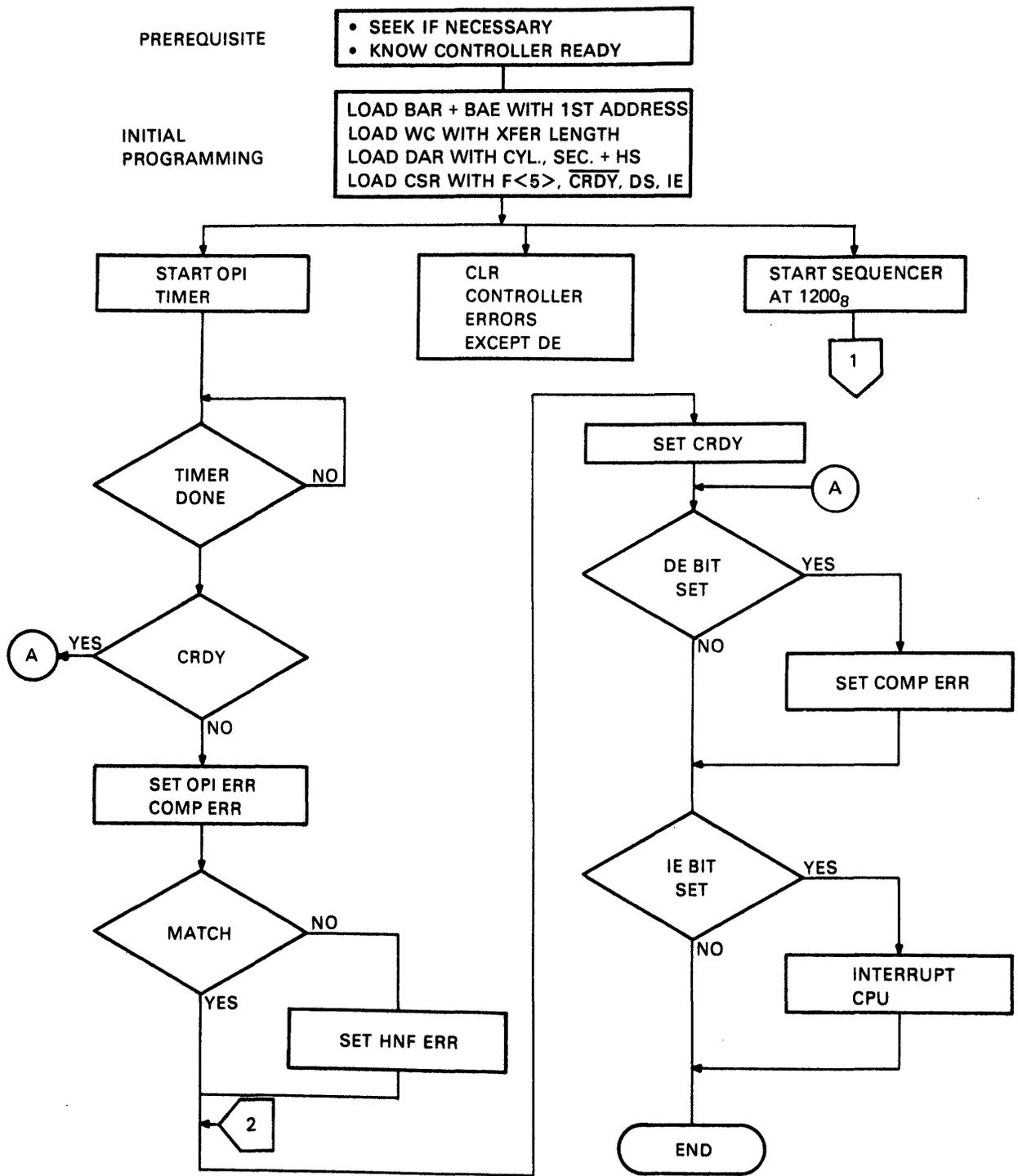


Figure 4-11 Write Data Functional Block Diagram



CZ-0493

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

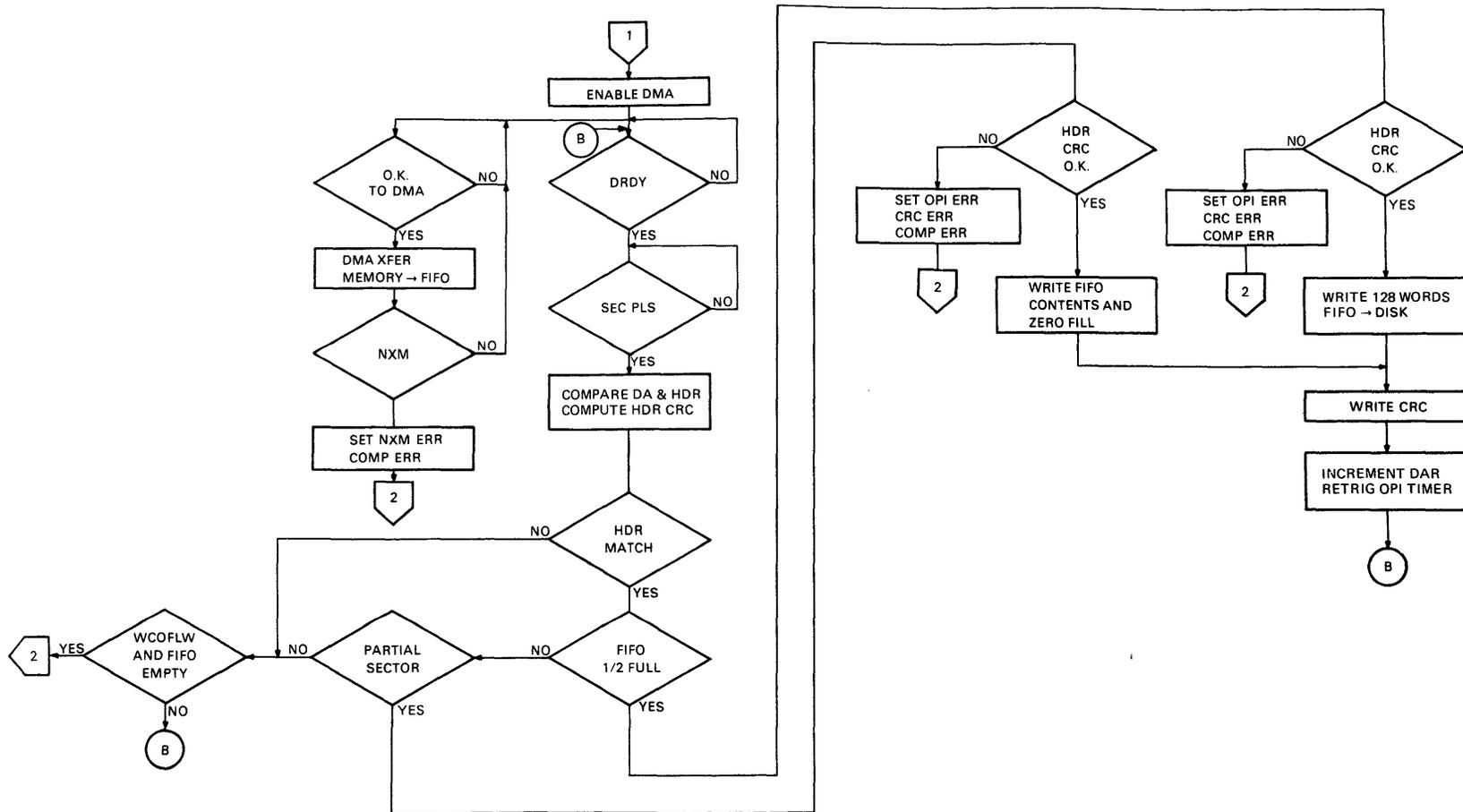


Figure 4-12 Write Data Command Flowchart  
(Sheet 2 of 2)

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 yet half full, the sequencer confirms that a partial sector write is desired, or that 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, the command will not start until the header match 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 MUX 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 multiplexer, 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 are then incremented by one 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.

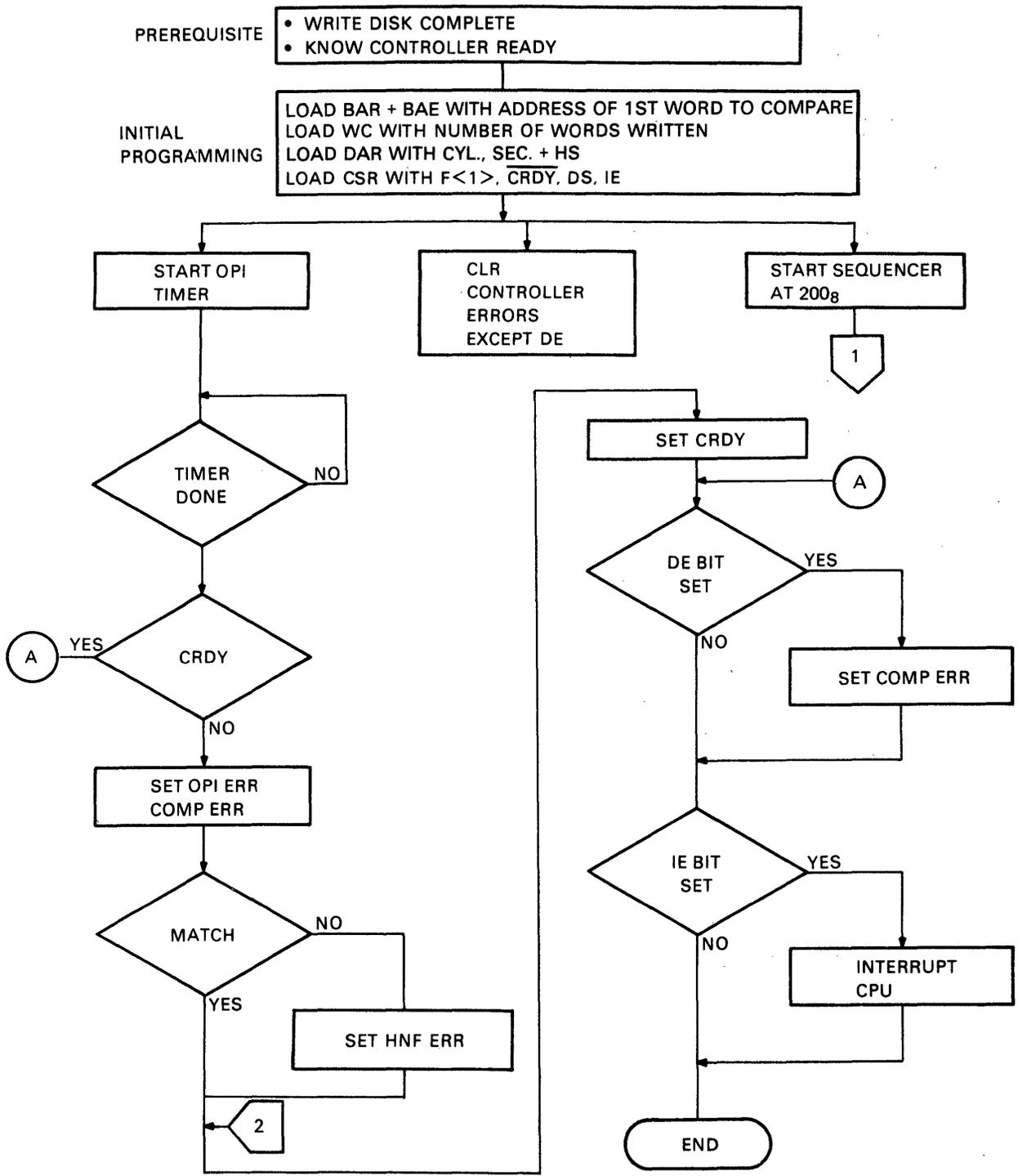
#### **4.8 WRITE CHECK FUNCTIONAL FLOW DIAGRAM**

The write check command is used to verify that data has been written on the disk correctly. It is accomplished by first writing a block of data on the disk using the write command. The write check command then reads this same block of data off the disk and compares it with the contents of its source data buffer area in main memory. This comparison is performed in the RLV12 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 and BAE registers 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, BAE, 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, the data buffer area of main memory begins filling the controller FIFO under DMA control. 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 by the compare logic. Either a compare error or a data CRC error sets the DCRC bit in the CSR.





CZ-0495

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

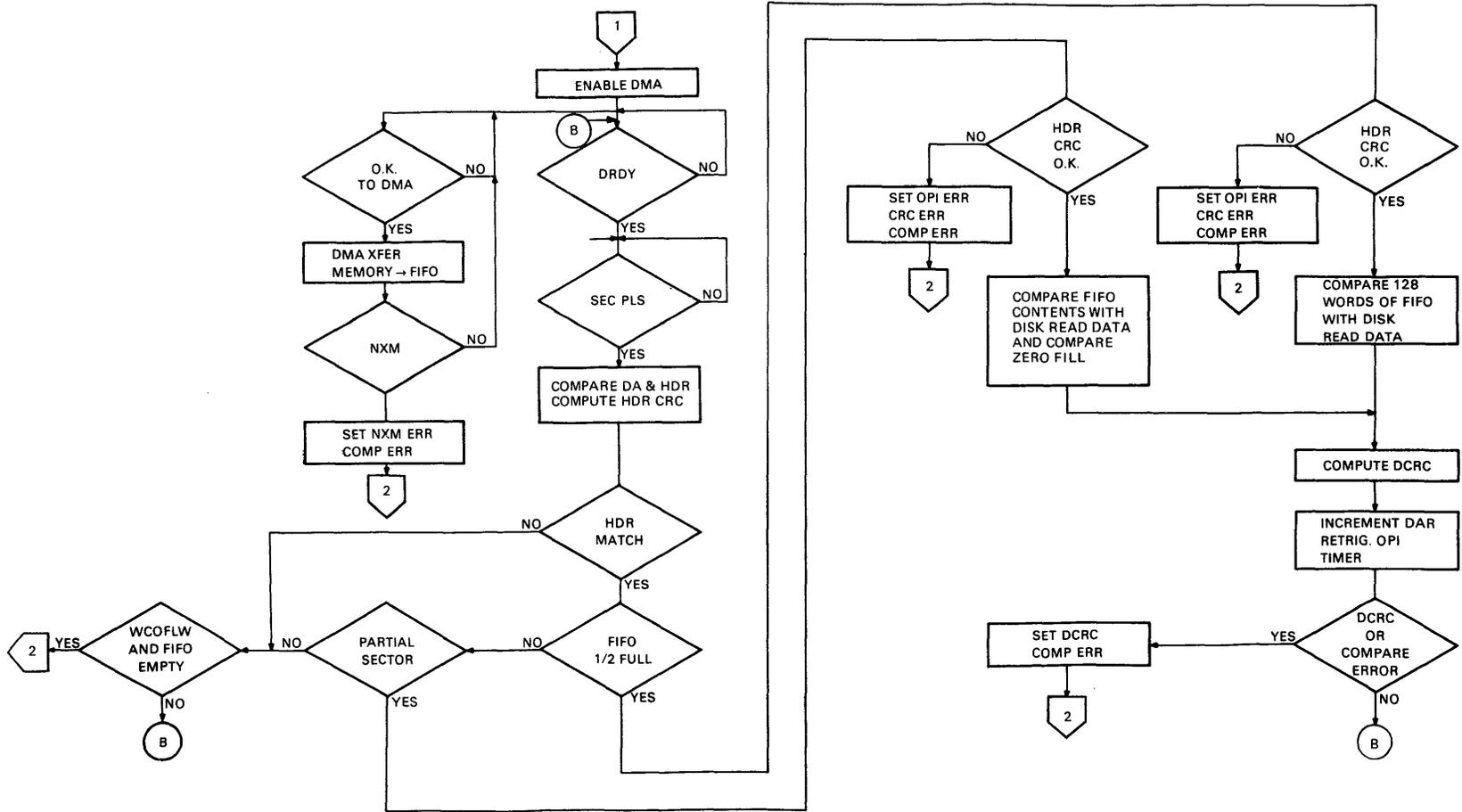


Figure 4-14 Write Check Command Flowchart  
(Sheet 2 of 2)

#### **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 to position the heads at the proper place. Read data is read off the disk in full sectors; however, software may desire only a portion of the sector from the controller FIFO. Figures 4-15 and 4-16 are the functional flow diagrams that illustrate this operation.

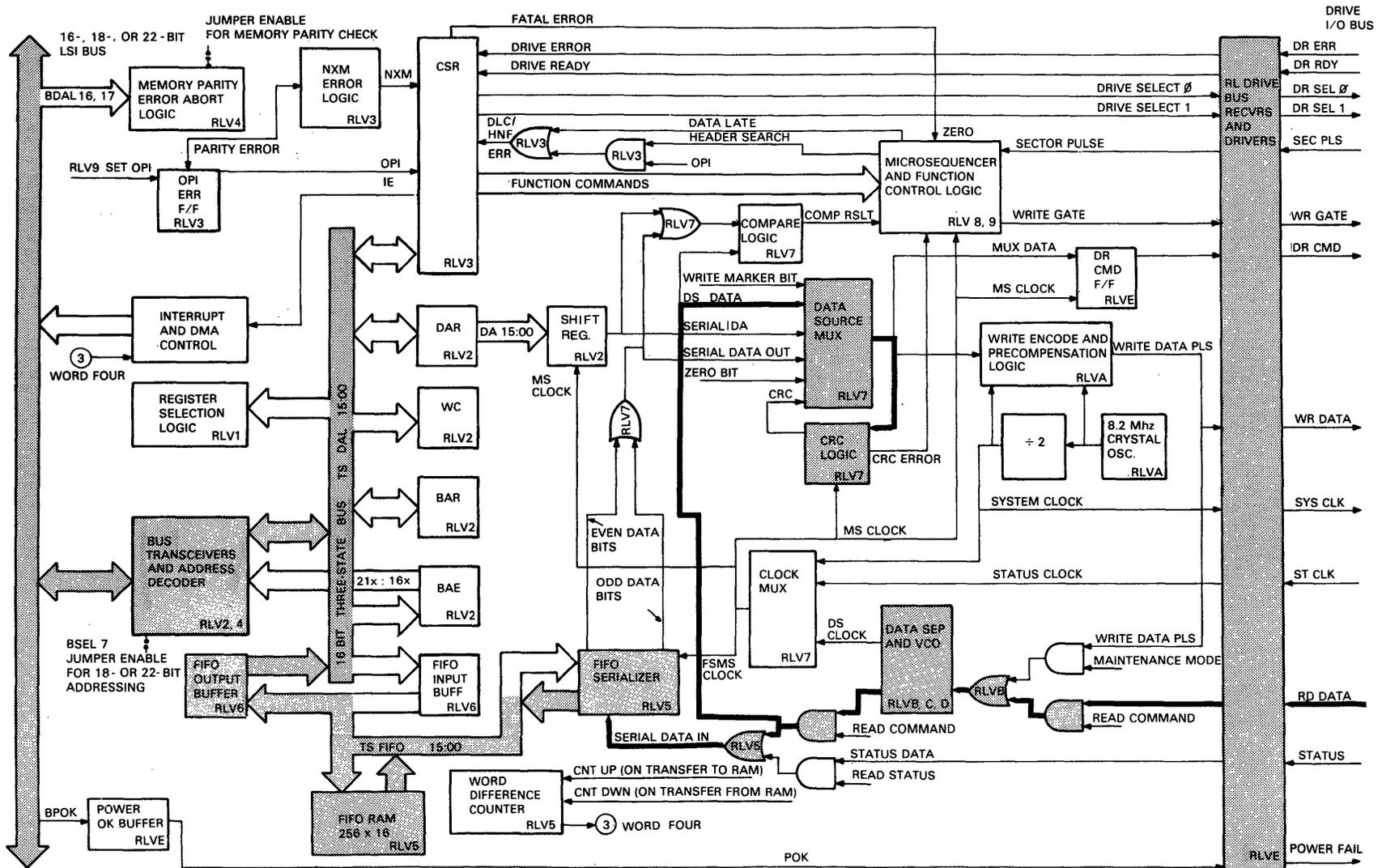
Prior to issuing the read data command, the BAR and BAE registers are loaded with the address of the first memory location of the read buffer. The word counter stores the number of words to be transferred and the DAR contains the cylinder and sector address of the first sector to be read off the disk.

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. With the next available sector pulse, the controller begins reading each header off the disk. As each header enters the controller through the data separator circuit, it is channelled through the data MUX 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 MUX 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 this 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 data separator circuit 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 count overflow occurs. The data CRC word is checked at the end of each sector.

When the word count 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.



C2-0484

Figure 4-15 Read Data Functional Block Diagram

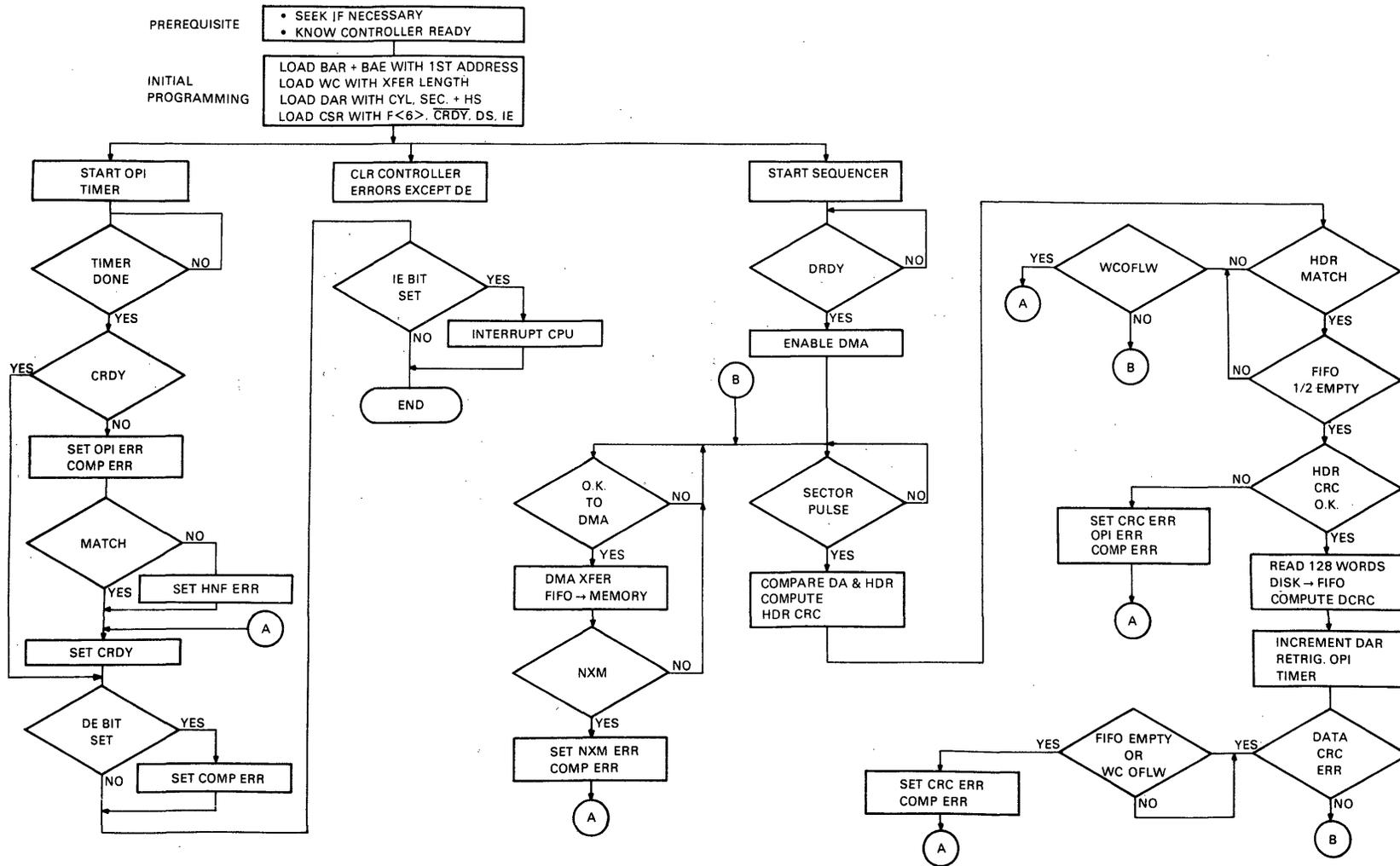


Figure 4-16 Read Data Command Flowchart

#### **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, the BAR and BAE registers are 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. System software then locates the sector preceding the bad sector by performing successive read header commands. When the header is found, the CSR is loaded with the head data without the header check function.

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 data separator circuit on the 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.

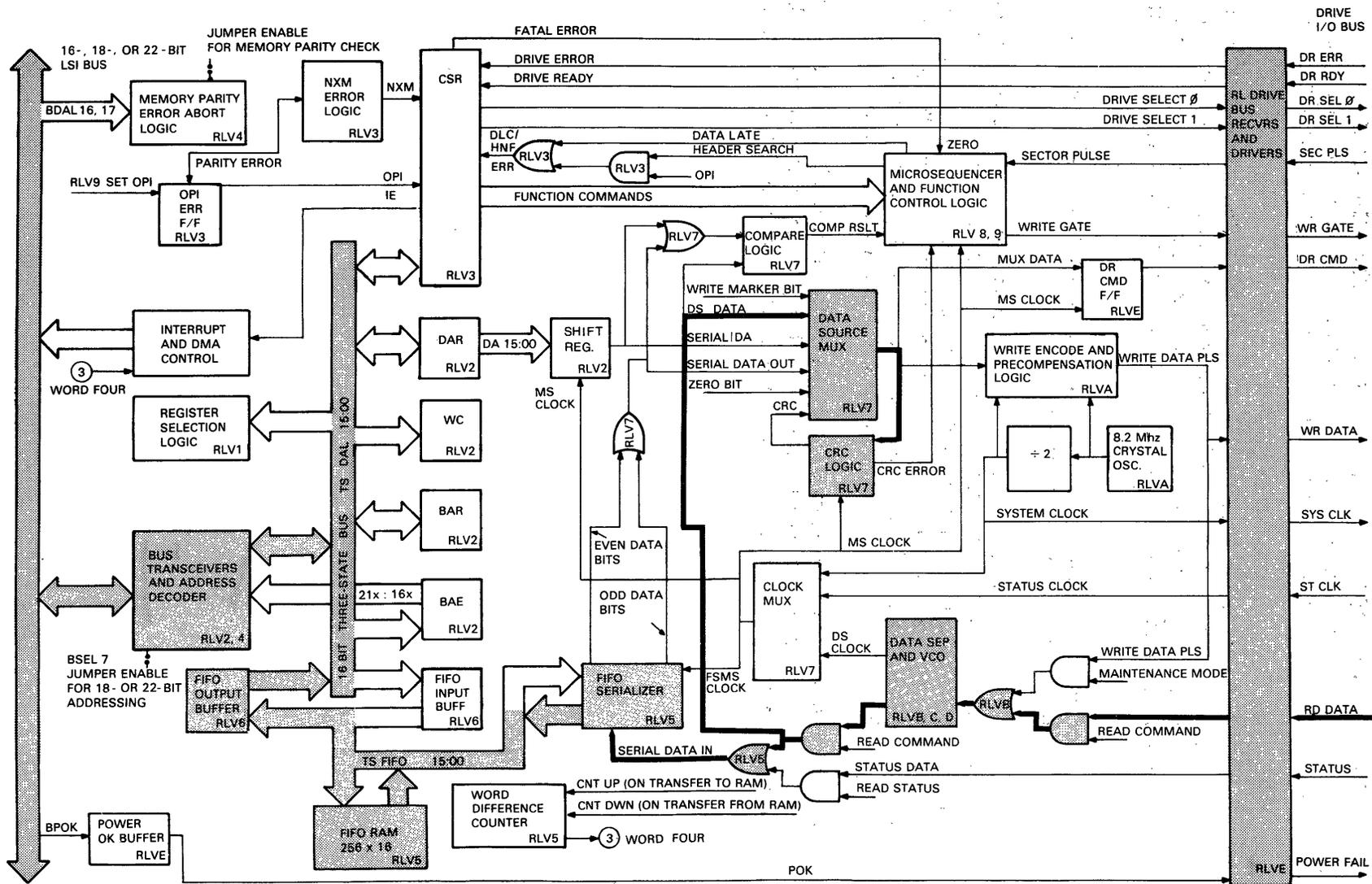


Figure 4-17 Read Data without Header Check Functional Block Diagram

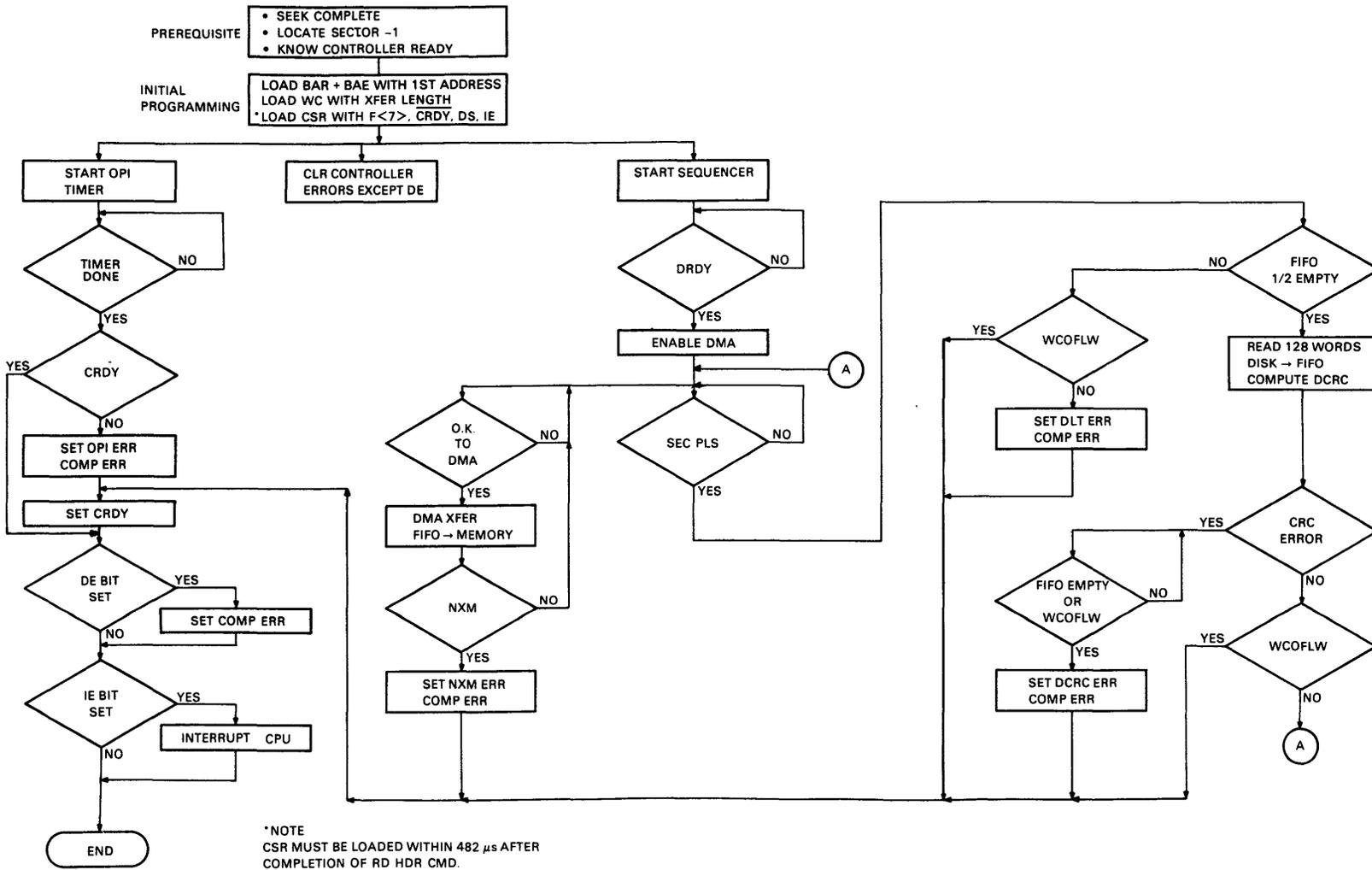


Figure 4-18 Read Data without Header Check Command Flowchart



## CHAPTER 5 UNIT LEVEL DESCRIPTION

### 5.1 INTRODUCTION

The RLV12 controller interfaces the RL01 and RL02 disk drives to a 16-, 18-, or 22-bit LSI-11 bus. One RLV12 can support up to four RL01 or RL02 disk drives in any combination. The RLV12, (an M8061 module), has the LSI-11 bus transceivers and decoders, programmable registers, the controller timing and sequence logic, and the data formatting circuits necessary to read and write on the disk.

The main sections of the RLV12 are shown in Figure 5-1. The RLV12 has five programmable registers:

- Control/status register (CSR),
- Bus address register (BAR),
- Disk address register (DAR),
- Multipurpose register (MPR),
- Bus address extension register (BAE) (22-bit addressing only).

These registers are addressed like any memory location. Of these five registers, the CSR is written last because it starts the microsequencer operation.

An RLV12 program can select 16-, 18-, or 22-bit LSI-11 bus addressing. The M8061 module is software compatible with, and can replace, the RLV11 when not enabled for 22-bit addressing.

To issue a command to the RLV12, the processor first places the address of a register on the LSI-11 bus. Then it places the data to be loaded into the register on the bus. The RLV12 controller decodes the address and channels the data to the correct register. The processor loads bits 0 through 15 of the bus address register (BAR) with the address in memory that the first transfer is to take place. If 22-bit addressing is used, the processor loads the bus address extension register (BAE) with extended address bits 16 through 21. For 18- or 22-bit addressing the software must also load bits 4 and 5 of the CSR with BAE bits 16 and 17.

Once the command is written into the control/status register, the RLV12 starts a microsequencer routine. The microsequencer decodes the command and branches to an address in the control store PROMs. There the microsequencer finds a routine for the command issued. The microsequencer then generates the control signals needed to channel the data through the controller.

Included on the controller are error detection features such as the memory parity error abort feature for use with memories that have parity error checking. When reading system memory, data bits 16 and 17 from the bus are checked for a parity error. If an error is detected, the current command to the controller is aborted.

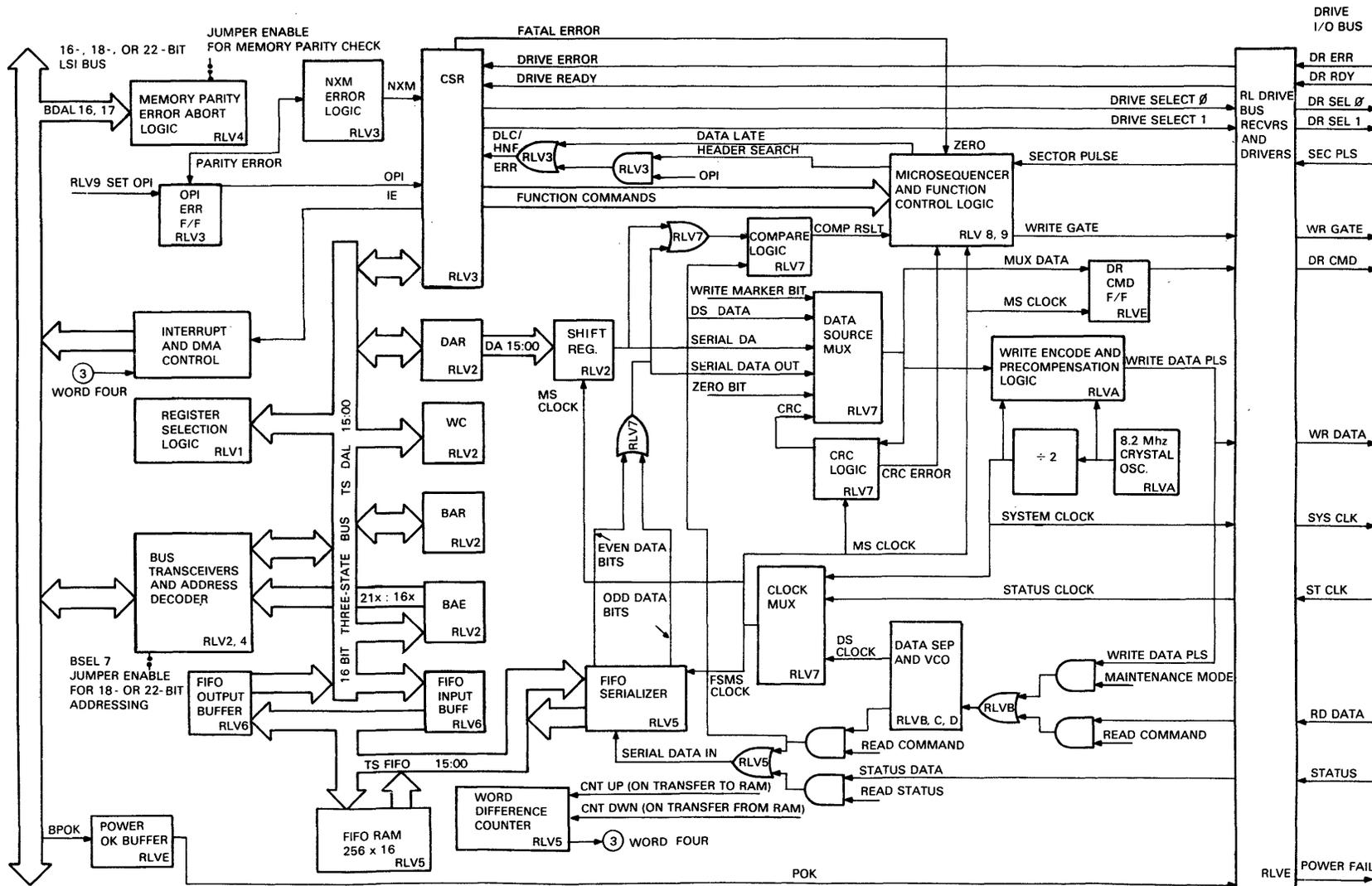


Figure 5-1 RLV12 Block Diagram

Cyclic redundancy checking (CRC) is used to check the serial data integrity from the disk drive. A CRC check word is created from the data being sent to the disk and this check word is written immediately following the data. When a header or data field is read from the disk, the incoming bit stream and CRC check word are checked for errors. If an error is detected, a header CRC or a data CRC error flag is set in the CSR.

The RLV12 has a  $256 \times 16$ -bit RAM to store data for DMA transactions. The RAM functions as a first-in, first-out (FIFO) memory that can store up to 256 words of data.

During a write command, a FIFO serializer is used with the FIFO RAM to convert parallel data into serial format to be written on the disk. During a read command, the FIFO serializer converts the serial data into parallel data to be loaded into the FIFO RAM.

## **5.2 BUS PROTOCOL**

The bus protocol logic, Figure 5-2, generates the control signals to read from, or write to, the controller. This logic uses the DC004 bus protocol chip. Two negative logic decoders and one positive logic decoder, provide the read and write signals to the five registers of the RLV12.

At register addressing time, R SYNC H clocks in the address bits (TSDAL 1, 2, and 3). These address bits are decoded to read or write to the five registers.

The DC004 generates a slave reply signal, SRLPY H, that becomes BRPLY L to the processor and completes the LSI-11 Q-bus protocol.

A single rank synchronizer monitors controller ready (CRDY) to enable the slave device (the addressed register). MRPLY L clocks in CRDY and generates S DEV EN H.

When CRDY is asserted, the RLV12 is ready to accept another command.

The signals XMIT H and REC H go to the DC005 transceivers that interface the LSI-11 bus and the 16-bit three-state DAL bus.

## **5.3 BUS TRANSCEIVERS**

The RLV12 uses DC005 bus transceivers as shown in Figure 5-3. These transceivers transmit and receive both data and address information. They interface the LSI-11 bus BDAL 0–15H signals and the RLV12 TS DAL 0–15H bus/address signals. BBS7 L must be asserted during address time to enable the transceivers. The transceivers are controlled by the signals XMIT H and REC H from the bus protocol logic.

The jumper pins connected to the transceivers select the device address and the interrupt vector of the RLV12.

## **5.4 PROGRAMMABLE REGISTERS**

The five programmable registers of the RLV12 interface to a three-state bus (TS DAL BUS). These registers receive address, data, and control information via the bus, and they return data and status information on the same bus.

### **5.4.1 Bus Address Register (BAR)**

The BAR, shown in Figure 5-4, has two DC006 binary counters. The BAR is loaded with the 16-bit bus address to which the first word of a DMA transfer is to be made. The signal WR BAR L enables the register to load this address.

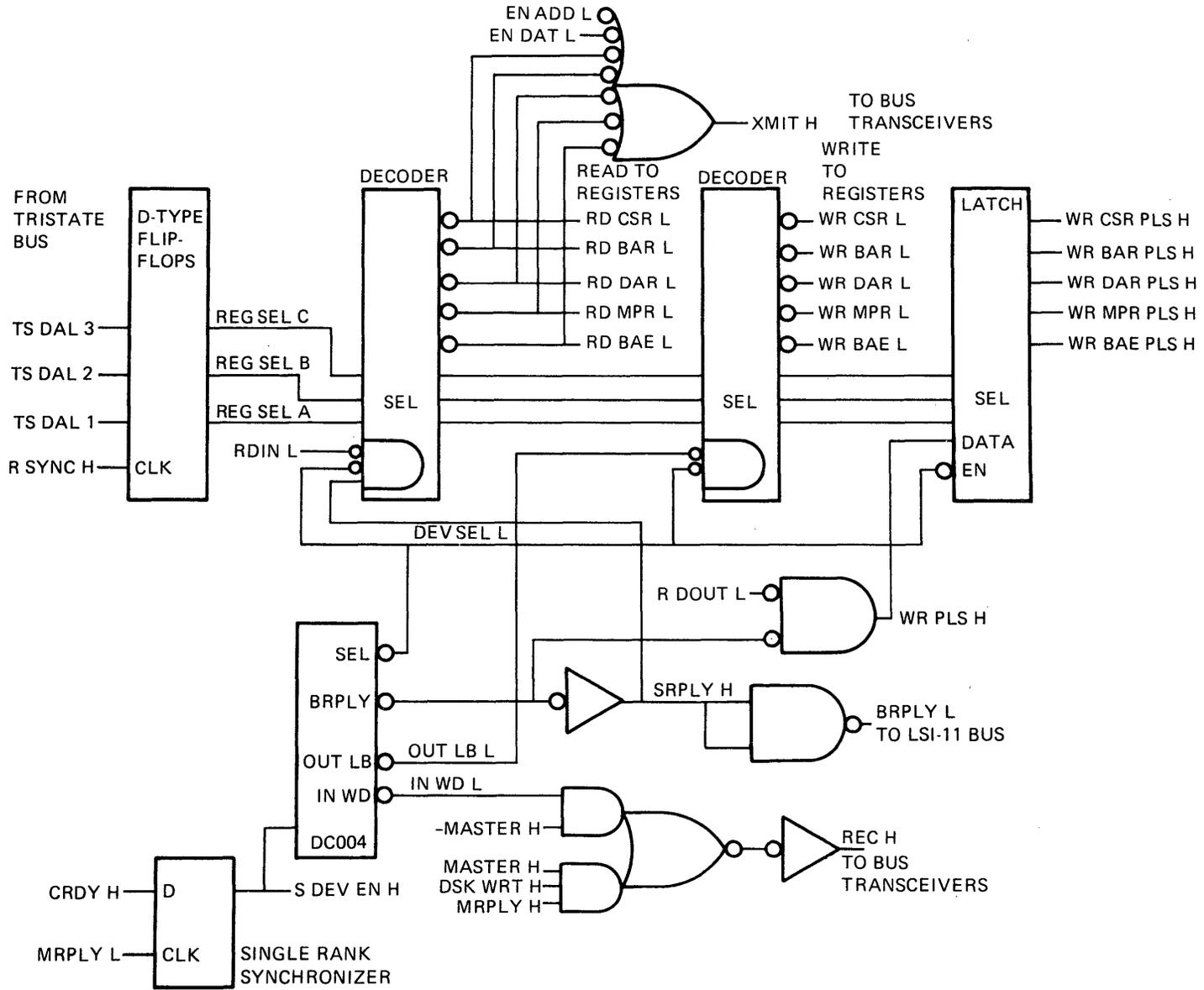
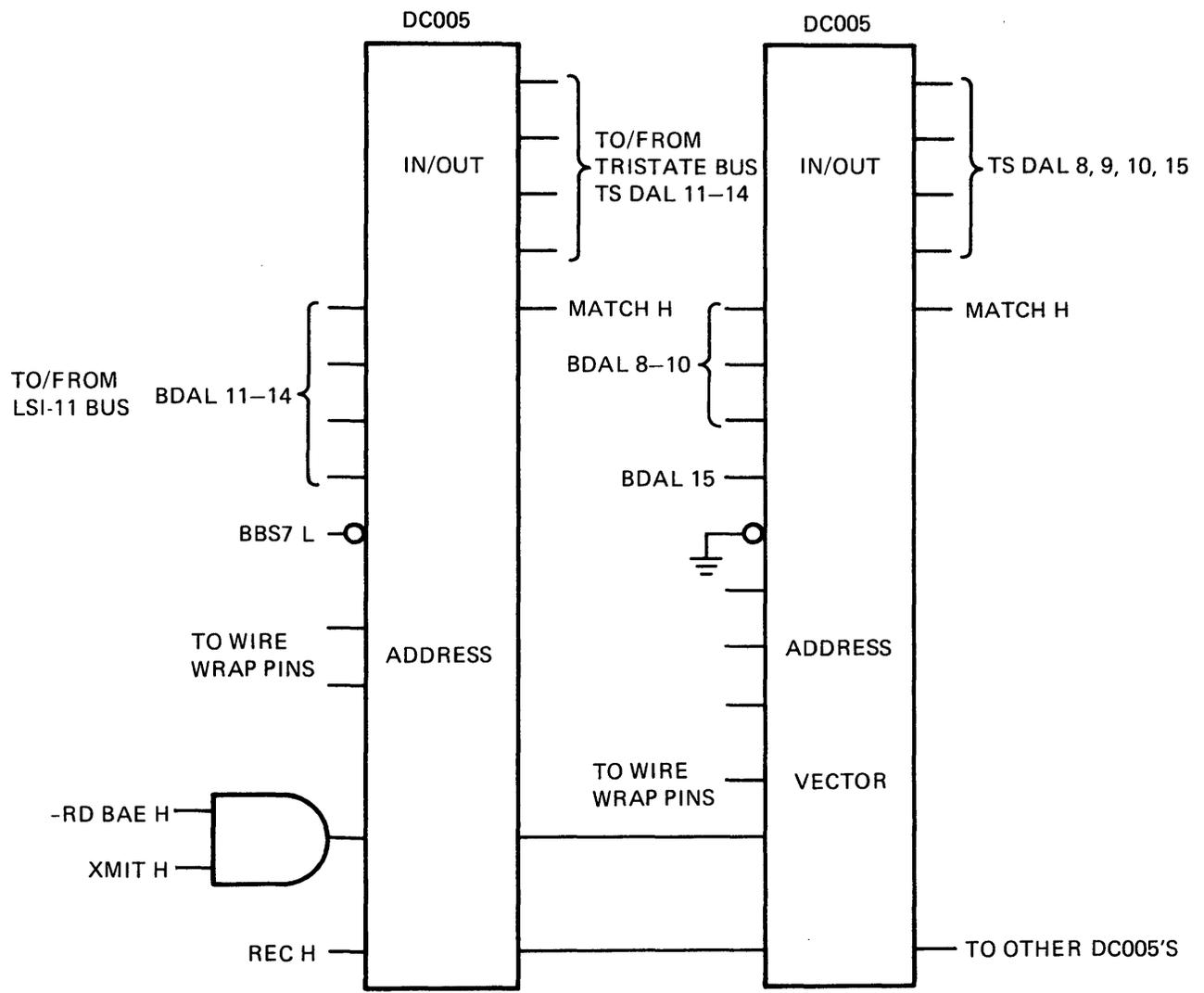
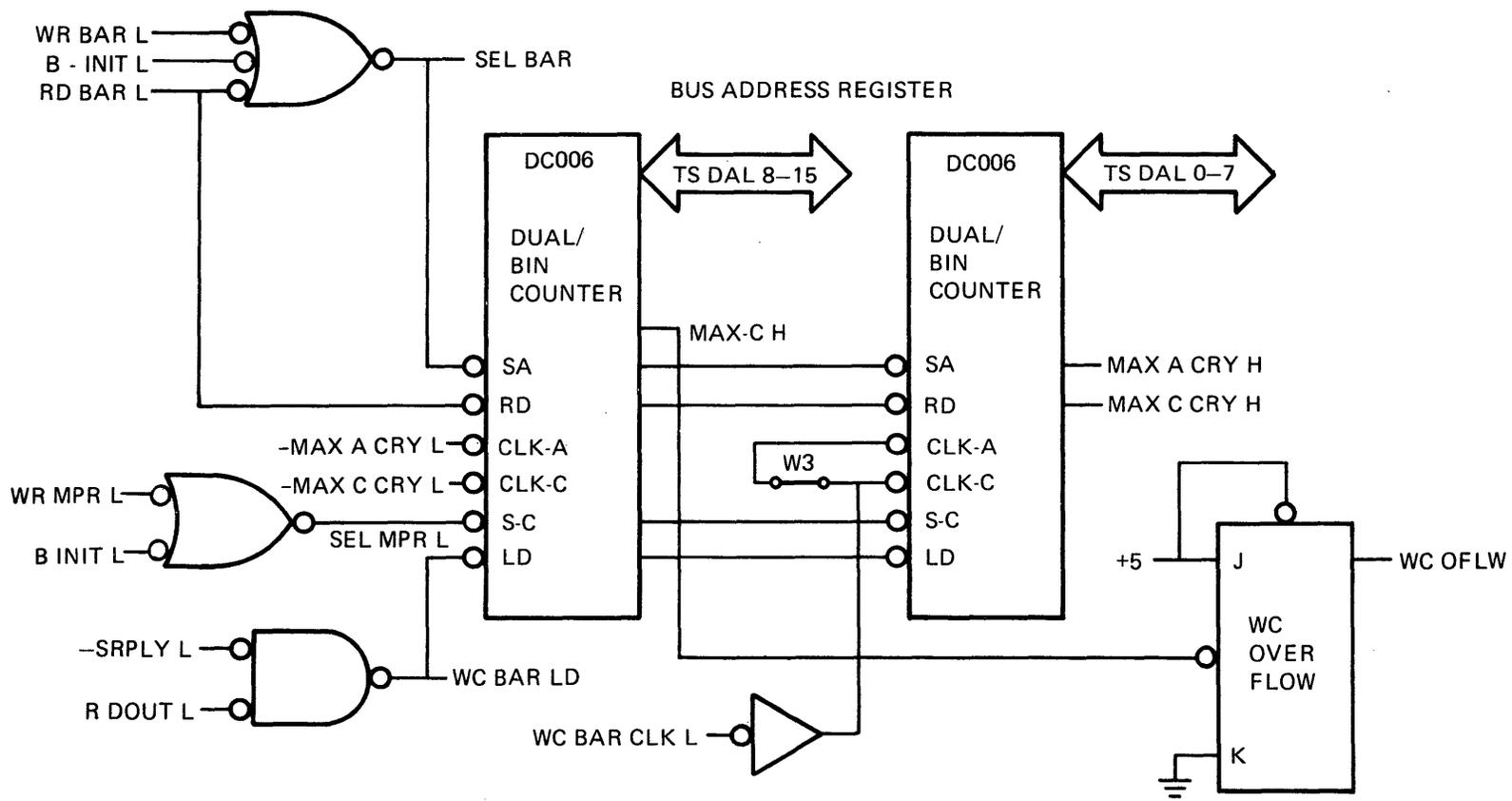


Figure 5-2 Bus Protocol Logic



MR-5739

Figure 5-3 Bus Transceivers



CZ-0518

Figure 5-4 Bus Address Register (BAR) Circuit

### 5.4.2 Bus Address Extension Register (BAE)

The BAE is a 6-bit register for the extended address bits, 16 through 21. (See Figure 5-5.) For 22-bit addressing, the BAE is loaded from TS DAL 0-5 using a write BAE command. Extended address bits 16 and 17 (BAE 0 and 1), must be loaded from the CSR via bits 4 and 5.

For 18-bit addressing, the extended address bits 16 and 17 must also be loaded into CSR bits 4 and 5.

#### NOTE

**Writing CSR bits 4 and 5 modifies BAE bits 0 and 1 and vice versa.**

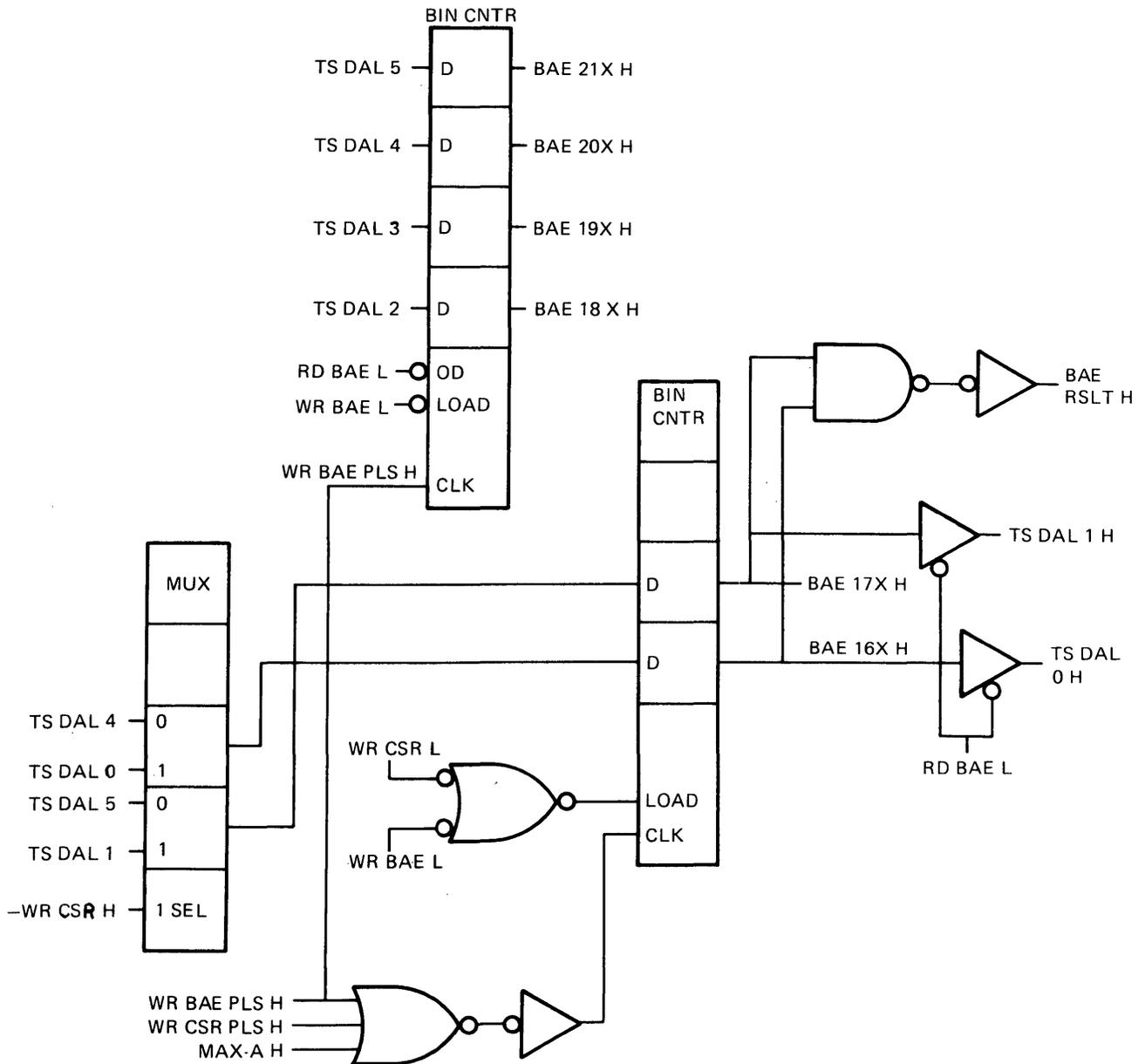


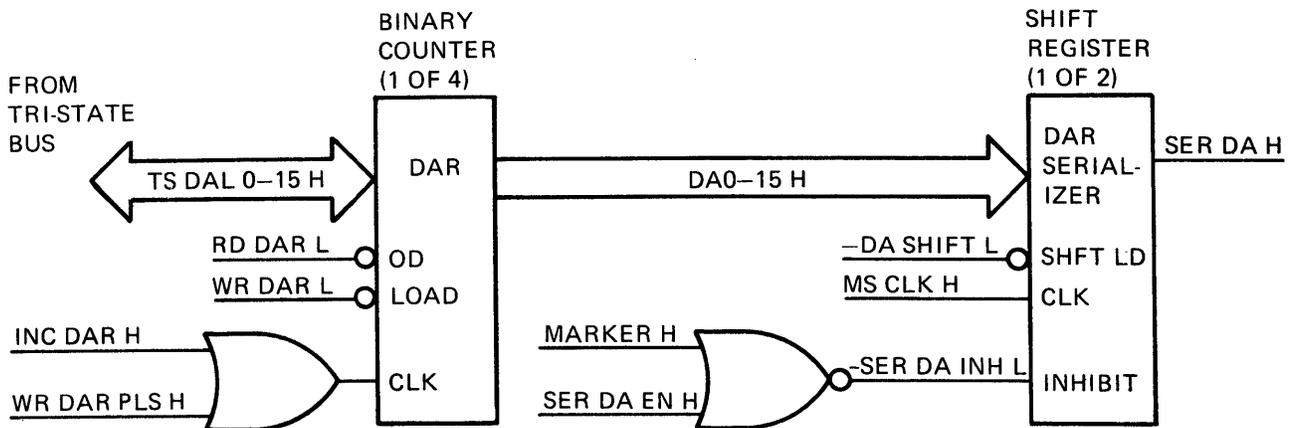
Figure 5-5 Bus Address Extension Register (BAE) Circuit

MR-5741

### 5.4.3 Disk Address Register (DAR)

The disk address register (Figure 5-6) holds the next sector address to read or write data on the disk. After reading or writing each sector, the contents of the DAR are incremented by one. The output of the DAR goes to the DAR serializer.

The DAR serializer has two 8-bit shift registers that receive parallel data in and shift serial data out. The DAR serializer sends the data to the header compare circuit.



MR-5742

Figure 5-6 Disk Address Register (DAR) Circuit

### 5.4.4 Control/Status Register (CSR)

The control/status register (Figure 5-7) is a holding register for the command to the microsequencer. The register also holds the interrupt enable bit, the controller ready signal, the drive-select bits, and error flags. A command to read the CSR gets the status information found in Table 5-1.

The controller ready bit (CRDY) is set by the hardware and indicates that the RLV12 is ready to accept a command. The CRDY bit must be cleared by software to initiate any command. After this bit is clear, the firmware generated signal PLS OPI H, starts the OPI watchdog timer.

The watch dog timer allows 550 ms for the controller to complete an instruction. The timer prevents the controller from taking too much time to perform an instruction and keeping out other instructions. If the instruction is not complete within 550 ms, the timer clocks the OPI flip-flop enabling OPI H, which turns off the controller.

Some of the CSR status error signals have two meanings depending on the state of the OPI flip-flop. When the D/H CRC flag is set without OPI H set, a data CRC error occurs; with OPI H set, a header CRC error occurs.

When the DLT/HNF flag is set without OPI H set, a data late error occurred; with OPI H set, a header not found error occurred.

During a DMA transfer, the NXM one-shot allows 10 microseconds for the addressed memory location to send and return BRPLY L. This one-shot prevents the RLV12 from indefinitely holding the LSI-11 Q-bus. If the one-shot times out, it clocks the NXM flip-flop setting NXM H and releases the LSI-11 bus.

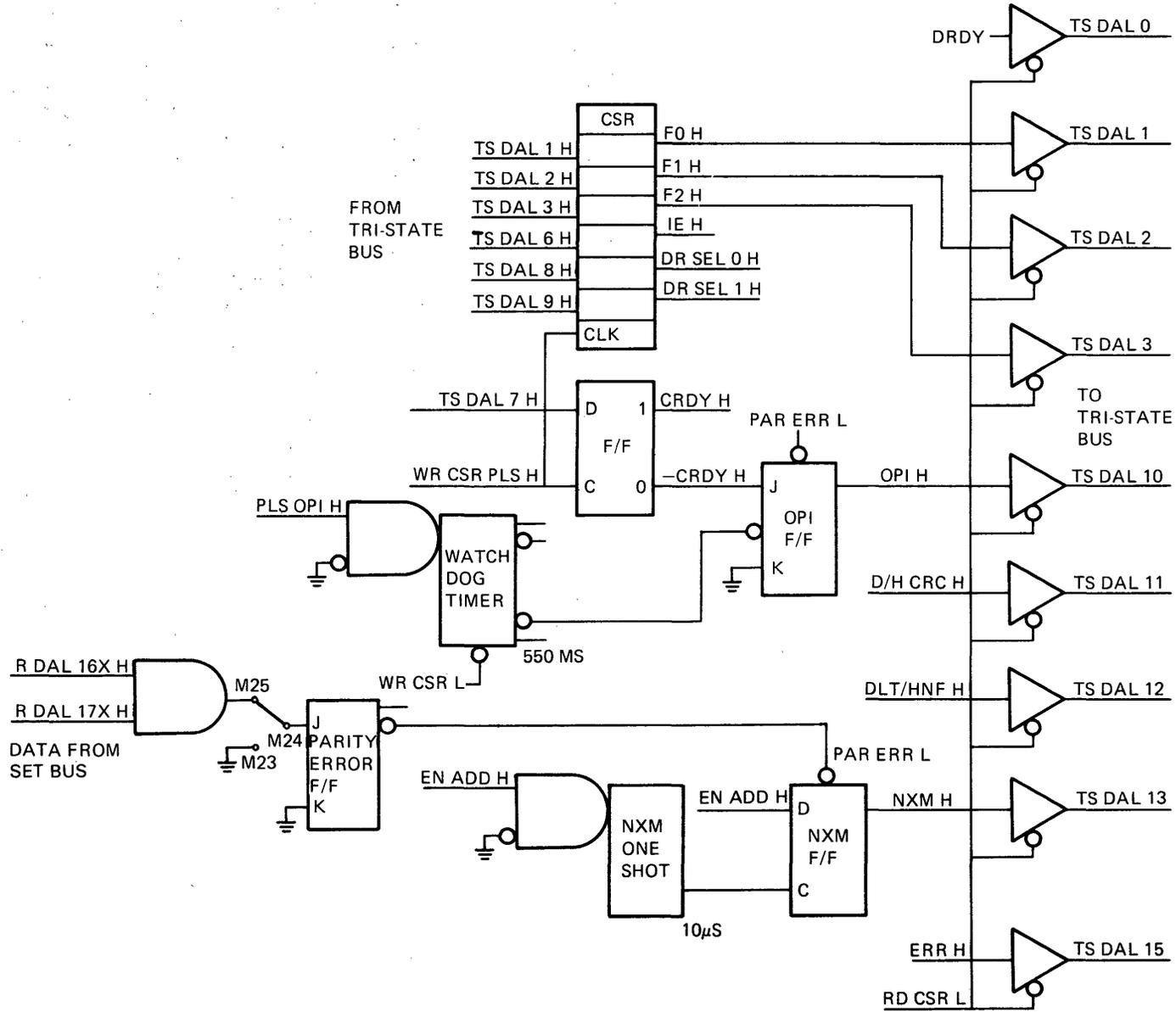


Figure 5-7 Control/Status Register (CSR) Circuit

**Table 5-1 Control/Status Register Bit Summary**

<b>CSR Bit(s)</b>	<b>Status</b>
0	Drive ready (DRDY)
1-3	Command (F0, F1, F2)
4,5	Extended address bits 16 and 17 (DAL 16-17)
6	Interrupt enable (IE)
7	Controller ready (CRDY)
8,9	Drive selected (DS)
10	Operation incomplete (OPI)
11	Data CRC error (DCRC)
10,11	Header CRC error (HCRC)
12	Data late (DLT)
10,12	Header not found (HNF)
13	Nonexistent memory (NXM)
10,13	Parity error abort (PAR ERR)
14	Drive error (DE)
15	Error flag (ERR)

If NXM H is set without OPI H set, a nonexistent memory error occurs. If NXM H is set with OPI H set, a memory parity error occurs. (A memory parity error forces both the NXM flip-flop and the OPI flip-flop set, providing the system is using parity memory.)

Any error that occurs also sets status bit 15.

#### **5.4.5 Multipurpose Register (MPR)**

The multipurpose register is made up of the FIFO buffers (shown in Figure 5-8) and the word count register (WC). (The WC register uses the same hardware as the bus address register shown in Figure 5-4.)

During a DMA transfer, the FIFO buffers function as DMA word buffers. The two's complement of the number of words to be transferred is loaded into the word count register (MPR). At the end of each DMA cycle, the word count is incremented. Referring to Figure 5-4, when the data transfer is complete, the WC register overflows creating MAX-C H and terminates the DMA transfer.

During a get status command or a read header command, the controller places the status information or header in the FIFO RAM. Reading the MPR places this data into the FIFO output buffer and on the TS DAL bus.

**FIFO Memory** – The FIFO memory is a first-in/first-out  $256 \times 16$ -bit RAM that can store up to 256 data words. A FIFO serializer converts serial data from the disk into parallel and places it in the FIFO memory (READ DATA operations). The FIFO serializer also converts parallel data from the FIFO memory into serial and sends it to the disk (WRITE DATA operations).

A word difference counter keeps track of the number of words coming from the disk to the FIFO buffer. After 4 words are read from the disk, the word difference counter signals the microsequencer to start a DMA transaction.

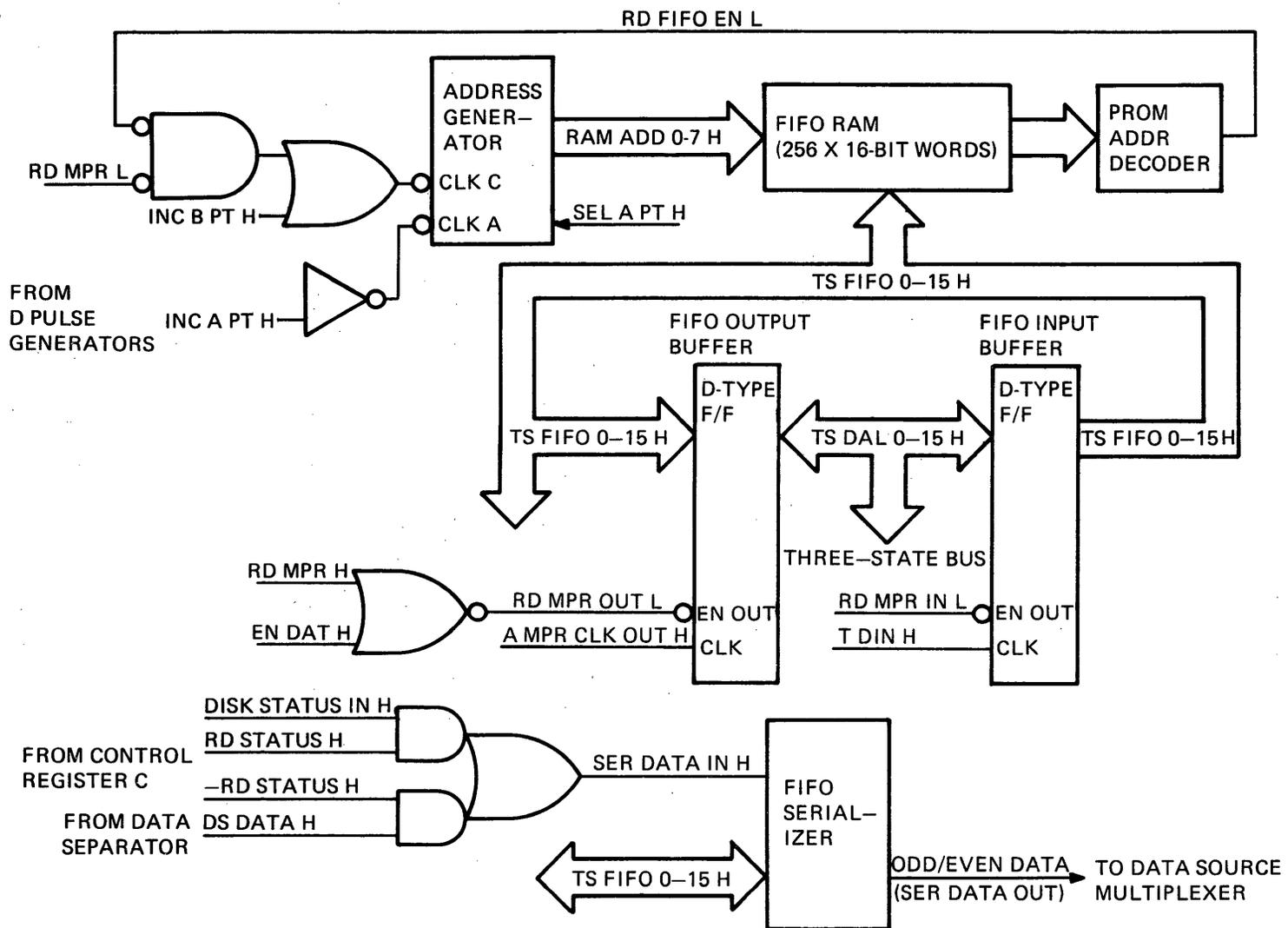


Figure 5-8 FIFO Buffer

## 5.5 DATA SOURCE MULTIPLEXER AND CRC GENERATOR

Data that is to be written on the disk goes to a data source multiplexer. (See Figure 5-1.) MUX SEL 0, 1, and 2 from the microsequencer determine which of the following inputs reaches the multiplexer output.

Serial Input	Source
SER DA (disk address)	DAR (disk address register)
SER DATA OUT	FIFO serializer
DS DATA	Data separator
CRC	CRC checker/generator
ZERO BIT	Ground
WRITE MARKER BIT	+5V

During WRITE DATA operations, the serial output of the multiplexer (MUX DATA H) goes to the write encoder precompensation circuit to be written on the disk. At the same time, a CRC checkword is being created by the CRC checker/generator. This checkword is then added to the end of the data field of the sector.

When the header or sector is read from the disk, the data is again sent through the CRC checker/generator. Any errors in the data or in the CRC word are detected, and a data CRC (DCRC) or a header CRC (HCRC) error bit is set in the control/status register.

During GET STATUS or SEEK operations, the output of the DAR shift register is passed through the data source mux. The MUX DATA output is passed through the drive command flip flop for transmission to the drive.

During the write data operations, the zero bit input to the data source mux causes the preamble zeros to be written on the disk. At the completion of the preamble zeros, a marker (synchronizing) bit must be written. The data source mux then selects the WRITE MARKER BIT input to accomplish this.

## 5.6 MICROSEQUENCER

The microsequencer decodes the function commands of the CSR and points to an address in the control store PROMs, where the routine resides to execute the command. The microsequencer sends an address (PR ADD 0-9 H) to the control store PROMs. (See Figure 5-9.)

**Control Store PROMs and Buffer Register** – The control store PROMs receive an address from the microsequencer and generate a 24-bit microinstruction at the outputs (PR OUT 0-23 H). The PROM outputs go to a buffer register, which is divided into 5 fields as follows.

1. Instruction field
2. T MUX SEL field
3. T FLAG X L (test flag don't care)
4. Constant field
5. LD CTRL register field

**Instruction Field** – The instruction field signals (INSTR 0, 1, 2, and 4) go to the conditional branch multiplexer to provide the microsequencer with the next address to access. These instruction signals generate the select inputs (S0 H and S1 H) and the enable inputs (FE L and RE L) to the microsequencer. INSTR 3 goes directly to the push/pop input of the microsequencer.

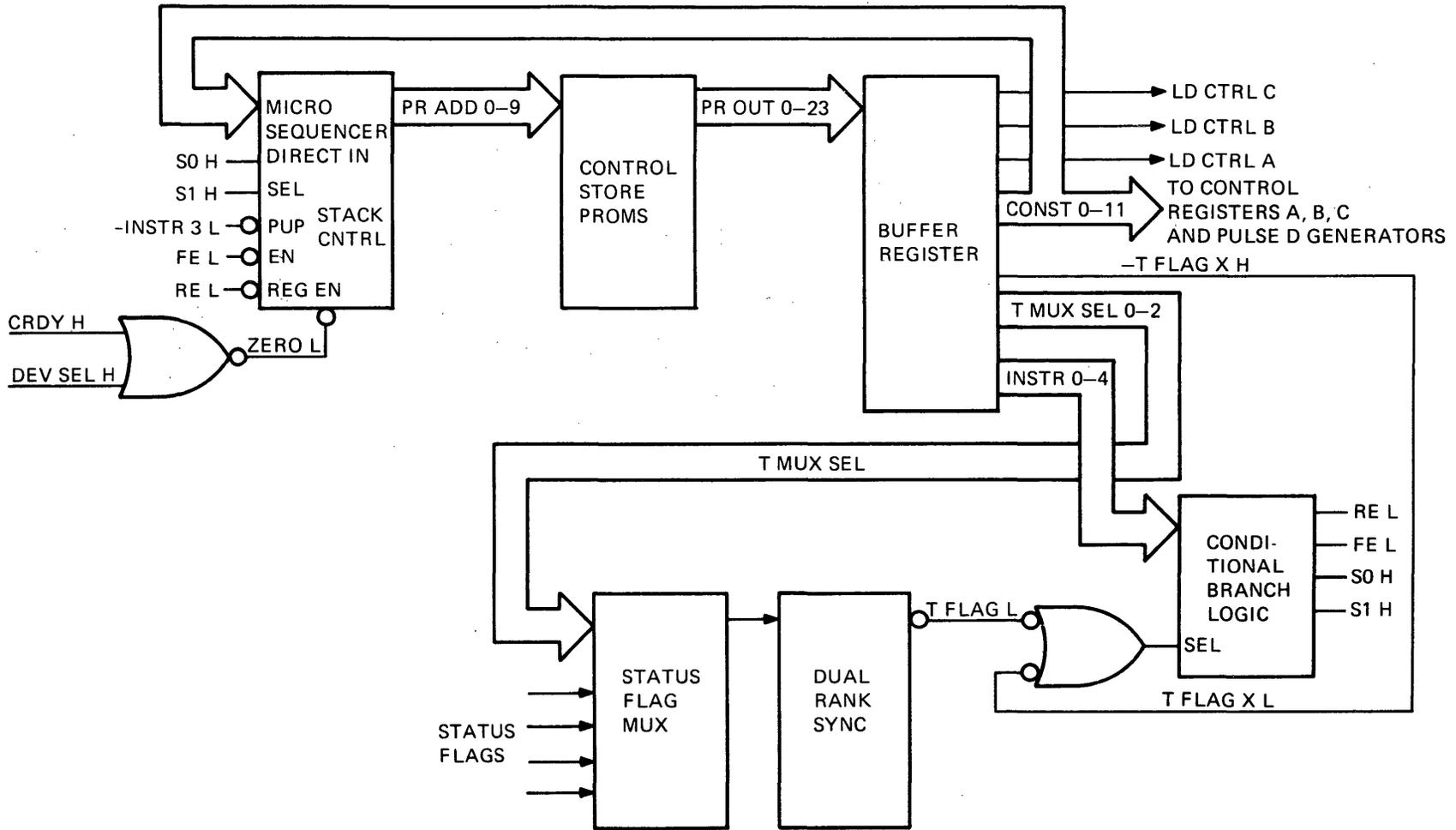


Figure 5-9 Microsequencer Logic

**T MUX SEL Field** – The T MUX SEL field signals select one of the status flags to enable the instruction from the conditional branch multiplexer. One of the status flags that go to the status flag multiplexer is enabled to pass to the dual-rank synchronizer. The status flag becomes T FLAG L and goes to the select input of the conditional branch multiplexer selecting the instruction field signals from the buffer register.

**T FLAG X L** – The T FLAG X L signal from the control store buffer register allows the micro-code to branch on a specific flag as follows.

1. When T FLAG X L is low, the instruction in the instruction field is executed unconditionally. (The state of T FLAG L is a don't care condition.)
2. When a status flag appears on the dual-rank synchronizer, it asserts T FLAG L. If, at the same time, T FLAG X L is high (unasserted), the microsequencer conditionally executes the instruction in the instruction field.
3. If both T FLAG X L and T FLAG L are high, the microsequencer skips to the next instruction in the Control Store PROMs.

**Constant Field** – The constant field has two purposes. It provides a direct input to the microsequencer and it provides inputs to load one of three control registers; (A, B, and C) and the two D pulse generators. (See Paragraph 2.7.)

**LD CTRL A, B, C** – When loading a control register or pulse generator, the signals LD CTRL A, B, or C are decoded to determine which register or pulse generator to load.

**Fatal Error Clearing Logic** – If a fatal pulse occurs, it halts the clock on the RLV12 and sets CRDY H. CRDY H generates ZERO L, which resets the microsequencer to location zero, where it stays until the controller is restarted (CRDY is cleared). When the controller is accessed, DEV SEL H clocks the microsequencer, which initializes the microsequencer.

## 5.7 CONTROL REGISTERS AND PULSE GENERATORS

The control signals for the RLV12 logic, such as clock selection, FIFO control, and data path control come from 3 control registers (A, B, and C) and two D pulse generators. These registers and D pulse generators are loaded from the constant field of the microsequencer's control store buffer.

Register A provides clock selection, multiplexer selection, and some enable signals.

Register B provides register selection and FIFO control.

Register C provides data path control.

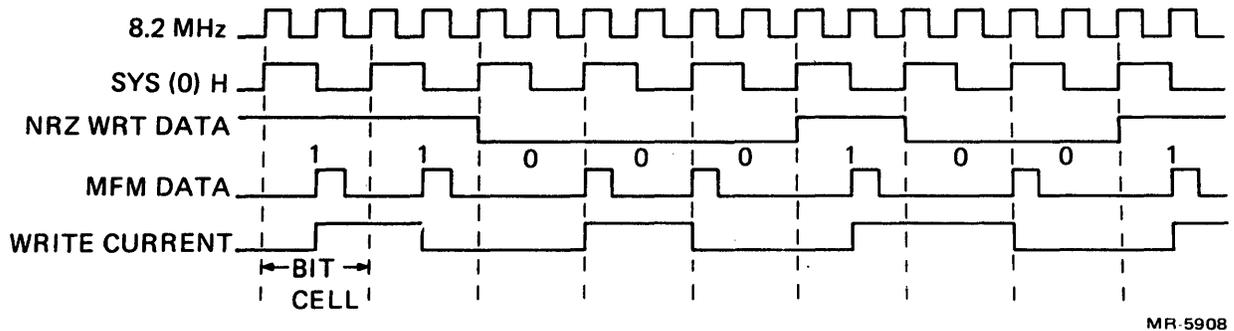
Two D pulse generators – one positive and one negative – provide pulses for clearing, incrementing, and decrementing the logic.

## 5.8 WRITE ENCODER AND PRECOMPENSATION LOGIC

The write encoder converts binary data into modified frequency modulated (MFM) data, which is recorded on a disk.

MFM is a magnetic recording method for disk drives, in which a clock signal is encoded in the flux transitions recorded on the disk. Therefore, when reading data from the disk, one can synchronize on the data transitions. Then with a phase-locked loop and MFM decoder one can recover the clock and data accurately.

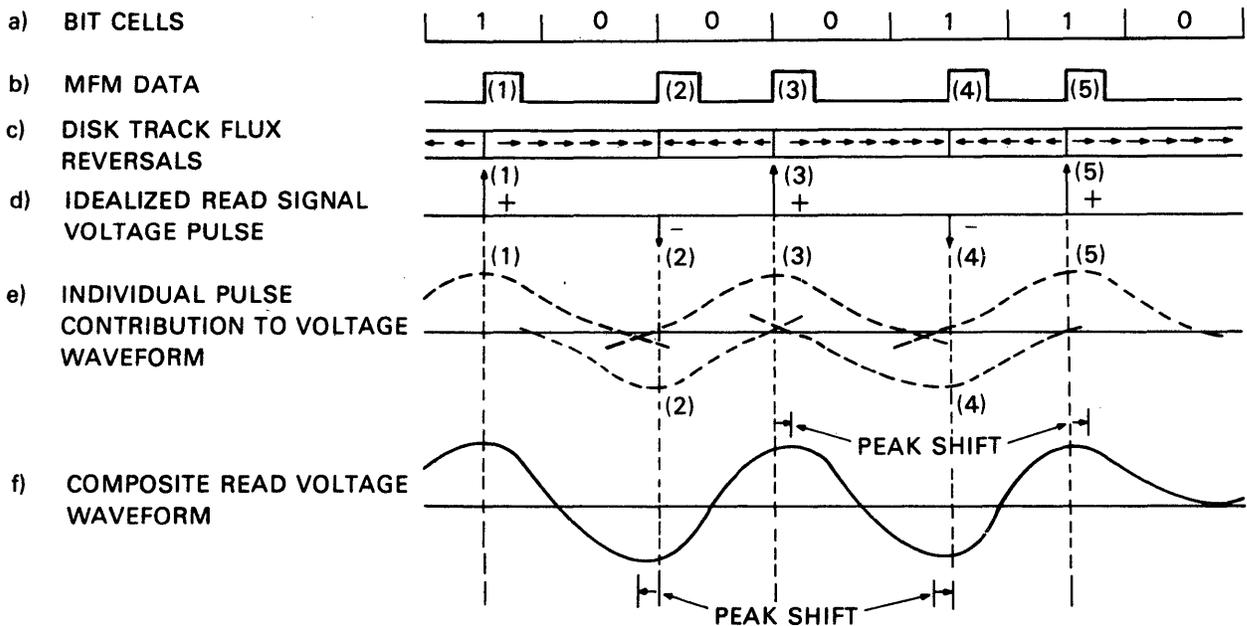
Each bit cell, shown in Figure 5-10, can have a transition at its beginning or at its center or may have no transition at all. Each 1 produces a transition at the center of the bit cell time, a 0 preceded by a 1 produces no transition, and a 0 preceded by a 0 produces a transition at the beginning of the bit cell time. Therefore, with MFM, encoding flux transitions are always present even with an all 0s or all 1s data pattern.



MR-5908

Figure 5-10 MFM Encoding

A phenomenon associated with magnetic recording is that adjacent flux transitions appear to be moved from where they were written (called peak shifting). The direction of the peak shift is linked to the position of the MFM pulses. Two pulses close together shift the peak of the read voltage away from each other. (See Figure 5-11.)



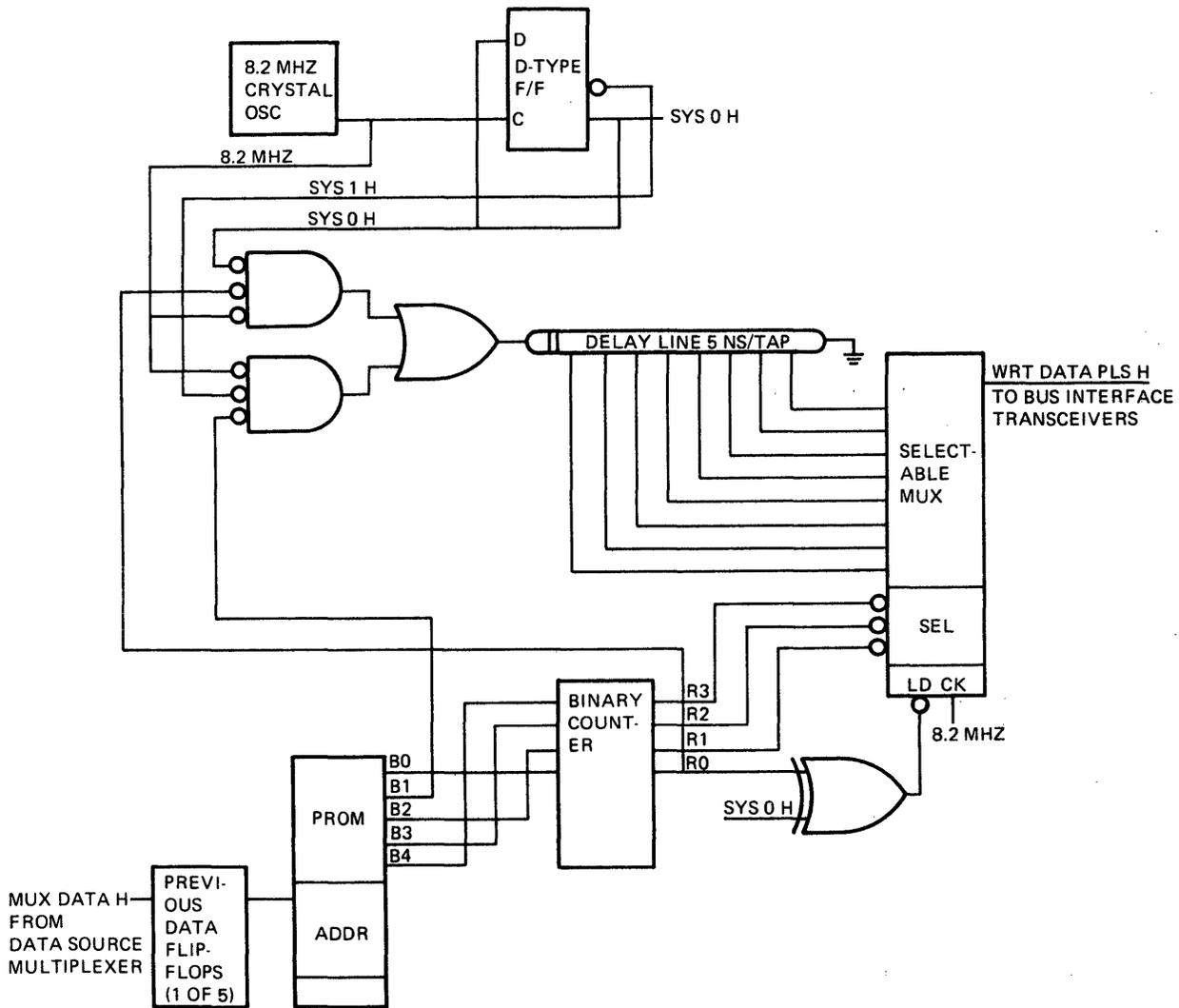
MA-1160

Figure 5-11 Peak Shift Waveforms

To offset this peak shifting, the write encoder uses a delay line to shift the data in the opposite direction to that expected by the peak shift. This shifting of the data is called precompensation.

The delay line has 9 taps. Each tap delays the data input 5 ns more from its entry point. (See Figure 5-12.) All 9 taps are inputs to a multiplexer. (The center tap is a reference line.)

The select lines to the multiplexer come from a PROM and binary counter which keeps a history of the previous data. The select lines determine whether to advance or delay the new data from the previous data, creating precompensated MFM data.

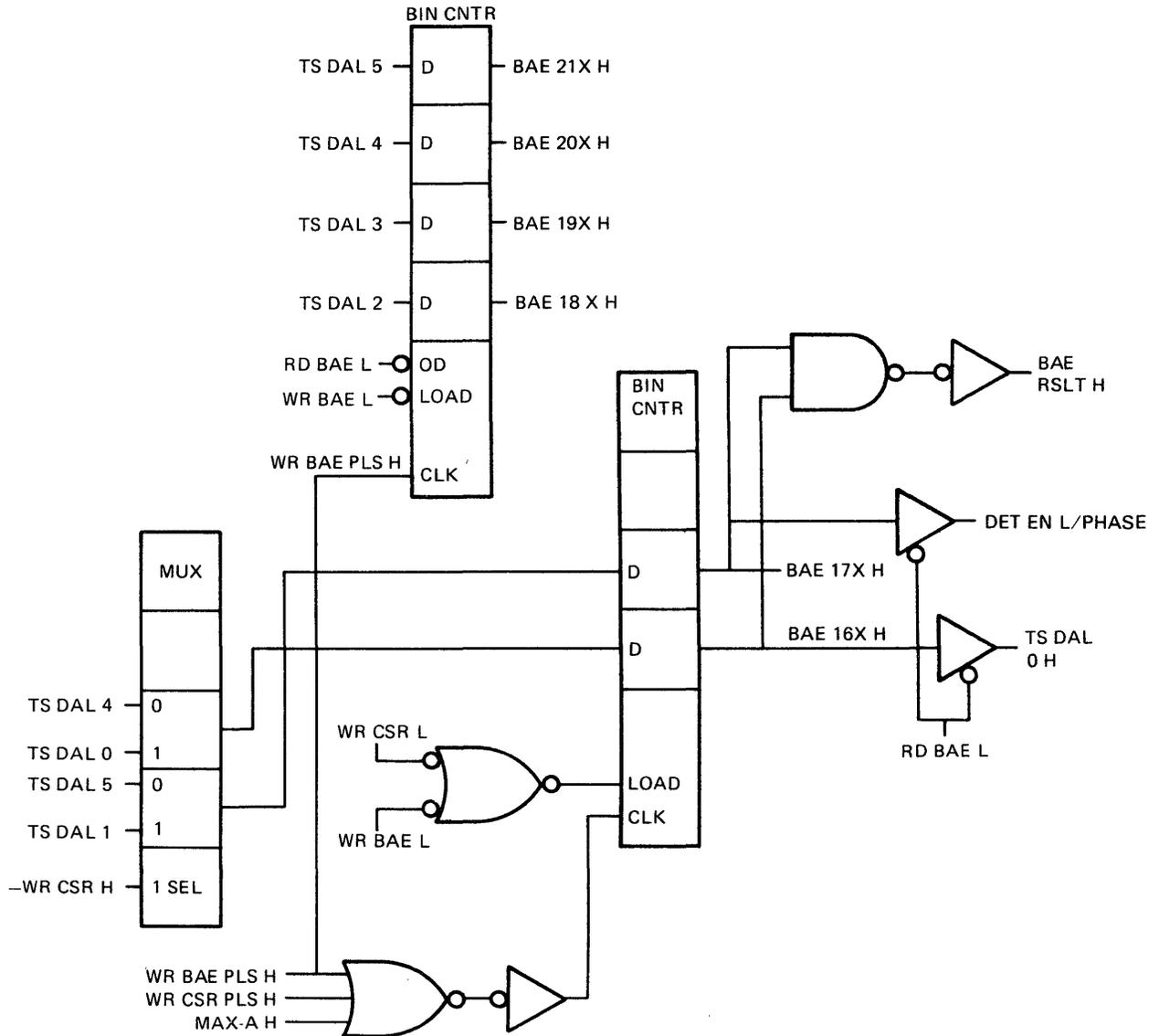


MR-5746

Figure 5-12 Write Encoder and Precompensation Circuit

### 5.9 DATA SEPARATOR READ CIRCUIT

The data separator read circuit takes the MFM data from the disk drive and produces binary data and a clock. This circuit, shown in Figure 5-13, uses a phase-locked loop to generate a clock signal to synchronize to the MFM data. (A variable capacitor sets the free-running frequency of the VCO. This frequency is set at the factory and should not be changed.) Then, the read circuit decodes the MFM data. The serial binary (NRZ) data then goes to the FIFO serializer, as DS DATA H, and is clocked in by DS CLK.



MR 5741

Figure 5-13 Data Separator Read Circuit



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-RLV12-TD-001

Fold Here

Do Not Tear - Fold Here and Staple

**digital**



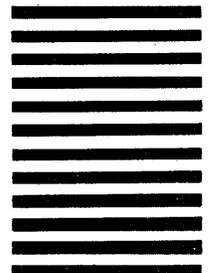
No Postage  
Necessary  
if Mailed in the  
United States

**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO. 33 MAYNARD, MA.

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation  
301 Rockrimmon Boulevard South  
Colorado Springs, Colorado 80919



Educational Services  
Development and Publishing

