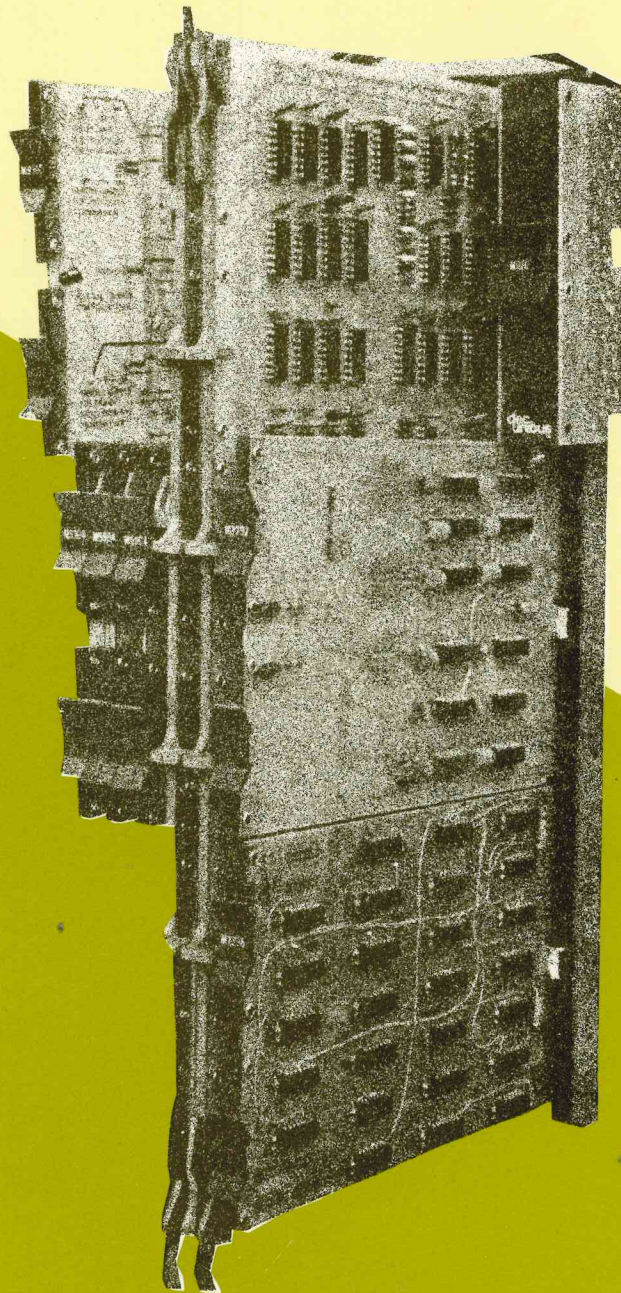


RH11/RH70 MASSBUS Controllers

A Self-Paced Course

Workbook 3



digital



RH11/RH70 MASSBUS CONTROLLERS SELF-PACED COURSE

WORKBOOK III

Document EY-D3023-WB-001
A Portion of Course EY-D3038-SP-001

Data Path Block Diagrams



RH11 Functional Block Diagrams



RH70 Functional Block Diagrams



Data Bus Interfacing



educational services development and publishing
digital equipment corporation
colorado springs, colorado

Copyright © 1979 by Digital Equipment Corporation

The information 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.

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

DEC
LSI-11
PDP

DECUS
Massbus
Q-Bus
VAX

Digital Logo
Omnibus
Unibus

OUTLINE

Data Path Block Diagrams

Unibus Receivers/Drivers	DP-1
Bus Address Register	DP-1
Data Multiplexer	DP-1
Input Multiplexer	DP-1
Silo	DP-3
Word Count Register	DP-4
Drive Word Count Register	DP-4
Start Counter	DP-4
Parity Checker/Generator	DP-4
Write Check Circuitry	DP-5
RH70 Differences	DP-5

RH11 Functional Block Diagrams

Write Operation	11-1
Read Operation	11-3
Write Check Operation	11-6

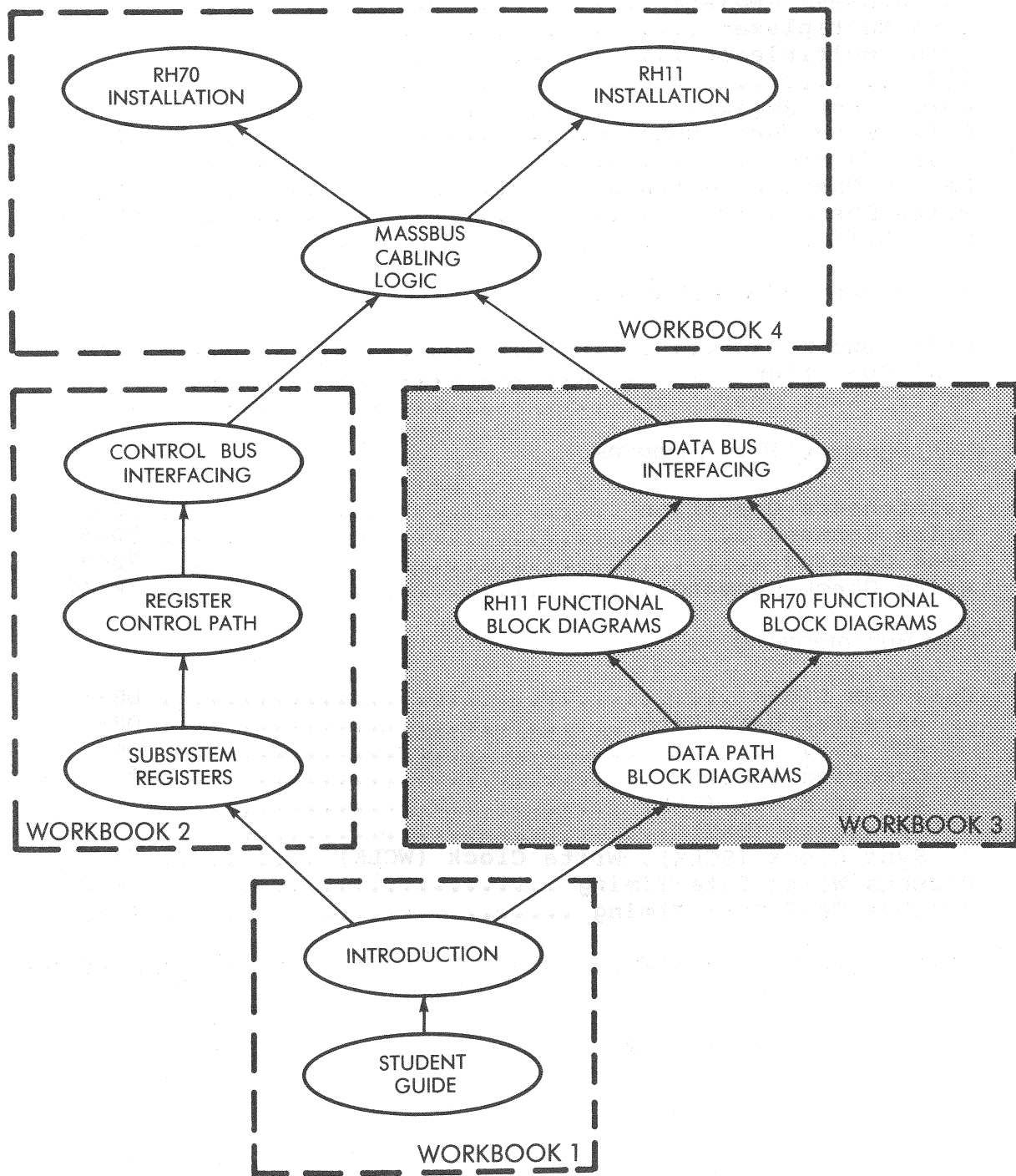
RH70 Functional Block Diagrams

Maintenance Operation	70-1
Write Command	70-5
Read Command	70-9
Write Check Command	70-12

Data Bus Interfacing

Data Bus Lines	DB-1
Parallel Data Path	DB-1
RUN	DB-1
Occupied (OCC)	DB-2
End of Block (EBL)	DB-2
Exception (EXC)	DB-2
Sync Clock (SCLK), Write Clock (WCLK)	DB-2
Massbus Write Data Timing	DB-2
Massbus Read Data Timing	DB-4

COURSE MAP



Data Path Block Diagrams

DATA PATH BLOCK DIAGRAMS

INTRODUCTION

The "Introduction" module gave you a brief look at the data path of the RH controller. Now, you will study a detailed description of the data path.

This module is divided into two sections: A functional description of each component in the RH11 data path block diagram and an explanation of how the RH70 data path differs from that in the RH11.

As you read through the module, use the block diagram as a reference. Each component of the block diagram has a mnemonic representing a page number in the Field Maintenance Print Set for easy correlation.

OBJECTIVES

Given a list of data path block diagram components and functions, be able to match the blocks with the function.

ADDITIONAL RESOURCES

RJS04 Maintenance Manual, Pages 4-3 through 4-11

RJP04 Maintenance Manual, Pages 5-3 through 5-11

RJP05/06 Maintenance Manual, Pages 5-3 through 5-11

TJU16 Maintenance Manual, Pages 5-3 through 5-11

RWP04 Maintenance Manual, Pages 5-3 through 5-21

DATA PATH BLOCK DIAGRAMS

UNIBUS RECEIVERS/DRIVERS

The Unibus A and B receivers/drivers are located on the left side of Figure 1. The drivers strobe information from the address and data lines onto the Unibus. The address lines are strobed from the Bus Address Register shown at the top of the figure. The data lines are an output of the Silo output buffer. The upper (Unibus A) or lower (Unibus B) receivers/drivers are enabled by the signal SEL BUSA. SEL BUSA is generated from the Port Select Bit in CS1 (Bit 10). The receivers are used during a Write or Write Check command. The output of the receivers is sent to the Data Multiplexer (DMX).

BUS ADDRESS REGISTER

This register and Bits 8 and 9 of the Control Status Register 1 comprise an 18-bit memory address. This register is loaded by the program with the starting memory address for a data transfer operation. Each time a Direct Memory Address (DMA) transfer is made, the register is incremented or decremented (reverse operation on a tape drive) twice. The register is strobed onto the address lines via the Unibus drivers during the DMA.

DATA MULTIPLEXER

The Data Multiplexer (DMX) receives data from Unibus A or Unibus B. This data is multiplexed by the signal SEL BUSA (not shown on the block diagram). During a Write command, the data is accepted by the Input Multiplexer (IMX). During a Write Check command, the data is routed to the XOR gate at the top of the block diagram.

INPUT MULTIPLEXER

The Input Multiplexer (IMX) receives data from either memory or the device. During Write commands, the Data Multiplexer supplies the write data. This data is originally from the memory address supplied by the BAR. During a Read or Write Check command, data from the device is sent to the Input Multiplexer. The output of the multiplexer is routed to the input buffer of the Silo.

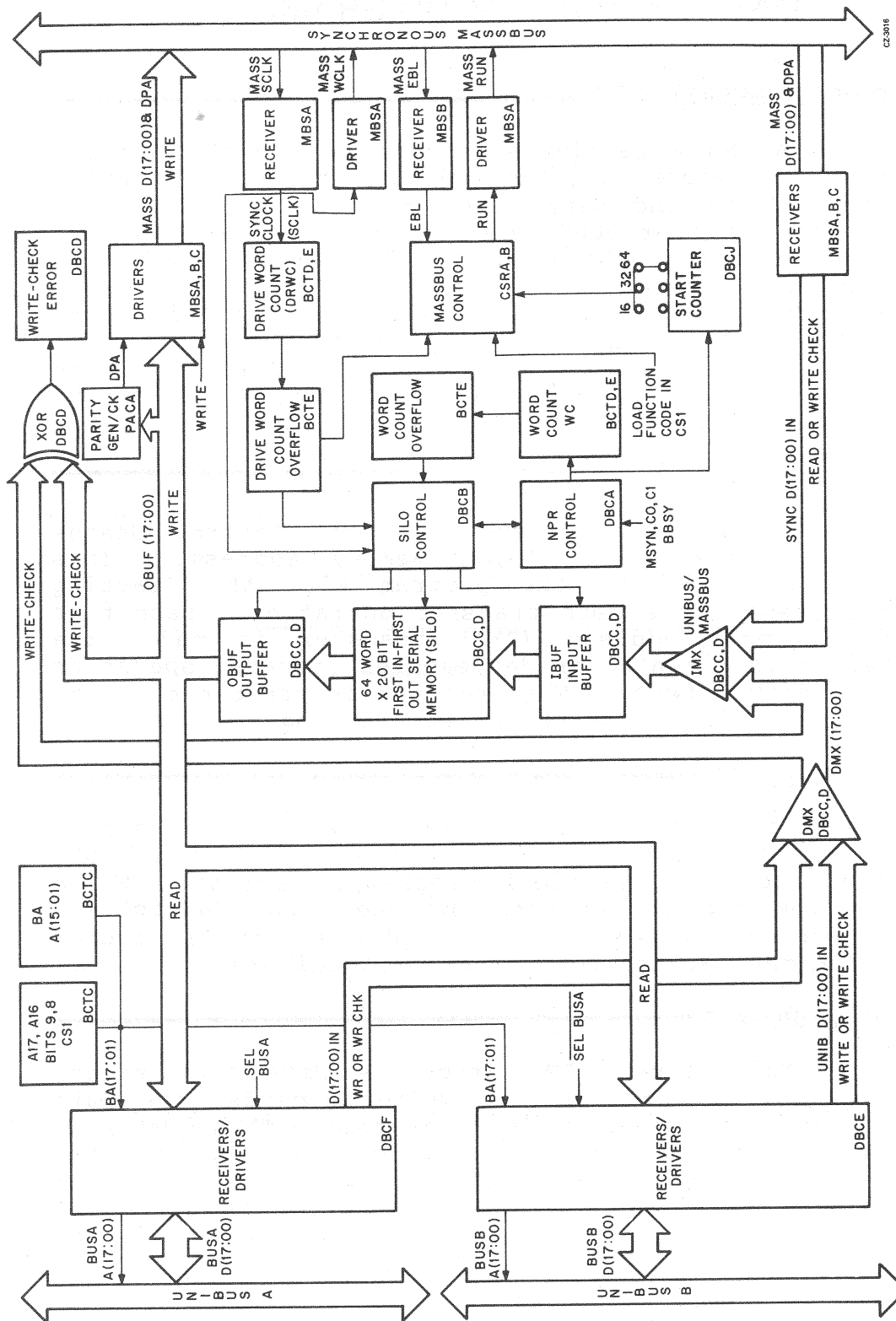


Figure 1 RH11 Data Path Block Diagram

SILO

The Silo is a 64 X 20-bit first-in, first-out memory. The Silo consists of five 64 X 4-bit MOS-type (DEC 3341) memory chips wired in parallel to obtain the desired length of 20 bits. Sixteen bits are used for PDP-11 words and two more bits are used for the Unichannel 15 system (18-bit) or DECSYSTEM-20 usage. One bit is used for parity and one bit is used for control of errors.

The Silo chips require -12 volts for operation. There is a regulator which drops the available -15 volts from the system, to -12 volts for the Silo. If this circuit should fail, the wrong voltage could cause erratic operation or destroy the memory chips. Incorrect system voltage could cause the regulator to malfunction. Before installing a new Data Buffer and Control Module (M7294), check for the presence of -15 volts. If the voltage is not correct, adjust it and then test the RH11 with the diagnostics. There are routines within the diagnostics that rigorously test the Silo for proper operation.

The Silo acts as a buffer area between the synchronous Massbus and the asynchronous Unibus. Data is fed into the input buffer from the Input Multiplexer. The Silo control logic then moves the data into the Silo (unless it is already full). The Silo control logic sequences data through the 64 locations to the output buffer. The Silo, and the two buffers, hold a total of 66 words.

The buffer area is filled before data is transferred to the Massbus during a Write operation. This gives the slower Unibus a head start over the faster Massbus.

The data words are then transferred over the Massbus to the peripheral. When a word is strobed out of the output buffer, another one is sequenced into it from the Silo. The input buffer is then gated into the Silo and a Non-Processor Request (NPR) is made to fill the input buffer.

During a Read operation, the Input Multiplexer receives data from the drive. When the word reaches the output buffer, an NPR is made to transfer the data to memory.

The data flow and NPR process will be discussed in detail in the "Functional Block Diagrams" modules.

WORD COUNT REGISTER

The Word Count Register is loaded by Software with the two's complement of the number of words to be transferred. Each time an NPR transfers a data word, the Word Count Register is incremented toward zero by the NPR control logic. When word count overflow is asserted during a Write operation, NPR transfers are terminated. When word count overflows during a Read, data transfers from the drive to the RH are terminated. In both cases, the Silo empties before the operation is terminated.

DRIVE WORD COUNT REGISTER

The Drive Word Count Register and Word Count Register are loaded at the same time with the same value. The drive word count is incremented toward zero each time a word is transferred over the Massbus. The Drive Word Count Register keeps the RH active until the proper amount of words have been transferred over the Massbus.

Here is an example of the process. During a Write command, the Word Count Register overflows with thirty words remaining in the buffer. The drive word count keeps the Silo control and Massbus control active until the thirty words are transferred over the Massbus. When drive word count overflow is asserted, the Write command terminates.

START COUNTER

This counter is an 8-bit up-counter used during a Write operation. It is cleared at the beginning of the operation. The counter is incremented at Master Sync (MSYN) time as the words are transferred into the buffer. The output, which is strappable, is asserted after 16, 32 or 64 counts. Most installations use the 64-word jumper. RUN (the signal to start the operation in the drive) will not be asserted from the Massbus control, until the start counter reaches the proper count. This ensures that the Unibus receives a 64-word head start before the drive begins writing.

PARITY CHECKER/GENERATOR

The parity circuit generates or checks for odd parity on Massbus data transfers. By setting the Parity Test flip-flop (Bit 04 of CS2) the parity circuit will check and generate even parity. The drive can check and generate only odd parity. The diagnostics use this parity check capability to test the circuits in the RH and the drive.

WRITE CHECK CIRCUITRY

The Write Check Circuitry consists of eighteen XOR gates. The data from memory reaches the XOR gates via the Data Multiplexer. The data from the drive goes through the Silo. These two words are compared bit for bit for equality. The Write Check command is covered in detail in the "Functional Block Diagrams" modules.

RH70 DIFFERENCES

Figure 2 is the data path block diagram for the RH70. A comparison of Figure 1 and Figure 2 shows that the RH11 and RH70 have the same components in a different arrangement.

The eight-word data buffer consists of eight registers. These registers are configured as a first-in first-out buffer. The buffer is designed to accept or provide memory with single- (16 bits) or double-word (32 bits) data. The buffer also converts double words to single words for transfer to the Massbus. Parity checking and generating is also provided for Massbus and memory data words.

During maintenance, the data is transferred over the Unibus to test the data buffer.

The 22-bit address needed by the PDP-11/70 is provided by the Bus Address and Bus Address Extension registers.

The Start Counter in the RH70 counts four data transfers. If these transfers are double-word, the buffer should be full.

SUMMARY

The following is a summary of the function of the components in the Data Path Block Diagram:

- Unibus Receiver/Drivers - receive and transmit data to/from the Unibus
- Data Multiplexer - selects data from Unibus A or Unibus B to be fed to the Silo or Write-Check circuitry
- Input Multiplexer - selects data from memory or drive
- Buffer

RH11 - 66 words of first-in first-out memory to buffer data between the asynchronous memory and the synchronous device

RH70 - 8 words of first-in first-out buffer

- Word Count Register - keeps track of the number of words transferred between memory and RH
- Drive Word Count register - keeps track of the number of words transferred between the RH and the device
- Parity Generator/Checker Circuit - checks and generates odd parity for data transfers on the Massbus
- Write Check Circuit - checks the data word from memory against the previously written word on the drive for equality during a Write Check command
- Start Counter - allows the buffer to fill before RUN is asserted to the drive

EXERCISES

Answer the following questions using any of your reference materials. Discuss any problems you have with another student or your course administrator. The solutions are on the next page.

1. How is either Unibus input selected on the RH11?
2. During a RH11 Read operation, from where does the Input Multiplexer receive its data?
3. What is the purpose of the data buffer?
4. What function does the Drive Word Count Register perform?
5. What function does the Word Count Register perform?
6. The parity circuitry only generates or checks for odd parity.
 - A. True
 - B. False

--- SOLUTIONS ---

1. By the Port Select flip-flop (Bit 10 of CS1) generating the signal SEL BUSA
2. From the drive
3. To buffer the data between the synchronous device transfers and the asynchronous memory transfers
4. To keep track of the number of words transferred over the Massbus
5. To keep track of the words transferred between memory and the RH
6. B. False By setting PAT in CS2, even parity can be generated or checked.

RH11 Functional Block Diagrams

RH11 FUNCTIONAL BLOCK DIAGRAMS

INTRODUCTION

The "Data Paths Block Diagrams" module described each of the components that make up the Data Path. The module you are about to read utilizes three separate block diagrams in the description of the 3 basic data commands:

- Write
- Read
- Write Check

The heart of the Data Path is the Silo. This course module explains Silo operation during each of the data commands. Knowing how the Silo operates will help you troubleshoot the controllers because the Silo is a high failure item.

As you read through the module, use the block diagrams as a reference. Each component of the block diagram has a mnemonic representing a page number in the Field Maintenance Print Set.

OBJECTIVES

Using the supplied block diagrams, you will be able to describe the three basic functions:

1. Write
2. Read
3. Write Check

To demonstrate proficiency, answer all multiple-choice questions correctly.

ADDITIONAL RESOURCES

RJS04 Maintenance Manual, Pages 4-3 through 4-19

RJP04 Maintenance Manual, Pages 5-3 through 5-19

RJP05/06 Maintenance Manual, Pages 5-3 through 5-19

TJU16 Maintenance Manual, Pages 5-3 through 5-19

RH11 FUNCTIONAL BLOCK DIAGRAMS

WRITE OPERATION

A Write operation transfers data from memory to the drive. Before data is transferred, the buffer area (Silo) is filled, giving the slower Unibus a head start on the faster Massbus. When the Drive Word Count Register overflows, the Write command is terminated.

Below is an explanation of the Write command (see Figure 1). The program loads the Unibus Word Count Register and Drive Word Count Register with the 2's complement of the number of words to be transferred. (Other local and remote registers are also loaded.) Then the Control and Status Register 1 is loaded with the Write command and GO is set.

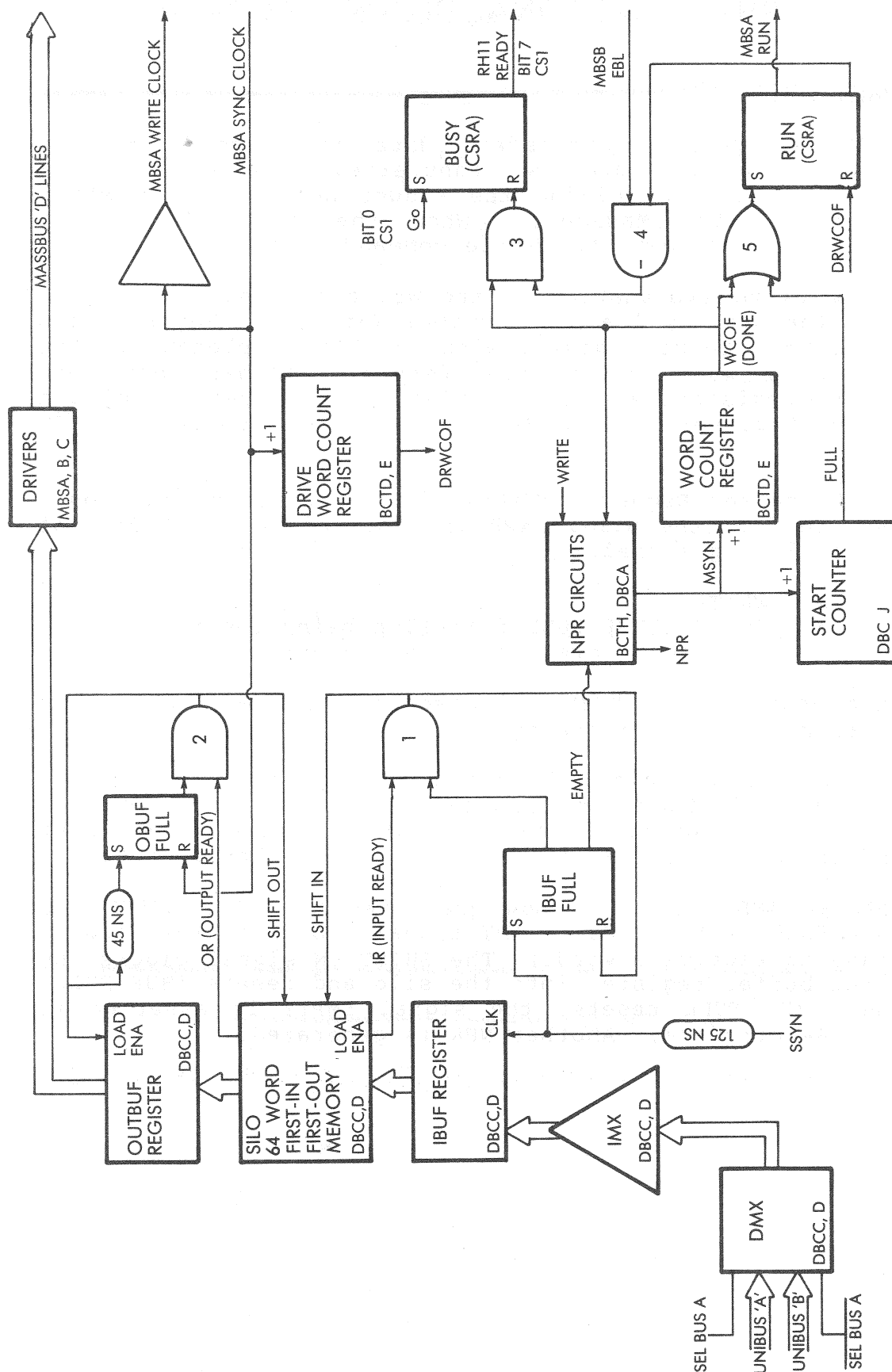
Non-Processor Requests (NPRs) are initiated by the logic made to fill the Silo. NPR is asserted by the ANDing of the following signals:

- WRITE
- EMPTY (IBUF FULL flip-flop being reset)
- Word count overflow is not asserted

When Master Sync (MSYN) is asserted, the word count and start counter are incremented.

The data from memory is gated through the Data Multiplexer and Input Multiplexer to the input buffer register. During Slave Sync (SSYN) time, the input buffer is clocked and the IBUF FULL flip-flop is set.

Labeled AND Gate 1 is now qualified by INPUT READY and IBUF FULL. (The INPUT READY signal indicates the Silo is ready to receive a word.) The SHIFT IN signal clocks the input buffer register into the silo and resets IBUF FULL. When IBUF FULL resets, the signal EMPTY is asserted to the NPR circuits. Another NPR is generated.



CZ-3018

Figure 1 Write Data Functional Block Diagram

When the first data word reaches the last cell of the Silo, the signal OUTPUT READY is asserted. When OBUF FULL is reset, it qualifies AND Gate 2. The signal SHIFT OUT clocks the data into the output buffer register, negates OUTPUT READY from the Silo, and sets OBUF FULL.

When the second data word reaches the last cell of the Silo, the signal OUTPUT READY is again asserted. SHIFT OUT will not be asserted because OBUF FULL is set.

The buffer will fill in this manner until the start counter FULL signal is asserted. The FULL signal is fed to OR Gate 5 and sets the RUN flip-flop. The other input to OR Gate 5 is word count overflow. Word count overflow sets the RUN flip-flop if the word count overflows before the start counter. This takes care of starting the operation for data transfers of less than 64 words. The signal RUN is responsible for starting the Write operation.

The drive asserts the signal SYNC CLOCK when it is ready to receive a word. SYNC CLOCK is routed back to the drive as WRITE CLOCK. WRITE CLOCK is returned to the drive to strobe the data into the buffer of the drive. SYNC CLOCK also resets the OBUF FULL flip-flop and increments the drive word counter.

The Write operation is stopped when word count and drive word count overflow. The word count register overflows first, which stops NPRs and enables AND gate 3.

When the Drive Word Count Register overflows, the RUN flip-flop is reset. The negation of the RUN signal informs the drive the last word has been sent. When the drive reaches the end of the sector (or record if a tape transport), it sends the signal End of Block (EBL). AND Gate 4 is then qualified and the output is fed to AND Gate 3. AND Gate 3 resets the BUSY flip-flop which terminates the Write operation.

READ OPERATION

The Read operation transfers data from the device to memory. The data from the device enters the buffer and sequences through to the output register. An NPR is made and the data is transferred to memory. The Silo buffers the data words from the faster Massbus to the slower Unibus. When word count overflows, the Read operation is terminated.

Below is an explanation of the Read command (See Figure 2). The program loads the Unibus Word Count Register and the Drive Count Register with the two's complement of the number of words to be transferred. (Other local and remote registers are also loaded.) Then the Control and Status Register 1 is loaded with the READ command and GO is set. GO sets the BUSY flip-flop. READ sets the RUN flip-flop. RUN being asserted, indicates that the RH is prepared to receive data from the drive.

The data from the drive passes through the receivers, the Input Multiplexer, and then through the buffer. After the drive places the data word on the Massbus D lines it generates SYNC CLOCK. The signal SYNC CLOCK strobes the data into the input buffer register, sets IBUF FULL and increments the drive word count.

The Silo works in the same manner as it did during the Write command. Each data word from the device sets IBUF FULL and then the word is strobed into the Silo. As the word is strobed into the Silo, IBUF FULL is reset. The data word sequences through the Silo into the output register. When the data word is loaded into the output register, the OBUF FULL flip-flop is set.

The signal FULL is gated to the NPR circuits where it is ANDed with READ to assert NPR. At MSYN time, the word count is incremented. At SSYN time, the OBUF FULL flip-flop is reset, letting another word sequence into the output register.

The Read operation is terminated with the overflow of drive word count and word count registers. The Drive Word Count Register will overflow first, which disables AND gate DBCB and resets the RUN flip-flop.

When RUN is negated, it signals the drive to terminate the Read operation. The Word Count Register keeps the RH active until the buffer is empty. When word count overflow and End of Block is received, the BUSY flip-flop is reset, terminating the Read operation.

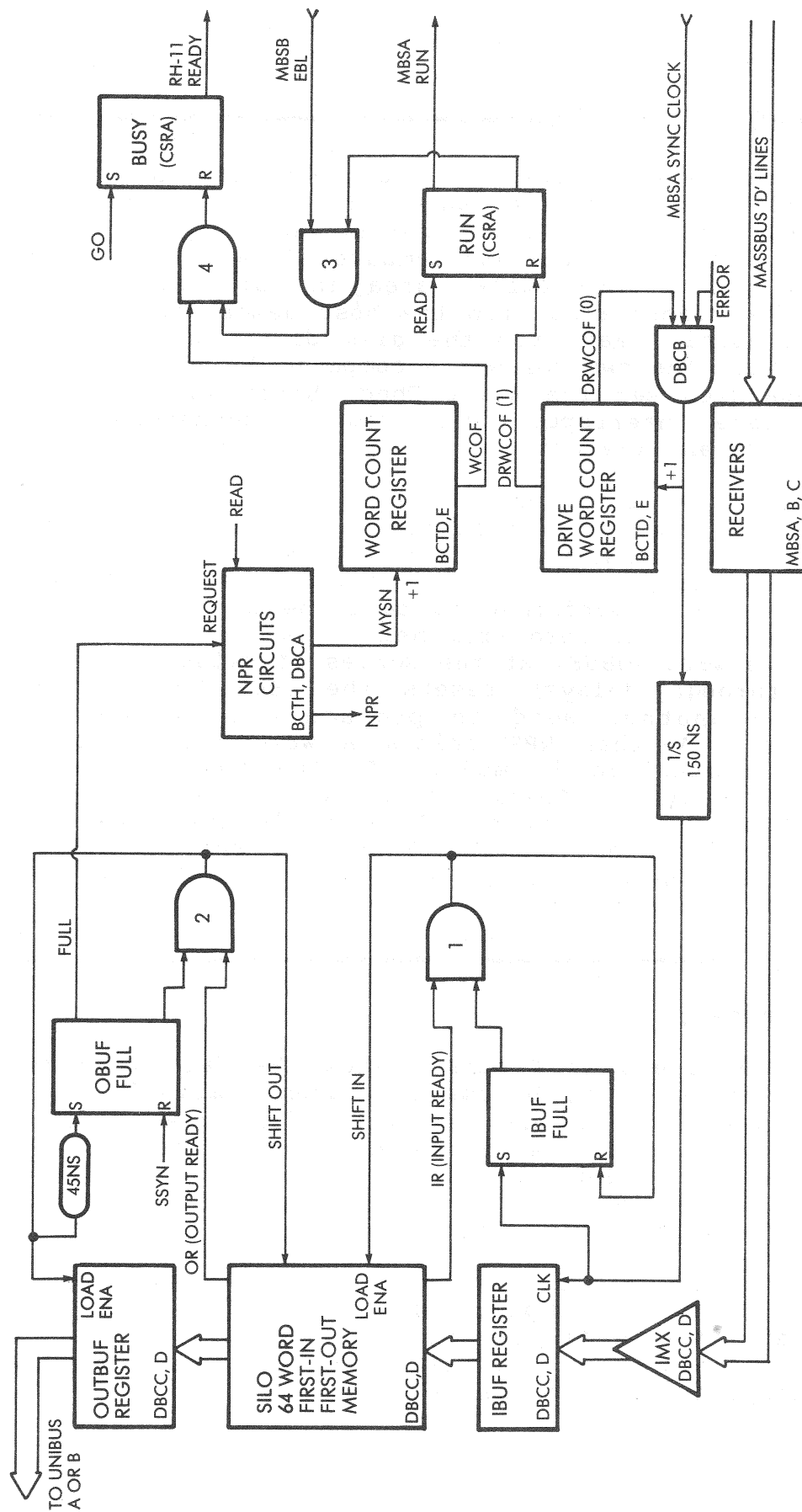


Figure 2 Read Data Functional Block Diagram

WRITE CHECK OPERATION ---

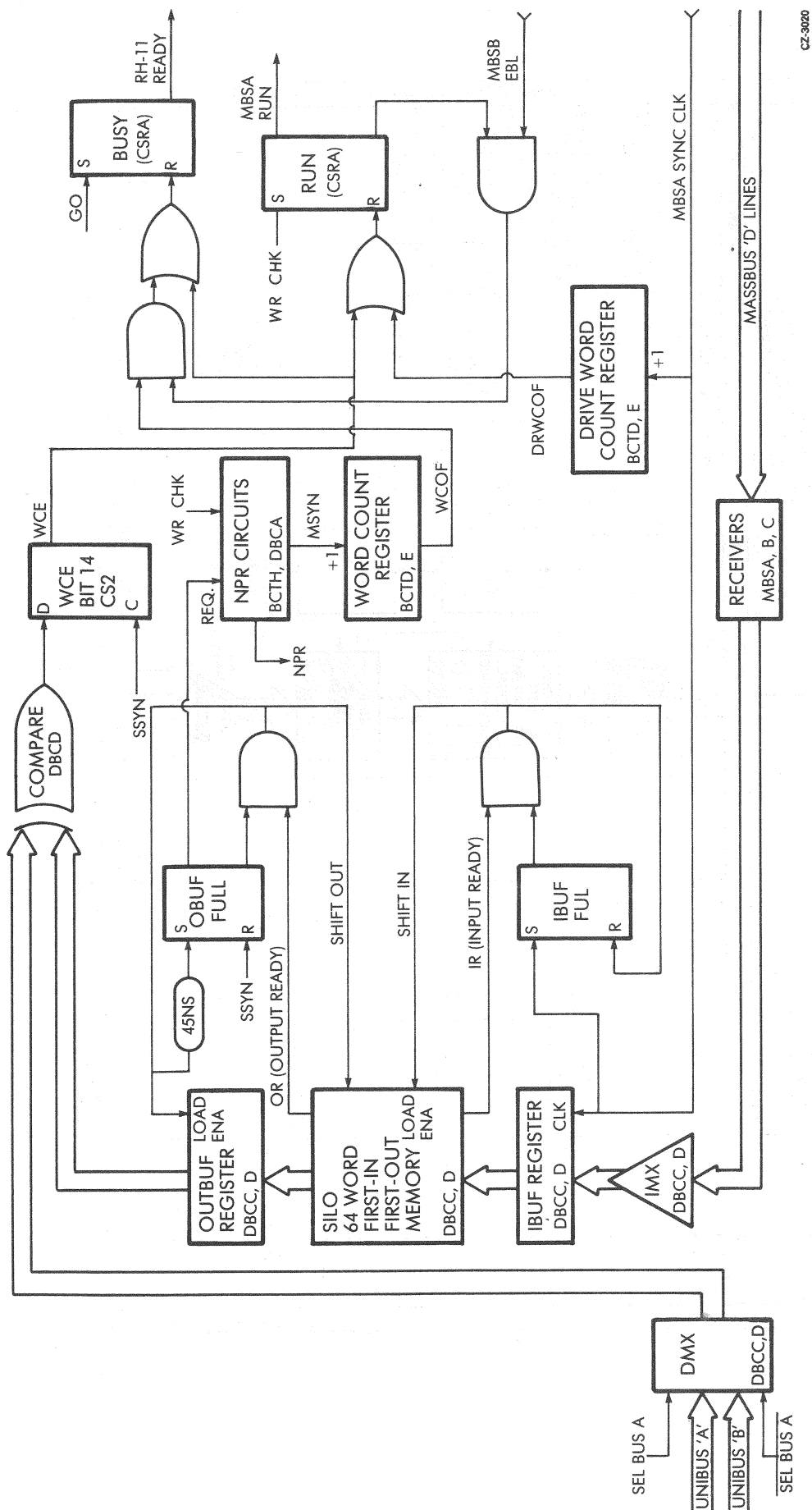
The Write Check operation is used to verify that data was correctly written on the media. It is used after a Write command. The Write Check command reads this same block of data from the drive and compares it with the contents of its source data buffer area in main memory. Data words are transferred from the host memory to the RH and simultaneously read from the disk drive and transferred to the RH. The two words are compared in the RH, and any discrepancies set the Write Check Error which causes an appropriate interrupt. Data remains unchanged in both the memory and the drive.

Below is an explanation of the Write Check command (See Figure 3). The Write Check operation reads data from the drive via the Massbus and stores the data in the Silo. When the data propagates through the Silo to the OBUF, a Unibus cycle is performed to read the corresponding word from memory. The word from memory is then compared with the drive word (OBUF) at the series of Exclusive-OR gates. SSYN (through delays) resets the OBUF FULL flip-flop allowing another word to propagate into the output register. Another NPR brings a word from memory and another comparison is made. If the two word are not equal, the Write Check Error (WCE) bit is set and generates Transfer Error. This holds the word in the output buffer, and the OBUF FULL flag remains asserted. Write Check Error clears the RUN and BUSY flip-flops, terminating the command.

SUMMARY ---

Write Command

The shaded areas in Figure 4 show the data path for a Write command using Unibus B. Upon initiation of the Write command, NPRs are issued to fill the Silo. The data from Unibus B is channeled through the Data Input Multiplexer and into the Silo. When the Start Counter reaches the proper count, RUN is asserted to the drive. The drive asserts SYNC CLOCK to transfer the data words to the drive. Word count overflows and stops the data transfers from memory. The drive word count overflows when the Silo is empty. Then RUN is negated, which terminates the Write operation in the drive. The drive then responds with End of Block to reset the BUSY flip-flop, terminating the Write command.



CZ-3020

Figure 3 Write Check Functional Block Diagram

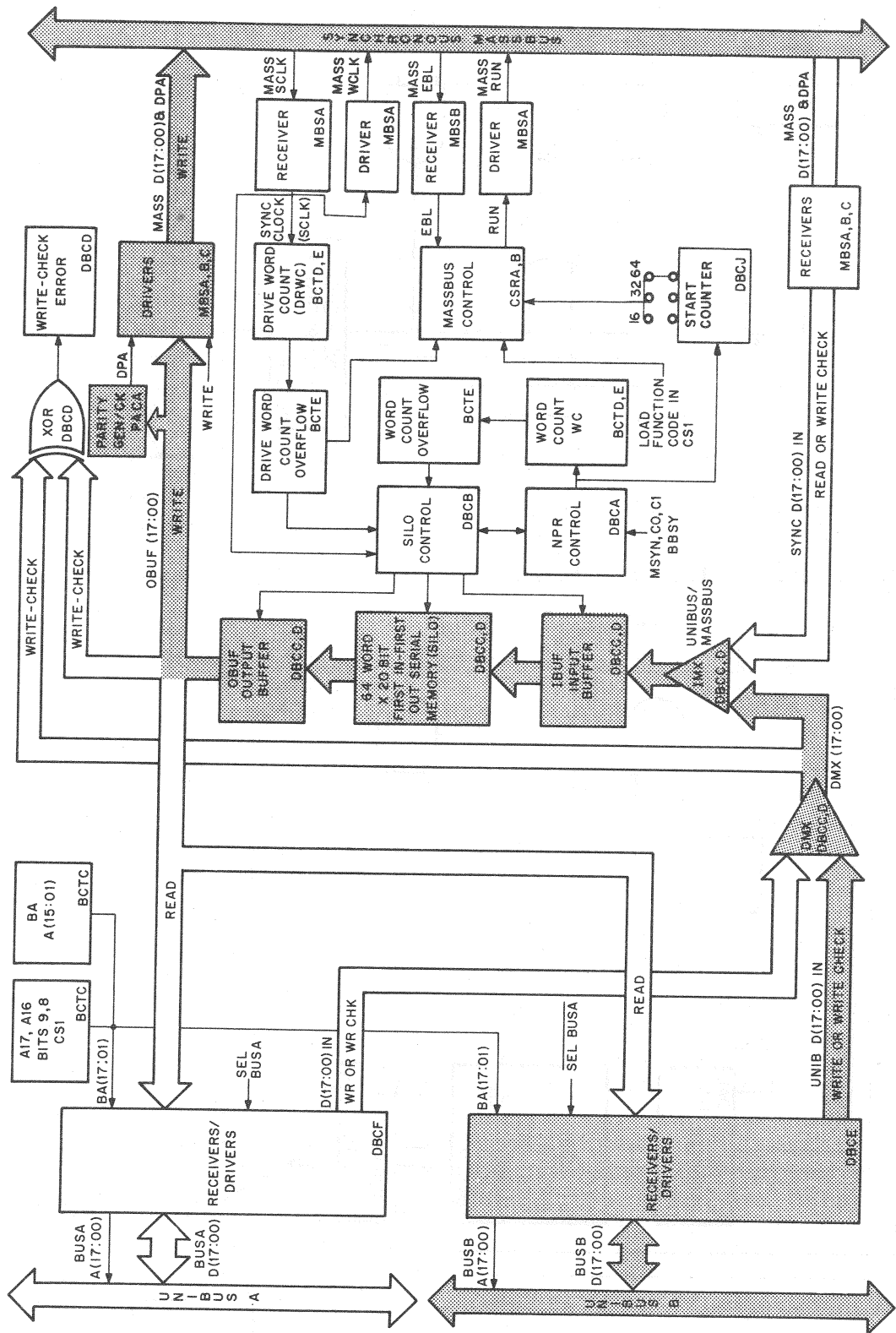


Figure 4 DMA Data Path for a Write Command

Read Command ---

The shaded areas in Figure 5 show the data path for a Read command using Unibus A. Upon initiation of the Read command, RUN is asserted to the drive. The drive then sends data over the Massbus into the Input Multiplexer to the buffer. When the data reaches the Output Buffer Register, an NPR is made. The data is then routed out through the Unibus drivers to the memory. When the drive word count overflows, RUN is negated, terminating the read data from the drive. When word count overflows, the Read command is terminated by resetting BUSY.

Write Check Command ---

The shaded area in Figure 6 shows the data path for a Write Check command using Unibus A. When the Write Check command is recognized, RUN is asserted to the drive. The drive then sends data over the Massbus through the Input Multiplexer to the Silo. When the data reaches the output buffer, a NPR is made. The word from memory enters the receivers and then is routed through the data multiplexer to the XOR gates. The two data words are then compared for equality. The command is terminated when word count overflows.

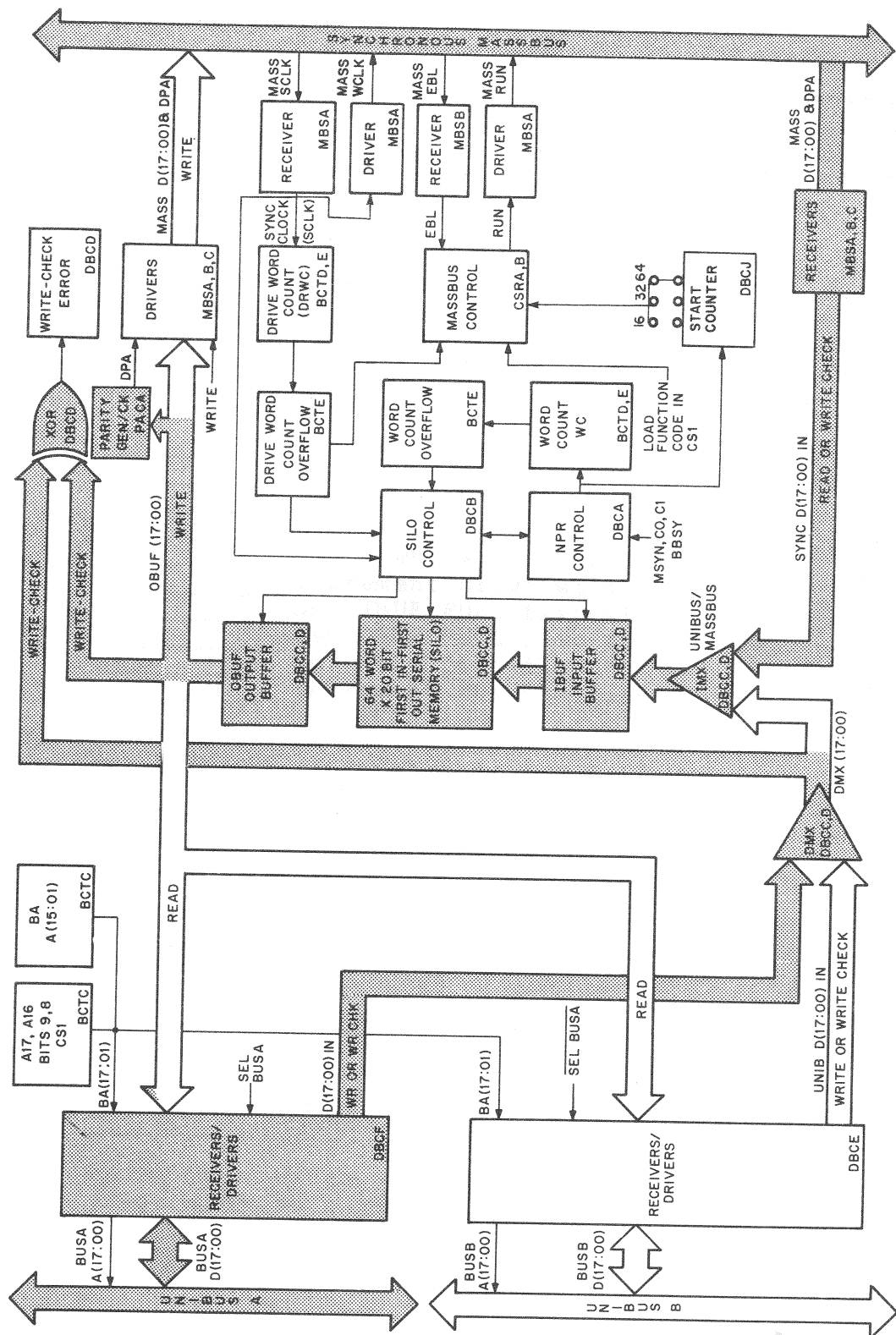
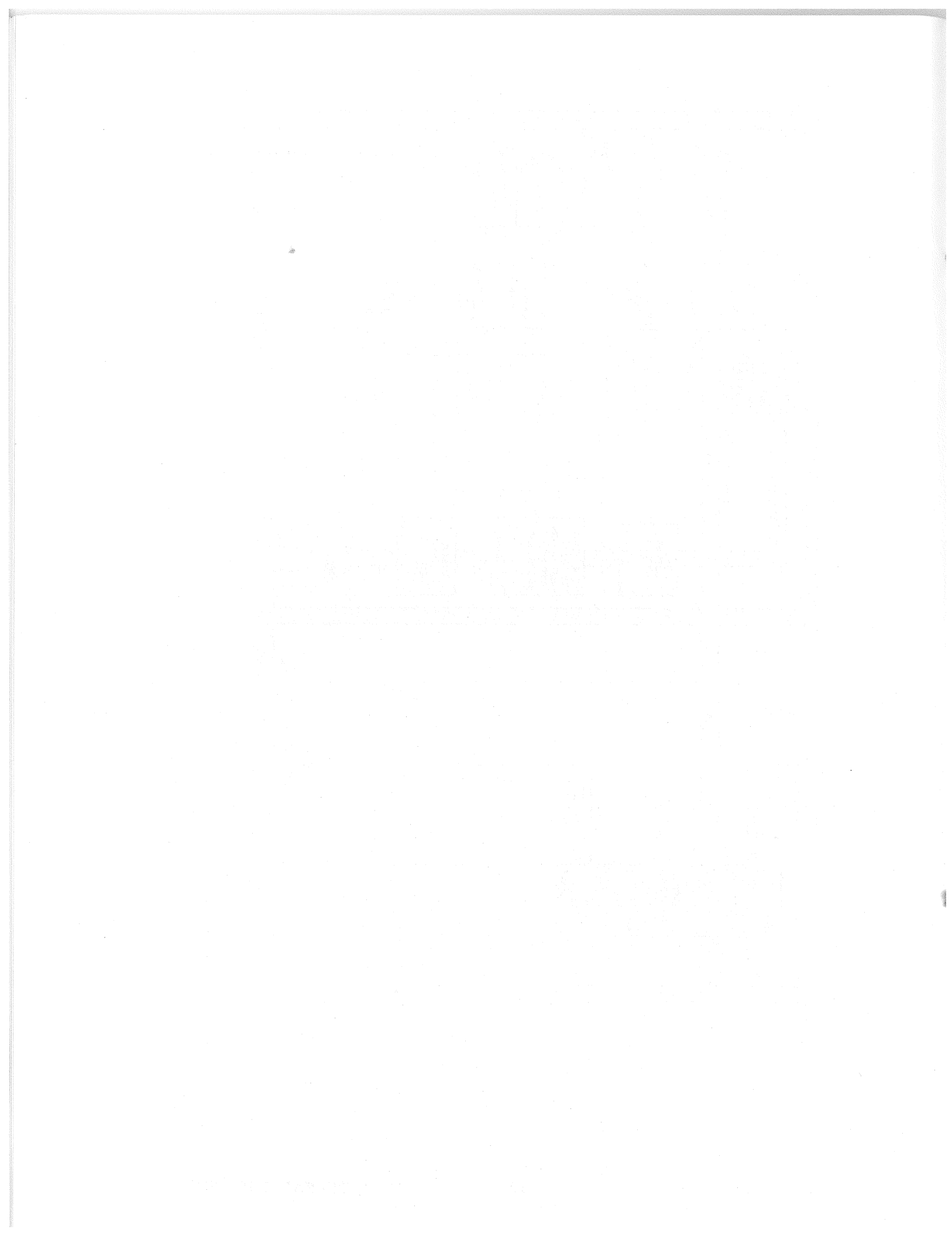


Figure 6 DMA Data Path for a Write Check Command



EXERCISES

Answer the following questions using any of your reference materials. Discuss any problems you have with another student or your course administrator. The solutions are on the next page.

1. Using the Write Data Functional Block Diagram (Figure 1) as a reference, number the following steps in the order in which they happen. Some steps happen many times before other steps; however, indicate just the first time they happen.

_____ Set RUN flip-flop
_____ Reset RUN flip-flop
_____ Assert NPR
_____ Reset BUSY flip-flop
_____ Increment word count
_____ Increment drive word count
_____ Set IBUF FULL flip-flop
_____ Set OBUF FULL flip-flop
_____ Set BUSY flip-flop
_____ Increment start counter

2. Using the Read Data Functional Block Diagram (Figure 2) as a reference, number the following steps in the order in which they happen. Some steps happen many times before other steps; however, indicate just the first time they happen.

_____ Set RUN flip-flop
_____ Reset RUN flip-flop
_____ Assert NPR
_____ Reset BUSY flip-flop
_____ Increment word count
_____ Set IBUF FULL flip-flop
_____ Set OBUF FULL flip-flop
_____ Set BUSY flip-flop
_____ Increment drive word count

3. Using the Write Check Functional Block Diagram (Figure 3) as a reference, briefly describe how the RH11 handles data for a Write Check function.

SOLUTIONS

1.	<u>7</u>	Set RUN flip-flop
	<u>9</u>	Reset RUN flip-flop
	<u>2</u>	Assert NPR
	<u>10</u>	Reset BUSY flip-flop
	<u>3 or 4</u>	Increment word count
	<u>8</u>	Increment drive word count
	<u>5</u>	Set IBUF FULL flip-flop
	<u>6</u>	Set OBUF FULL flip-flop
	<u>1</u>	Set BUSY flip-flop
	<u>3 or 4</u>	Increment start counter

2.	<u>2</u>	Set RUN flip-flop
	<u>8</u>	Reset RUN flip-flop
	<u>6</u>	Assert NPR
	<u>9</u>	Reset BUSY flip-flop
	<u>7</u>	Increment word count
	<u>4</u>	Set IBUF FULL flip-flop
	<u>5</u>	Set OBUF FULL flip-flop
	<u>1</u>	Set BUSY flip-flop
	<u>3</u>	Increment drive word count

3. Data is read from the drive into the Silo buffer. When the words reach the Output Buffer Register an NPR is generated. Data from memory is routed through the Data Multiplexer to the XOR gates. At the XOR gates, the data from memory is compared with the data from the drive for equality.

RH70 Functional Block Diagrams

RH70 FUNCTIONAL BLOCK DIAGRAMS

INTRODUCTION

This module describes data flow in the RH70. The description includes block diagrams of the following functions:

- Maintenance
- Read
- Write
- Write Check

Each component in the block diagram has a Field Maintenance Print Set page mnemonic associated with it, providing a quick reference to the print set. As you read through this module, refer to the block diagrams.

This lesson will help you gain an understanding of how the RH70 Controller operates. Knowing the functionality of the printed circuit modules will serve as a valuable troubleshooting aid when you encounter controller problems.

OBJECTIVES

Using the supplied block diagrams, you will be able to describe the four basic functions of the RH70:

1. Maintenance
2. Read
3. Write
4. Write Check

ADDITIONAL RESOURCES

RWP04 Maintenance Manual, EK-RWP04-MM

RWP05/06 Maintenance Manual, EK-RWP56-MM, Paragraph 5.3

RH70 FUNCTIONAL BLOCK DIAGRAMS

MAINTENANCE OPERATION

Due to the complexity of the data buffer design, a built-in maintenance feature has been incorporated. This feature allows the programmer to write into the Data Buffer (DB) as a register, thereby allowing a succession of data words to be sequenced through the data buffer and read back via a read register from the DB. In this way, the majority of the logic in the buffer and parity circuits is verified.

Figure 1 shows the data path when the programmer writes and reads the Data Buffer register. When writing to the data buffer, the data from the Unibus is routed through the receivers to the buffer. The data then sequences through the buffer to be available at the output. The data buffer stores up to eight words. Reading of the data buffer routes the words through the drivers to the Unibus.

Figure 2 shows the buffer configuration used during a maintenance operation. The RA, RB, RC, RD, RE and OBUF registers contain associated flags (RA FULL through OBUF FULL) to designate the status of their respective registers. The RA or RB FULL flags are presented to the programmer via the Input Ready (IR) status bit (bit 06) in the Control and Status Register 2. When the hardware selected flag (RA FULL or RB FULL) is asserted (register is full), the Input Ready Status is negated. This indicates that the register is not available to receive a word. The OBUF FULL flag is presented to the programmer via the Output Ready (OR) status bit (bit 07) of the Control and Status 2 register. When the OBUF FULL flag is asserted, it indicates that the contents of OBUF are full (has a valid data word) and that the programmer can read the data buffer.

The data path also contains three steering signals used to alternate the flow of data between the right and left-hand sides of the data path. These signals are RB ENA, RD ENA, and MXR SEL. When the system is first initialized by asserting INIT on the Unibus, or setting the CLR bit in the CS2 register:

- The RB ENA flip-flop is pointing to the RA register
- The RD ENA flip-flop is pointing to the RC register
- The MXR SEL logic is pointing to the RC register.

This primes the right-hand side of the data path to accept the first data word.

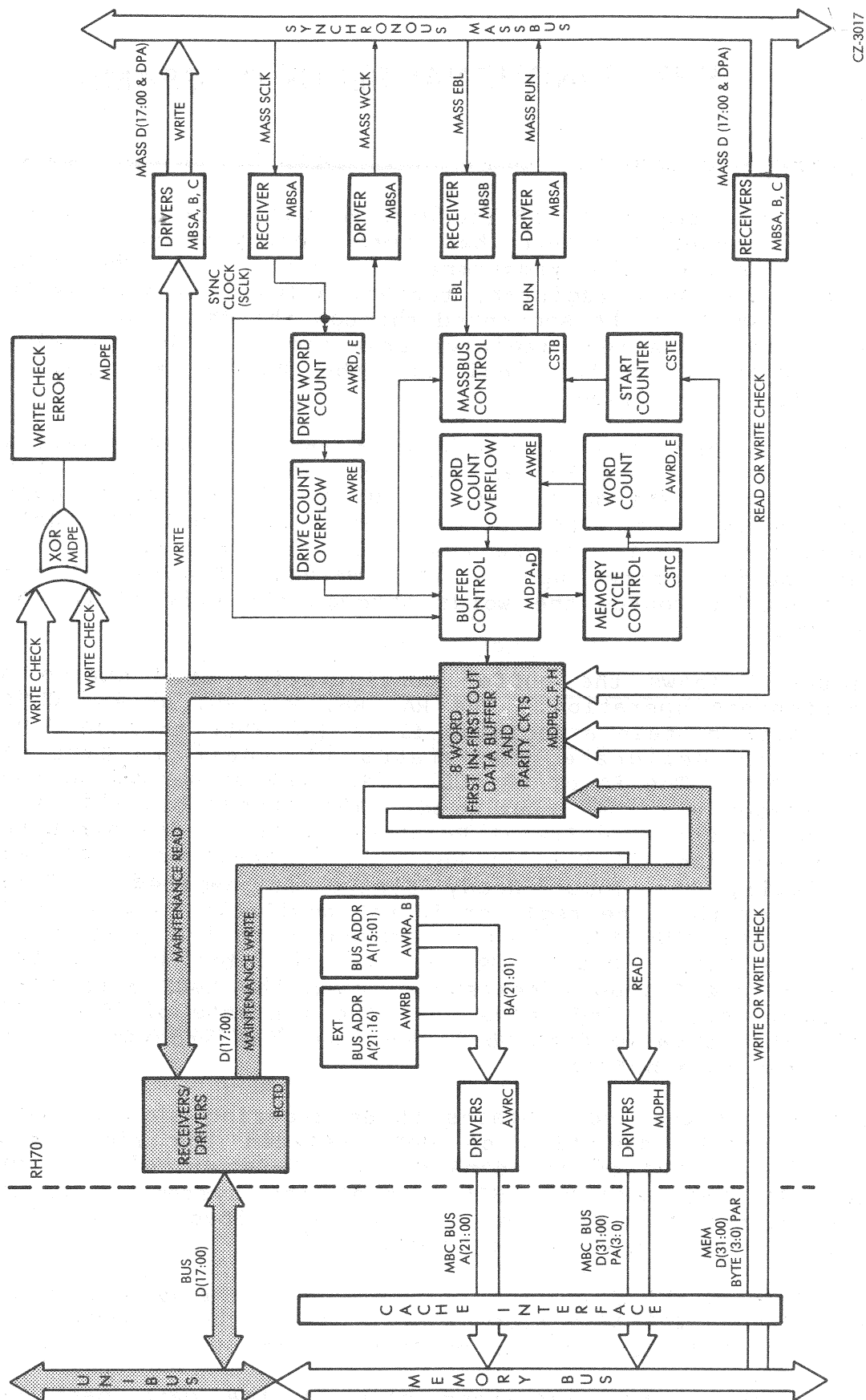
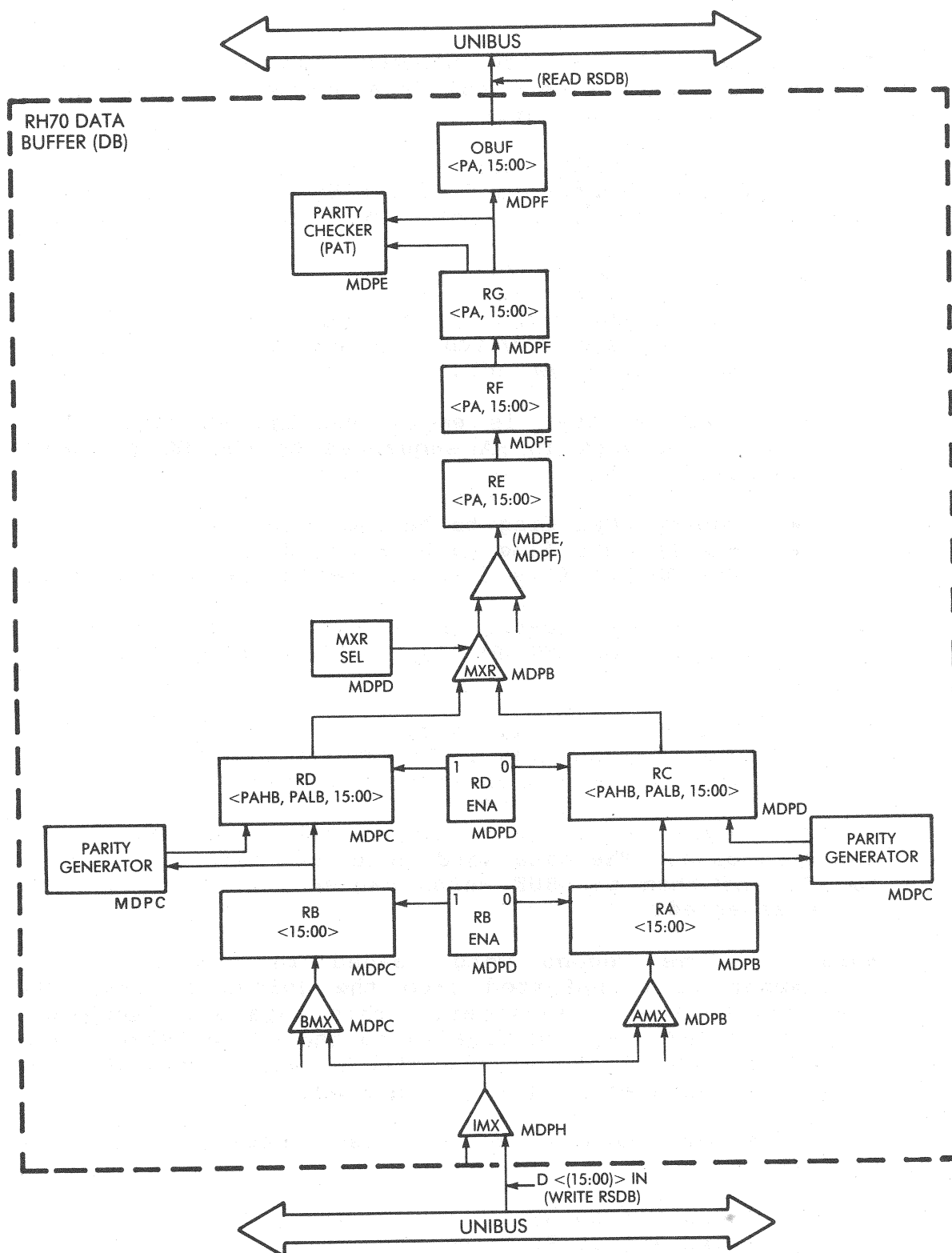


Figure 1 Maintenance Operation Data Path



CZ-3040

Figure 2 Data Buffer Maintenance Operation

The following description assumes eight words are to be sequentially loaded into the data buffer and none of the words are to be read out.

Word 1 - The first word loaded in the DB by the programmer is transferred from the Unibus to IMX, AMX and then to the RA register. This action causes two events:

- The RA FULL flag is asserted (which negates the IR Status in CS2)
- The RB ENA flip-flop is toggled to point to the RB register (which, causes the IR Status to again be asserted).

Since the RC register is empty and the RA register is full, the data word in RA sequences to the RC register. This causes:

- The RC FULL flag to be asserted
- The RA FULL flag to be negated
- The RD ENA flip-flop to toggle the RD register.

The data word then sequences to the RE register through the MXR since the MXR SEL was pointing to RC. This causes:

- The RC FULL flag to clear
- The RE FULL flag to be asserted
- The MXR SEL to point to the RD register.

Now the logic is set up to enable the left-hand side of the data path. The data word in RE is transferred to RF, to RG, and then to OBUF which causes the OBUF FULL flag to be asserted.

Word 2 - The second word loaded in the DB by the programmer is transferred from the Unibus to IMX, BMX, and then to the RB register. This data word sequences through RD, RE, RF, and finally to the RG register. Word 2 remains there since data word 1 is still in OBUF. The right-hand data path is again enabled.

The remaining words are sequenced into the following registers:

- Word 3 into RF
- Word 4 into RE
- Word 5 into RC
- Word 6 into RD
- Word 7 into RA
- Word 8 into RB

The RB ENA flip-flop is toggled to point to RA. Since a word is in RA, the IR status is negated.

When the programmer does a read from the DB register, the contents of OBUF are supplied to the Unibus as data. This allows the words to sequence through the buffer. As successive reads are performed, the words are sequenced out of the DB in the same order in which they are loaded into the DB.

Whenever a word is clocked into RC or RD, the parity check generator circuit shown, generates odd parity for each byte of the word. As the data word sequences to the RE, the parity bits are X-ORed, resulting in one parity bit for the word. This parity bit is transferred along with the data word until the data word reaches OBUF. There is a third parity generator/check prior to OBUF. This circuit checks the parity of the data word to ensure that it is odd. If it is not, the parity checker will cause the MDPE (Massbus Data Parity Error) condition (bit 8 of CS2) to be asserted. This indicates that a hardware failure occurred between inputting the data word into the data buffer and outputting the data word from the data buffer.

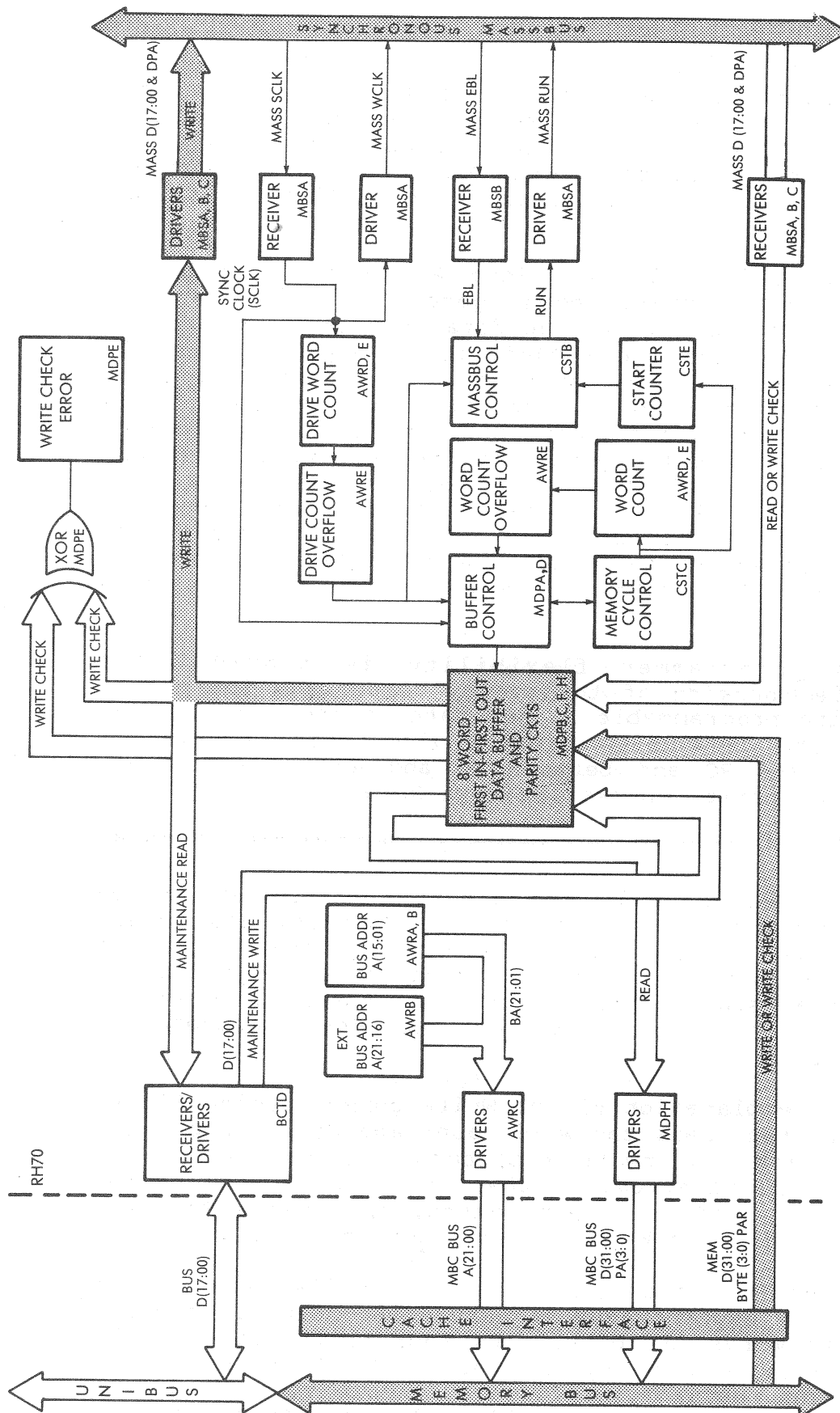
Additional programmer flexibility is provided for maintenance checking of the parity circuits. The parity circuits are programmable to generate or check bad (even) parity. The IPCK bits control the parity generators between RA and RC and between RB and RD. The PAT bit controls the parity checker between RG and OBUF.

WRITE COMMAND

In a write operation, data is transferred from memory to the RH70 Massbus controller, to the Massbus, and subsequently to the drive. Before data is transferred to the drive the buffer is filled. This gives the asynchronous memory a head start on the synchronous Massbus. When the drive count register overflows the Write command is terminated.

Below is an explanation of the Write command using Figure 3. The program loads the word count and drive word count registers with the two's complement of the number of words to be transferred. (Other local and remote registers are also loaded.) Then Control and Status Register 1 is loaded with the Write command and Go is set.

Memory requests are made to fill the buffer area. The transfers are asynchronous in nature and normally consist of two data words (32 bits, plus four parity bits). The Cache interfaces with up to four RH70 Controllers and contains the necessary priority arbitration logic to select the appropriate controller when multiple requests are initiated. The data is routed from memory through the Cache Interface to the data buffer.



CZ-3017

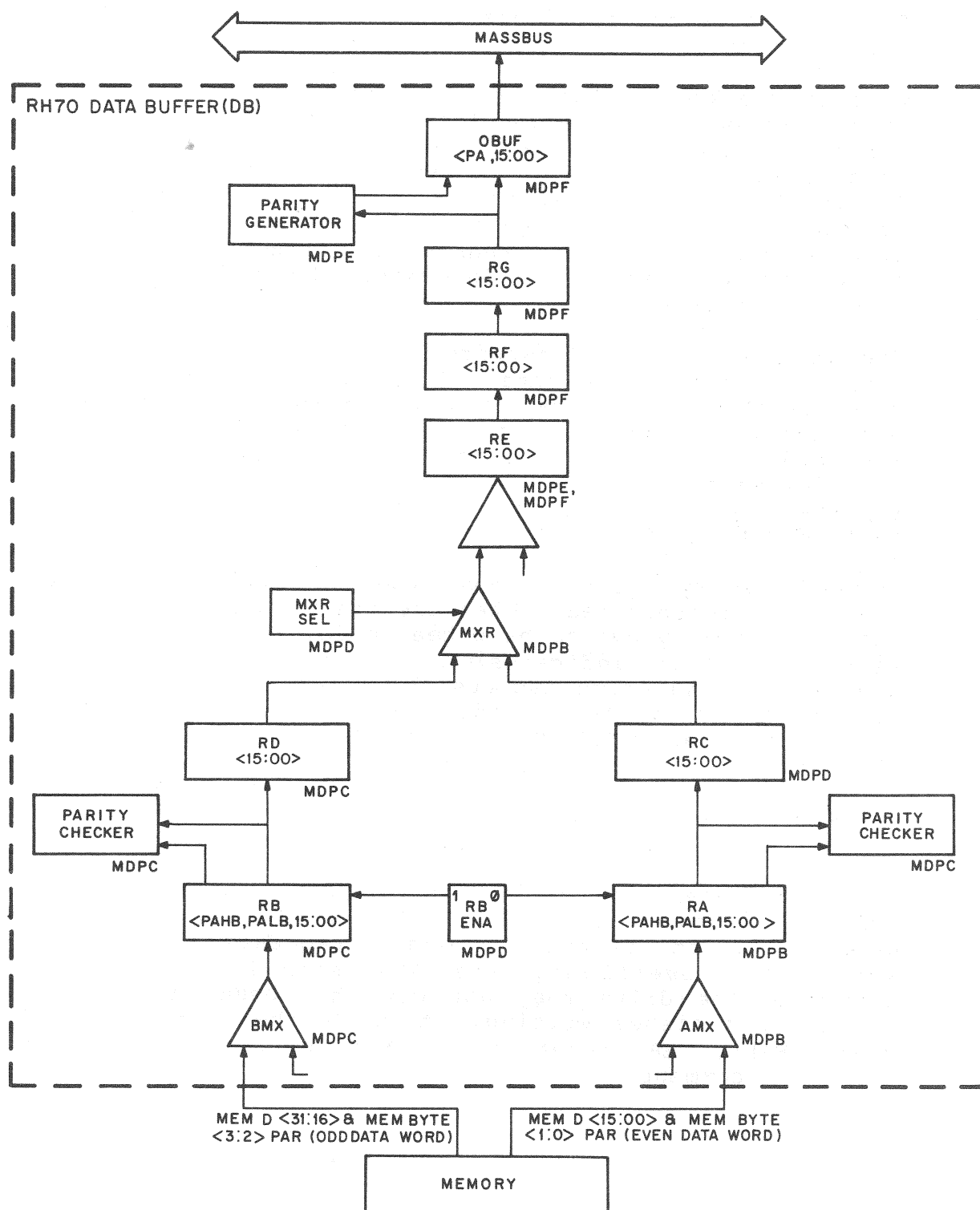
Figure 3 Write Command Data Path

Figure 4 shows the configuration of the data buffer for a Write Command. The double word from memory is clocked into RA and RB. Since RC and RD are empty, the double word is clocked into RC and RD. Each data word consists of two bytes plus a parity bit for each byte. The byte parity is checked by a parity checker as the double word is transferred from RA to RC and from RB to RD. If the parity is correct, a new memory cycle will be initiated. If the parity is incorrect, the data parity error is flagged and the memory transfer is frozen in order to allow the programmer to determine the address of the incorrect data word.

The word in RC is transferred to RE. The word in RE sequences through the data buffer and is followed by the word in RD which is now transferred to RE. The double word has thus been converted into two single words - the low word followed by the high word. As the data word is transferred from RG to OBUF, a parity bit is generated if required (for odd parity). This ensures correct parity for the data as it is clocked onto the Massbus.

As each memory cycle is made the word count and start counter is incremented. When the start counter indicates that four memory cycles have been made, the RUN signal is asserted. This indicates to the drive that data is available. The drive asserts the signal SYNC CLOCK when it is ready to write a word. SYNC CLOCK is routed back to the drive as WRITE CLOCK. WRITE CLOCK is returned to the drive to strobe the data into the drive buffer. SYNC CLOCK also increments the drive word counter and sequences the next word into the OBUF. This sequence repeats itself until drive word count overflows.

The write operation is terminated when word count and drive word count overflow. The word count register overflows first, terminating memory requests. When drive word count overflows, the RUN signal is negated, informing the drive the last word has been sent. When the drive finishes writing, it sends the signal End of Block (EBL). EBL resets the BUSY flip-flop, terminating the Write command.



II-2911

Figure 4 Data Buffer Write Command

READ COMMAND

The read operation transfers data from the device to memory. The data from the drive enters the buffer and sequences through to the output. When two words are assembled at the output of the buffer, a memory request is made. When word count overflows, the read operation is terminated.

Below is an explanation of the Read command using Figure 5. The program loads the word count and drive word count registers with the two's complement of the number of words to be transferred. (Other remote and local registers are also loaded.) Then Control and Status Register 1 is loaded with the Read command and Go is set. Go sets the BUSY flip-flop. READ sets the RUN flip-flop, asserting the signal RUN to the drive. RUN asserted indicates that the RH70 is prepared to receive data from the drive.

The data from the drive is placed on the D lines and then SYNC CLOCK is asserted which:

- Strokes the data into the RE register
- Increments the drive word count

Figure 6 shows the configuration of the data buffer for a Read command. The data is clocked into the RE register by SYNC CLOCK and then sequenced through to the OBUF register. Between the RG and OBUF, the data is checked for odd parity. The data word is then transferred to the RA register. This word stays in RA until the second word is clocked into RB. When these two registers are full, the two words are clocked into RC and RD. Parity is generated for each byte of the words. A memory request is then made. At the completion of the memory cycle, the word count is incremented the proper amount. As the words are transferred to memory, RA and RB are sequenced to the output.

The read operation is terminated with the overflow of drive word count and word count. The drive word count register will overflow first negating the RUN flip-flop. RUN signals the drive to terminate the read operation. Word count has not yet overflowed keeping the RH active until the buffer is empty. When word count overflows, the memory transfers are terminated. When the RH receives EBL, the BUSY flip-flop is reset, terminating the read operation.

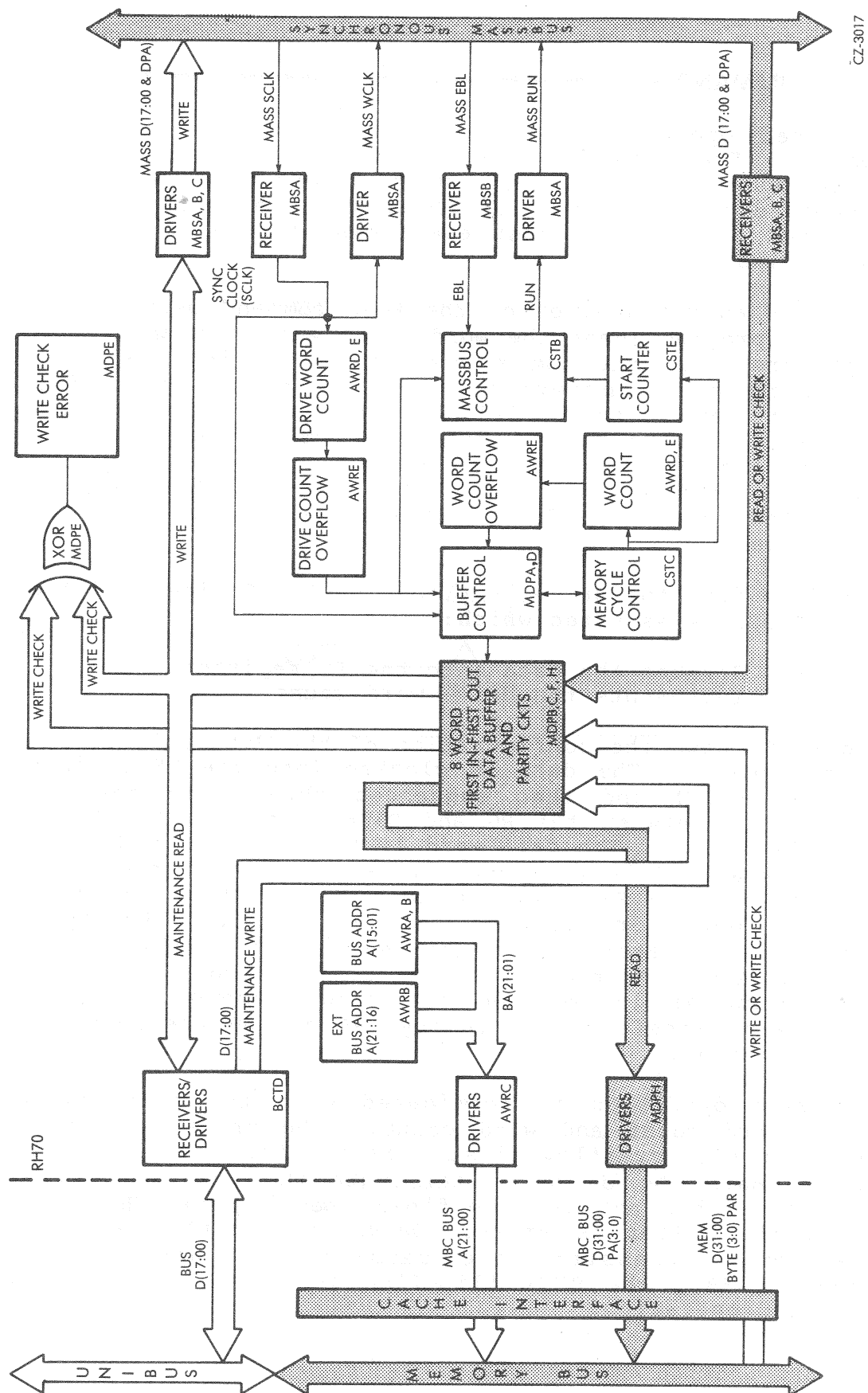
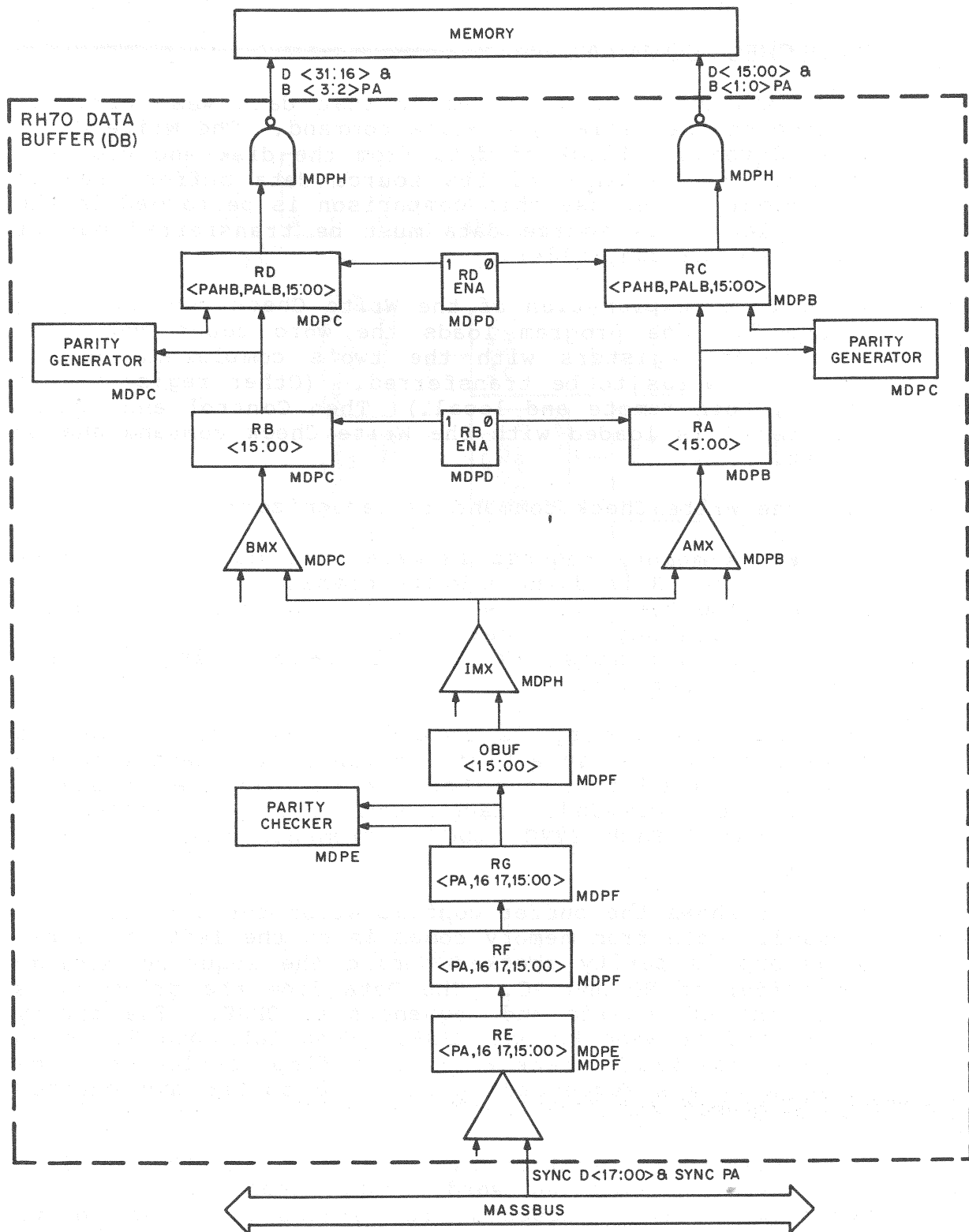


Figure 5 Read Command Data Path



TR-0907

Figure 6 Data Buffer Read Command

WRITE CHECK COMMAND

This command is used to verify that data was correctly written on the media by a Write command. The Write Check command reads a block of data from the disk and compares it with the contents of its source data buffer area in main memory. Because this comparison is performed in the controller, this source data must be transferred out of memory to the controller.

Below is an explanation of the Write Check command using Figure 7. The program loads the word count and drive word count registers with the two's complement of the number of words to be transferred. (Other registers are loaded, both remote and local.) Then Control and Status Register 1 is loaded with the Write Check command and Go is set.

When the Write Check command is recognized:

- A memory request is made (the memory acts like the RH is doing a Write command)
- The RUN signal is asserted, indicating the drive is ready to receive words from the drive (the drive acts as if it were doing a read operation).

The data from memory is routed to the buffer via the Cache Interface as in a Write command. The data from the drive is routed over the Massbus through the receivers as in a Read command. Each memory cycle increments the word count. Each SYNC CLOCK increments the drive word count.

Figure 8 shows the buffer configuration for a Write Check command. Data from memory comes in on the left in double words and is parity checked during the sequence through the buffer to RD and RC. The Data from the drive comes in on the right side and sequences to OBUF. The parity is checked between RG and OBUF. When OBUF and RC or RD is full, the Write Check Error flip-flop is clocked. The data words are sequenced through the buffer and checked one at a time.

The write check operation is terminated with the overflow of drive word count and word count. When the word count overflows, memory requests are terminated. When drive word count overflows, the RUN signal is negated. When the buffer is empty and EBL is received from the drive, the BUSY flip-flop is reset, terminating the Write Check command.

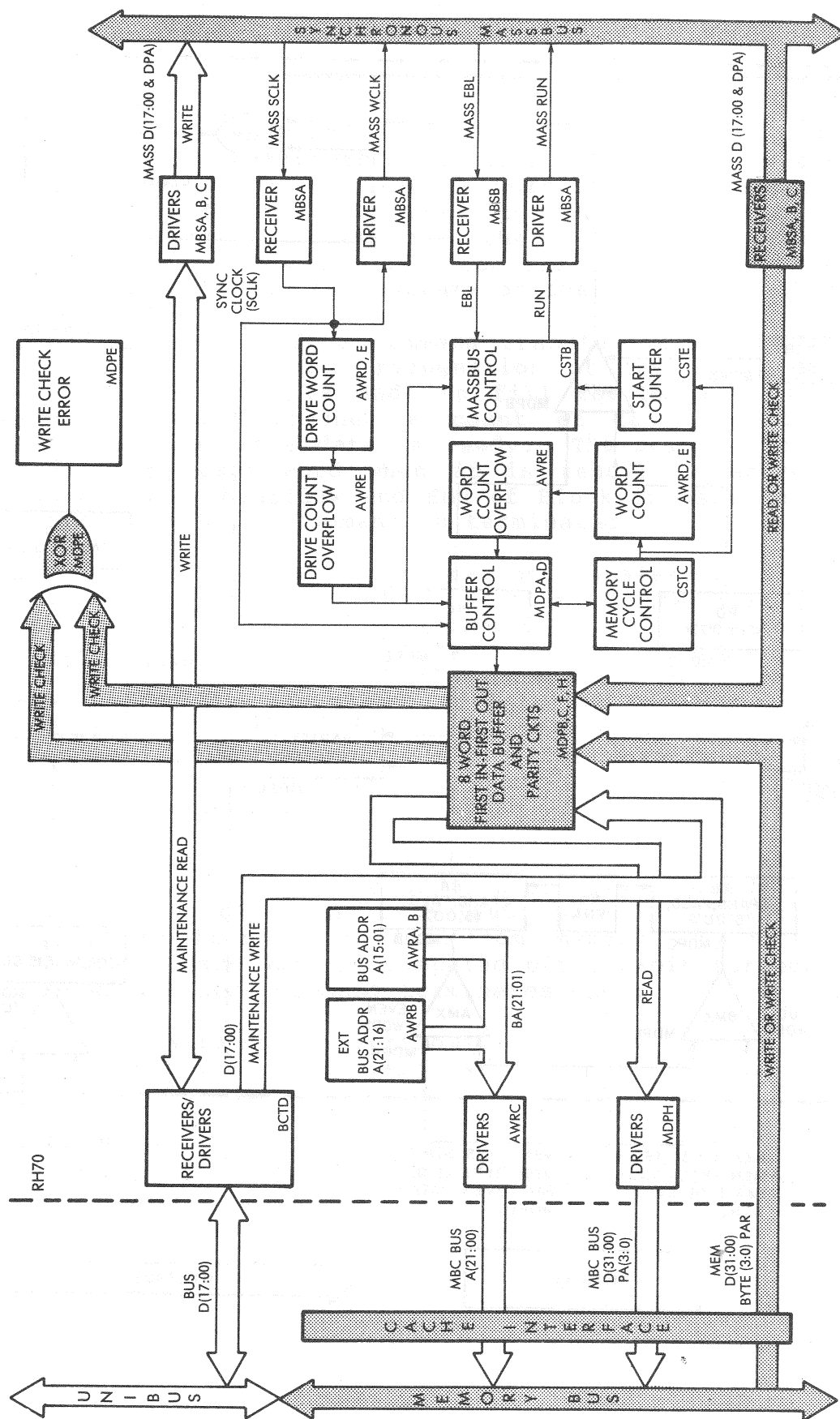
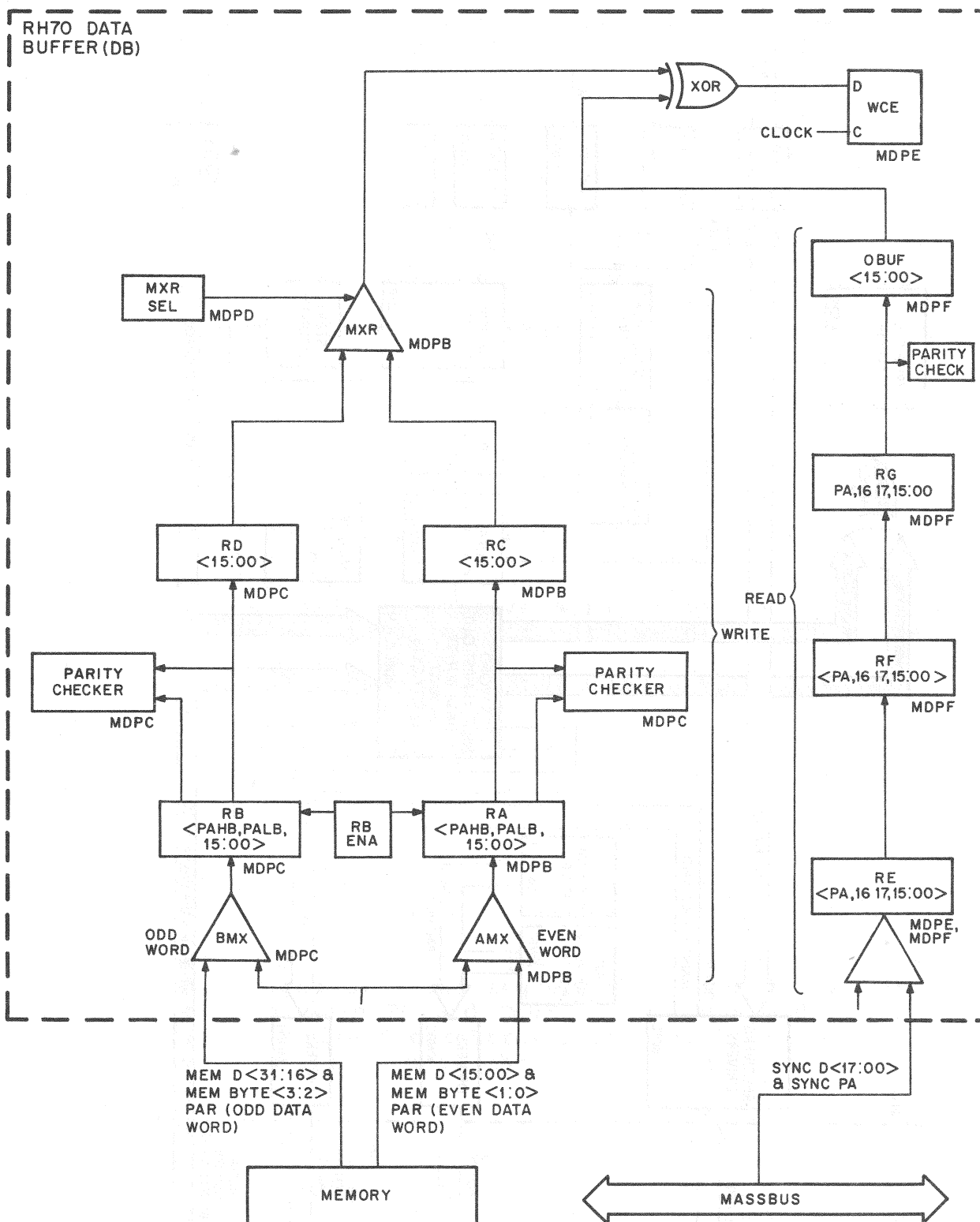


Figure 7 Write Check Command Data Path



11-2914

Figure 8 Data Buffer Write Check Command

SUMMARY

Maintenance Operation - This operation is used by diagnostics to test the data buffer. Writing to the data buffer register (DB) loads a word into the buffer. The word then sequences through the buffer to the output. The programmer can then read the DB to retrieve the word. The buffer parity circuits can also be checked using the IPCK and PAT bits to generate or check for even parity.

Write Command - This command transfers data from memory to the drive. Upon recognition of the Write command, memory requests are made to fill the buffer. When the start counter reaches a count of four the drive is informed that the data is ready. The drive asserts SYNC CLOCK for each word when it is ready. When word and drive count overflow and End of Block is asserted by the drive, the Write command is terminated.

Read Command - The Read operation transfers data from the drive to memory. Upon recognition of the Read command, the drive is informed the controller is ready to receive data. As each word is received, the drive word counter is incremented. The single data words sequence through the buffer and are assembled into two words for transfer to memory. As each memory transfer takes place the word counter is incremented. When drive word count and word count overflow and End of Block is received, the Read command is terminated.

Write Check Command - This command is used to verify that data was written on the media correctly. The drive fills half of the buffer as in a Read command. The other half of the buffer is filled from memory as in a Write command. Each word is compared bit-for-bit for equality. The Write Check command terminates when:

- Word count overflows
- Drive count overflows
- The buffer is empty
- End of Block is received from the drive.

The first step in the process is to identify the data to be processed. This is done by the user, who provides a list of data items and their locations. The system then searches for these items and returns a list of their locations. This list is then used to generate a set of instructions for the processor.

The second step is to process the data. This is done by the processor, which takes the instructions from the first step and processes the data accordingly. The processor then returns a list of results, which is then used to generate a set of instructions for the output device.

The third step is to output the results. This is done by the output device, which takes the instructions from the second step and outputs the results accordingly. The output device then returns a list of results, which is then used to generate a set of instructions for the processor.

The fourth step is to store the results. This is done by the storage device, which takes the instructions from the third step and stores the results accordingly. The storage device then returns a list of results, which is then used to generate a set of instructions for the processor.

The fifth step is to delete the results. This is done by the processor, which takes the instructions from the fourth step and deletes the results accordingly. The processor then returns a list of results, which is then used to generate a set of instructions for the output device.

EXERCISES

Answer the following questions using any of your reference materials. Discuss any problems you have with another student or your course administrator. The solutions are on the next page.

1. During a Maintenance operation, what signal informs the programmer that the data has sequenced through the buffer to the output?

2. Using the Write Data Functional Block Diagram as a reference, number the following steps in the order in which they first happen.

- _____ Set the RUN flip-flop
- _____ Reset the RUN flip-flop
- _____ Request a memory transfer
- _____ Increment word count
- _____ Increment drive word count
- _____ Increment the start count
- _____ Reset BUSY flip-flop

3. Using the Read Data Functional Block Diagram as a reference, number the following steps in the order in which they first happen.

- _____ Set the RUN flip-flop
- _____ Reset the RUN flip-flop
- _____ Request a memory transfer
- _____ Reset BUSY flip-flop
- _____ Increment word count
- _____ Increment drive word count

4. Using the Write Check Functional Block Diagram as a reference, briefly describe how the RH70 handles data for a Write Check Function.

SOLUTIONS

1. Output Ready (OR) Bit 07 of CS2
2. 4
6
1
2-3
5
2-3
7
3. 1
5
3
6
4
2
4. Data is read from the drive to fill half of the buffer. Data is read from memory to fill the other half. The data words are then compared for equality in the XOR gates.

Data Bus Interfacing

DATA BUS INTERFACING

INTRODUCTION

This module defines the Massbus signals used in the transfer of data. This module, coupled with the Control Bus Interfacing module, makes the description of communication between controller and peripheral complete.

Once you know these signals, you will find it easier to understand Massbus peripherals and subsystems.

OBJECTIVES

Given a list of Massbus interfacing signals and definitions, be able to match the signal with the definition.

ADDITIONAL RESOURCES

RJS04 Maintenance Manual, Page 2-1, Paragraph 2.2

TJU16 Maintenance Manual, Page 2-1, Paragraph 2.2

RJP04 Maintenance Manual, Page 2-1, Paragraph 2.2

RJP05/06 Maintenance Manual, Page 2-1, Paragraph 2.2

RWP04 Maintenance Manual, Page 2-1, Paragraph 2.2

DATA BUS INTERFACING

DATA BUS LINES

The Data Bus section of the Massbus consists of a 19-bit parallel data path and 6 control lines. The Data Bus is described in the following paragraphs and illustrated in Figure 1.

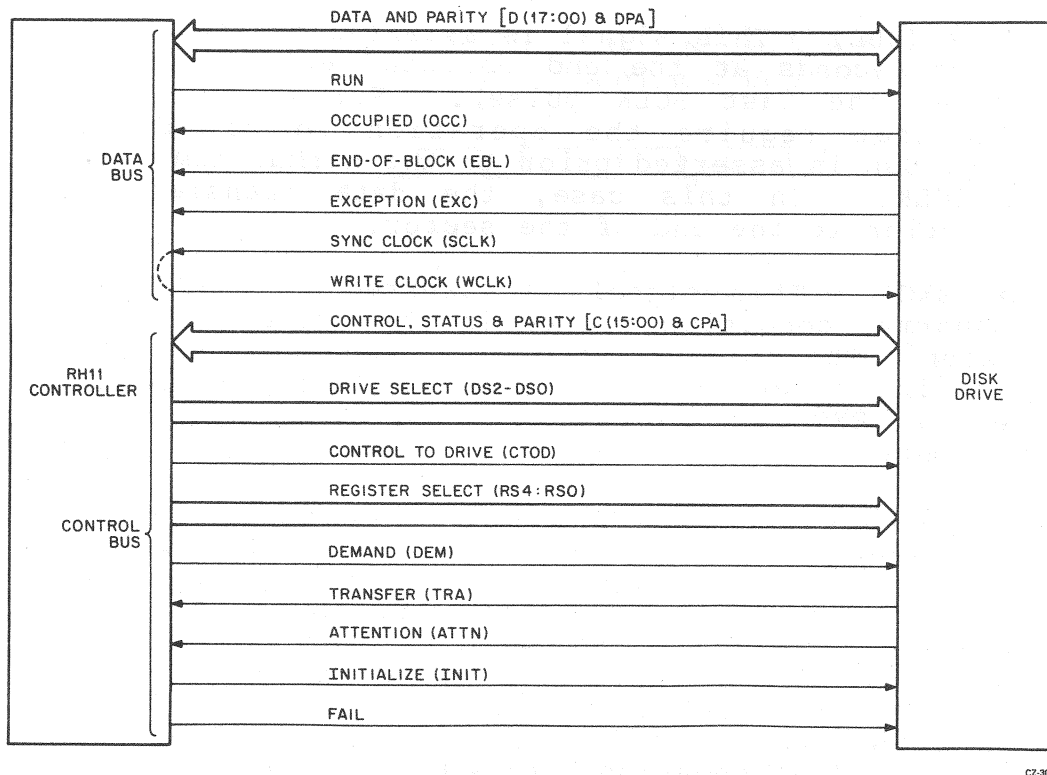


Figure 1 Massbus Interface Lines

Parallel Data Path - The 18 data bits of the Parallel Data Path are designated D00 through D17. The associated parity bit is referred to by the mnemonic DPA. The data path is bidirectional and employs odd parity. Data is transmitted synchronously, using a clock generated in the drive.

RUN - After a data transfer command has been written into the control register of a drive, the drive connects to the data bus. The controller asserts the RUN line to initiate the function. At the end of each sector or tape record the trailing edge of the End of Block pulse strobes the RUN line. If this signal is still asserted, the function continues for the next sector. If it is negated, the function is terminated.

Occupied (OCC) - This signal is generated by the drive to indicate that the data bus is busy. As soon as a valid data transfer command is written into a drive, the drive asserts OCC. Various errors may prevent a drive from executing a command. In this case, the controller will time out due to nonassertion of either OCC or Sync Clock (SCLK), and the Missed Transfer (MXF) error will be set in the controller. OCC is negated at the trailing edge of the last End of Block pulse of a transfer.

End of Block (EBL) - This signal is asserted by the drive for two microseconds at the end of each sector or tape record (after the last SCLK pulse). For those error conditions that require the operation to terminate immediately, EBL is asserted prior to the normal time for the last SCLK. In this case, the data transfer is terminated prior to the end of the sector.

Exception (EXC) - This signal is asserted by the drive when an abnormal condition occurs in the drive during a data transfer. The drive asserts this signal to indicate an error during a data transfer command (Read, Write or Write Check). EXC is asserted at or prior to assertion of EBL and negated at the negation of EBL.

Sync Clock (SCLK), Write Clock (WCLK) - These signals are the timing signals used to control the strobing of the data into the controller and/or into the drive. During a Read operation, the RH Controller strobes the data lines on the negation of SCLK and the drive changes the data on the assertion of SCLK. During a Write operation, the controller receives SCLK and echoes it back to the drive as WCLK. On the assertion of WCLK, the drive strobes the data line; on the negation of WCLK, the controller changes the data on the data lines.

MASSBUS WRITE DATA TIMING ---

Figure 2 illustrates the timing between the RH and peripheral device during a Write Data transfer.

1. The drive asserts OCC when it recognizes the Write command.
2. When the RH decodes the Write command:
 - The data lines are enabled to the Massbus.
 - Memory requests are made to start filling the buffer.

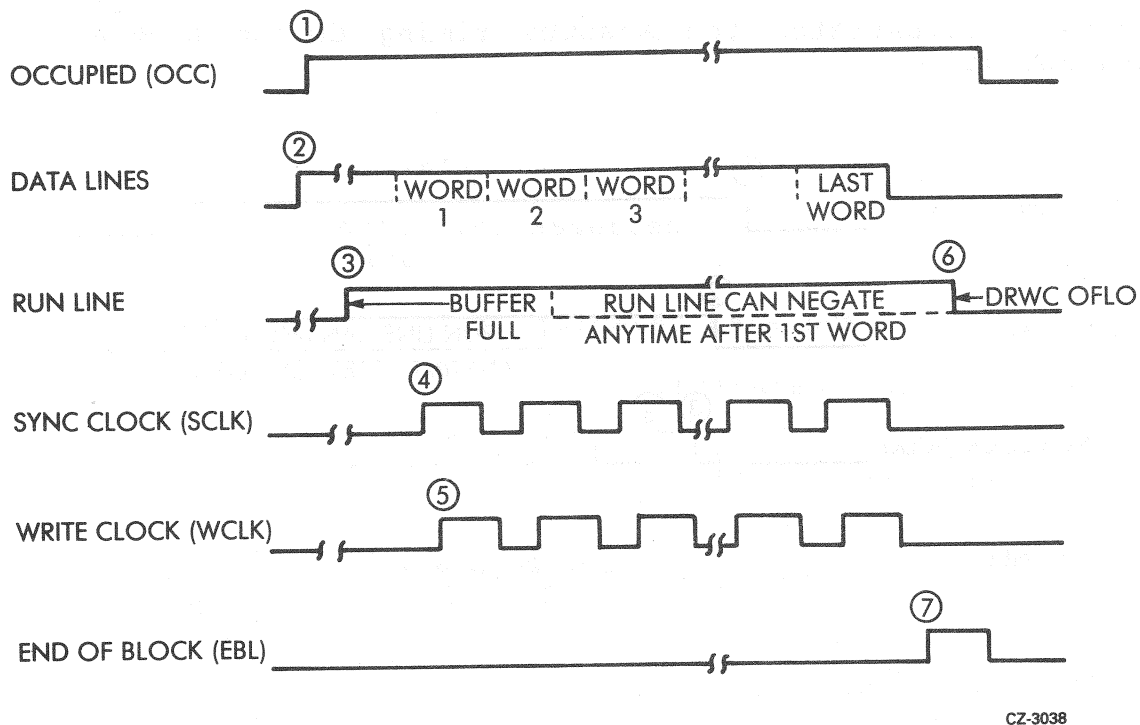
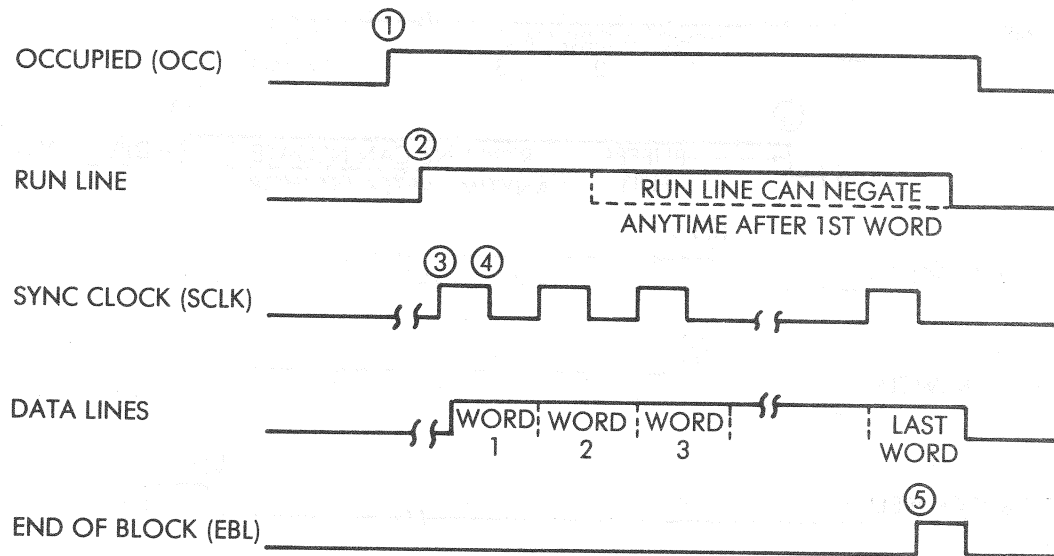


Figure 2 Massbus Write Data Transfer Timing Diagram

3. RUN is asserted as soon as the buffer is full, indicating that the RH is ready to send data. When RUN is asserted to a disk drive, the drive begins to look for the proper sector. When RUN is asserted to a tape drive, the tape begins to move.
4. The drive asserts SCLK when it is ready to receive the data.
5. WCLK is generated from SCLK and sent back to the drive to strobe the data lines into the buffer in the device.
6. When Drive Word Count overflows, the RUN signal is negated. This stops the word transfer to the drive. The drive will finish writing the CRC or ECC words.
7. The drive generates EBL when it finishes that block of data.

MASSBUS READ DATA TIMING

Figure 3 illustrates the Massbus timing during a Read Data operation.



CZ-3039

Figure 3 Massbus Read Data Transfer Timing Diagram

1. The drive asserts OCC after decoding the Read command.
2. The RH asserts RUN when it decodes the Read Data operation.
3. The drive asserts SCLK when a word is available to be transferred to the RH.
4. The trailing edge of SCLK strobes the data into the buffer of the RH.
5. The drive sends EBL to the RH after all the data has been transferred for that block of data and the error words have been checked.

SUMMARY

The following is a list of the signals on the data bus.

- D Lines - 18 bidirectional data lines
- DPA - data parity line
- RUN - controller-asserted to initiate function in the drive
- Occupied (OCC) - indicates data bus is busy
- End of Block (EBL) - indicates End of Block (sector)
- Exception (EXC) - indicates an abnormal condition occurred in the drive during a data transfer
- Sync Clock (SCLK) - strobes data into the RH during a read
- Write Clock (WCLK) - strobes data into drive during a write (an echo of SCLK)

EXERCISES

Answer the following questions using any of your reference materials. Discuss any problems you have with another student or your course administrator. The solutions are on the next page.

1. The following signals are on the data portion of the Massbus. Place a C before those that are generated in the RH controller. Place a D before those that are generated in the drive.

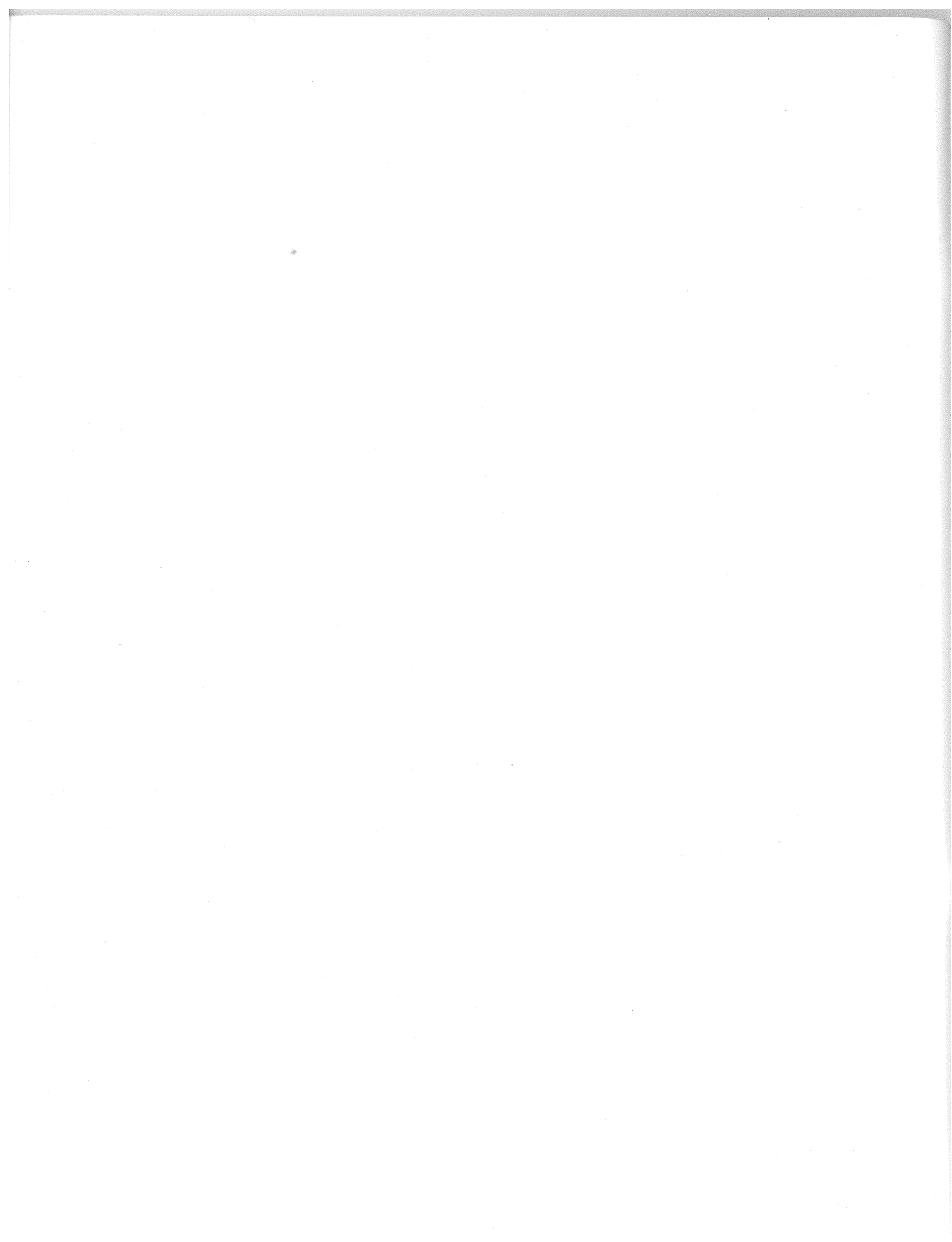
- _____ DPA during a Read command
- _____ RUN
- _____ Occupied
- _____ End of Block
- _____ Exception
- _____ Sync Clock

2. Which signal is asserted if an error occurred in the drive during a data transfer?
3. Which clock strobes the data into the drive buffer during a Write command?
4. Which signal is asserted when the data bus is busy?
5. Which signal initiates a function in the drive?

SOLUTIONS

1.

<u>D</u>	DPA during a Read command
<u>C</u>	RUN
<u>D</u>	Occupied
<u>D</u>	End of Block
<u>D</u>	Exception
<u>D</u>	Sync Clock
2. Exception
3. Write Clock
4. Occupied
5. Run





EDUCATIONAL SERVICES