Honeywell

SERIES 60 (LEVEL 6)

TYPE MSC9101/9102 MEDIUM PERFORMANCE DISK CONTROLLER MANUAL

Doc. No. 71010423-200 Order No. FM54, Rev. 1

Honeywell

SERIES 60 (LEVEL 6)

TYPE MSC9101/9102 MEDIUM PERFORMANCE DISK CONTROLLER MANUAL

Doc. No. 71010423-200 Order No. FM54, Rev. 1

RECORD OF REVISIONS

REVISION	DATE	AUTHORITY	AFFECTED PAGES
-100	Aug. 1976	--	--
-200	May 1977	BLCO60922 BLCO61160 BLCO61264 BLCO70306	--

Hardware Publications, M&TO, Billerica, MA 01821

Printed in the United States of America

0423/FM54

CONTENTS

CONTENTS

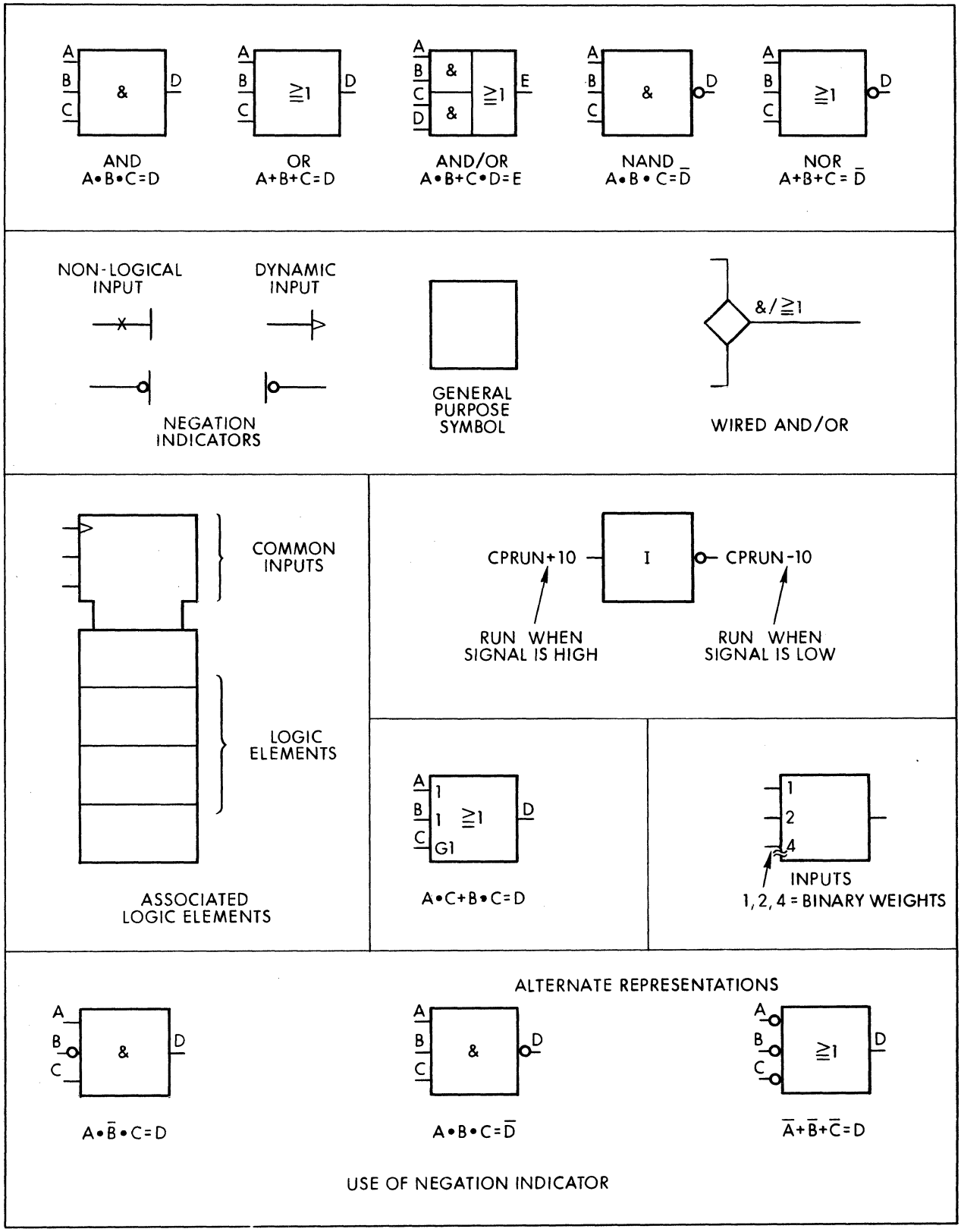
CONTENTS

ILLUSTRATIONS

ILLUSTRATIONS

TABLES

LOGIC SYMBOLOGY



1 INTRODUCTION

1.1 SCOPE AND PURPOSE OF THIS MANUAL

This product manual describes the functionality and operation of the Type MSC9101/9102 Medium Performance Disk Controller (MPDC) and the attached Read Only Store (ROS) adapter. A device adapter for either a cartridge disk or a storage module is used with the MPDC and is described in this manual from a functional standpoint only. The MPDC, the ROS adapter, and the device adapter make up the MPDC subsystem.

The software command structure for the MPDC is also described. For more detailed information pertaining to the software command structure and programming, refer to the System Overview Handbook, Order No. AS22, or the Peripherals Handbook, Order No. AT04.

The theory of operation presented in this manual is designed to acquaint the reader with the functional hardware elements of the MPDC at a major and an intermediate level. To analyze or troubleshoot the MPDC subsystem, logic block diagrams are available in the appropriate controller or adapter reference manual (see Table 1-1).

For information pertaining to device adapter operation and subsystem firmware, refer to the appropriate device adapter manual (see Table 1-1).

1.2 GENERAL DESCRIPTION

The MPDC is a microprogrammed controller that provides the Series 60 Level 6 System with the facility to store and retrieve data from mass storage media. An illustration of the MPDC with

attached adapters is shown in Figure 1-1. Two adapters attach to the MPDC: the read only store adapter, which contains the firmware microprogram instructions which support the MPDC and a specific device; and a device adapter, which contains the channel specific logic to control mass storage devices.

The MPDC is programmed to accept input/output (I/O) commands across the Megabus that are directed to the nonbusy I/O channels configured to the controller. Information transfers from the MPDC to the device adapter are in byte form. The device adapter supplies information to and retrieves information from the devices in bit serial form. Upon retrieval of bit serial information from the device, the device adapter assembles the information into a register for transfer to the MPDC in byte form.

Table 1-1 Reference Documents

TITLE	DOCUMENT NUMBER	ORDER NUMBER
Model 34/36 System Manual	71010200-202	FL35B
Model 34/36 Central Processor Manual	71010201-202	FL36B
Model 06 System Manual	71010210-200	FL37
Model 06 Central Processor Manual	71010211-200	FL38
Type MSC9101/9102 Medium Performance Disk Controller Reference Manual	71010241-200	FM35
Type CDM9101 Cartridge Disk Adapter Manual	71010240-200	FM08
Power System Manual	71010290-200	FL34
Type MSC9102 Storage Module Adapter Manual	71010429-100	FN80
System Overview Handbook	--	AS22
Peripherals Handbook	--	AT04

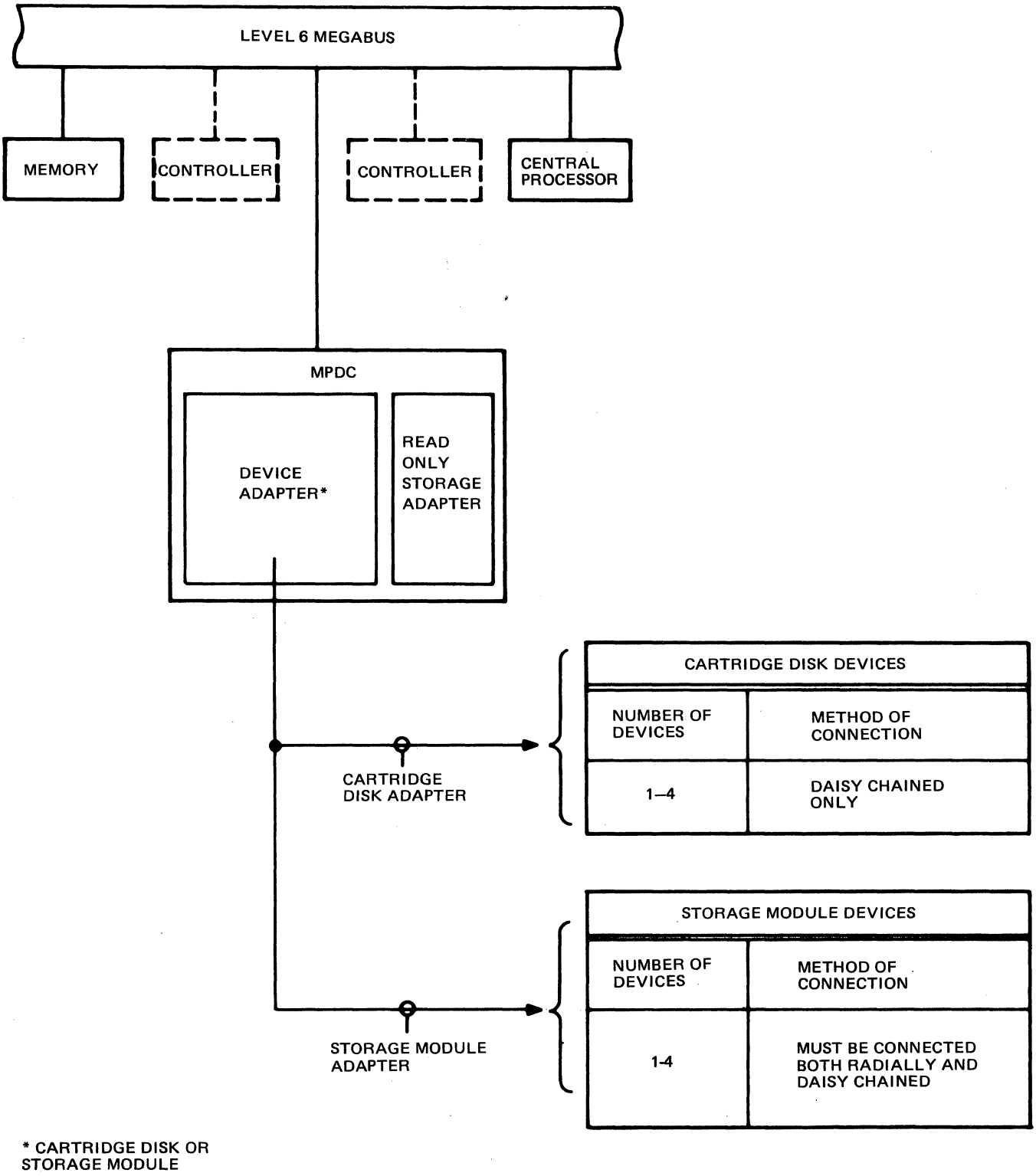


Figure 1-1 Medium Performance Disk Controller Subsystem

1.3 FUNCTIONAL DESCRIPTION

The major functional components of the MPDC subsystem are the MPDC, the read only store adapter, and the device adapter, which are described in the following text.

1.3.1 Medium Performance Disk Controller

The MPDC performs the following general control functions:

- Interfaces a maximum of four devices to the Megabus network
- Services the devices by multiplexing the appropriate data through the arithmetic logic unit and performs the action determined by the command set
- Executes Megabus command sequences
- Decodes commands
- Transfers data between the mass storage device and main memory via the Megabus
- Stores status and control register information received from the device adapter
- Controls the direction of data flow during command execution.

1.3.2 Read Only Storage (ROS) Adapter

The ROS adapter stores resident control firmware and diagnostic microprograms. The ROS adapter contains the following elements:

- A clock oscillator for the MPDC subsystem timing requirements
- A memory consisting of 2048 X 16 locations for the storage of control firmware and diagnostic microprograms
- The hardware elements necessary to retrieve the information stored in the programmable read only memory (PROM).

1.3.3 Cartridge Disk Adapter (CDA)

The CDA controls from one to four cartridge disk devices. Functions of the CDA are to:

- Control device interface dialogs
- Detect sync words
- Generate and verify check words
- Monitor the device states
- Execute data recovery not performed by the device
- Convert data formats from bit serial to byte, and byte to bit serial.

Seven types of cartridge disk devices may be connected to the MPDC subsystem as follows:

- Type CDU9101 Cartridge Disk (removable disk pack only, 100 TPI, 2.5 MBytes)

- Type CDU9102/9114 Cartridge Disk (removable disk pack plus a fixed disk, 100 TPI, 5.0 MBytes)
- Type CDU9103/9115 Cartridge Disk (removable disk pack only, 200 TPI, 5.0 MBytes)
- Type CDU9104/9116 Cartridge Disk (removable disk pack plus a fixed disk, 200 TPI, 10.0 MBytes).

1.3.4 Storage Module Adapter (SMA)

The SMA controls from one to four storage module devices. Functions of the SMA are to:

- Control device interface dialog
- Detect sectors
- Generate/detect address marks
- Execute error detection and correction (EDAC) generation and verification of corrections
- Generate and verify Cyclic Redundancy Check (CRC)
- Monitor the device states
- Execute data recovery that is not performed by the device
- Convert data formats from bit serial to byte and from byte to bit serial.

Four types of storage module devices may be connected to the MPDC subsystem as follows:

- Type 9760-1 Storage Module Drive (40 MBytes 3600 RPM)
- Type 9762-1 Storage Module Drive (80 MBytes 3600 RPM)
- Type 9764-1 Storage Module Drive (150 MBytes 3600 RPM)
- Type 9766-1 Storage Module Drive (300 MBytes 3600 RPM).

1.4 PHYSICAL CHARACTERISTICS

The MPDC consists of dual in-line packages (DIPs) mounted on a full-sized Series 60 Level 6 module manufactured using printed wiring assembly techniques. The MPDC board has two 50-pin connectors (Z01 and Z02) to insert the MPDC into the Megabus backplane, and four 25-pin connectors (W01 through W04) to connect the device adapter and the ROS adapter to the MPDC. The tables in Figure 1-2 indicate the manner in which the connectors are labeled in the MPDC subsystem and show the connections between the units. Cables to external devices connect via the back edge of the device adapter rather than the MPDC.

Three switches on the MPDC board must be set during installation. The two hex rotary switches, positioned in DIP locations C16 and C17 of the MPDC, are set to the channel number of the MPDC as specified in the system configuration. The rocker switch, positioned in DIP location P20 of the MPDC, is the device present switch. As shown in Figure 1-2, positions 5 through 8 of this switch correspond to devices 0 through 3 and are set to the ON position for each device that is connected to the device adapter (positions 1 through 4 are not used).

1.5 INTERFACE NETWORKS

The MPDC subsystem supports three interface networks, as shown in Figure 1-3. The Megabus/MPDC interface transfers all the signals that are required to permit the MPDC to function in the Series 60 Level 6 System. Across this interface the MPDC receives priority information, and sends and receives control, data and address information between the MPDC and other subsystems connected to the Megabus.

The MPDC/device adapter interface transfers data, strobes, and control information between the MPDC and the device adapter.

The MPDC/read only store adapter interface transfers control information and microinstructions between the MPDC and the ROS adapter. The MPDC addresses a location in the firmware read-only memory and starts the execution of MPDC/device firmware routines.

1.6 REFERENCE DOCUMENTS

The reference documents listed in Table 1-1 supplement the information contained in this manual.

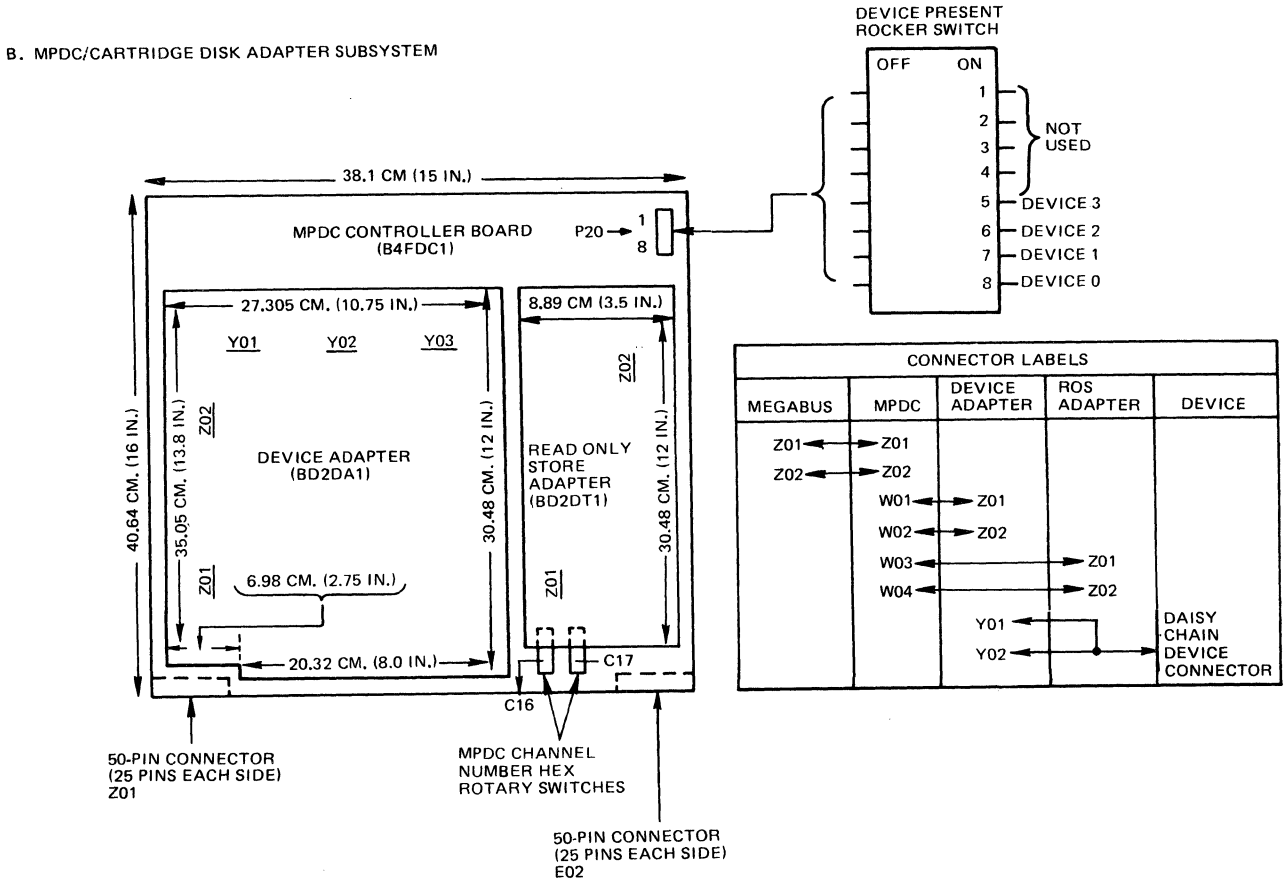


Figure 1-2 MPDC Subsystem Dimensions and Switch Placement (Sheet 1 of 2)

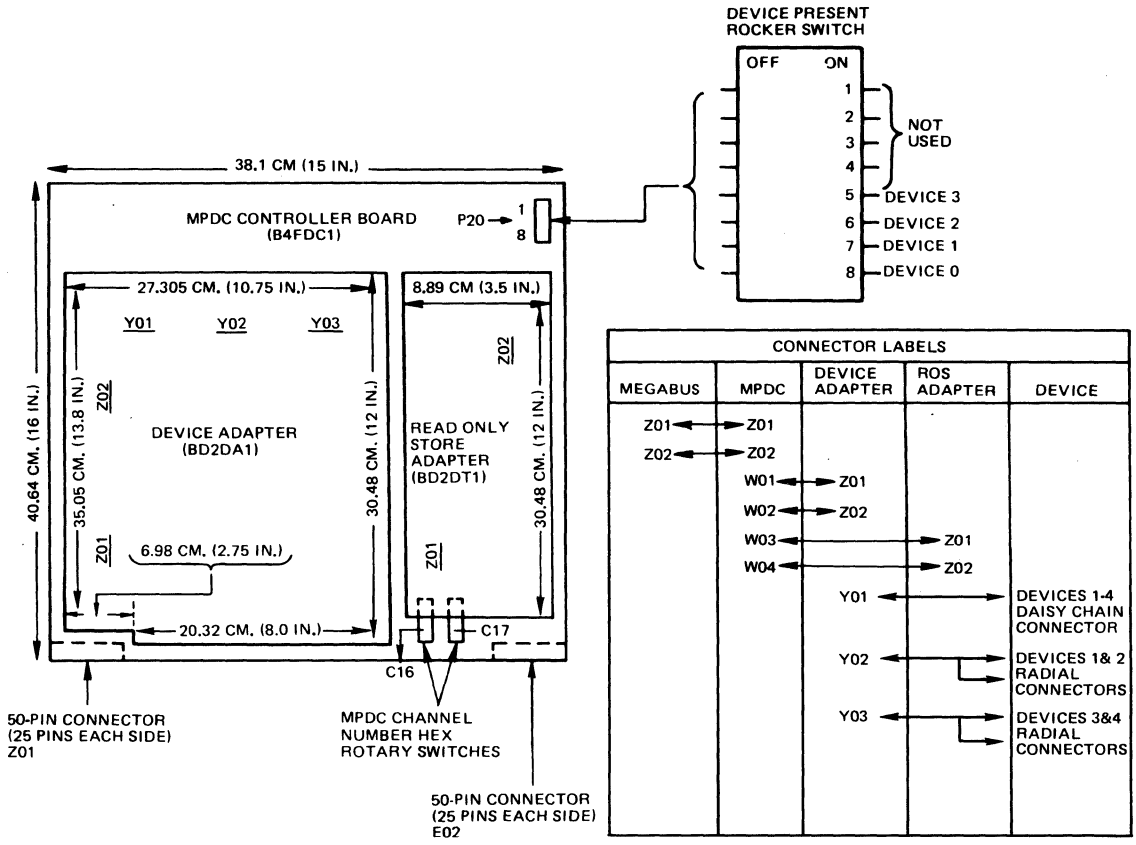


Figure 1-2 MPDC Subsystem Dimensions and Switch Placement (Sheet 2 of 2)

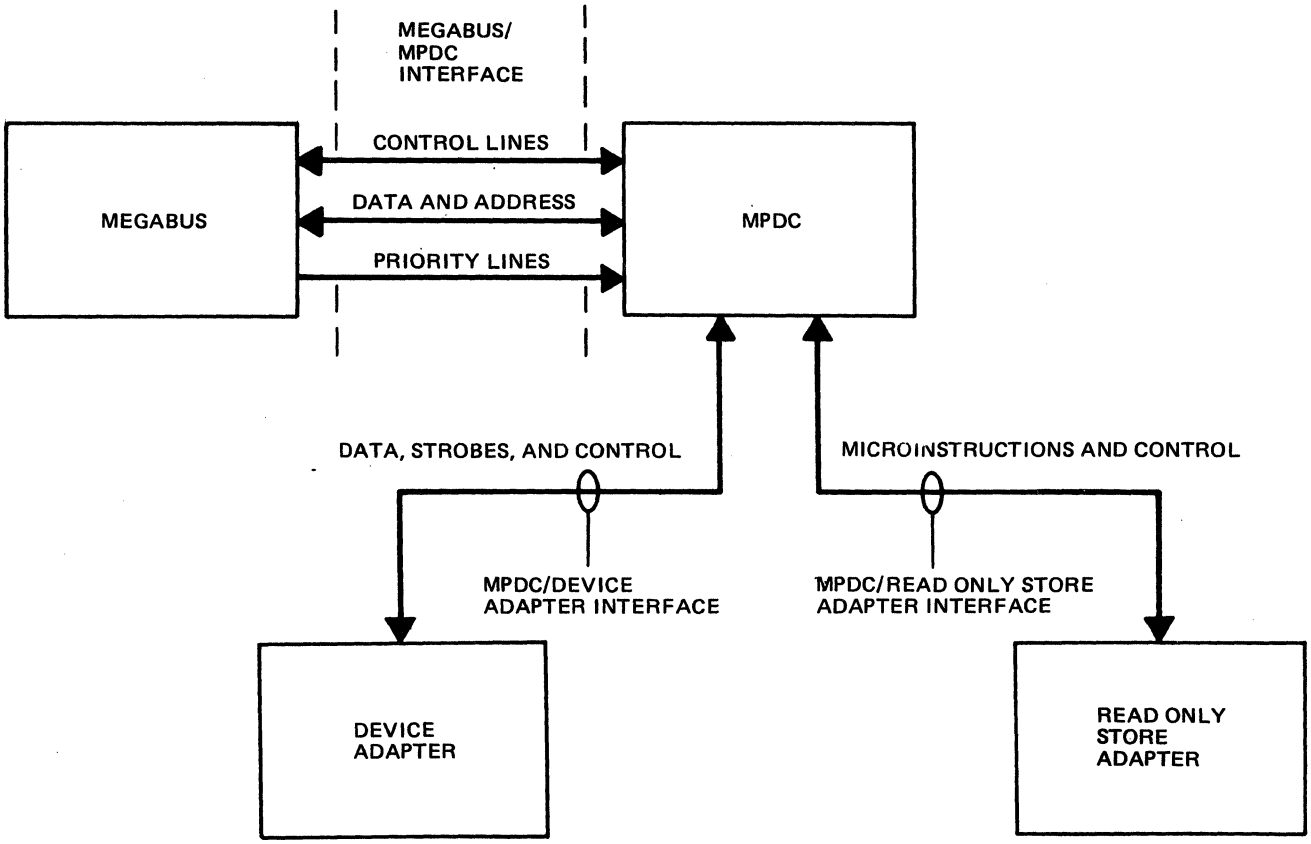


Figure 1-3 MPDC Interfaces

II THEORY OF OPERATION - OVERVIEW

The medium performance disk controller (MPDC), along with its associated adapters, is a firmware-oriented peripheral device controller which can communicate with any other controller on the Level 6 Megabus network.

Three functional components are necessary for the proper operation of the MPDC subsystem: software, firmware, and hardware. The following subsections provide a description of the software operations performed between the MPDC and other controllers on the Megabus network, and an overview description of the firmware control and hardware logic elements associated with the MPDC subsystem.

2.1 SOFTWARE

The MPDC operations are a direct result of an input or output instruction from the central processor (CP). The common types of operations which can occur across the Megabus and the software commands associated with the MPDC subsystem are described in this subsection.

2.1.1 Megabus Operations

The Series 60 Level 6 system allows any available unit (e.g., CP or MPDC) to independently establish communications with another unit via a shared signal path defined as the Megabus. The unit desiring communication requests a Megabus cycle. When access to the Megabus

is granted, via a priority network, the requesting unit becomes the master (information-transmitting unit) and is allowed to place specific parameters on the Megabus. Since the parameters consist of a unit address (channel number) accompanied by information (or data), the desired unit is selected as a slave (information-receiving unit) and the operation is completed in one Megabus cycle.

Certain types of master-to-slave communications require the initiation of a response or a second Megabus cycle, to complete the operation. When a slave is required to respond, it assumes the role of the master, initiates the response, delivers the parameters, and completes the operation during the second Megabus cycle. The duration of this operation depends on slave response time; therefore, the intervening time between the two Megabus cycles may be used by two other units not involved in the interchange.

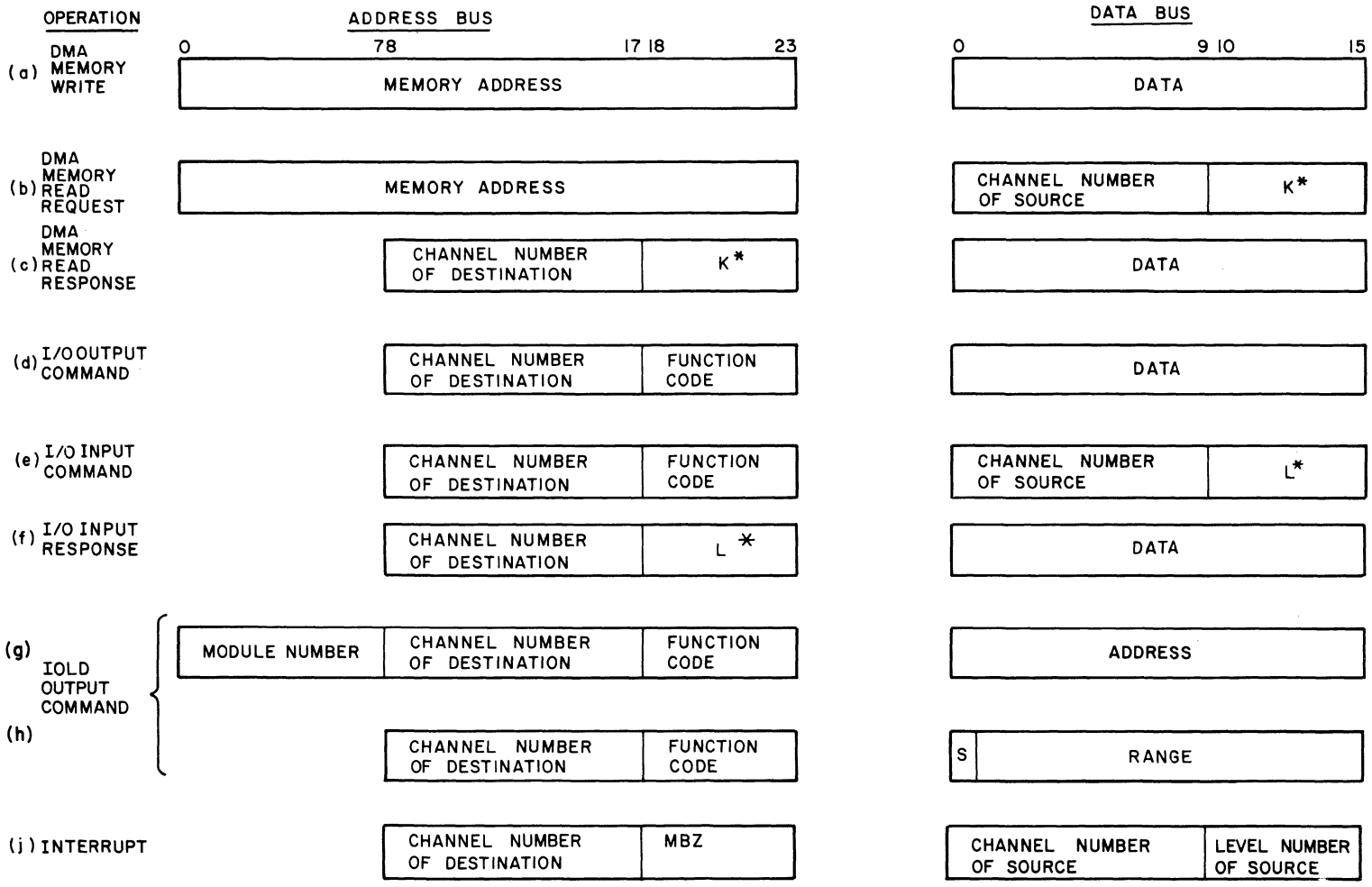
Table 2-1 lists the MPDC operations performed via the Megabus, the source of a request cycle, the destination, and the number of Megabus cycles required to complete the operation. Notice that write operations require only one cycle, whereas read operations require an additional Megabus cycle for the response.

2.1.1.1 Megabus Formats

Software utilizes the address bus and data bus to execute operations across the Megabus. Figure 2-1 shows the specific parameter formatting that is required to execute the Megabus operations listed in Table 2-1.

Table 2-1 MPDC Megabus Operations

TYPE OF OPERATION	SOURCE	DESTINATION	NUMBER OF BUS CYCLES
DMA Read	MPDC	Memory	2
DMA Write	MPDC	Memory	1
I/O Output Command	CP	MPDC	1
I/O Input Command	CP	MPDC	2
Interrupt	MPDC	CP	1



*INDICATES THAT INFORMATION CAN BE TRANSFERRED ACROSS THE DATA BUS IN THE INDICATED BIT POSITIONS DURING A READ REQUEST OR I/O INPUT COMMAND, AND THE INFORMATION WILL BE RETURNED ACROSS THE ADDRESS BUS IN THE INDICATED BIT POSITIONS DURING THE CONTROLLER RESPONSE CYCLE. AN EXCEPTION IN FIELD L IS THE GENERAL PURPOSE INTERFACE (GPI) AND THE 716 BUS ADAPTER, WHICH RETURN ZEROS.

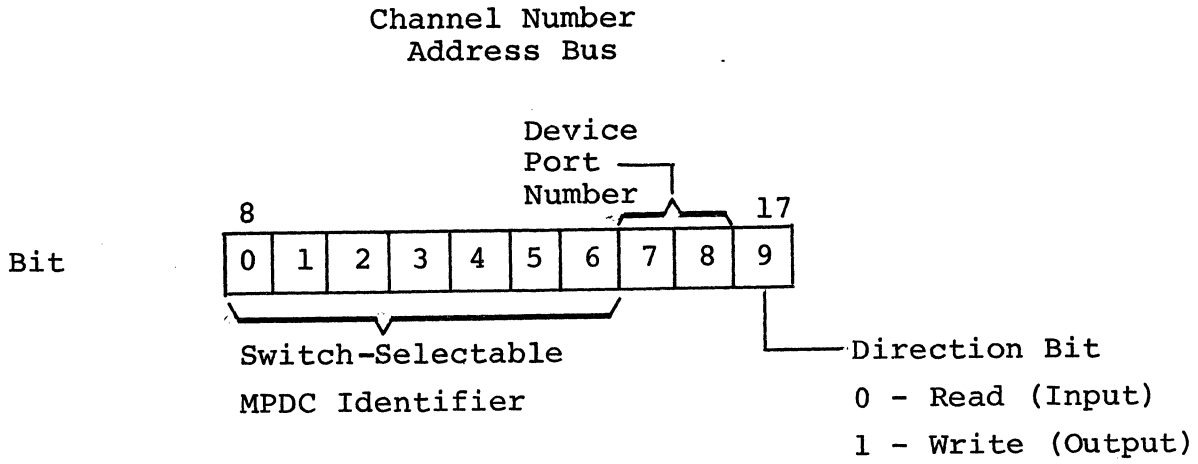
Figure 2-1 Data and Address Bus Formats

2.1.1.2 Unit Addressing

Devices attached to the MPDC are software addressable via channel numbers. Each individual device (drive) has two channel numbers assigned, differing from each other only in the low order bit position (called the direction bit). When a data transfer Input/Output (I/O) command is issued to a bidirectional device (e.g., disk), the direction bit of the channel number specifies whether it is an input or output data transfer. For all other software commands, the direction bit is ignored by the MPDC hardware. Software visibility of the attachments to an MPDC is such that the device subsystems (adapter plus associated device or devices) are in general independent of each other. For example, operations on one disk are independent of any activity on another disk except that the controller response to Megabus sequences addressed to one device (channel number) may be stalled while the MPDC is busy servicing another device (i.e., a command sequence is accepted but not initiated).

If a device adapter is configured with fewer than four devices, it responds to channel numbers associated with the installed devices only.

The bit position utilization of the channel number is as follows:



2.1.1.3 Device Identification

The functionality of the Megabus requires the assignment of a unique device identification number for every type of device indirectly connected to the Megabus via a control unit. The device ID consists of 16 bits and is placed on the data bus when requested by software via the Input Device Identification command (see subsection 2.1.2.2). Device ID numbers find usage in the following ways:

- In the MPDC, during initialization time, firmware can identify the specific complement of devices attached.
- Test and verification programs can use the ID numbers to identify the specific device features so that they can properly test them.

- The device drivers can use the ID number to set certain parameters within the driver as a function of the device features.
- It is possible to determine the actual system configuration by polling each channel number in turn to determine the ID numbers.

2.1.1.4 Function Codes

When an I/O command format is placed on the Megabus, the requested operation is identified by the configuration of the accompanying six-bit function code (FC). This function code appears on the address bus lines (bits 18 through 23) only.

The low order function code bit (bit 23) determines if the operation is an input or output command. If the bit is a One, an I/O output command is indicated; if a Zero, an I/O input command is to be executed. With this consideration, the function codes and the types of I/O commands to which the MPDC responds are listed in Table 2-2.

The function code configurations are stored in predefined scratch pad memory (SPM) locations. During I/O Megabus commands, the SPM can be accessed to write into or read from a specific SPM location. Certain function codes are related to work locations in the SPM that are assigned to the user, allowing independent organization of these SPM locations for user-defined I/O commands.

Table 2-2 Input/Output Commands

COMMAND TYPE	FUNCTION CODE	INSTRUCTION
Output	01 (Hex)	Control Word
	03	Interrupt Control
	07	Task Word
	09	Input/Output Load Address (09) Range (0D)
	0F	Offset Range
	11	Configuration Word A
	13	Configuration Word B
Input	02	Interrupt Control
	06	Task Word
	08	Memory Byte Address
	0A	Memory Module Address
	0C	Range
	10	Configuration Word A
	12	Configuration Word B
	18	Status Word 1
	*1A	Status Word 2
	26	Identification Word
Diagnostic	Any Even Code	Read Scratch Pad Memory
	Any Odd Code	Write Scratch Pad Memory

*This command is used with the Storage Module adapter only.

2.1.2 Input/Output Commands

The I/O commands supported by the MPDC are described in the following subsections. Because many of the commands are device dependent, detailed command descriptions appear in the appropriate device adapter manual.

2.1.2.1 Output Commands

As shown in Table 2-2, there are seven output commands related to the MPDC which enable predefined operations required by the MPDC. This subsection describes the output commands, depicting the operations performed in the MPDC controller. Output commands are identified by odd numbered function codes.

1. Control Word (FC01) - This command (Figure 2-2) loads a control word into the SPM for the device channel referenced by software. The Control Word command is unconditionally accepted by the channel regardless of its busy state, with one exception. If the MPDC is currently executing a data transfer operation on another channel, software may receive a Wait response from the MPDC until the current transfer is completed.

   Two control operations (Initialize and Stop I/O) may be executed by the MPDC through the Control Word command. The operation performed is determined by the logic state of bits 0 and 1 on the data bus.

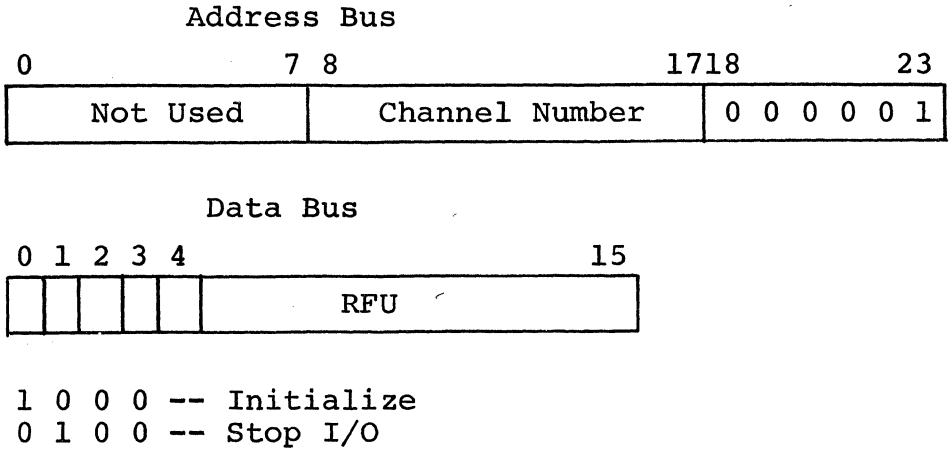


Figure 2-2 Control Word Format

   a. Initialize - The Initialize operation causes the MPDC to run its basic logic test; clear the bus interface; block interrupts; reset any Busy conditions; and clear all device channels.

Any operations that are being executed at the time of initialization are abruptly terminated, and all registers are cleared. Information pertaining to the terminated operations cannot be recovered, and the interrupt level for all channels is set to Zero (i.e., no interrupts are generated for the terminated operations).

b. Stop I/O - The Stop I/O operation causes any operation currently active on the specified device channel to be abruptly terminated. If a data transfer operation is in progress, it is not completed and no error checking is performed. If the device is actively performing an operation when the Stop I/O is received, position-oriented information cannot be recovered. An interrupt is generated for the operation terminated by the Stop I/O as if the operation had terminated normally. Status, address, and range information in the MPDC remains unaltered. The Stop I/O function has no effect on any channel other than the one specified.

2. Interrupt Control (FC03) - The Interrupt Control command (Figure 2-3) loads the SPM with the referenced device channel, the interrupt level, and the channel number of the CPU to which subsequent interrupts are to be delivered.

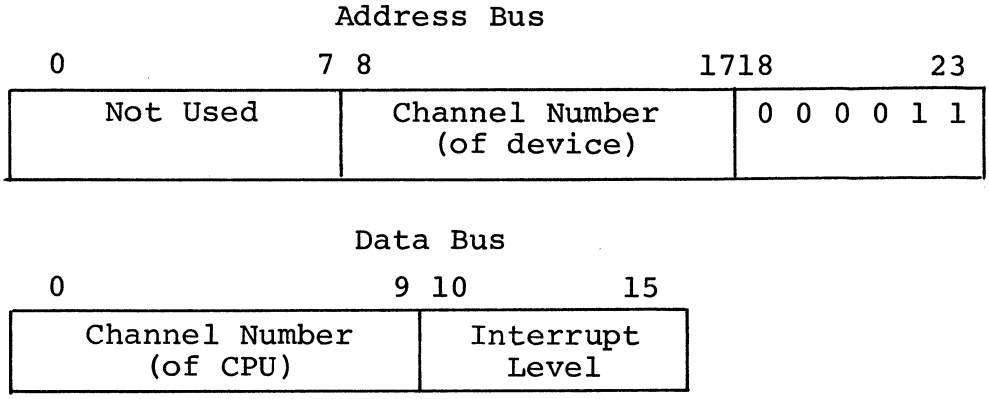


Figure 2-3 Interrupt Control Format

The interrupt level is a 6-bit quantity and is positioned on the data bus in bit positions 10 through 15. The channel number of the CPU loading the interrupt level is located in bit positions 0 through 9 of the data bus. The interrupt level is set to Zero whenever the MPDC subsystem is initialized. If an interrupt level of Zero is loaded in the SPM, interrupts are not generated or saved for any event that occurs while the interrupt level is set to Zero.

3. Task Word (FC07) - The Task Word command (Figure 2-4) transmits device-specific information to a referenced channel.

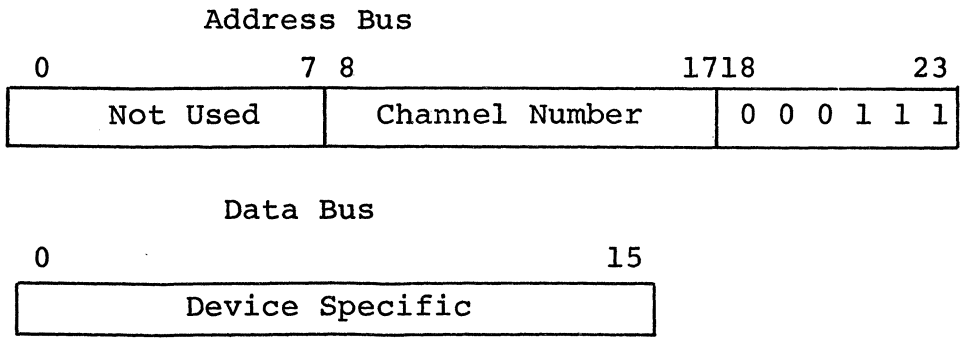


Figure 2-4 Task Word Format

The coding of bits 0 through 15 of the data bus represents the operation that is to be performed. When this command is accepted, the task word is stored in the scratch pad memory, the device channel enters the busy state, and the command is stacked. If no channel is executing an operation, the task is executed the next time that firmware enters the polling loop. If other channels have commands stacked, the priority determines when the task is executed. All address, range, and configuration information must be loaded prior to execution of this command. The direction of data transfer is indicated in the low order bit of the channel number used in the most recent output address command. Refer to the appropriate device adapter manual for specific operations that may be executed via the Task Word command.

4. Input/Output Load (IOLD) (FC09) - The I/O Load command is transformed by the CPU into the Output Address and Output Range commands on the Megabus. Each IOLD command results in an Output Address command (Figure 2-5) followed by an Output Range command (Figure 2-6). The programmer only needs to specify the first function code (FC09) for the initial transfer of a 24-bit main memory address to the MPDC. The CPU hardware/firmware generate the second function code (FC0D) for the transfer of the 16-bit range number. In this manner, the MPDC decodes two separate commands from the transfer as follows:

   a. Output Address Command -

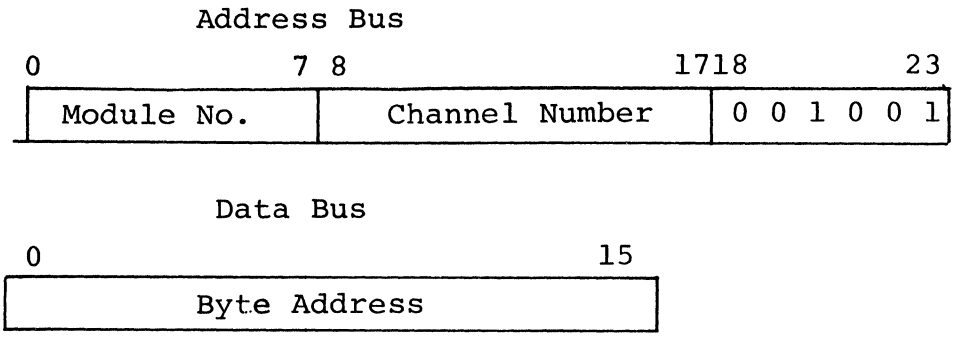


Figure 2-5 Output Address

This command loads a 24-bit address into the address register associated with the referenced channel. The address refers to the starting (byte) location in the main memory where the MPDC commences input or output data transfers. Bits 0-7 of the address bus (module number) are the most significant bits of the address; the data bus contains the 16 least significant address bits. Data transfers to or from memory normally are on a word basis, but byte mode transfers can occur (associated with the first and/or last memory cycle of a particular data transfer) if the main memory buffer (identified by this instruction) begins or ends on an odd byte boundary.

b. Output Range Command -

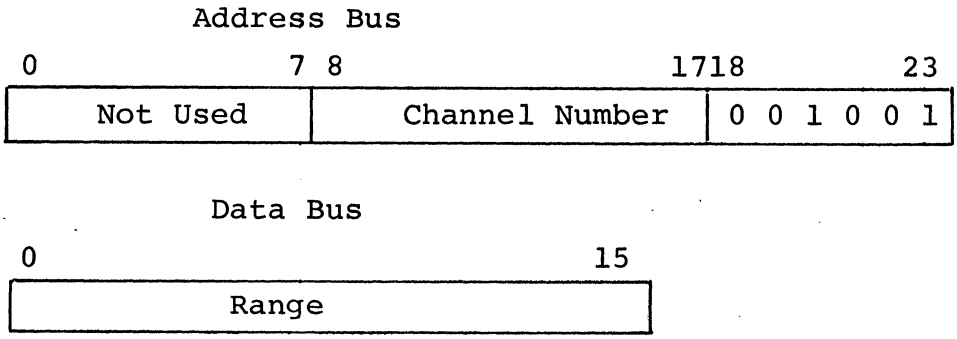


Figure 2-6 Output Range Format

This command loads the range register associated with the referenced device channel. The 16-bit quantity loaded (data bus) is the number of bytes to be transferred during the data transfer that is being set up. The number is a positive binary quantity (bit 0 must be Zero) and is decremented by the controller after each memory transfer. A range of zero results in a premature end-of-operation termination for any read or write command that may be subsequently issued. Any range register by another Output Range command.

5. Offset Range (FC0F) - This command (Figure 2-7) loads the output range register in the SPM location associated with referenced device channel. The 16-bit quantity loaded (data bus) is the number of bytes to be discarded from the beginning of the data transfer prior to the transfer of any data to the main memory. The address bus and data bus configurations are as shown in Figure 2-7.

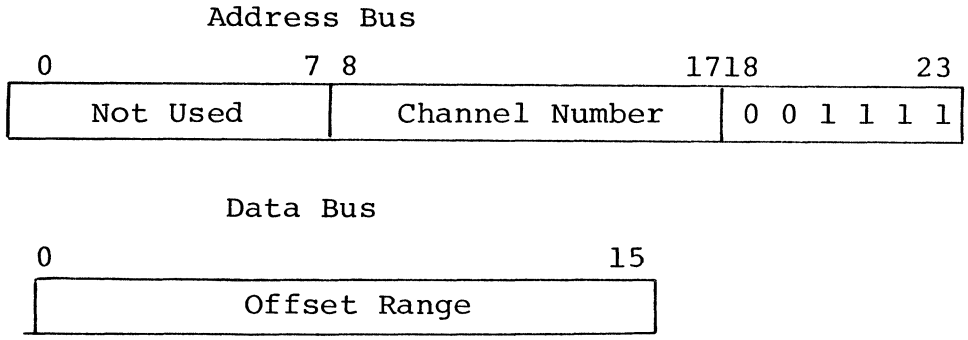


Figure 2-7 Offset Range Format

The output range is decremented only after the offset range register is decremented to zero. The output offset range is used only in conjunction with read operations (ignored for write operations) and must be set for each data transfer using an offset. The offset is a positive binary quantity and is decremented by the MPDC after each byte read from the device. Any offset range register residue is applied to the next command unless reset by another Output Offset Range instruction.

6. Configuration Word A (FC11) - The Configuration Word A command (Figure 2-8) transmits device-specific information to a referenced channel.

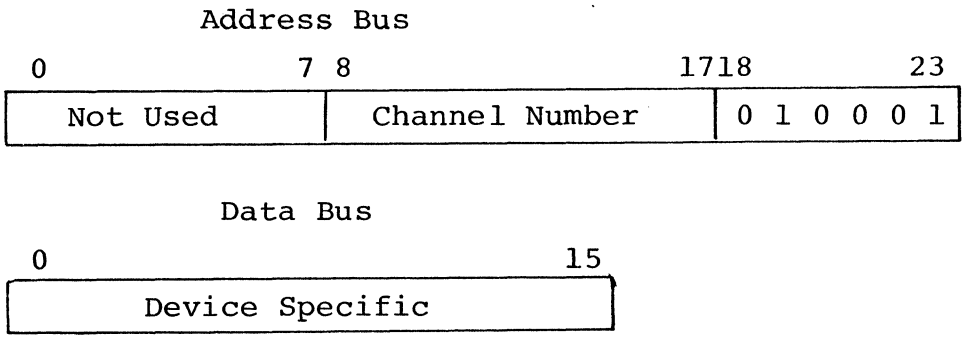


Figure 2-8 Configuration Word A Format

The information contained in Configuration Word A is stored in the SPM for the device corresponding to the referenced channel. This word provides a seek address for the attached disk or storage module devices. Refer to the appropriate device adapter manual for specific operations that may be executed via the COnfiguration Word A command.

7. Configuration Word B (FC13) - The Configuration Word B command (Figure 2-9) transmits device-specific information to a referenced channel.

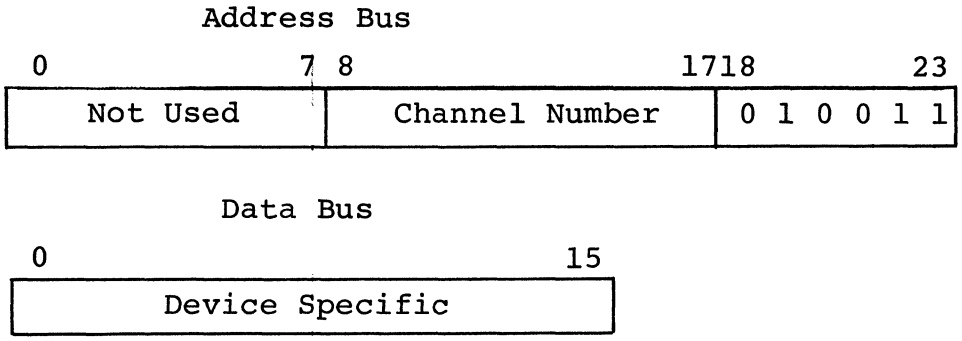


Figure 2-9 Configuration Word B Format

Configuration Word B is used as the low order two bytes of a Sector ID field to be searched for during a data field read or write operation on the specified device. The sector number is incremented by One after each data field during Read or Write operations and points to the next sector after completing the operation. Refer to the appropriate device adapter manual for specific operations that may be executed via the Configuration Word B command.

2.1.2.2 Input Commands

Ten input commands related to the MPDC are shown in Table 2-2. They enable predefined operations to occur in the MPDC and are identified by even numbered function codes. This subsection describes the input commands, depicting the operations performed in the MPDC.

Each input command requires two Megabus cycles to complete. The first Megabus cycle is the request cycle which sets up the MPDC for the transfer of requested information. The second Megabus cycle, the response cycle, delivers the requested information to the appropriate controller.

1. Interrupt Control (FC02) - This command (Figure 2-10) causes the device channel's interrupt level to be transferred to the requesting channel.

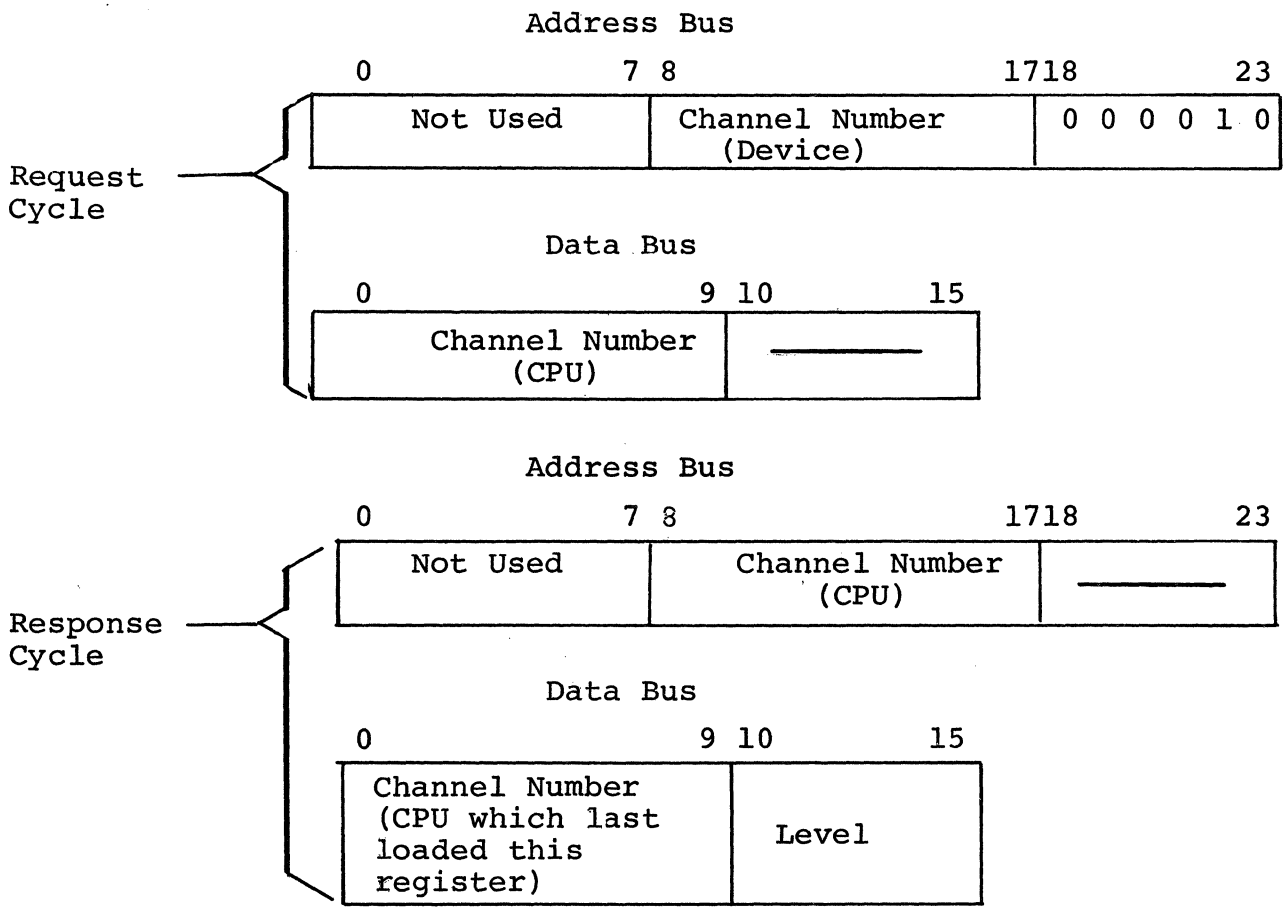


Figure 2-10 Interrupt Control Format

During the response cycle (second half read), the level value of the requested device channel is placed on the data bus in bit positions 10 through 15. This quantity is the value previously received in an Output Interrupt Control command or is a default value of Zero (the interrupt level assumed by the device channel when initialized). The channel number returned in bit positions 0 through 9 of the data bus may differ from the channel number of the CPU executing the command if more than one CPU is attached to the Megabus.

The information received by the MPDC in bit positions 0 through 15 of the data bus during the request cycle is returned to the requesting controller in bit positions 8 thorugh 23 of the address bus during the response cycle.

2. Task Word (FC06) - This command (Figure 2-11) causes the Task Word of the referenced device channel to be transferred to the requesting channel. The Task Word transferred contains the code of the last operation executed by the channel (unless an Initialize has occurred). The request and response cycles are shown in Figure 2-11.

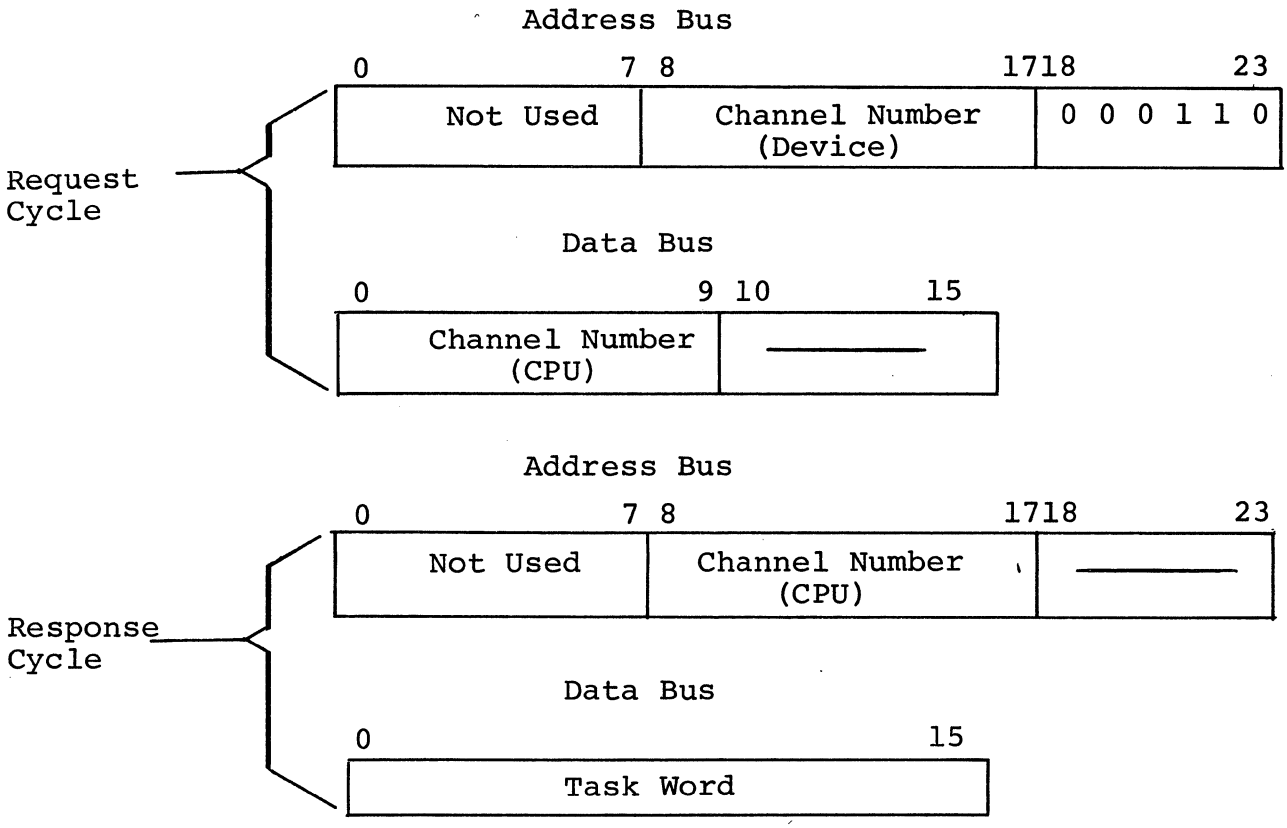


Figure 2-11 Task Word Format

During the response cycle (second half read) the MPDC returns in bit positions 8 through 23 of the address bus the same information that was received in bit positions 0 through 15 of the data bus during the request cycle.

3. Memory Byte Address (FC08) - This command (Figure 2-12) causes the current contents of the referenced device channels Memory Byte Address to be transferred to the requesting channel.

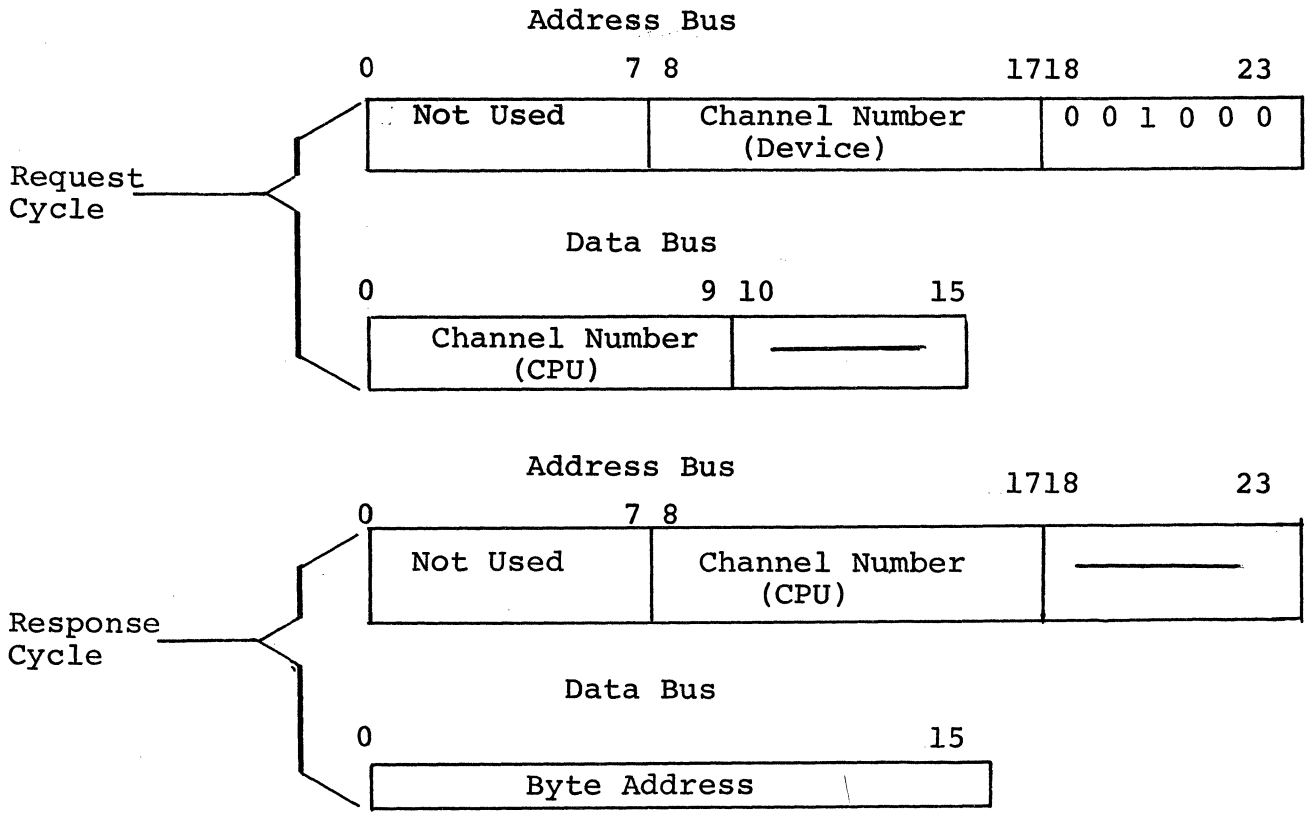


Figure 2-12 Memory Byte Address Format

During the response cycle (second half read), the MPDC returns in bit positions 8 to 23 of the address bus the same information that was received in bit positions 0 to 15 of the data bus during the request cycle. The data bus contains the low order 16 bits of the Memory Byte Address currently stored for the specified channel in the MPDC. Note that if a Write command ended at a byte coundary (high order eight bits of a word), the memory address reflects the next word (not the low order eight bits of the previous word).

4. Memory Module Address (FC0A) - This command (Figure 2-13) causes the current contents of the referenced device channels Memory Module Address to be transferred to the requesting channel.

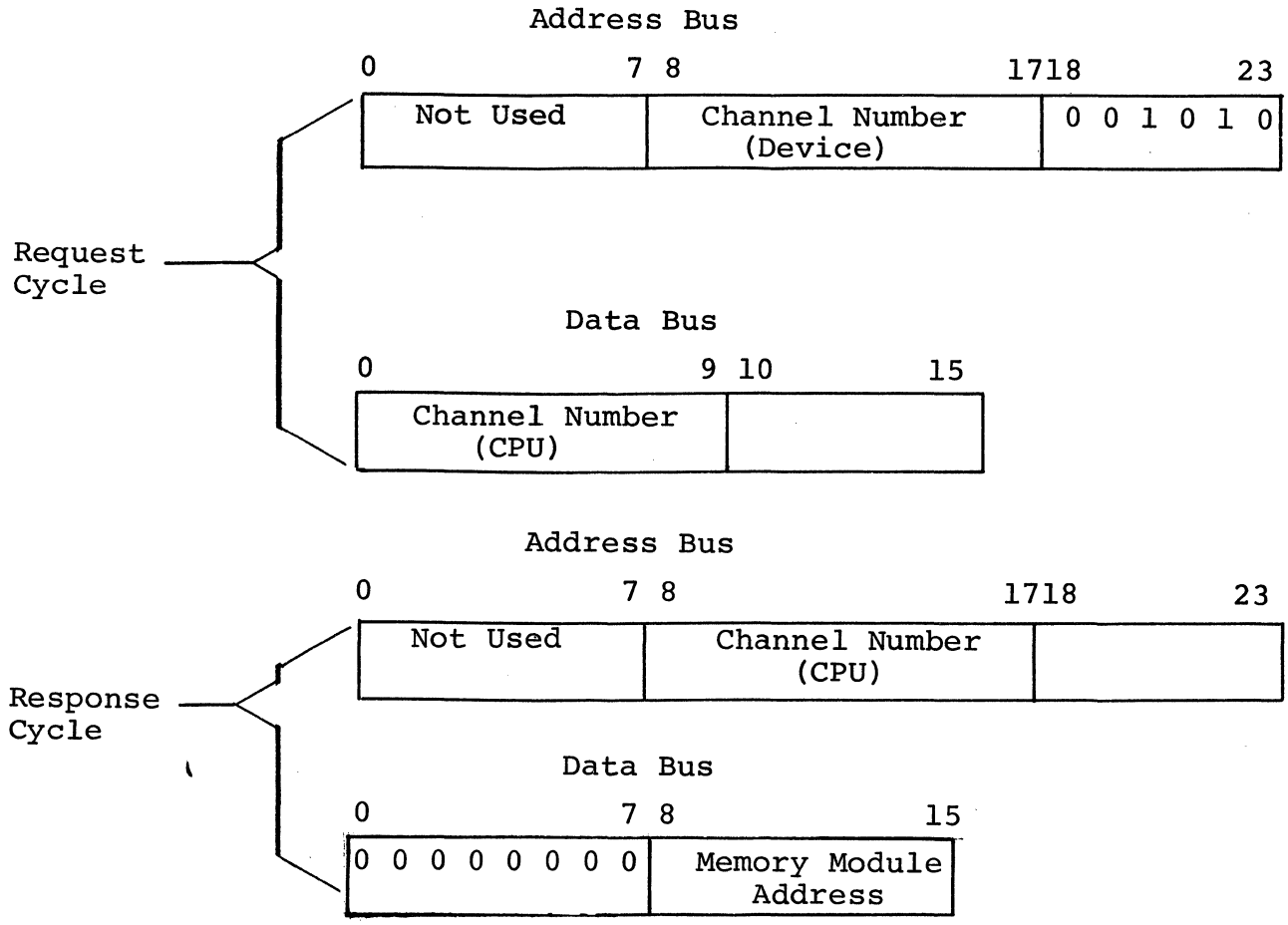


Figure 2-13 Memory Module Address Format

During the response cycle (second half read), the MPDC returns in bit positions 8 to 23 of the address bus the same information that was received in bit positions 0 to 15 of the data bus during the request cycle. The data bus contains the high order eight bits of the Memory Word Address currently stored for the specified channel in the MPDC.

5. Range (FC0C) - This command (Figure 2-14) causes the current contents of the referenced device channels Range register to be transferred to the requesting channel.

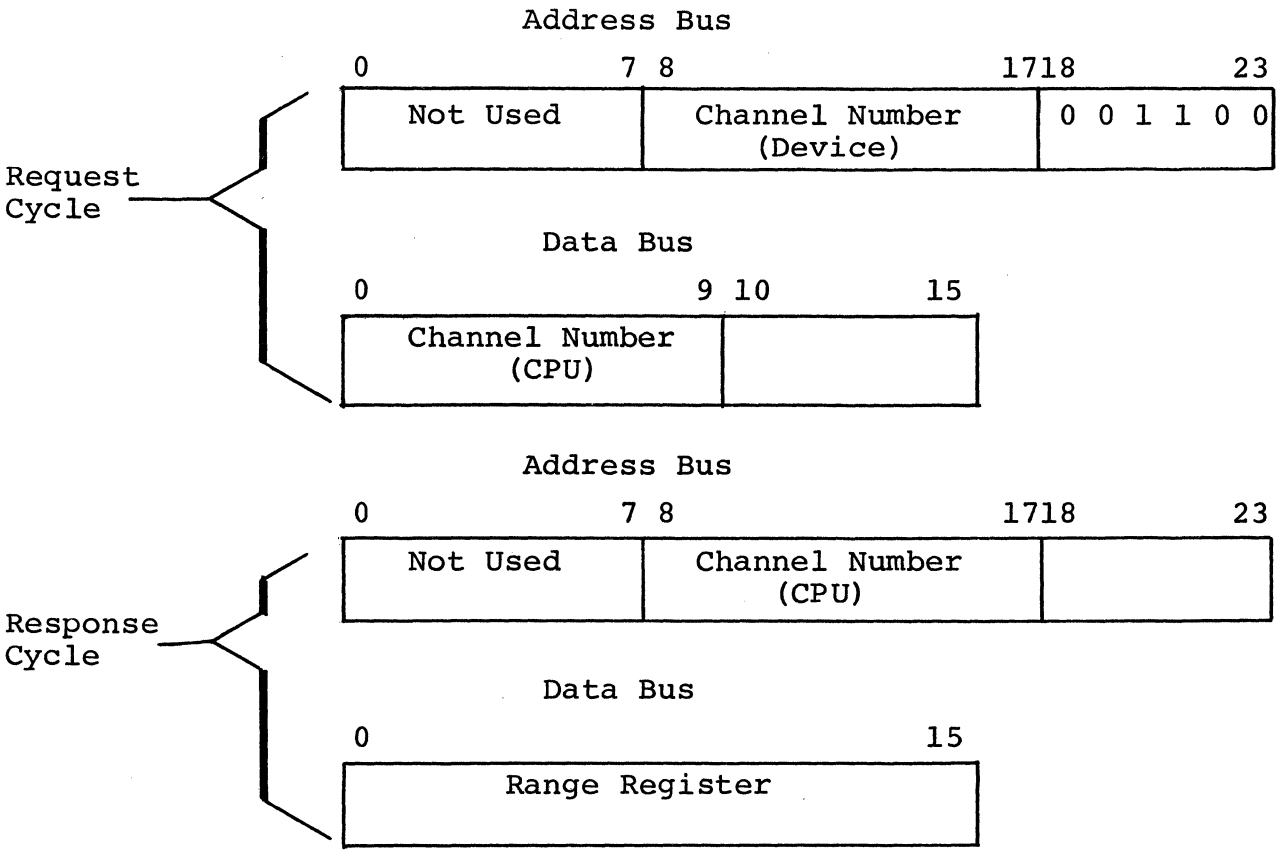


Figure 2-14 Range Format

During the response cycle (second half read), the MPDC re-returns in bit positions 8 through 23 of the address bus the same data that was received in bit positions 0 through 15 of the data bus during the request cycle.

6. Configuration Words A and B (A=FC10, B=FC12) - The operations performed by the MPDC address bus and data bus are identical for the transfer of Configuration Words A and B (except for the function codes); therefore, their operation is explained under one heading. These two commands (Figure 2-15) cause the current contents of the device channels Configuration Word A (B) to be transferred to the requesting channel.

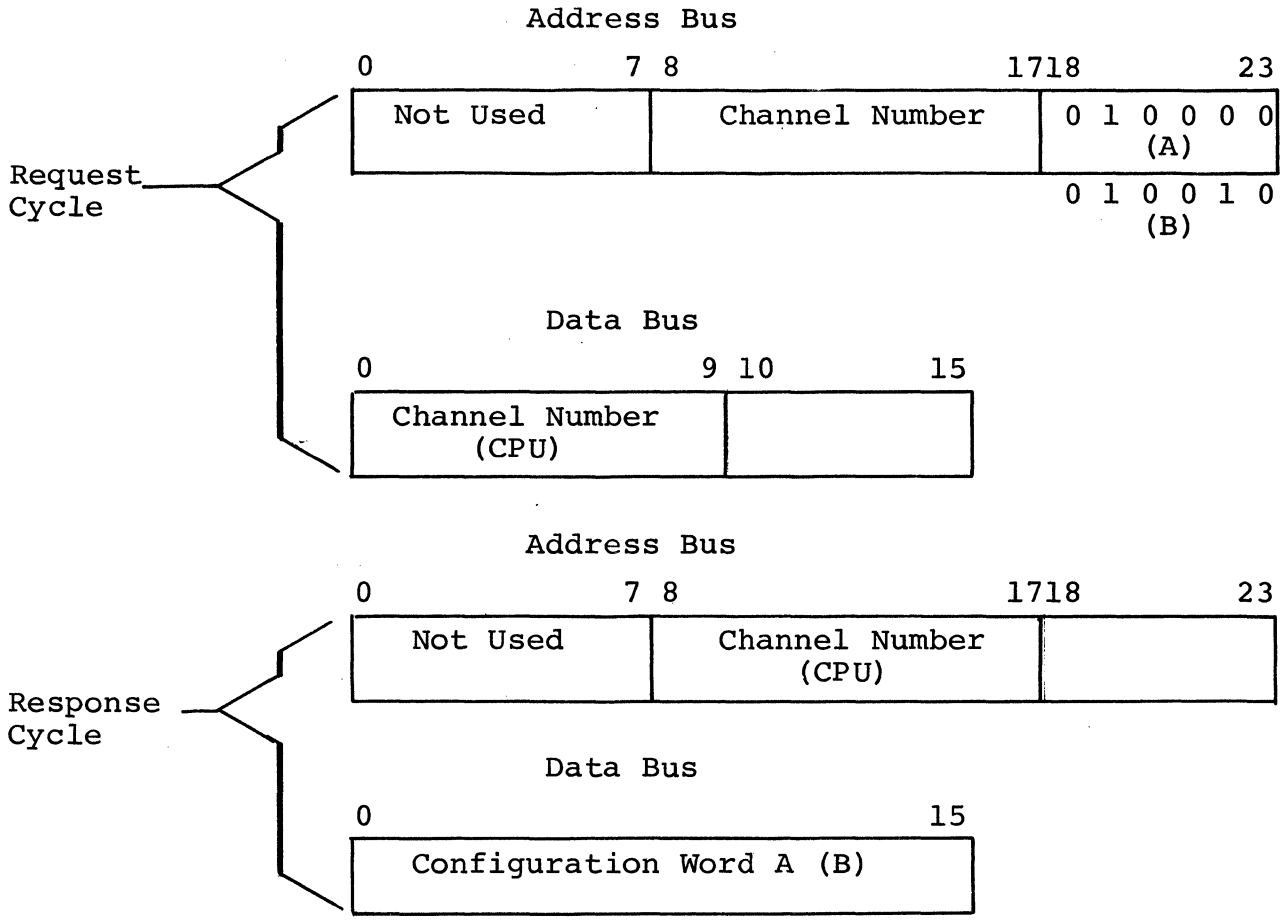


Figure 2-15 Format for Configuration Words A and B

During the response cycle (second half read), the MPDC returns in bit positions 8 through 23 of the address bus the same information that was received in bit positions 0 through 15 of the data bus during the request cycle.

7. Status Word 1 (FC18) - This command (Figure 2-16) causes the referenced device channel's Status Word 1 to be transferred to the requesting channel. During the response cycle (second half read), the MPDC returns in bit positions 8 through 23 of the address bus the same data that was received in bit positions 0 through 15 of the data bus during the request cycle. The request and response cycles are shown in Figure 2-16. The relationship between the status word 1 bits and the MPDC is explained in the following text.

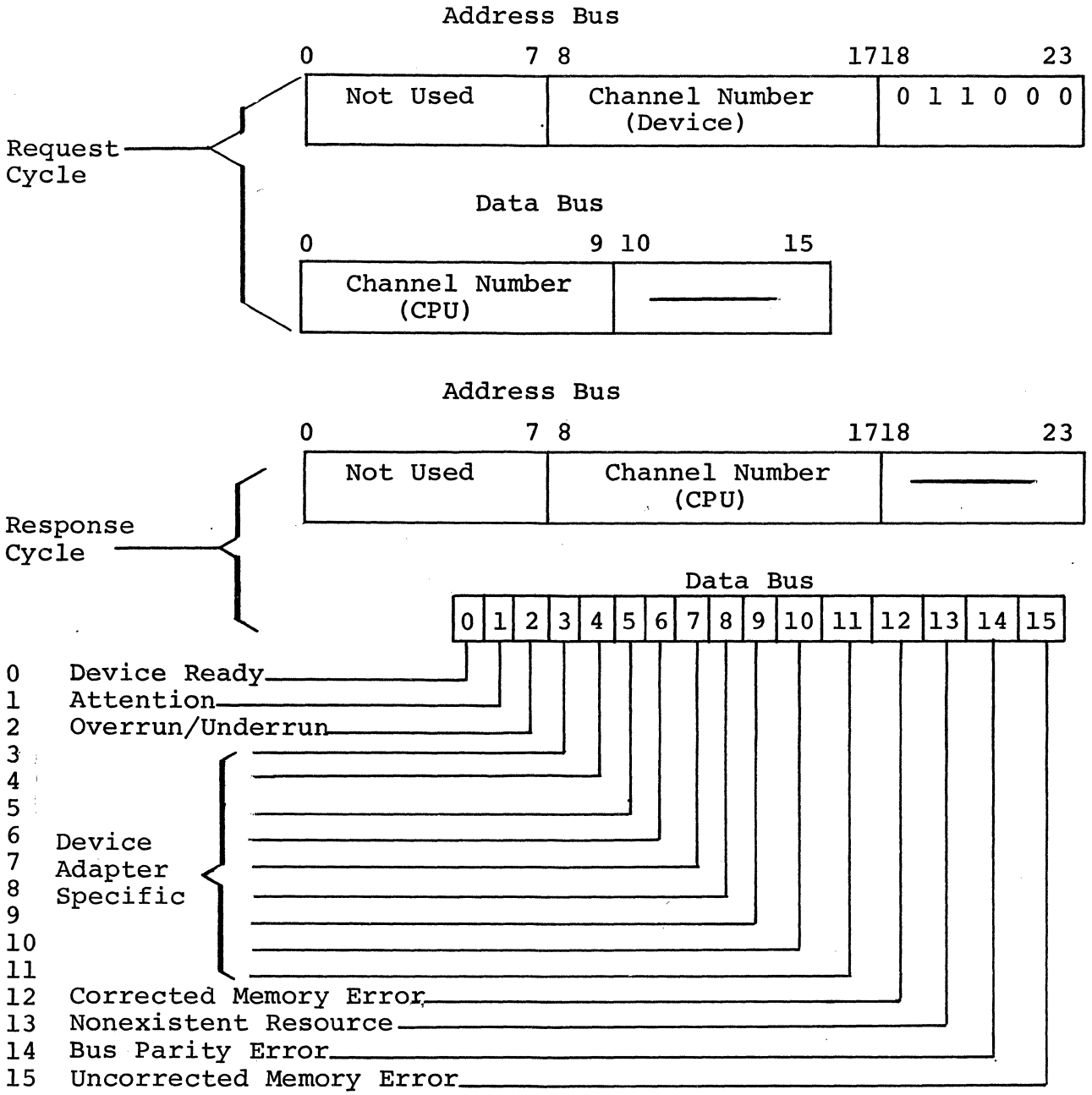


Figure 2-16 Status Word 1 Format

a. Device Ready (bit 0) - This bit indicates that the device is on line with the medium loaded and that no further manual intervention is required to place it under program control. A change in state of this bit causes the Attention bit (bit 1) to be set, resulting in an interrupt (if the interrupt level is nonzero).

b. Attention (Bit 1) - This indicator is set whenever the Device Ready bit (bit 0 of the status word) changes state. Any change in the operational status of the device (e.g., load/unload of media) will be indicated to software by setting the attention bit.

   Whenever the attention bit is set, an interrupt is attempted (if the interrupt level is non-zero). If a previously initiated operation is in progress when a device state change is sensed, the resultant interrupt (with the attention bit set) serves as notification of both the end of the operation and the device state change.

   This bit is reset by an Initialize command (Output Control Word), an Output Task Word command, a Master Clear on the Megabus, or an error free Input Status Word command.

c. Overrun/Underrun (Bit 2) - This bit is set during a read or write operation when the data transfer to/from the main memory cannot be maintained at a high enough rate (as required by a specific device). Either data was lost on input because of failure to keep up with device demands or data was unavailable on output when required by the device. This bit is reset by an Initialize command (Output Control Word), an Output Task Word command, or a Master Clear on the Megabus.

d. Bits 3 through 11 - These bits are device specific. Refer to the appropriate device adapter manual for the assignment of these bits in Status Word 1.

e. Corrected Memory Error (Bit 12) - This bit indicates that during execution of the previous operation, the main memory detected and corrected a memory read error. The data delivered to the MPDC was assumed to be correct. This bit is reset by an Initialize command, an Output Task Word command, an error-free* Input Status Word 1 command, or a Master Clear on the Megabus.

f. Nonexistent Resource (Bit 13) - This bit is set whenever the MPDC attempts any Megabus cycle (except interrupt) and receives a Negative Acknowledge response from the memory.

   This bit is reset by an Initialize command (Output Control Word), an Output Task Word command, a Master Clear on the Megabus, or an error-free Input Status Word 1 command.

g. Bus Parity Error (Bit 14) - This bit is set whenever the MPDC detects a parity error on either byte of the data bus during any bus cycle or when a parity error is detected in bits 0 to 7 of the address bus.

*An Input Status Word 1 command from the storage module device only resets the Corrected Memory Error bit.

This bit is reset by an error-free Input Status Word 1 command or an Initialize (via a Master Clear or an error-free Output Control Word command).

h. Uncorrected Memory Error (Bit 15) - This bit indicates that, during the execution of the previous operation, the main memory detected a memory read error which the error detection and correction (EDAC) algorithm could not correct. The data that was delivered to the MPDC was incorrect. The occurrence of this condition does not cause a termination of the operation in progress but may result in bad data written on the medium. This bit is reset by an Initialize command (Output Control Word), an Output Task Word command, or a Master Clear on the Megabus.

8. Status Word 2 (FC1A) - This command (Figure 2-17) causes the referenced device channel's Status Word 2 to be transferred to the requesting channel. Status Word 2 is used with the Storage Module adapter only.

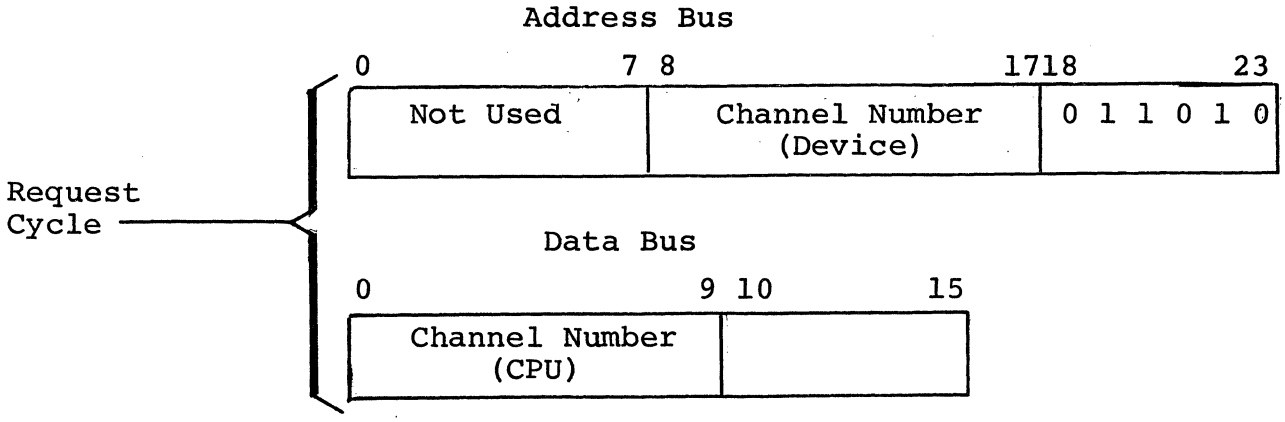


Figure 2-17 Status Word 2 (Sheet 1 of 2)

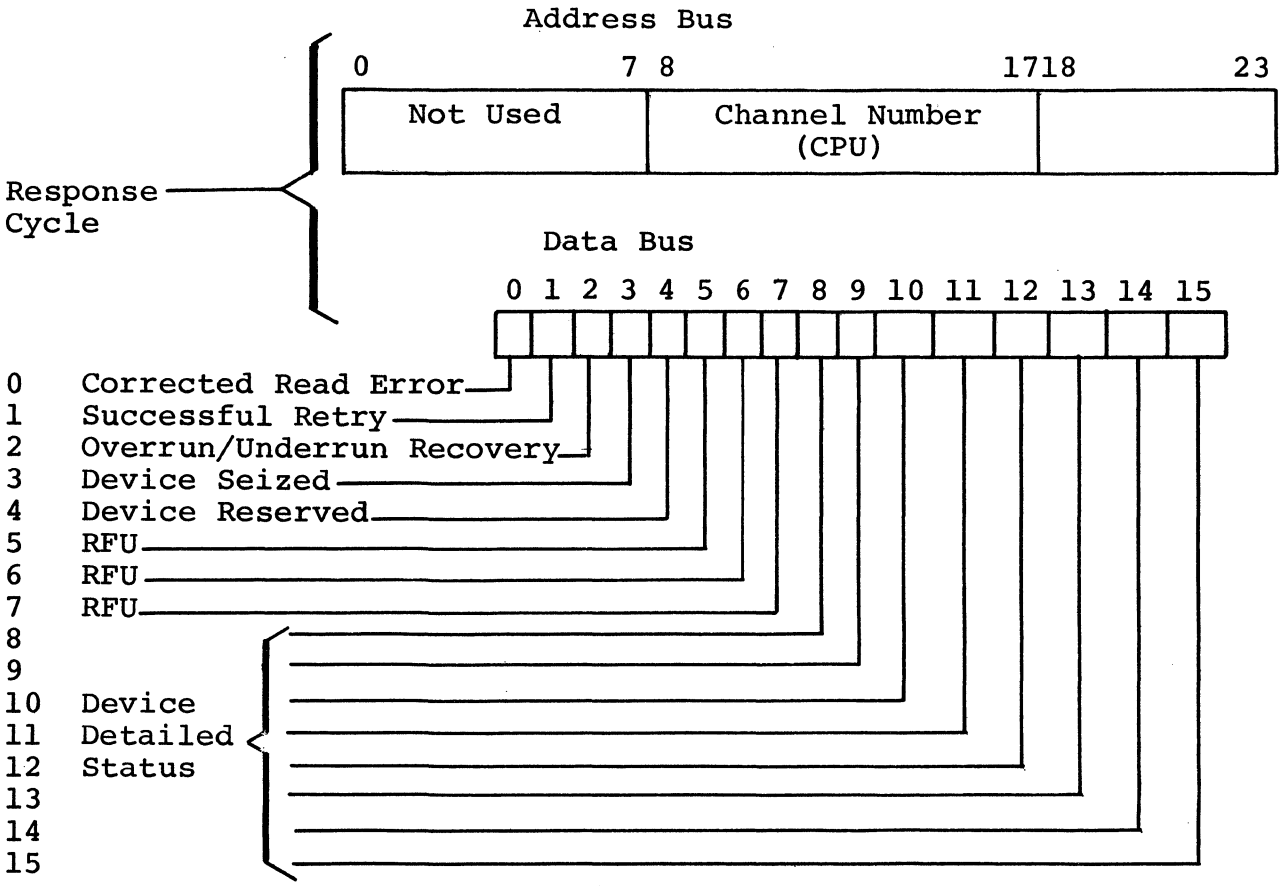


Figure 2-17 Status Word 2 (Sheet 2 of 2)

During the response cycle (second half read), the MPDC returns in bit positions 8 to 23 of the address bus the same information that was received in bit positions 0-15 of the data bus during the request cycle. For a description of the Status Word 2 bit structure, refer to the Type MSC9102 Storage Module Adapter Manual (Document No. 71010429-100, Order No. FN80).

9. Identification Code (FC26) - This instruction (Figure 2-18) causes the referenced device channel to transfer its identification (ID) code to the requesting channel.

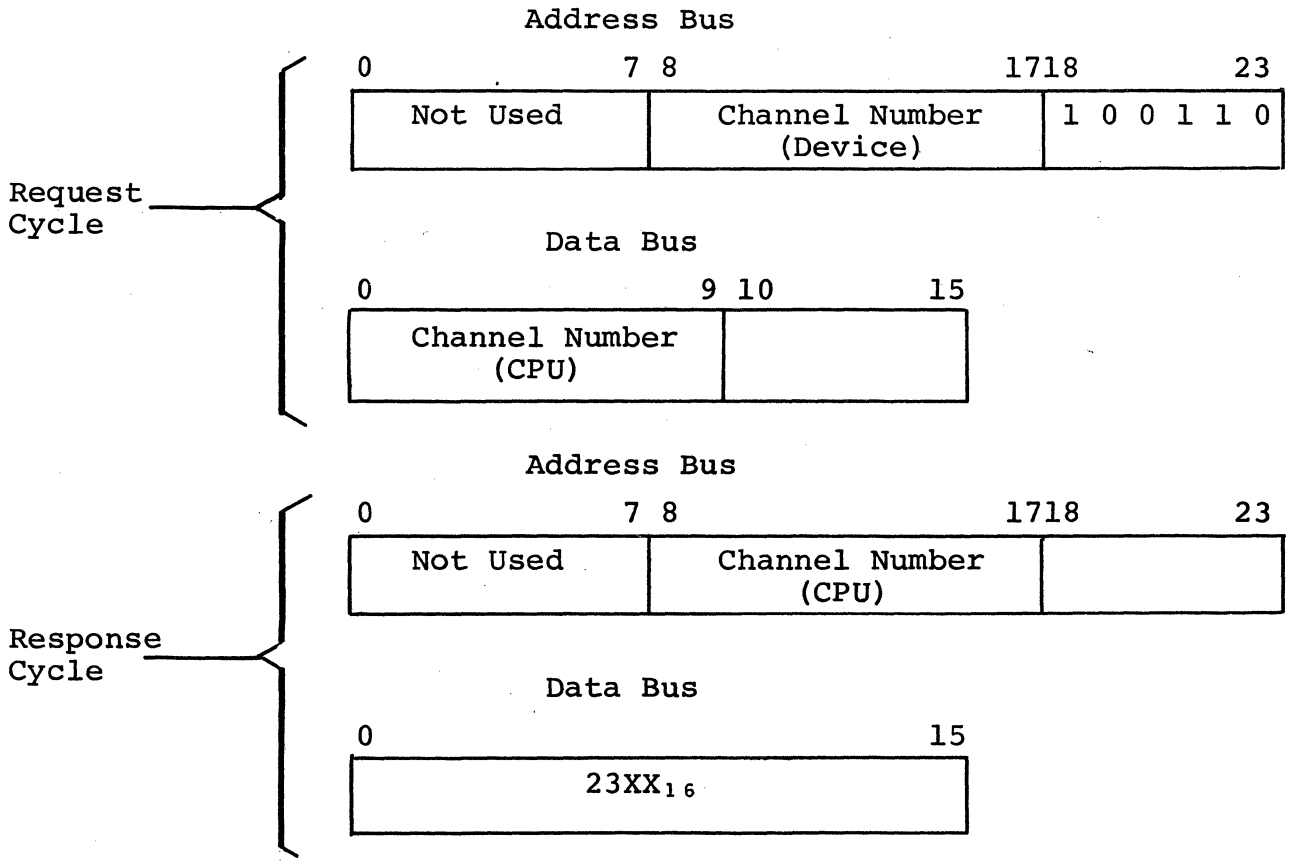


Figure 2-18 Identification Code Format

During the response cycle (second half read), the MPDC returns in bit positions 8 through 23 of the address bus the same data that was received in bit positions 0 through 15 of the data bus during the request cycle.

The MPDC is assigned the ID range 2300_{16} to $23FF_{16}$. Within this range, the following IDs have been assigned to the different device types attached to the MPDC:

DEVICE TYPE	ID CODE (HEX)	MODEL
Cartridge Disk	2330	CDU9101 - 100 TPI, no fixed disk
	2331	CDU9102 - 100 TPI, with fixed disk
	2332	CDU9103 - 200 TPI, no fixed disk
	2333	CDU9104 - 200 TPI, with fixed disk
Storage Module	2360	9760-1 - 40 MBytes
	2361	9762-1 - 80 MBytes
	2362	9764-1 - 150 MBytes
	2363	9766-1 - 300 MBytes

2.1.2.3 Diagnostic Commands

The MPDC maintains 64 scratch pad memory (SPM) locations for each device. Scratch pad memory locations are addressed during read or write commands by combining two bits of the channel number and the high order five bits of the function code, as shown in Figure 2-19.

To perform a diagnostic read command in the scratch pad memory, the user may use any even function code not listed under the input commands.

To perform a diagnostic write command in the scratch pad memory, the user may use any odd function code not listed under the output commands.

Complete software visibility to the SPM is provided for diagnostic purposes. An output bus sequence addressed to one of the devices causes the information on the data bus (16 bits) to be loaded into the device-specific SPM location specified by the device port number and the high order 5 bits of the function code.

Any input bus sequence addressed to a device causes the SPM location specified by the port number and the high order 5 bits of the function code to be returned via the data bus (during the second half read cycle).

The output address command is a special case. When an output address command is executed (on port 0 for example) SPM locations 04 and 05 (hex) are loaded with the low order 16 bits of the address. The high order 8 bits of the address are loaded into SPM location 06.

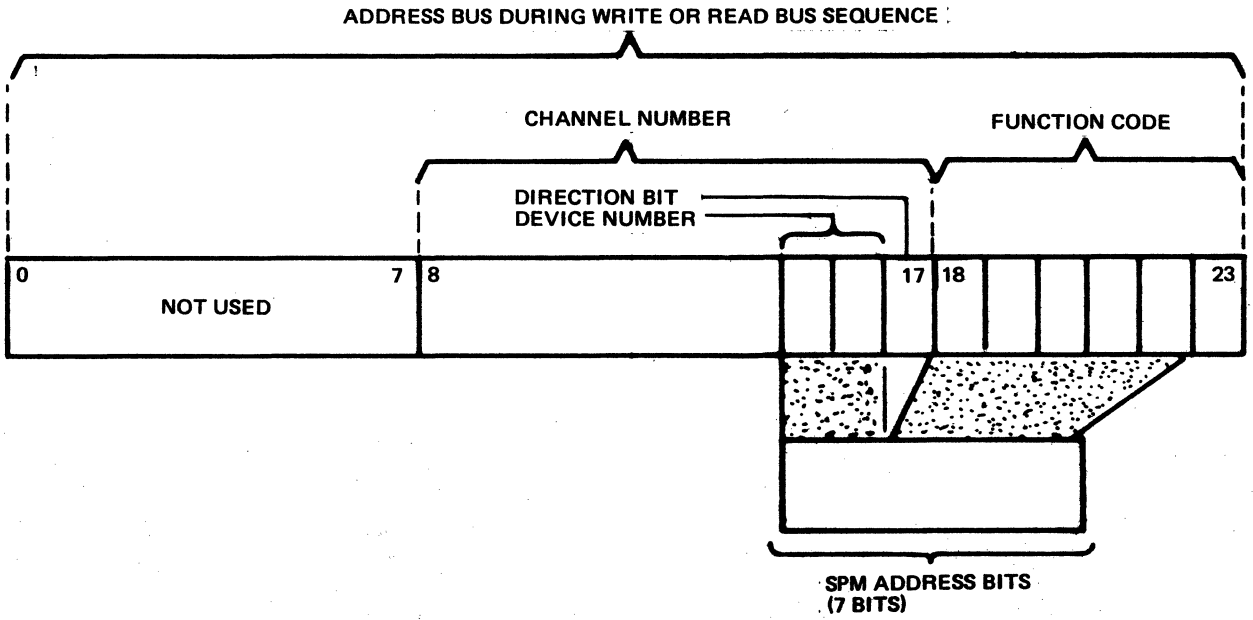


Figure 2-19 MPDC Device-Specific SPM Addressing

2.1.3 Direct Memory Access (DMA) Operation

All data transfers associated with the MPDC subsystem are executed in the DMA mode. Data transfers are normally in word mode, but byte mode transfers can occur. If the main memory buffer begins or ends on an odd byte boundary during the first and/or last memory cycle of a particular data transfer, a byte mode transfer is executed.

Figure 2-20 illustrates the address and data bus configurations for read and write data transfers. During DMA write sequences, the address bus is loaded with the memory address, and the data bus is loaded with the data to be written into memory.

During the request cycle of a memory read sequence, bits 10 through 15 of the data bus may contain the address of a scratch pad memory location in the MPDC into which the returned data (from memory) is to be delivered. During the response (second half read) cycle, memory places on the address bus (bits 8 through 23), the contents of the entire data bus (bits 0 through 15) as received during the request cycle.

If a Negative Acknowledge (NAK) response is received at the MPDC during a memory Write or Read request, the data transfer continues to its normal termination with a nonexistent resource posted in Status Word 1. If a Wait response is received at the MPDC for a memory Write or Read request, the MPDC hardware interface retries the bus cycle. Retries continue until either a NAK or acknowledge (ACK) is received at the MPDC. While a bus cycle is pending in the MPDC hardware interface, no bus cycles from the CPU to the MPDC are accepted (including the Output Control Word). This also applies to the interval between a memory Read request and the Read response (Second Half Read) cycle. Once the MPDC is conditioned to do a memory transfer (either Read or Write), all other bus cycles addressed to the MPDC are responded to with either a NAK or a Wait (depending on channel busy status) until the memory reference is complete. Read response (Second Half Read) cycles from memory to the MPDC are always completed with an Acknowledge (NAK and Wait responses are never used for these cycles).

2.1.4 Interrupt Operation

If the MPDC interrupt level of a channel is zero (either via initialize process or loaded to Zero) no interrupt is attempted for that channel. If a condition or event occurs which would normally cause an interrupt, the appropriate bits in the Status Word are set, but no interrupt is attempted or saved.

If the interrupt level is set to zero when an interrupt is pending (via an Output Control Word or a Master Clear), the pending interrupt is discarded.

Whenever the MPDC interrupt level is nonzero, and an operation initiated by an Output Task Word or Output Control Word command is completed, or the Attention bit is set in the Status Word, an interrupt is attempted. If a negative acknowledge (NAK) response is received during an interrupt cycle, the MPDC stores the interrupt until it detects a pulse on the Resume Interrupt interface line.

The interrupt is then retried. Once an interrupt has been stored, the MPDC is capable of receiving commands and/or conducting data transfers on any of the other channels. The channel with the pending interrupt remains busy, and the MPDC does not accept commands issued to that channel (except Output Control Word).

The channel number supplied on the data bus during an interrupt is the channel number used in the most recent Output Address command to that device. If no Output Address command has occurred at the time of an interrupt, the low order bit of the channel number is a Zero.

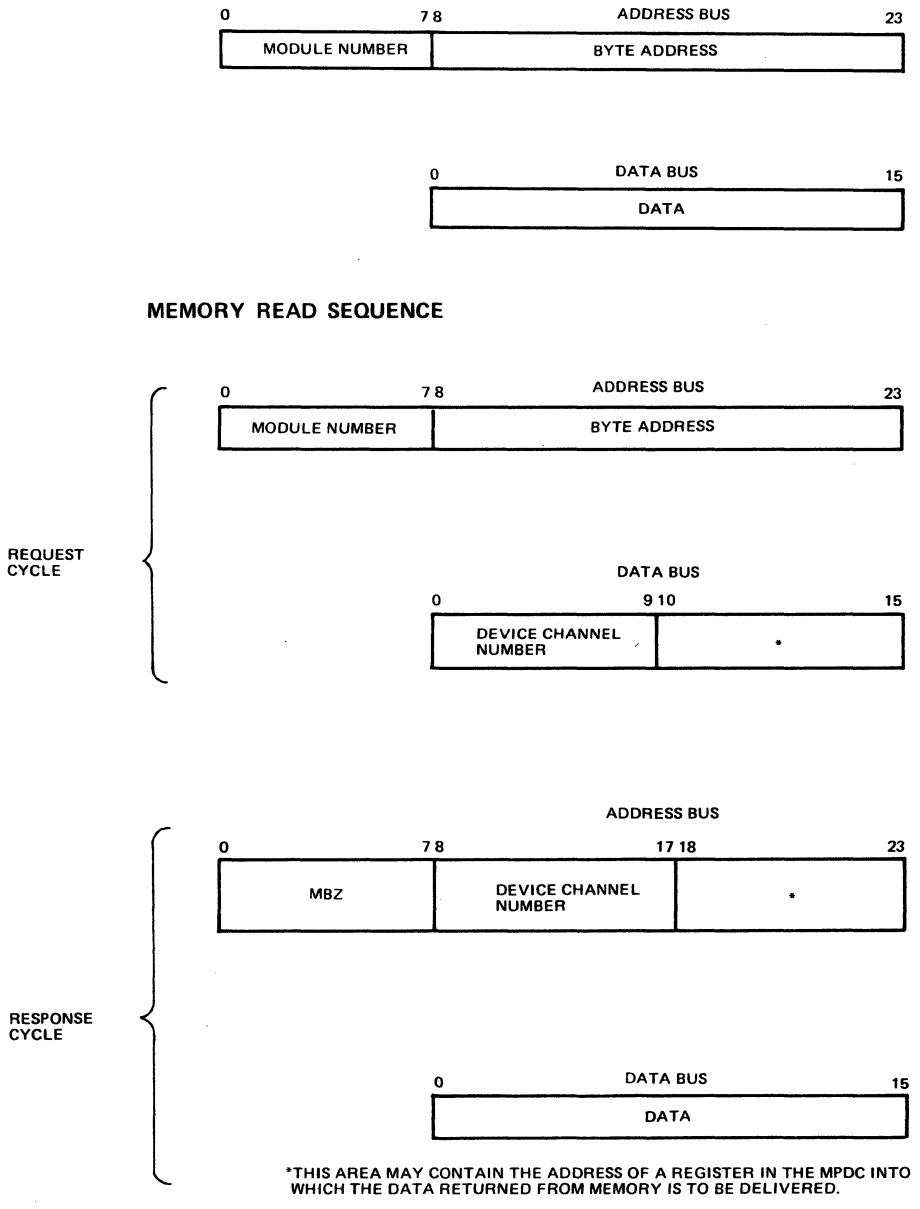


Figure 2-20 Address Bus and Data Bus Configurations for Read and Write Data Transfers

2.1.5 Device Data Access

Associated with each device attached to the MPDC subsystem is a set of registers which are loaded by software and specify the parameters required for an operation. In addition to the range and address registers, there are two configuration registers which contain record location and identification information and a task register which contains operation codes. To perform a specific operation, software first loads the address, range, and appropriate configuration registers (if any). The task register is loaded last and specifies the operation to be performed. The MPDC begins command execution when it receives the task word.

Commands addressed to a nonbusy channel are always accepted, but exeuction may be delayed by a Wait response. All commands addressed to a busy device channel are rejected (NAK response) except for the Output Control Word from software which may receive a Wait response.

For a detailed description of data access to disk devices, refer to the appropriate device adapter manual.

2.2 FIRMWARE

Firmware, a sequence of microinstructions resident in microprogram control stores, is the primary control element of the MPDC/device adapter. The main function of the firmware is to interpret external and internal events or conditions and to react in a prescribed manner (i.e., setting or resetting of hardware functions). Efficient data transfer is also a result of firmware control of hardware components in the data path.

Each time the MPDC is cleared, a series of firmware operations called the Basic Logic Tests is performed. These tests are used to verify the integrity of the MPDC hardware. Upon successful completion of the Basic Logic Tests, the firmware proceeds to set up the hardware for execution of software instructions. This is accomplished by setting up control information in a read/write memory and enabling the adapter. After the adapter is enabled, the firmware enters a routine which waits for a request from the Megabus or from the adapter. When a request occurs, firmware analyzes status and priority before proceeding to the appropriate operation for processing the request.

The firmware load (in the Read Only Store Adapter) is peculiar to the type of device adapter attached to the MPDC. Therefore, the theory of operation for the MPDC/device adapter firmware is described in Section III of each device adapter manual. The firmware commands, scratch pad memory topology, and the cycle flow of the firmware routines are described at an intermediate level.

2.3 HARDWARE

The MPDC hardware logic (see Figure 2-21) is organized into nine areas: interface, Megabus logic, address control, data logic unit, scratch pad memory, and the range and offset range counters. An overview of each of these major areas is presented in this section (for detailed information refer to Section III).

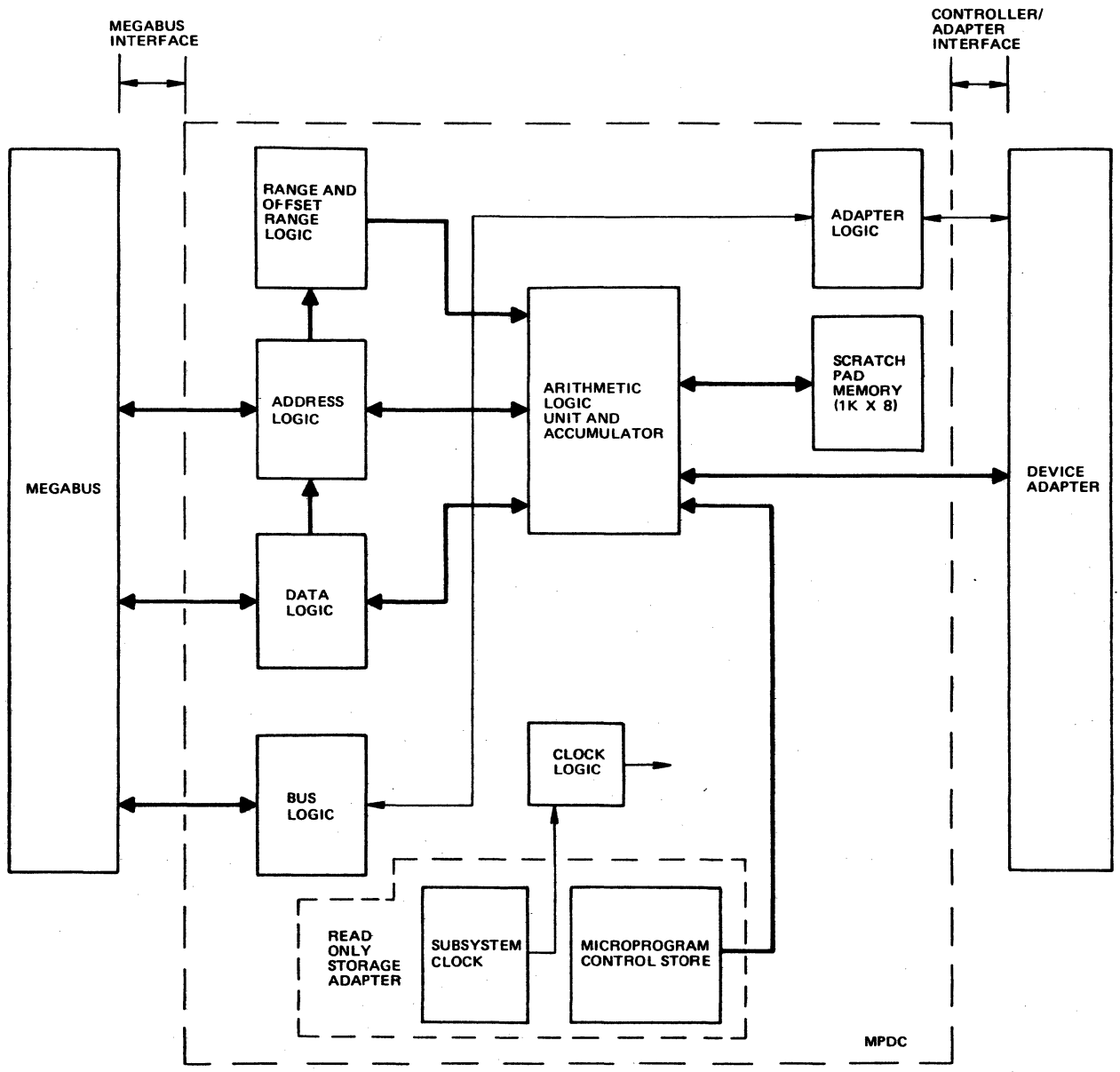


Figure 2-21 MPDC Major Block Diagram

2.3.1 Interface Description

Four interface networks are associated with the MPDC, three of which are described in this manual: Megabus, Read Only Storage, and MPDC/device adapter. The fourth interface network, the device level interface, is described in the appropriate device adapter manual.

2.3.1.1 Megabus/MPDC

The interface network between the Megabus and the MPDC (shown in Figure 2-22) provides a communications path for control, address, and data transfers between the MPDC and other controllers connected to the Megabus. This interface also supplies the paths for determining priority of a request from any controller attached to the Megabus. The name, mnemonic, and a description of each signal associated with the Megabus/MPDC interface are provided in Table 2-3.

2.3.1.2 MPDC/Device Adapter

The interface network between the MPDC and the device adapter provides a communications path for control, timing, and data transfers between these two subsystems. Figure 2-23 depicts the MPDC/device adapter interface lines and indicates the direction of signal transfer across these lines. Table 2-4 lists the MPDC/device adapter interface lines and provides the name, mnemonic, and a description of each line.

Across thes interface firmware may select and perform designated operations in the device adapter and service request and data may be sent from the device adapter to the MPDC.

2.3.1.3 MPDC/Read Only Storage (ROS) Adapter

The MPDC/ROS adapter interface provides a path for communication between the MPDC and the control store area of the ROS adapter. The control store may be accessed to execute firmware-controlled device routines and to set up the MPDC for information transfers. The signals that may be monitored across this interface are shown in Figure 2-24 and defined in Table 2-5.

2.3.2 Megabus Logic

The Megabus logic contains the circuitry involved in generating and accepting Megabus cycles.

The MPDC interfaces with the Megabus via transceivers which provide the electrical characteristics required of all Megabus connections, allowing data, address, and control signals to be routed to and from the MPDC. However, for certain control responses and tiebreaking signals, separate circuits (standard TTL) are used to interface the MPDC with the Megabus.

The MPDC, as with all controllers that interface with the Megabus, requires a tiebreaking network. This network resolves simultaneous requests and grants Megabus cycles to controllers on a positional priority basis. In the Series 60 Level 6 system the main memory and the central processor reside at opposite ends of the

Megabus, with memory reflecting the highest priority and the CP the lowest. Under these conditions, the MPDC and other controllers occupy intermediate priority positions with the priority increasing toward the memory end of the Megabus.

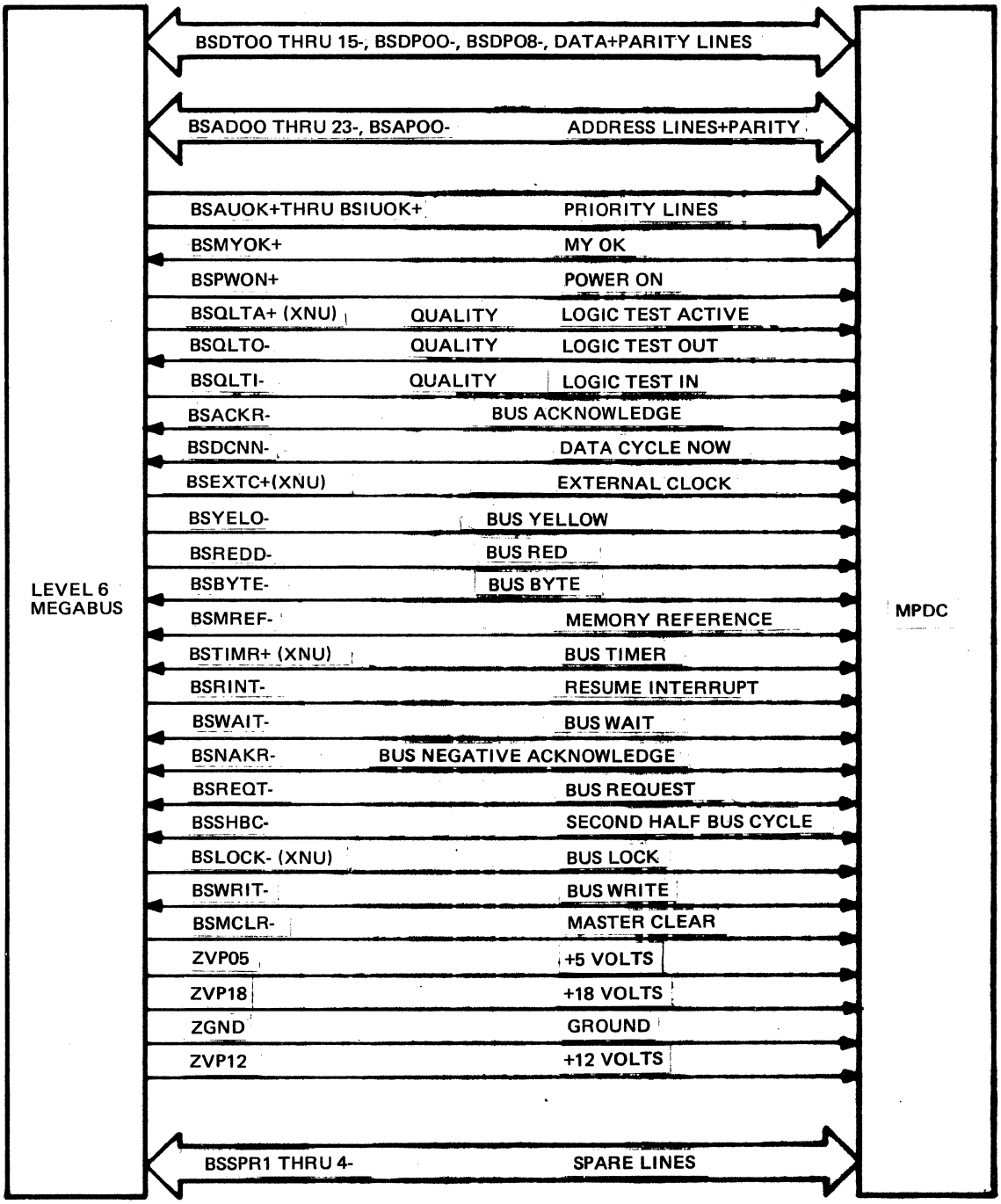


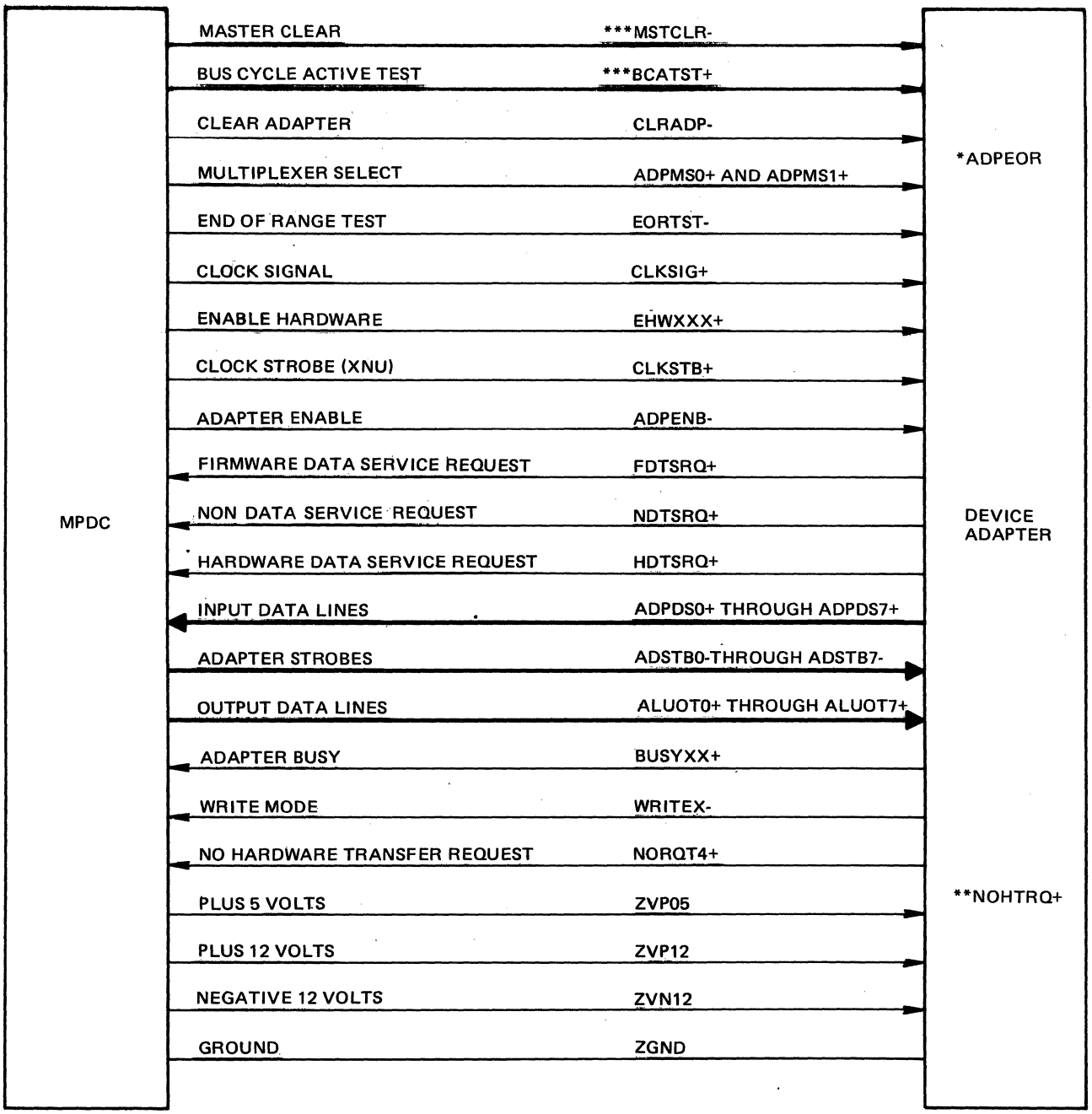
Figure 2-22 Megabus/MPDC Interface

Table 2-3 Megabus/MPDC Interface Signal Lines
(Sheet 1 of 2)

TERM/MNEMONIC	DESCRIPTION
Data Bits 0 through 7 (BSDT00 through BSDT07)	These lines represent the most significant byte of data.
Data Bits 8 through 15 (BSDT08 through BSDT15)	These lines represent the least significant byte of data.
Data Parity-Left Byte (BSDP00)	This signal contains odd parity for data bits 0 through 7.
Data Parity-Right Byte (BSDP08)	This signal contains odd parity for data bits 8 through 15.
Address Bus Bits 0 through 23 (BSAD00 through BSAD23)	These lines contain an address to be used by memory or by a controller or central processor.
Address Parity (BSAP00)	This signal contains odd parity for the most significant byte of the address bus, bits 0 through 7.
Priority Lines (BSAUOK through BSIUOK)	These lines are used to establish priority of the units attached to the bus.
MY OK (BSMYOK)	This signal indicates the unit that is presently using the bus.
Power On (BSPWON)	This signal is true when all power supplies in the system are operating correctly.
Logic Test In (BSQLTI)	This signal initiates the Internal Logic Test in a unit attached to the bus.
Logic Test Out (BSQLTO)	This signal indicates the unit has successfully completed running its Internal Logic Test and is used as the logic test in signal for the next unit attached to the bus.
Acknowledge (BSACKR)	This signal indicates that the information on the bus has been accepted.
Data Cycle Now (BSDCNN)	This signal indicates that the information on the bus is valid.
Yellow (BSYELO)	This signal indicates that the accompanying transferred information is correct, but that a correction operation was performed.
Red (BSREDD)	This signal indicates that the accompanying transferred information is in error.

Table 2-3 Megabus/MPDC Interface Signal Lines
(Sheet 2 of 2)

TERM/MNEMONIC	DESCRIPTION
Byte (BSBYTE)	This signal indicates that the current transfer is a byte transfer rather than a word transfer.
Memory Reference (BSMREF)	This signal indicates that the address leads contain a memory address.
Wait (BSWAIT)	This signal indicates that the transfer will be accepted when the bus data register is available.
Negative Acknowledge (BSNAKR)	This signal indicates that the information on the bus has been refused.
Bus Request (BSREQT)	This signal indicates that one or more units on the bus have requested a bus cycle.
Second Half Bus Cycle (BSSHBC)	This signal identifies the second bus cycle in response to a memory read request.
Bus Write (BSWRIT)	This signal indicates that information on the bus is ready to be transferred.
Master Clear (BSMCLR)	This signal initializes the units attached to the bus.
Resume Interrupt (BSRINT)	This signal is a 200-nanosecond pulse which is issued by the central processor when it is capable of receiving interrupts again.



* THE MNEMONIC FOR THIS LINE CHANGES AT THE DEVICE ADAPTER CONNECTOR FOR A CARTRIDGE DISK SUBSYSTEM

** THE MNEMONIC FOR THIS LINE CHANGES AT THE DEVICE ADAPTER CONNECTOR FOR A CARTRIDGE DISK OR A STORAGE MODULE SUBSYSTEM

*** USED IN STORAGE MODULE SUBSYSTEM ONLY

Figure 2-23 MPDC/Device Adapter Interface

Table 2-4 MPDC/Device Adapter Interface Lines (Sheet 1 of 2)

NAME	MNEMONIC	DESCRIPTION
Clear Adapter	CLRADP-	This signal is transferred to the device adapter whenever a Master Clear signal is received by the MPDC or when a Reset Device Adapter operation is issued from firmware.
Multiplexer Select	ADPMS0+, ADPMS1+	These lines determine the type of information (Device Data, Device ID, Status 1 or 2) to be transferred from the device adapter to the MPDC by selecting one of four registers in the device adapter.
End of Range Test	EORTST-	This signal notifies the device adapter that the range has been exhausted.
Clock Signal	CLKSIG+	This signal is used as a test clock in a storage module subsystem only.
Enable Hardware	EHWXXX+	This signal enables the write or read path between the MPDC and the device adapter.
Clock Strobe	CLKSTB+	This signal is not used by the adapter.
Adapter Enable	ADPENB-	This signal enables the device adapter input multiplexer to transfer information to the MPDC.
Firmware Data Service Request	FDTSRQ+	This signal notifies the MPDC that the adapter has reached the end of its ID or data field transfer.
Nondata Service Request	NDTSRQ+	This signal notifies the MPDC of any catastrophic type errors.
Hardware Data Service Request	HDTSRQ+	This signal indicates to the MPDC that the adapter has a data byte ready to be taken.
Input Data Lines	ADPDS0-7	These lines provide a path for the information transfer from the device adapter to the MPDC.

Table 2-4 MPDC/Device Adapter Interface Lines (Sheet 2 of 2)

NAME	MNEMONIC	DESCRIPTION
Adapter Strobes	ADSTB0-7	The strobe signals enable logic areas of the adapter for the transfer of data or control information from the MPDC.
Output Data Lines	ALUOT0-7	These lines handle the information transfer from the MPDC to the device adapter.
Adapter Busy	BUSYXX+	Indicates to the MPDC that the device adapter is busy.
Write Mode	WRITEX-	This signal indicates to the MPDC that the device adapter is in the write mode and is used in the MPDC to check for end of range.
No Hardware Transfer Request	NORQT4+	This signal notifies the MPDC that the device adapter cannot handle a hardware transfer at this time.
Plus 5 Volts	ZVP05	The +5 volt lines supply the device adapter with the voltage level needed for adapter operation.
Plus 12 Volts	ZVP12	This voltage is available at the device adapter but is not used.
Ground	ZGND	The ground lines provide the device adapter with a system ground level.
Master Clear*	MSTCLR-	This signal is enabled whenever the MASTER CLEAR button on the system control panel is depressed or when a power fail sequence occurs.
Bus Cycle Active Test*	BCATST+	This signal notifies the device adapter that the MPDC is executing a bus cycle or that the MPDC has issued an Acknowledge response.
Negative 12 Volts	ZVN12	This voltage is not used in a Cartridge Disk subsystem. In a Storage Module subsystem this voltage is used to generate a Negative 5 volts for the device drivers/receivers.

*Used in Storage Module subsystem only.

Table 2-4 MPDC/Device Adapter Interface Lines (Sheet 1 of 2)

NAME	MNEMONIC	DESCRIPTION
Clear Adapter	CLRADP-	This signal is transferred to the device adapter whenever a Master Clear signal is received by the MPDC or when a Reset Device Adapter operation is issued from firmware.
Multiplexer Select	ADPMS0+, ADPMS1+	These lines determine the type of information (Device Data, Device ID, Status 1 or 2) to be transferred from the device adapter to the MPDC by selecting one of four registers in the device adapter.
End of Range Test	EORTST-	This signal notifies the device adapter that the range has been exhausted.
Clock Signal	CLKSIG+	This signal is used as a test clock in a storage module subsystem only.
Enable Hardware	EHWXXX+	This signal enables the write or read path between the MPDC and the device adapter.
Clock Strobe	CLKSTB+	This signal is not used by the adapter.
Adapter Enable	ADPENB-	This signal enables the device adapter input multiplexer to transfer information to the MPDC.
Firmware Data Service Request	FDTSRQ+	This signal notifies the MPDC that the adapter has reached the end of its ID or data field transfer.
Nondata Service Request	NDTSRQ+	This signal notifies the MPDC of any catastrophic type errors.
Hardware Data Service Request	HDTSRQ+	This signal indicates to the MPDC that the adapter has a data byte ready to be taken.
Input Data Lines	ADPDS0-7	These lines provide a path for the information transfer from the device adapter to the MPDC.

Table 2-4 MPDC/Device Adapter Interface Lines (Sheet 2 of 2)

NAME	MNEMONIC	DESCRIPTION
Adapter Strobes	ADSTB0-7	The strobe signals enable logic areas of the adapter for the transfer of data or control information from the MPDC.
Output Data Lines	ALUOT0-7	These lines handle the information transfer from the MPDC to the device adapter.
Adapter Busy	BUSYXX+	Indicates to the MPDC that the device adapter is busy.
Write Mode	WRITEX-	This signal indicates to the MPDC that the device adapter is in the write mode and is used in the MPDC to check for end of range.
No Hardware Transfer Request	NORQT4+	This signal notifies the MPDC that the device adapter cannot handle a hardware transfer at this time.
Plus 5 Volts	ZVP05	The +5 volt lines supply the device adapter with the voltage level needed for adapter operation.
Plus 12 Volts	ZVP12	This voltage is available at the device adapter but is not used.
Ground	ZGND	The ground lines provide the device adapter with a system ground level.
Master Clear*	MSTCLR-	This signal is enabled whenever the MASTER CLEAR button on the system control panel is depressed or when a power fail sequence occurs.
Bus Cycle Active Test*	BCATST+	This signal notifies the device adapter that the MPDC is executing a bus cycle or that the MPDC has issued an Acknowledge response.
Negative 12 Volts	ZVN12	This voltage is not used in a Cartridge Disk subsystem. In a Storage Module subsystem this voltage is used to generate a Negative 5 volts for the device drivers/receivers.

*Used in Storage Module subsystem only.

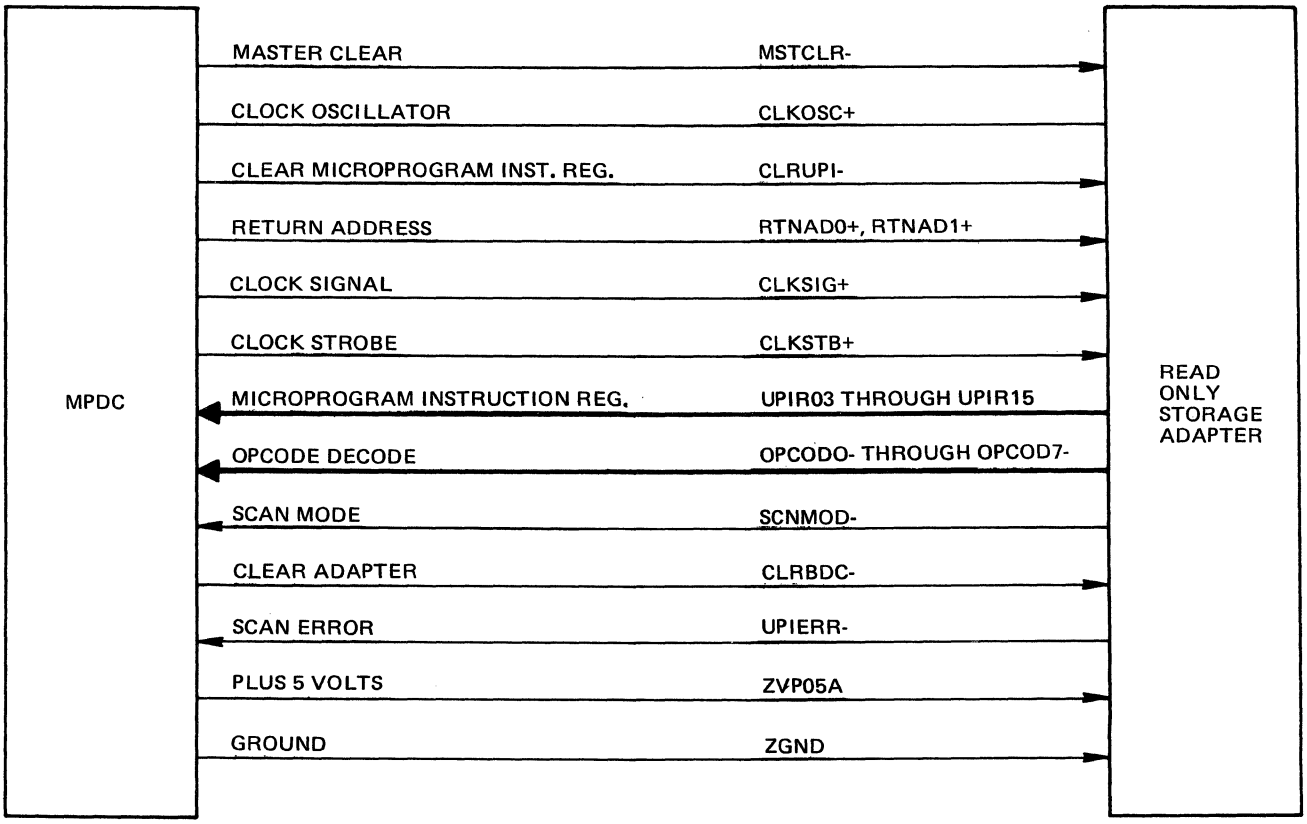


Figure 2-24 MPDC/Read Only Storage Adapter Interface

Table 2-5 MPDC/ROS Adapter Interface Lines (Sheet 1 of 2)

NAME	MNEMONIC	DESCRIPTION
Master Clear	MSTCLR-	This signal is enabled whenever the MASTER CLEAR button on the system control panel is depressed or when a power fail sequence occurs.
Clock Oscillator	CLKOSC+	This signal is generated from an 8 MHz crystall oscillator located on the ROS adapter and is sent to the MPDC, where the subsystem clock signals are derived and distributed.
Clear Microprogram Instruction Register	CLRUPI-	This signal clears the MPIR to Zeros and disables the op-code decoder.
Return Address	RTNAD0+, RTNAD1+	These signals are sent from the MPDC to the ROS adapter and define the address in the subroutine return address register to be read or written.
Clock Signal	CLKSIG+	This signal provides the clock pulses needed in the ROS adapter to drive the microprogram address counter, microprogram index register, and the scan error logic.
Clock Strobe	CLKSTB+	This signal provides the strobe which enables information to be written into the subroutine return address register.
Microprogram Instruction Register	UPIR03 through UPIR15	The MPIR signals provide firmware controller microprogram instructions to the MPDC and device adapters.
Op Code Decoder	OPCOD0 through OPCOD7	The op-code decoder signals supply the MPDC with the type of command that is to be executed.
Scan Mode	SCNMOD-	This signal notifies the MPDC that the ROS adapter is presently performing a scan operation.

Table 2-5 MPDC/ROS Adapter Interface Lines (Sheet 2 of 2)

NAME	MNEMONIC	DESCRIPTION
Clear Adapter	CLRBDC-	This signal clears the ROS adapter microprogram address counter and the scan bit address counter to Zeros.
Scan Error	UPIERR-	This signal notifies the MPDC that during scan mode an error was detected in the microprogram instruction register. This signal causes the MPDC clock to halt.
Plus 5 Volts	ZVP05	The +5 volt lines provide the ROS adapter with the voltage level needed for ROS adapter operation.
Ground	ZGND	The ground lines provide the ROS adapter with a system ground level.

2.3.3 Address Logic

The address logic handles the information transferred to or from the MPDC across the address bus. During I/O operations (except I/O Load), the information consists of the channel number of the slave unit and the function code of the operation. Bits 0 through 7 of

the address bus are not used for these operations. During an I/O Load operation, bits 0 through 7 of the address bus contain the identifier of the memory controller where I/O transfers are to begin; the remaining bits of the address bus contain the same information as any other I/O command.

For direct memory access (DMA) read or write commands from the MPDC, the address bus contains the 24-bit address of the memory location to be accessed. During a read response, the memory delivers the MPDC channel number across the address bus.

The major element of the address logic is the transceivers. They receive information from and send information to the Megabus. The address transceivers contain a path for the bidirectional flow of information to/from the arithmetic logic unit (ALU), and a unidirectional path to the scratch pad memory (SPM) for the storage of instruction information.

2.3.4 Data Logic

The data logic consists of the hardware elements that influence the transfer of information across the data bus. Information is transferred across the Megabus to/from the data transceivers.

For memory read requests, I/O input commands, and interrupts, the data bus contains the channel number of the MPDC. During the response cycle of memory read or I/O input commands, the data bus contains the data that the MPDC requested. The I/O Load command uses the data bus to send address and range information to the MPDC. For all other instructions, the data bus contains data.

The data transceivers contain a bidirectional path to the ALU for the transmission or reception of data. They also have a path to the address transceivers to deliver the channel number of the requesting source back to the source during response cycles.

The information transferred to the MPDC across the data bus is stored in a first-in-first-out (FIFO) buffer. The FIFO is 16 bits wide and provides parallel-in parallel-out operation. When requested by firmware, the FIFOs transfer either the left or right half of the data input to the ALU.

2.3.5 Subsystem Clock

The MPDC utilizes an 8-MHz oscillator, located on the read only store (ROS) adapter, to derive 4-MHz clock cycles for the subsystem. The clock signals are distributed to the control areas of the MDPC and the device adapter. A delay line in the clock circuitry provides the controller with the ability to generate critical timing signals which ensure error-free operation.

2.3.6 Read Only Store Adapter (Microprogram Control Store)

The microprogram control store (MPCS) provides permanent storage for resident control firmware and diagnostic microprograms. The storage capacity of the MPCS is 4096 locations X 16 bits wide. The output of the microprogram control store is derived from 16 pro-

grammable read only memory (PROM) chips that are 1024 X 4 bits wide. The control store has the ability to vary address sequencing to execute routines, depending on priority information, test conditions, or channel activity.

The microprogram control store logic is located on the read only store (ROS) adapter, and the outputs are distributed to control areas of the MPDC and the device adapter.

2.3.7 Scratch Pad Memory (SPM)

The scratch pad memory is a 1K-by-8-bit-wide read/write memory which stores information that is required by or generated by each channel (i.e., data, status, commands, etc.). The memory is segmented into two sections, indexed and nonindexed, each containing two quadrants.

The nonindexed section of memory (512 locations) uses 256 locations as work locations, and the other 256 locations are not used.

The indexed section of memory (512 locations) uses 256 locations for the storage of device specific information, and the other 256 locations are not used. The 256 locations for device-specific information are subdivided into four sections (64 locations per channel).

The SPM is addressed via 10 bits: the high order bit selects indexed or nonindexed mode, the second high order bit selects a 256-location quadrant, the next two bits select 64 locations within the quadrant, and the six low order bits select the address.

Data from the A-operand (AOP) wired-OR gate is written into the SPM during a firmware Memory Write command at the location specified by the address bits. The data out of the SPM is delivered to the AOP and B operand (BOP) multiplexers, and is transferred to the arithmetic logic unit (ALU) when selected.

2.3.8 Arithmetic Logic Unit

The arithmetic logic unit (ALU) performs eight-bit arithmetic and logic operations on the information located at its input gates and determines the destination of the result as a function of firmware. The ALU is considered the focal point of all data transfer operations in the MPDC because information pertaining to any data transfer must pass through the ALU before being delivered to its destination.

Two multiplexers (AOP and BOP) provide the input to the ALU and allow various combinations of registers to be used as operands within the ALU. Firmware may use the ALU to perform word mode operations by generating a carry-in equal to the carry-out of the previous ALU command. The ALU may perform bit operations via a data constant supplied by firmware. The constant may be loaded into the ALU via the BOP multiplexer. Status is generated as a result of an ALU operation and is stored in status flops for interrogation. The information remains in the status flops until the initiation of a subsequent ALU firmware command.

Considered as part of the ALU circuitry is the accumulator (ACU). The ACU is an eight-bit register used for temporary storage of the ALU outputs. An additional feature of the ACU is that its contents can be left shifted using the ALU.

2.3.9 Range and Offset Range Logic

This logic controls the number of bytes to be transferred during a data operation. The range logic is used during input or output operations and the offset range logic is restricted to input operations.

An Output Range instruction loads the range register associated with the referenced channel with a 16-bit quantity, identifying the number of bytes to be transferred during the data transfer that is being set up.

The Input Range instruction causes the contents of the referenced channel's range register to be transferred to the requesting channel.

The Output Offset Range instruction loads the offset range register associated with the referenced channel with a 16-bit quantity, identifying the number of bytes to be discarded from the beginning of the data transfer prior to the transfer of any data to the main memory.

The Output Offset Range register must be set for each data transfer requiring an offset. Any offset range residue is applied to the next command unless reset by another offset range instruction.

2.3.10 Adapter/Channel Control Logic

A channel consists of the MPDC, device adapter, and device. Complete channel control results from a combination of hardware and firmware.

Each channel can provide three types of requests: bus, data, or nondata. These requests are supplied to a channel request encoder where the presence of a data request signifies that a data byte read from the device is available or the need for a data byte to be transferred to the device exists. The nondata request indicates a change of device state or the detection of a device condition which requires attention.

The request encoder indicates when a channel request is active. It also sets up the priorities such as any Megabus request having priority over a channel data request. A similar request from channel 1 would have higher priority than that from any of the higher channel numbers.

Firmware can test the output of the priority encoder to determine if a request, or what type of request, has occurred. It also can test the encoder to ascertain if the request with the highest priority is a Megabus request or a channel request.

2.4 OPERATIONAL STATES AND MODES

2.4.1 Operational Overview

The primary functions of the MPDC are to execute Megabus sequences, provide command decoding, provide a storage area for status and control information, and control the direction of the general flow of command execution. The MPDC also controls the transfer of data between the main memory (visible at the Megabus) and the mass storage device. This data may be viewed as a sequential data stream with the MPDC providing the necessary control and transformation of that data into and out of the data formats of the connected mass storage devices.

Data transfers associated with the MPDC subsystem are executed in direct memory access (DMA) mode. The MPDC uses a combination of microprogram control (firmware), software-generated commands which rpovide address, range, configuration, and task information, and hardware logic to produce the appropriate action for each data element. These actions may consist of a data transfer, the delection or addition of data elements, or the recognition of certain data elements or sequence of data elements as special conditions.

At the Megabus side of the MPDC, software-generated I/O commands determine operations to be performed by the MPDC subsystem. The MPDC supports these I/O commands via the address bus and data bus and takes the required action necessary to support the commands.

At the device side of the MPDC, the device adapter provides both the device level interface (DLI) and parallel/serial conversion of the byte data stream from the Megabus into bit serial stream form for the mass storage device and vice versa.

As data passes through the MPDC from Megabus to device adapter or vice versa, the MPDC exercises its control over the contents and format of the data stream, generating appropriate status, interrupts, and control information.

Address, range, configuration, and task information generated by the CPU software is stored in the MPDC scratch pad memory. This information is used for reference during MPDC command execution and is up-dated when instructed by firmware.

2.4.2 Operational States

The MPDC initialized state can be entered by depression of the system control panel MASTER CLEAR pushbutton, as a result of applying system power, or by a programmed master clear command. Initialization clears the MPDC scratch pad memory and resets the MPDC/device adapter control logic. Whenever the MPDC is initialized, its basic logic test (quality logic test) is executed which functionally verifies the operational capabilities of the major logic components of the MPDC subsystem. Upon completion of the basic logic test, firmware enters the Clear SPM routine, which clears the SPM to Zeros. Firmware then enters the Unit Select routine, which loads the channel number for each device into the SPM. From this point, firmware enters the Setup routine, which loads the four segments of the SPM with device-specific information (provided that four devices

are attached to the subsystem). After the Setup routine is completed, the basic firmware used for verification and selection is completed and is not returned to unless an initialize (software or hardware) is received.

When the basic firmware is completed, the MPDC enters the Wait routine, which is considered to be the initial MPDC firmware routine. This routine is the idle loop in which the MPDC waits for a request from hardware or software. In the idle loop, firmware polls devices, looking for requests or changes in the states of the devices. If a Bus Request is detected, firmware enters the Bus Request routine; if a specific operation is detected, firmware enters the appropriate device support routine.

2.4.3 Operational Modes

The MPDC subsystem operates in either of two modes, input or output.

2.4.3.1 Input Mode

Incoming bit serial data from a mass storage device is converted by the device adapter into data elements of eight bits for transfer to the MPDC. Each time a data element is accumulated in the device adapter, a Channel Request Interrupt is issued to the MPDC. This informs the MPDC that the device adapter has a data byte ready for transfer to the MPDC.

When the MPDC detects a Channel Request Interrupt, it enables its hardware logic and firmware transfer routines. The MPDC then transfers the data bytes through the arithmetic logic unit to the Megabus as directed by software. Data transfers between the MPDC and the Megabus (to or from memory) are in word form except for possibly the first or last transfer, which may be in byte form. The length of the data transfer depends on the range that was established in the CPU software command.

2.4.3.2 Output Mode

Data transfers across the Megabus that are directed to a non-busy mass storage device are under the control of CPU software. The MPDC must be set up for the appropriate address, range, configuration, and task word. When the task word is detected by the MPDC, command execution begins.

For example, to Write Data on a mass storage device, information is read from the main memory and loaded into the data bus of the MPDC. Within the MPDC, the data is transferred to the first-in-first-out (FIFO) data buffers, where a maximum of 32 data bytes can be stored. When instructed by firmware, the parallel-output of the FIFO transfers a byte of data to the arithmetic logic unit (ALU). From the ALU, firmware-generated adapter strobes control the transfer of information from the MPDC to the device adapter. Data inputted to the device adapter is stored in the adapter FIFOs until it is transferred to the device. Several predefined fields are required to set up a transfer between the device adapter and the device before data can be sent to the device. Transfer of information according to the track format is essential for proper data storage.

III
THEORY OF OPERATION-INTERMEDIATE

The medium performance disk controller (MPDC) hardware is described in this section at an intermediate level. Figure 3-1 depicts the major logic elements (registers, counters, etc.) of the MPDC and shows how they are grouped together and interconnected to accomplish specific tasks.

3.1 DATA FLOW AND CONTROL PATHS

The data flow path between the Megabus and the MPDC is bi-directional and may contain either data or control information, depending on the type of command issued.

Information entering the MPDC from the Megabus (via an Output command) is transferred through the registers, buffers, and multiplexers in the MPDC, under firmware control, to the device adapter and finally to the device. The reverse procedure occurs when an Input command is issued.

The MPDC data paths are set up by the firmware routines which are resident in the ROS adapter control store area. The selected MPDC multiplexers determine the data or control path to be taken, and are under control of certain bits of the control store word.

The remainder of the information in this section describes how data and control transfers are set up and executed.

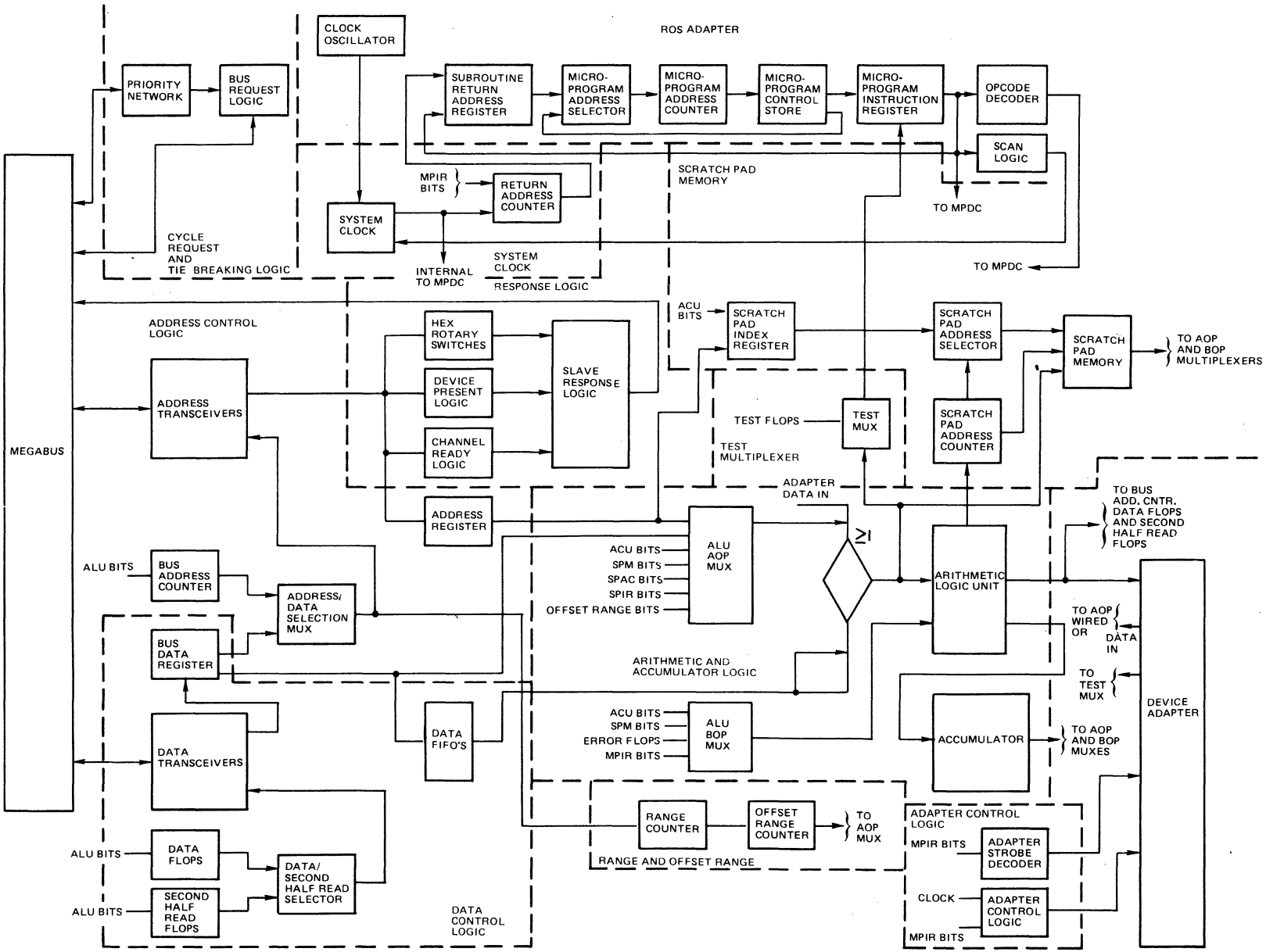


Figure 3-1 MPDC Intermediate Block Diagram

HONEYWELL PROPRIETARY AND CONFIDENTIAL

3.2 MEGABUS TIMING AND CONTROL

The Megabus timing and control logic consists of the cycle request logic, the tiebreaking network, and the slave response logic. In essence, the interfacing required between two units to set up a control or data transfer is contained in the Megabus timing and control logic. The basic functions of this logic are:

1. Compare the address on the Megabus with the state of the MPDC channel address switches to ascertain whether or not the MPDC is being addressed.
2. Monitor the MPDC to ascertain whether or not it is busy.
3. Determine whether the MPDC is to be acting as the master or the slave controller.
4. Generate an Acknowledge (ACK), Negative Acknowledge (NAK), or a Wait (WAIT) signal to reflect the condition of the MPDC).
5. Analyze the function code on the Megabus address lines to ascertain the function to be performed.
6. Analyze the port number on the Megabus address lines and manipulate, via the scratch pad memory (SPM) and counters, the addresses of device specific information located in the SPM.
7. Transfer address, range, configuration, and task information from the Megabus address and data lines into the appropriate areas of the SPM.

3.2.1 Cycle Request and Tiebreaking Network

Communication with the MPDC subsystem is handled through the Megabus network, which performs asynchronous handshaking operations as shown in Figure 3-2. A controller requesting a Megabus cycle sets Bus Request (BSREQT-) true. The BSREQT- signal is common to all controllers on the Megabus; therefore, a tiebreaking function must be completed to determine which controller has priority, if more than one controller has enabled its BSREQT- line. When the tiebreaking function is completed, and one controller has been granted a Megabus cycle, the controller acting as the master sets its Data Cycle Now (BSDCNN-) signal true and places the information to be transferred on the Megabus. Each controller develops an internal strobe from the BSDCNN- signal, and when the strobe delay is completed, the slave controller recognizes its channel number, or in the case of memory, its memory address. If the information on the Megabus address lines contains the MPDC channel number, the MPDC may respond in one of three ways: Bus Acknowledge (BSACKR-), Bus Negative Acknowledge (BSNAKR-), or Bus Wait (BSWAIT-). After the MPDC responds, the control lines return to the false state in the sequence shown in Figure 3-2.

The Megabus timing is fully asynchronous; each transition occurring only when the preceding transition has been received. The hardware logic utilized to generate the timing functions which provide the Megabus handshaking sequence is shown in Figure 3-3.

When the MPDC requests a Megabus cycle (see Figure 3-3), the Firmware Cycle command is executed, and the cycle parameters (see Table 3-1) are set according to the output of the accumulator. The firmware cycle requests (CYCREQ+) flip-flop sets, initiating a Megabus cycle. When the CYCRQT+ flip-flop sets and the Megabus is not busy (BSBUSY-), the set request (SETRQT) gate is enabled, forcing the my request (MYREQT+0 flip-flop to set. The MYREQT+ signal is gated to the MPDC priority network and to the Megabus to inhibit other units on the Megabus from initiating a new request.

If MYREQT is set and the MPDC has the highest priority on the Megabus, the set data cycle now (SETDCN) flip-flop sets. Priority is determined by the lines BSAUOK through BSIUOK being high, indicating that no unit on the Megabus with higher priority than the MPDC has a Megabus request active. The SETDCN signal sets the My Data Cycle Now (MYDCNN) signal, which stays active until the slave unit responds. The response from the slave clears the MYDCNN flip-flop. When the MYDCNN signal is enabled, the information to be transferred across the Megabus is loaded into the address and data registers.

The priority network consists of nine input signals (BSAUOK through BSIUOK) and one output signal (BSMYOK). The MPDC has priority over the Megabus when all of the inputs are high, a condition indicating that no other controller of higher priority than the MPDC has a Megabus request enabled. When the signal BSMYOK+ sets, it is transferred to the Megabus whereby it notifies all other controllers connected to the Megabus that the MPDC is presently utilizing the Megabus.

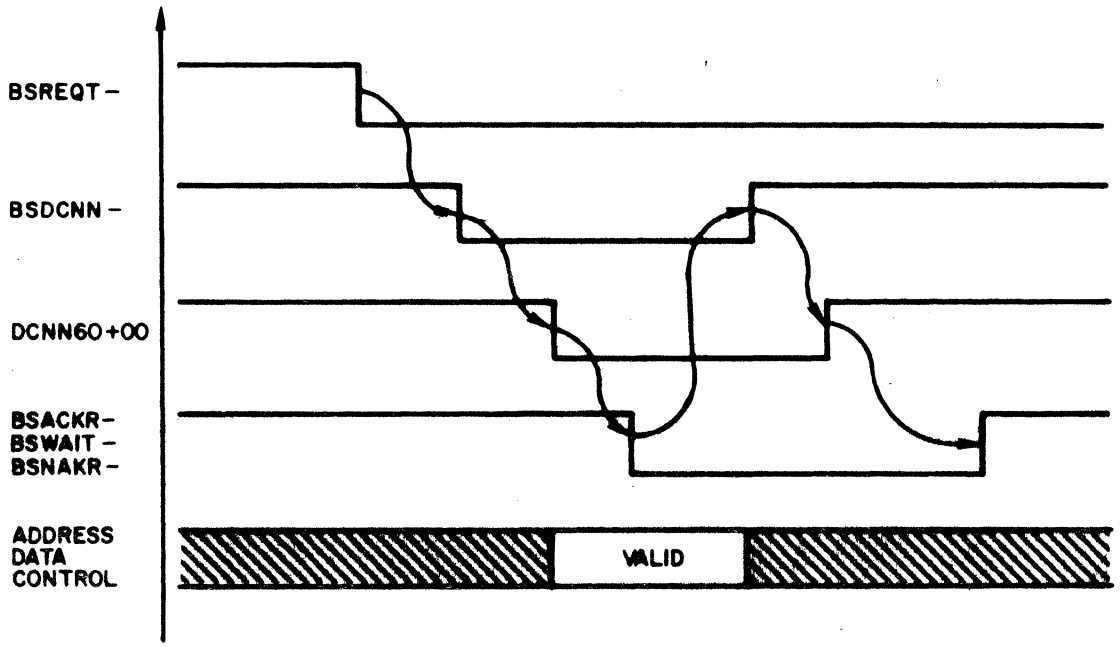


Figure 3-2 Megabus Handshaking Sequence

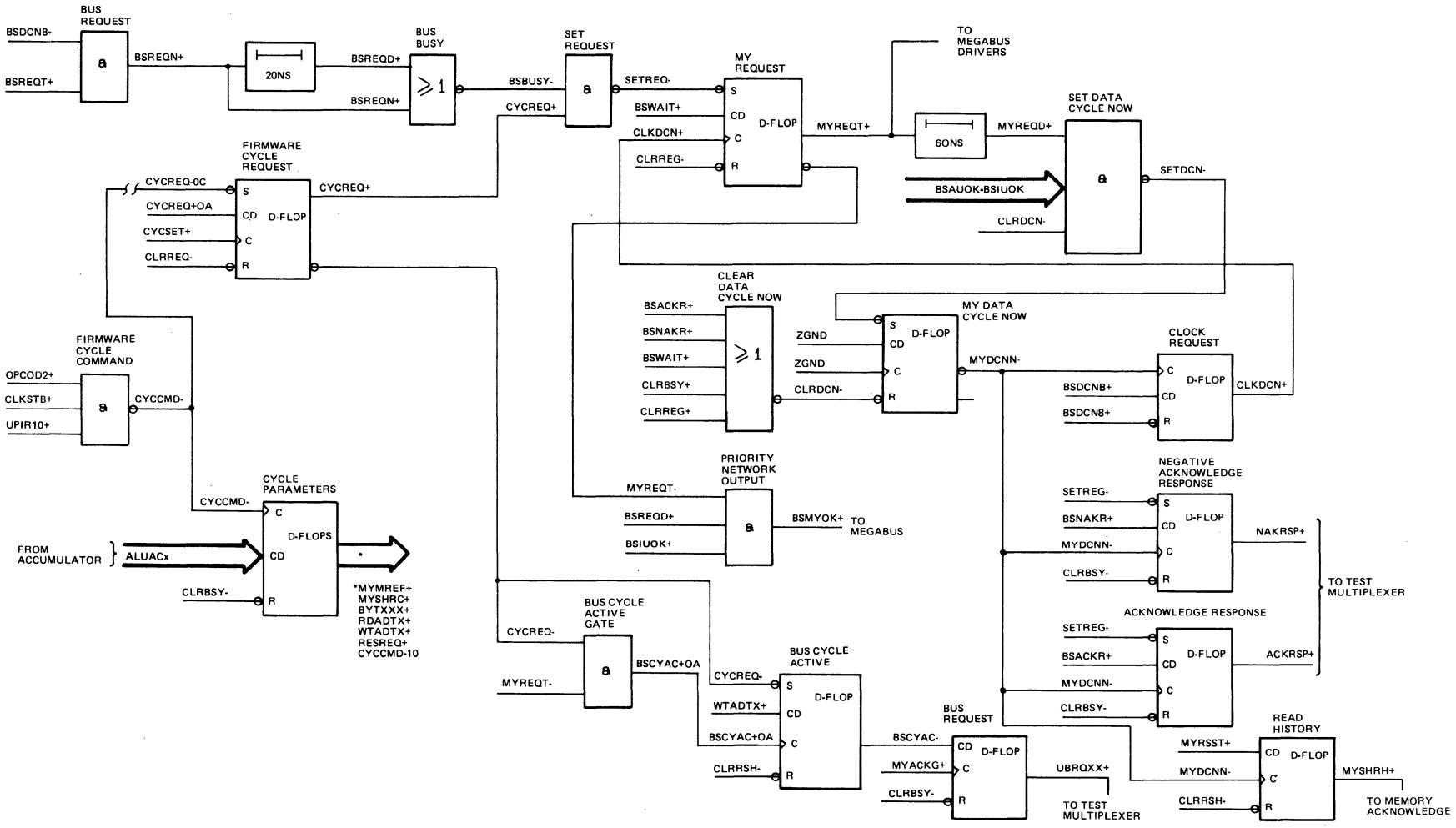


Figure 3-3 Cycle Request Logic and Tiebreaking Network

HONEYWELL PROPRIETARY AND CONFIDENTIAL

Table 3-1 Firmware Cycle Command Parameters

MNEMONIC	ACU BIT	CYCLE BYTE DEFINITION	CYCLE COMMAND REGISTER PARAMETERS					
			WRITE FIELD 1	WRITE FIELD n	READ FIELD 1	READ FIELD n	END OF READ WITH SINGLE BYTE STORED	INPUT RANGE; ADDRESS; ETC.
CYCCMD-10	0	Cycle	0	0	0	0	1	1
MYMREF	1	Memory Reference	1	1	1	1	1	0
RESREQ	2	Response Required	1	1	0	0	0	0
MYSHRC	3	Second Half Read	0	0	0	0	0	1
BYTXXX	4	Byte Mode	X	0	X	0	1	0
RDADTX	5	Read	0	0	1	1	1	0
WTADTX	6	Write	1	1	0	0	0	0
XNU	7	Not Used	0	0	0	0	0	0

X = No effect.

3.2.2 Response Logic

The response logic (Figure 3-4) generates the appropriate response for the MPDC, analyzes the address on the Megabus address lines to determine if the MPDC is being addressed, and generates a busy signal when required.

The MPDC responds to a Megabus cycle in one of four ways which are prioritized and have the following significance:

1. No response indicates that the addressed channel has no associated device attached.
2. The negative acknowledge response (MYNAKR+) indicates that the addressed channel is not in a ready condition. The channel may be busy doing a device operation.
3. A wait response (MYWAIT+) indicates that the addressed channel is ready, but the MPDC is busy performing another operation. The master unit should retry the Megabus cycle.

4. An acknowledge response (MYACKR+) indicates that the addressed channel is ready and that the MPDC has stored the information on the Megabus in the address and data registers.

Each channel of the MPDC has an output level (ADPGND) which indicates that a device is attached to that channel. The level for each channel is delivered to the device present multiplexer, which selects the level corresponding to the addressed channel. The output from the device present multiplexer (ADPGND-10) is utilized by the cycle response logic to generate a response in accordance with the MPDC condition. A high output from the device present multiplexer indicates that no device is attached to that channel of the MPDC and inhibits the MPDC response logic.

There are four channel ready (CHNRDY) flip-flops which firmware uses to indicate channel status. The state of the addressed channel, as reflected by the channel ready flip-flop, is selected by the channel ready multiplexer and transferred to the cycle response logic. The channel ready (CHNRDY) signal is augmented by the second half bus cycle (BSSHBC) signal which causes the addressed channel to appear ready for any second half read cycle. The channel ready multiplexer may be forced to a ready state whenever the function code on the address bus is set to a One state.

The register busy latch stores the status of the MPDC while it is busy clearing specific registers. The output of the register busy latch goes to the cycle response logic to generate a wait response.

The address decoder (BSASW1+ through BSAW7+) determines if the MPDC is being addressed by the Megabus. The decoder, consisting of two hex rotary switches and gating, compares the address on the Megabus against the address of the MPDC. The address of the MPDC is determined by the setting of the switches and is not visible to the MPDC until the first Megabus cycle acknowledged by the MPDC. The decoder output is active (low) when bits 8 through 14 of the Megabus agree with the switch settings and when the Memory Reference signal on the Megabus is low. Bits 15 and 16 of the Megabus determine which of the four MPDC channels is being addressed and are not examined by the decoder. These bits select adapter present and channel ready status.

The cycle response gating generates the ACK, NAK, or WAIT response if the MPDC is addressed and the addressed channel has a device attached. The output of the response gate is stored in the response latch 60 ns after the cycle was initiated with BSDCND coming high. The output of the response latch goes to the Megabus drivers and to various logic on the MPDC.

The response status register stores pertinent Megabus signals, including the channel numbers, byte mode, write mode, and red/yellow status signals. The channel number, stored during the ACK response, is decoded by the Megabus request decoder, which generates four outputs indicating which channel acknowledged the Megabus. These outputs go to the channel request priority encoder and serve to determine firmware execution.

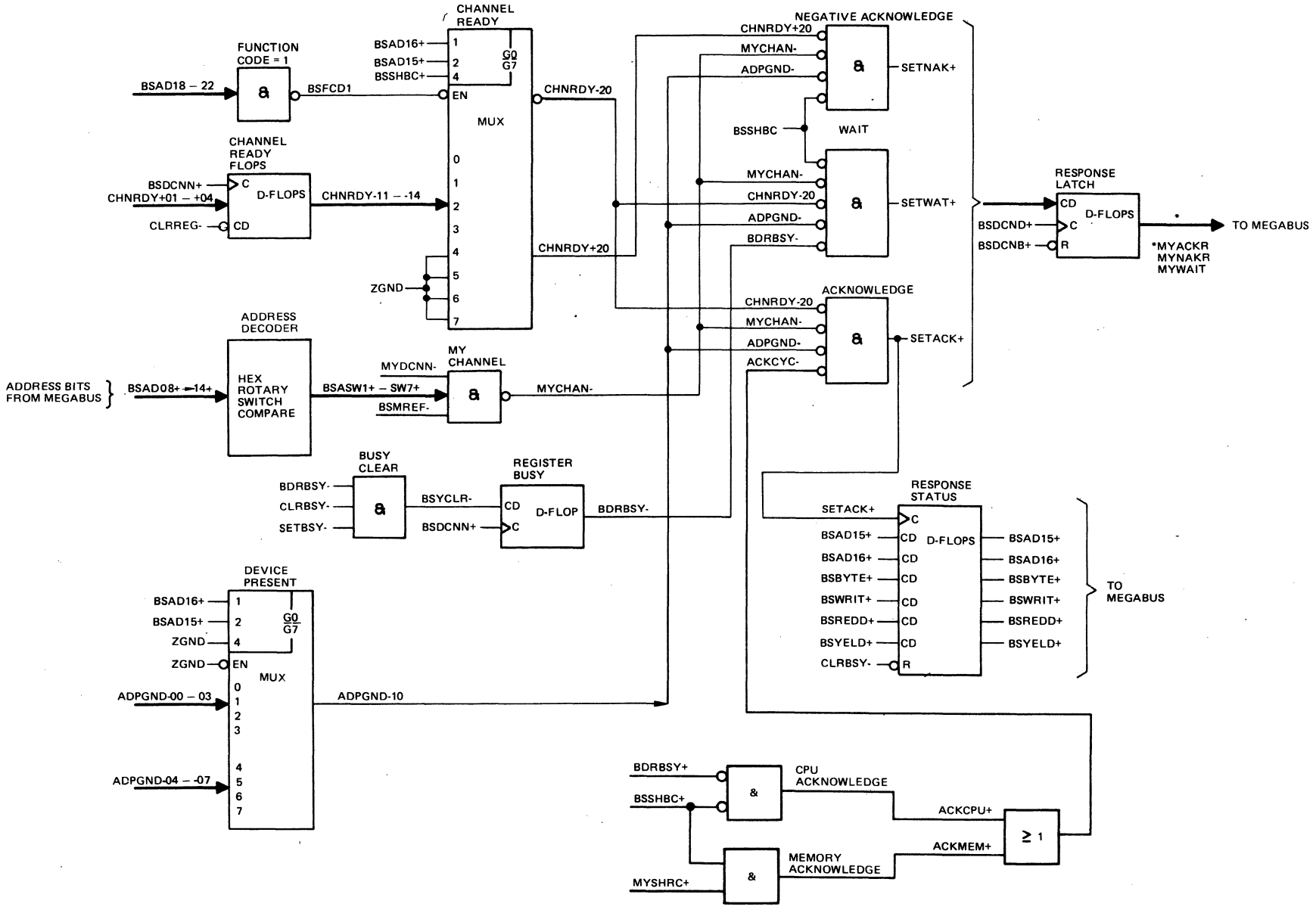


Figure 3-4 Response Logic

HONEYWELL PROPRIETARY AND CONFIDENTIAL

3.3 ADDRESS CONTROL LOGIC

The address control logic consists of the address transceivers, through which memory addresses, channel numbers, and function codes are transferred between the MPDC and the Megabus, and the control logic, which distributes the information contained on the address lines throughout the MPDC (see Figure 3-5).

The 24-bit address (BSAD00+ through BSAD23+) and the module parity bit (BSAP00+) that is transferred from the Megabus to the MPDC are distributed throughout the MPDC in the following manner:

1. BSAD00+ through BSAD23+ and BSAP00+ are transferred from the address transceivers to the address register when an Acknowledge Response (MYACKG+) signal and a shift bus data register (SFTBDR-) signal is valid at the input gate of the address register.
2. BSAD08+ through BSAD14+ are transferred from the address transceivers to the MPDC hex rotary switches to determine if the incoming information is for the MPDC.
3. BSAD15+ and BSAD16+ contain device port information and are sent to the device present logic (to determine if a particular device is existent in the system) and to the channel ready logic (to determine if the requested channel is busy).
4. BSAD18+ through BSAD22+ are sent to the function code equals one gate to determine if the incoming information is a control word transfer.
5. BSAD00+ through BSAD07+ and BSAP00+ are transferred to the bus address parity checker to determine if the incoming module parity is correct.

The 24-bit address, which is stored in the address register, is transferred to the ALU AOP multiplexer in byte parallel form when the clock input, Shift Bus Data Register (SFTBDR-) signal, is enabled. The SFTBDR- signal enables the low order eight bits of the address register (MYAD16+ through MYAD23+) to be transferred to the ALU AOP multiplexer and shifts the other two bytes of the address to the low order positions of the register. Thus, three transfers are required to empty the contents of the address register into the ALU AOP multiplexer. Bits 15 and 16 (MYAD15+ and MYAD16+) of the address register are gated to the scratch pad memory index register.

For operations in which the MPDC is to transfer information to the Megabus (memory reads and writes, interrupts, and input commands), the information is delivered to the address transceivers across lines MYAP00-, BBAD00+ through BBAD07+, and BDAD08+ through BDAD23+.

For memory read or write operations, the memory address is byte transferred from the arithmetic logic unit (ALUOT0+ through ALUOT7+) into the bus address counter. Three transfers from the ALU are required, the bus address counter shifting the byte input through the counter until it is full. The bus address counter outputs (BBAD08+ through BBAD23+) are transferred to the address/data selection

multiplexer, which delivers the address bits (BDAD08+ through BDAD23+) to the Megabus when the control gate, second half read cycle (MYSHRC+), is low. Bits 0 through 7 and the parity bit from the bus address counter (BBAD00+ through BBAD07+ and MYAP00-) are transferred to the Megabus transceivers, thus completing the memory address for a direct memory access read or write operation.

During an input interrupt command or an I/O input command, the information received by the MPDC during the request cycle across the data bus (channel number of source), is returned to the requesting controller during the response cycle across the address bus (channel number of destination)*. This is accomplished by transferring the contents of the data bus into the bus data register during the request cycle. The output from the bus data register (MYDT00+ through MYDT15+) is delivered to the address/data selection multiplexer, which has its control gate (MYSHRC+) enabled during a second half read cycle (response cycle). Therefore, the content of the data bus is transferred to the address transceivers across lines BDAD08+ through BDAD23+, completing the operation.

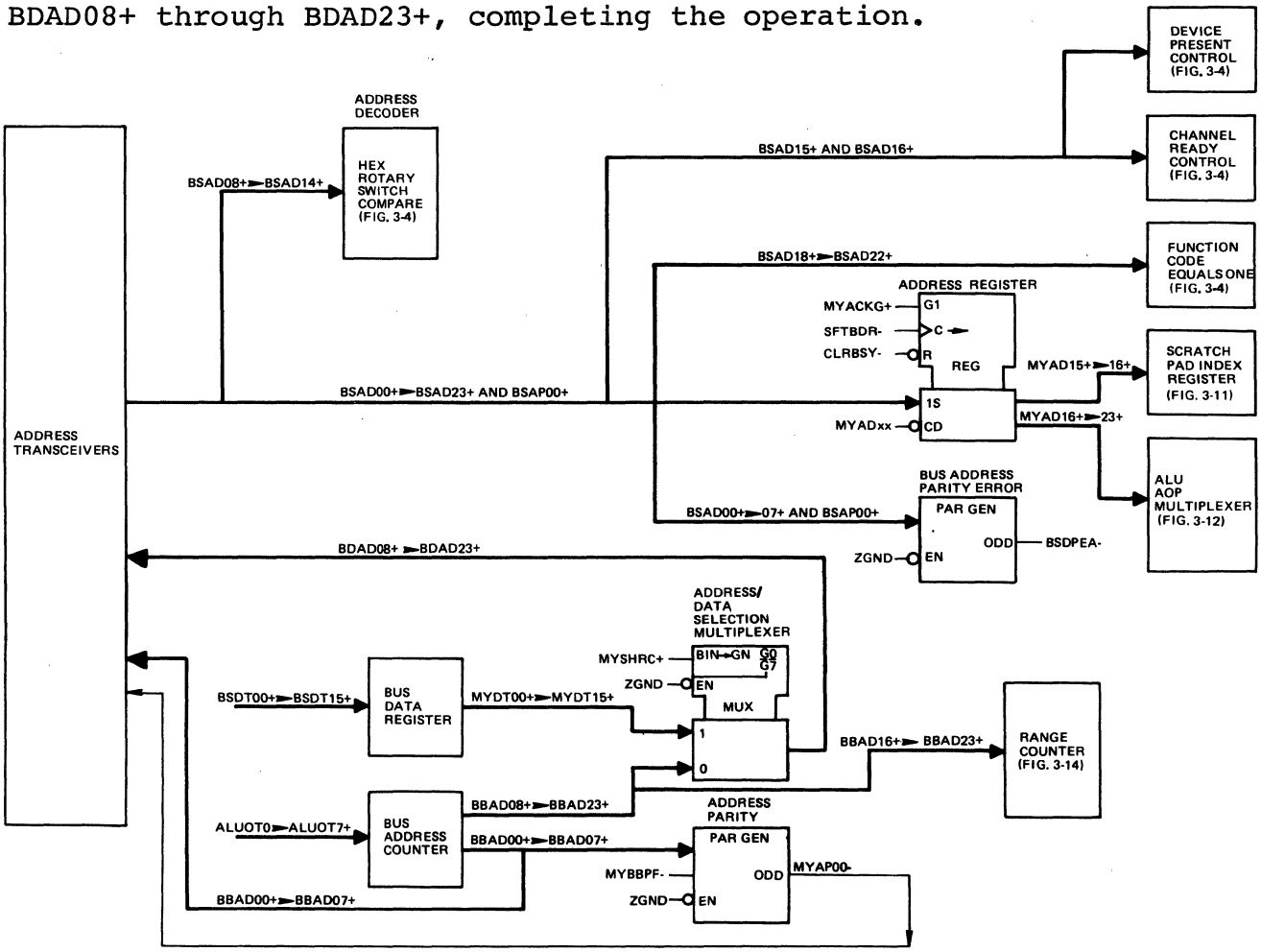


Figure 3-5 Address Control Logic

*The unit which initiated the request (source) becomes the destination during a response cycle.

3.4 DATA CONTROL LOGIC

The data control logic (shown in Figure 3-6) enables information to be transferred between the MPDC and the Megabus data lines. Sixteen data bits and two parity bits enter and exit the MPDC via the data transceivers, carrying input and output information between the MPDC and other controllers. The major logic elements of this area are the error checkers, first-in-first-out (FIFO) data buffers, and the data/second half read selector.

Information entering the MPDC from the Megabus data lines is gated through the data transceivers (BSDT00+ through BSDT15+) to the bus data register and the error-checking logic. The error-checking logic checks the parity on the module address bits, and the left half and right half bytes of the data bits. Any error in this area sets the data parity error gate (BSDPER+). The output from the bus data register (MYDT00+ through MYDT15+) is distributed to the ALU AOP multiplexer, where selection is determined by a 3-bit microprogram index register address; the address/data selection multiplexer, which permits information to enter from the data lines when the MPDC is responding to a second half bus cycle (MYSHRC+); and the data FIFOs, which stack incoming data words from the Megabus.

For operations in which the MPDC is to transfer information to the Megabus (memory reads and writes, interrupts, and input commands), the information is delivered to the data transceivers across lines MYDP00-, MYDP08-, and BDDT00+ through BDDT15+).

For a memory write operation, data is delivered from the ALU to the data flip-flops. The output of the data flip-flops (BBDT00+ through BBDT15+ is then sent to the data/second half read selector multiplexer, which reflects the data flip-flop input at its output. The output from the selector (BDDT00+ through BDDT15+) is delivered to the data transceivers to become the information content on the Megabus data lines.

For a memory read operation (request cycle), the same logic is used to deliver information to the Megabus, the only difference being that the information content sent to the Megabus from the ALU is the MPDC channel number.

For an I/O Input command, the MPDC must respond with a second half read cycle. The second half read flip-flop for the right data byte is firmware loaded with the contents of the ALU. The contents of the right half data flip-flops are shifted to the left data flip-flops, and the right data flip-flops are reloaded from the ALU. The output from the second half read data flip-flops (SHRD00+ through SHRD15+) is transferred to the data/second half read selection multiplexer. Being the response to the I/O Input command, MYSHRC+ is high, reflecting the second half read data input at the output. The output of the selector (BDDT00+ through BDDT15+) contains the information content to be transferred to the Megabus.

The first-in-first-out (FIFO) buffer memory collects data in the MPDC (Figure 3-7). The MPDC utilizes four FIFO chips organized as 16 words by 16 bits. These four FIFOs receive the bus data register output (MYDT00+ through MYDT15+), and a fifth FIFO is used

to prevent the MPDC from making a cycle request when the data FIFOs are full. The FIFO chips are capable of stacking 14 words, plus retaining one word in the input and output registers (total capacity 16 words).

The input and output mnemonics shown within the FIFO chips in Figure 3-7 have the following definitions:

- PL - Enable gate for a parallel load
- TTS - Transfer contents to stack input
- TOP - Transfer out parallel input
- EO - Enable output
- D - Parallel data inputs
- Q - Parallel data outputs
- IRF - Input register full
- ORE - Output register empty.

When firmware enables the write mode, it checks the starting address to determine if the first transfer is to be a word or a byte. If the address is on an odd byte boundary, firmware enables the byte mode (BYTXXX-). The byte mode disables the parallel load gate to the left data FIFO causing the first data transfer to be made to the right data FIFO. After the first transfer, byte mode is reset and the FIFOs are loaded until 16 words have been stored, or until the range has been exhausted, whichever occurs first. Each time information is loaded into the FIFOs, the input register full gate (IRF) sets. The input register full output (FIFOFL/R-) is gated to the FIFO input (TTS). The TTS input causes the information in the FIFO input register to be stacked in the chip. As long as there is a range, the FIFOs continue to stack data until 16 words are resident in the FIFOs. At this point, the fifth FIFO (not shown in Figure 3-7) prevents the MPDC from making another cycle request until the FIFOs have space for another word.

The FIFOs are emptied by enabling the transfer out parallel input gate (FIFOUL/R-). The information being extracted from the FIFOs (FIFO00+ through FIFO15+) is delivered to the AOP wired-OR gate. When the output register empty gate is low (FIFORL/R-), it signifies that the FIFOs are empty. Information can be simultaneously parallel loaded and extracted from the FIFOs as long as the input register is empty, which permits another byte to be transferred to the FIFO. The information is extracted from the FIFOs alternately as function FRBTAD is toggled.

3.5 SUBSYSTEM CLOCK (Figure 3-8)

The output of an 8 MHz crystal oscillator (CLKOSC+), located on the ROS adapter, is sent to the MPDC where the clock signals for the MPDC and the device adapter are derived and distributed. The 8-MHz signal, CLKOSC+, is fed to a JK flip-flop in the MPDC which is toggled every trailing edge to produce a 4-MHz square wave, CLKSIG±. The negation clock signal, after inversion, is fed to a 100 ns delay line. The delay line is tapped to produce timing signals which will meet data setup hold times sufficient for writing into the scratch pad memory; and signals to clock registers in the MPDC and the adapter.

The clock can be stopped by firmware with a miscellaneous micro-op and bit 9 of the microprogram index register set. This sets the clock half flip-flop (CLKHLT+) which stops the system clock until a Master Clear signal is issued on the Megabus. The clock is also stopped if the Scan Error (UPIERR-) signal goes low. Whenever a Master Clear signal is sent to the MPDC, a series of Basic Logic Tests (BLTs) are performed. One of the BLTs scans the micro-ops with a long check character. If this long check is in error, the ROS adapter sets UPIERR- low, which sets CLKHLT+ on the MPDC and stops the clock.

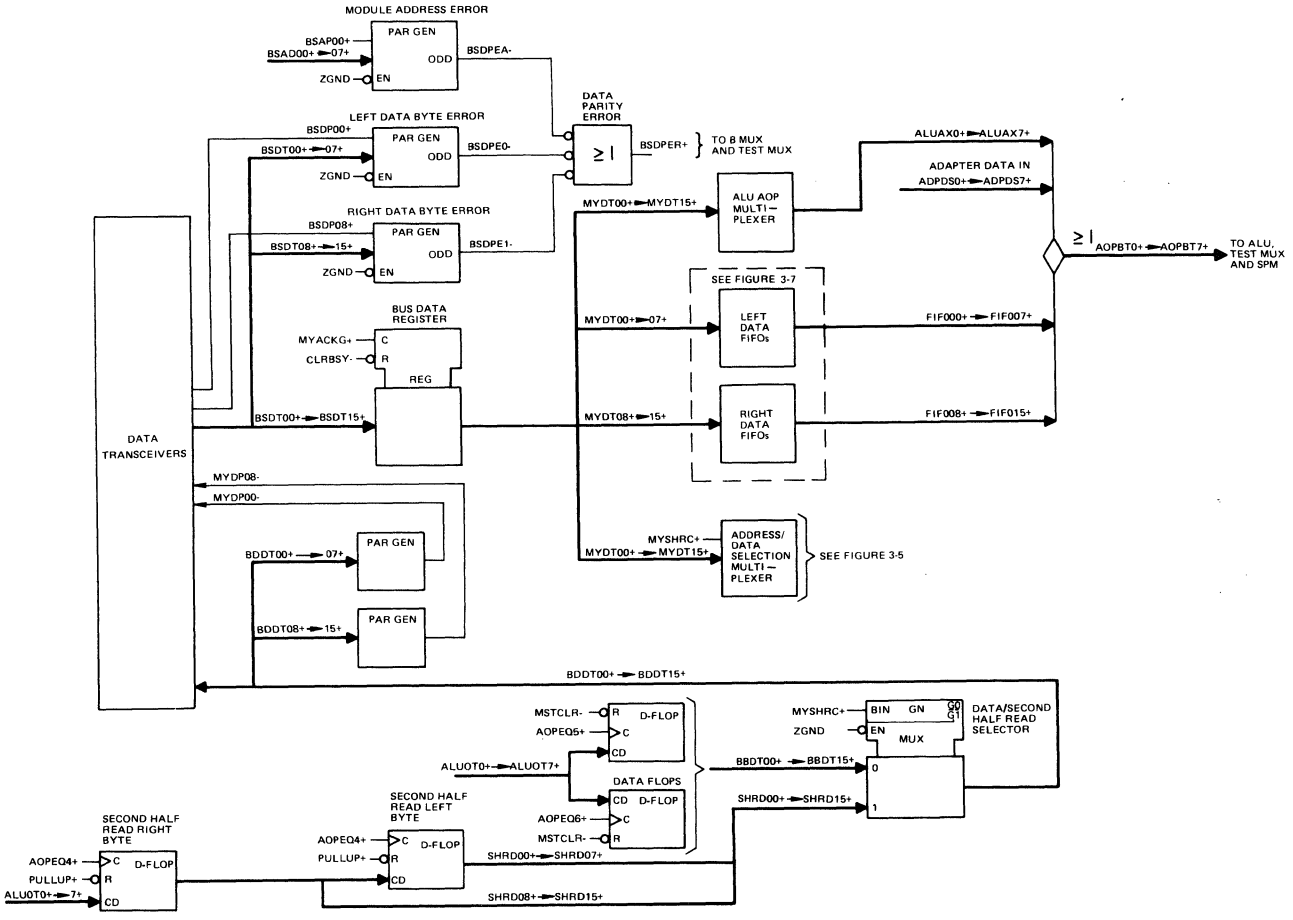


Figure 3-6 Data Control Logic

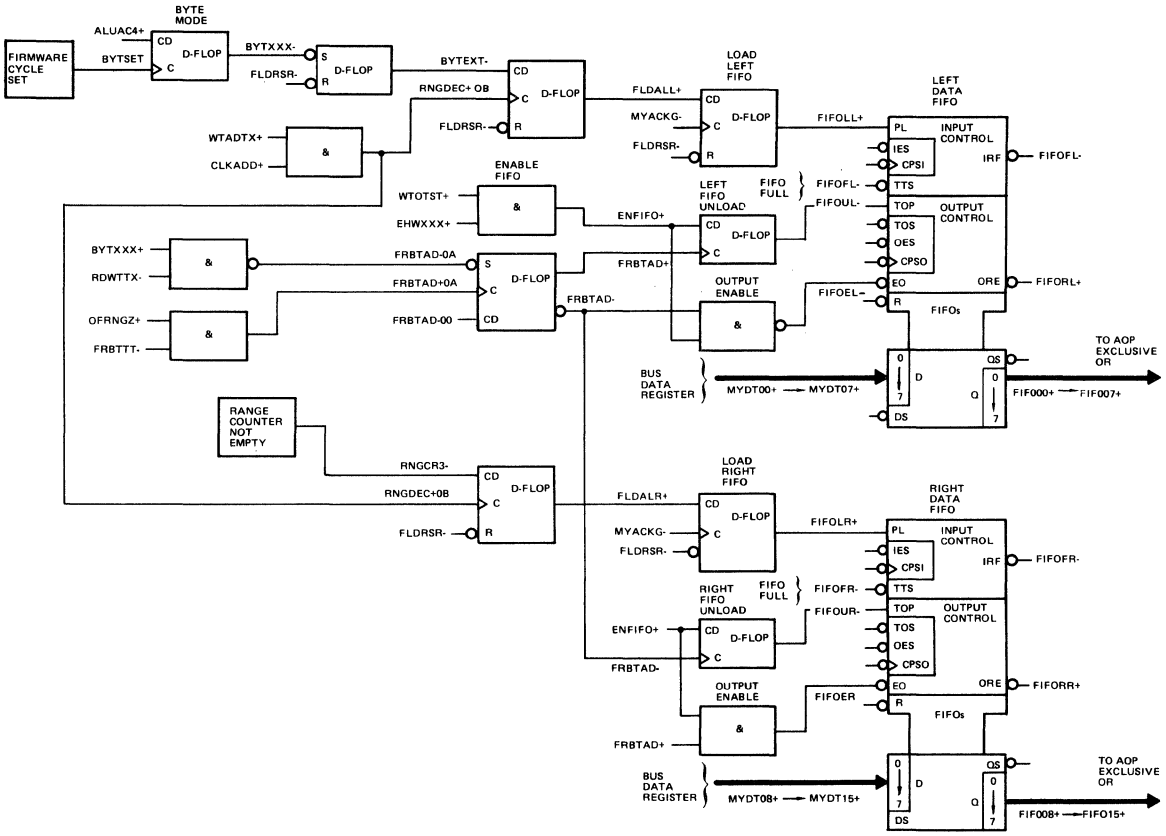


Figure 3-7 Data First-In-First-Out Buffer Memory

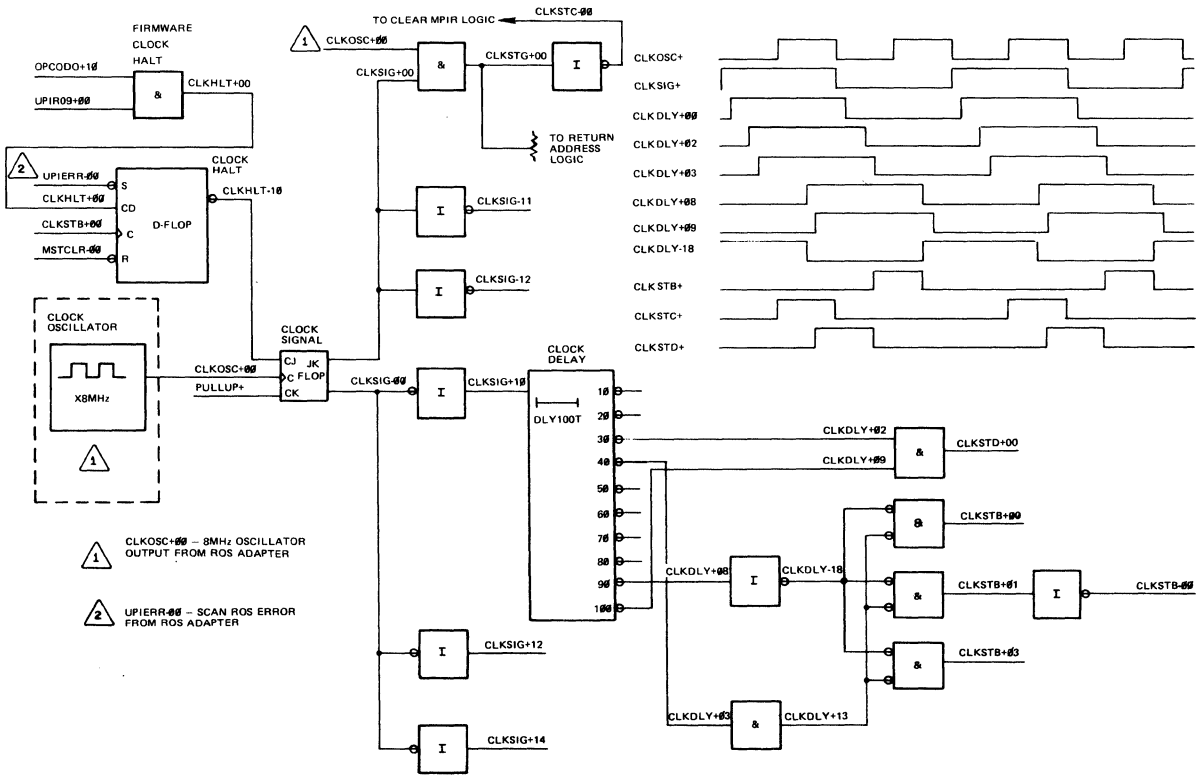


Figure 3-8 Subsystem Clock

3.6 READ ONLY STORE ADAPTER LOGIC

The read only store adapter (microprogram control store) logic provides permanent storage for resident control firmware and diagnostic microprograms. The logic for the microprogram control store is located on an adapter module which attaches to the MPDC via two 25-pin connectors. Across these connectors, the MPDC interfaces with the read only store adapter.

The major logic elements of the read only store adapter (shown in Figure 3-9) are the subroutine return address register (SRAR), the microprogram address selector (MPAS), the microprogram address counter (MPAC), the microprogram control store (MPCS), the microprogram instruction register (MPIR), the op code decoder, and the scan logic. Each logic area is described in this subsection.

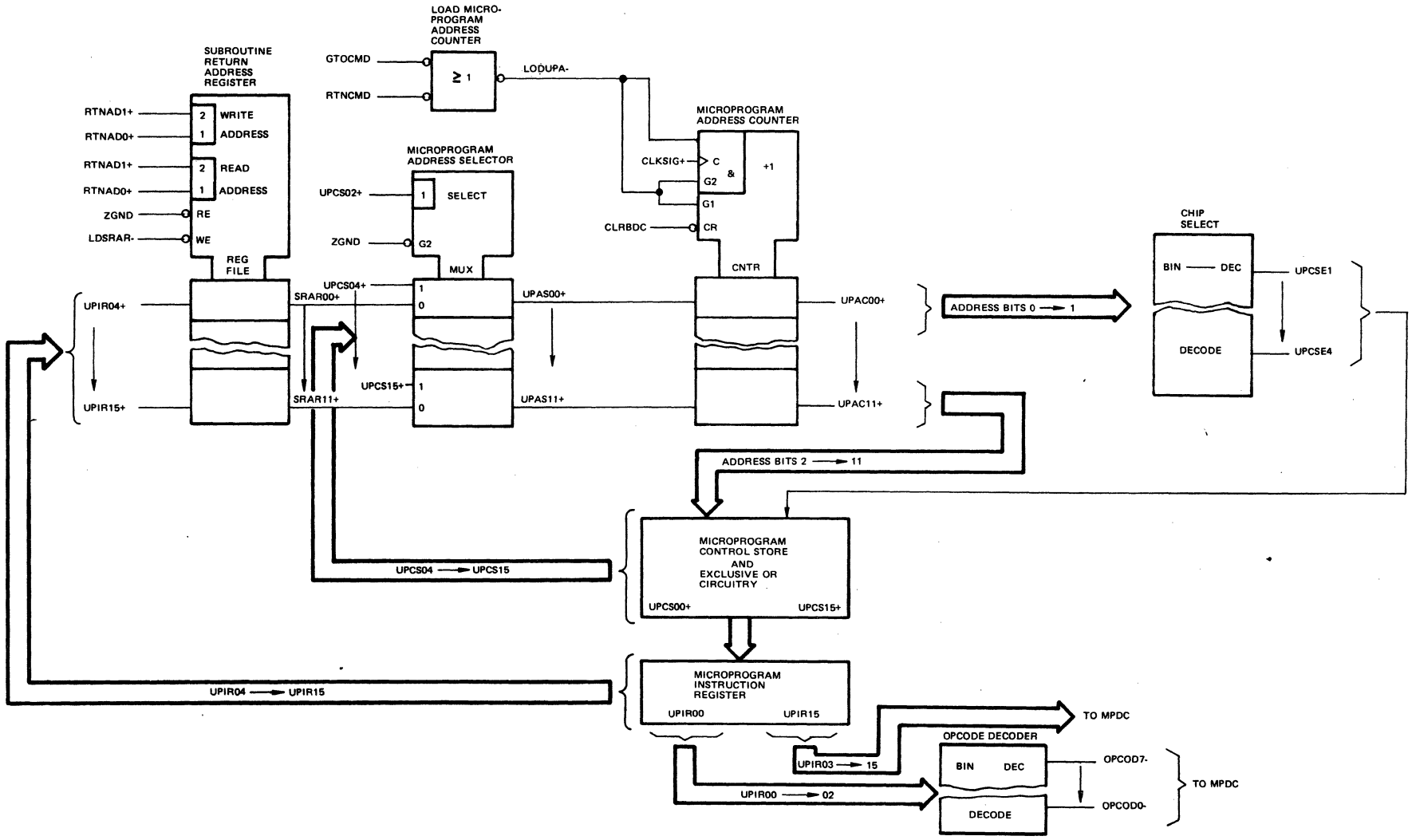


Figure 3-9 Microprogram Control Store Functionality

HONEYWELL PROPRIETARY AND CONFIDENTIAL

3.6.1 Subroutine Return Address Register (SRAR)

Firmware uses the SRAR to store an address which is to be branched to later by a firmware routine or subroutine. The SRAR is a 12-bit register file, the output of which reflects one of four locations within its storage area. The location of the SRAR that is being read or written is determined by the return address counter (RTNAD0-RTNAD1), which increments or decrements the SRAR.

To load a location with an address, the firmware issues a Load Return Address command, which generates the write enable function LDSRAR. This causes the address defined by bits 4 through 15 of the microprogram instruction register to be stored in the SRAR at the location specified by the return address counter and increments the SRAR address.

When firmware performs a Return Branch command, the SRAR address is decremented and the location defined by the return address counter is gated to the SRAR outputs. The reason for decrementing the SRAR address is to bring the SRAR back to the last location stored. The SRAR output is then transferred via the address selector to the address counter. Utilizing the Return Branch command (RTNCMD) makes it possible to preset the address counter to an address previously stored in the SRAR.

3.6.2 Microprogram Address Selector (MPAS)

The MPAS selects one of its two inputs to be used as a preset for the MPAC. When performing firmware operations and the output of the control store bit 2 (UPCS02) is high, indicating a Go To command (GTOCMD), the output of the MPAS (12 bits) reflects bits 4 through 15 of the control store. When UPCS02 is low, the 12 bits of the return address register are selected for the MPAS outputs.

3.6.3 Microprogram Address Counter (MPAC)

The MPAC is a 12-bit counter which sets up the chip select and location in the microprogram control store to be accessed. It is incremented once at the start of every clock cycle, except in the instance of a clear or load operation. The MPAC is cleared by CLRBDC, which is active only during an MPDC initialize (Master Clear). The MPAC is loaded when LODUPA is active. The load operation (LODUPA) is performed when a Go To command or a Return Branch command is decoded in the address control logic.

The low-order 10 bits (2 through 11) of the MPAC are gated directly to the address inputs of the control store PROM. The high-order two bits)0 and 1) are fed to the chip select decoder which determines which four of the 16 control store PROM chips to enable.

3.6.4 Microprogram Control Store (MPCS)

The microprogram control store provides permanent storage for resident control firmware and diagnostic microprograms. The MPCS consists of 16 PROM chips that are 1024 X 4 bits wide. The chip select input permits 4 of the 16 chips to be enabled, producing a 16-bit output. The output is derived from the 10-bit address code provided at the input to the chip.

The 16-bit output from the MPCS is delivered to the microprogram instruction register (MPIR), after passing through an Exclusive OR circuit, for further distribution. The MPCS output, bits 2 and 4 through 15, is also delivered to the microprogram address selector after passing through the exclusive-OR circuits.

3.6.5 Microprogram Instruction Register (MPIR)

The microprogram instruction register (MPIR) is a 16-bit wide register used to store the output of the control store for one clock cycle during a microinstruction execution. The MPIR is loaded at the leading edge of each cycle by the Clock signal (CLKSIG+) unless the clear microprogram instruction register (CLRUPI) is active, which causes a reset to Zero. CLRUPI is active during the Master Clear operation and during a skipped cycle due to a successful test instruction.

The output of the MPIR is distributed in the following manner:

- Bits 0 through 2 are delivered to the op code decoder.
- Bit 3 is transferred to the MPDC across the interface.
- Bits 4 through 15 are delivered to the subroutine return address register, and to the MPDC across the interface.

3.6.6 Op Code Decoder

The high-order three bits of the MPIR are fed to the op code decoder, which performs a 3-bit decode to one of 8 lines. These lines indicate the type of microinstruction being performed. The op code decoder is enabled unless the MPDC is in the process of performing a MPCS scan or unless the MPIR is being cleared via the CLRUPI function. The eight output lines of the op code decoder, in conjunction with various other MPIR bits, implement and control the MPDC and device adapter hardware.

To increase speed of operation during the firmware routines, branch type commands are decoded directly from the output of the control store rather than from the MPIR. For the same reason, the address for Go To commands is taken from the output of the control store. In the case of all other command types, the op code is decoded from the output of the MPIR.

3.6.7 Scan Logic

Scan logic in the MPDC ensures the integrity of the control stores (Figure 3-10). This logic performs a longitudinal check on each bit position of the control store and causes the MPDC clock to halt if an error is detected. If no errors are detected, the MPDC executes the firmware portion of the Basic Logic Test.

The scan mode flip-flop (SCNMOD) sets with Master Clear (MSTCLR), disabling the op code decoder and inhibiting branch commands. With the commands inhibited, the MPDC loads the MPIR with the output of the control store and increments the MPAC to the next location at each sequential clock cycle. This process continues through the MPAC from location 000 to location FFF, at which time a carry-out of the MPAC (UPAC2C) is detected. The MPAC then returns to zero and begins another scan.

The scan bit selectors (SCNBSH and SCNBSL) select one bit of the MPIR, using the output of the scan bit address counter as an address. The selected MPIR bit provides the input to the scan bit sum flip-flop (SCNBSM). SCNBSM half adds the selected bit at the start of each clock cycle with its previous contents. At the end of one scan through the control store, the scan bit sum flip-flop has added the specified bit of each location with an expected result of zero (no error). If the sum is not zero at the termination of each scan, the scan error gate (SCNERR) is high and sets the scan error flip-flop (UPIERR). UPIERR causes the MDC clock to halt, and the contents of the scan bit address counter indicates which bit of the control store is in error.

If an error is not detected at the time of a scan, the scan bit address counter increments due to the overflow of the MPAC, and the MPAC wraps around to location zero. The scan operation is repeated with the next bit of the MPIR as an input to the scan bit sum flip-flop. The scan operation continues until an error is detected or until all bits of the MPIR have been checked, at which time the scan bit address counter overflows (SCNBAC) and resets the scan mode flip-flop. When the scan mode flip-flop is reset, the op code decoder and branch commands are enabled, and normal execution of firmware begins.

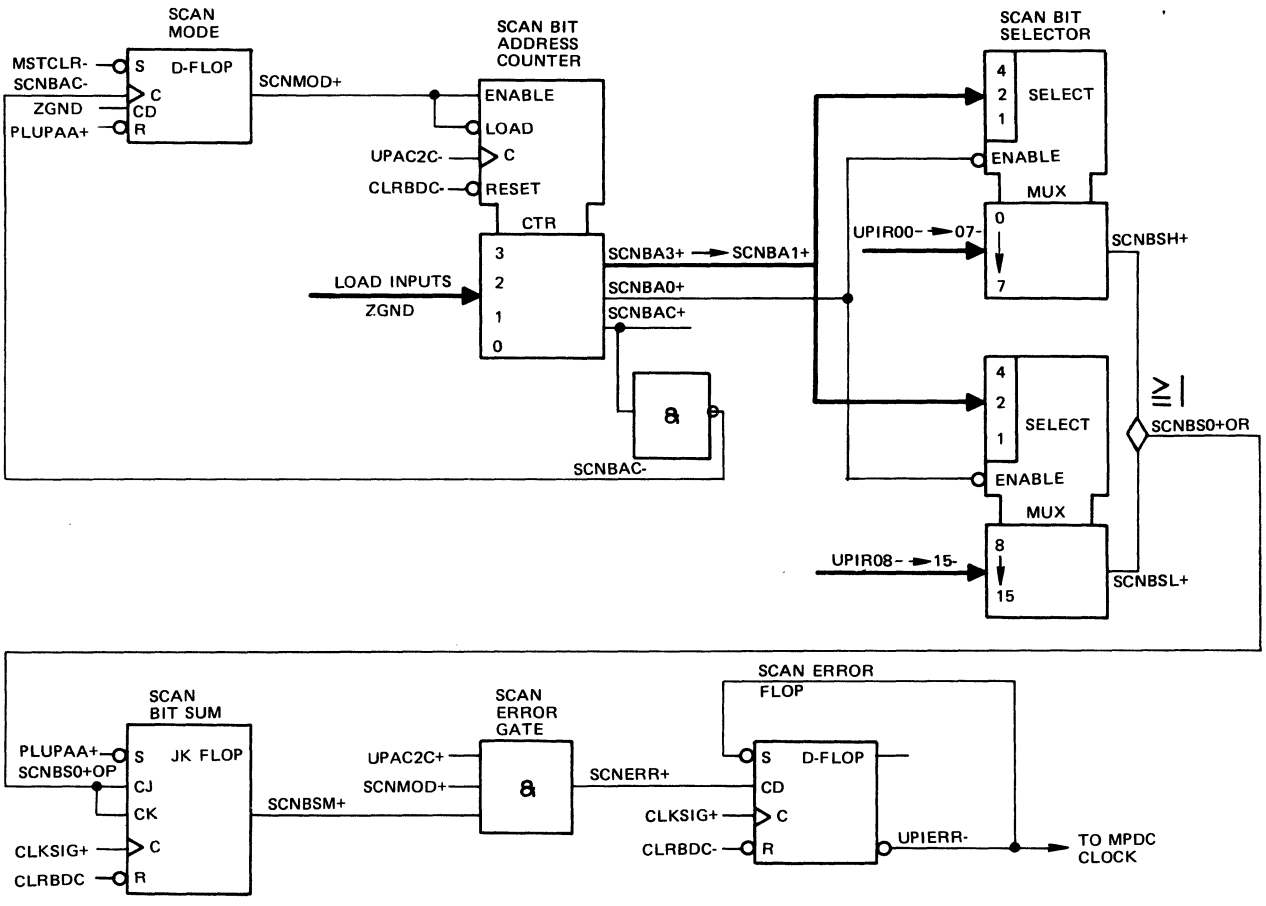


Figure 3-10 Scan Logic

3.7 SCRATCH PAD MEMORY LOGIC

The scratch pad memory logic consists of the index register, the address counter, the address selector, the scratch pad memory, and the logic elements that control the operation of these areas. The flow of information through the SPM is described in the following subsections, and the logic elements are depicted in Figure 3-11.

3.7.1 Scratch Pad Memory (SPM)

The SPM is a 1024-location by 8-bit read/write memory which is used for storage of information that is required by or generated by each channel. Data from the A operand exclusive-OR gate is written in the SPM during the Memory Write command at the location defined by bits 0 through 2 of the address selector and bits 2 through 7 of the address counter. The data out of the SPM is delivered to both the AOP and BOP multiplexers. The decode of a Memory Write command is ANDed with the clock signal CLKSTD, which produces the write pulse MWTCMD.

For information pertaining to SPM segmentation and topology charts, refer to the appropriate device adapter manual.

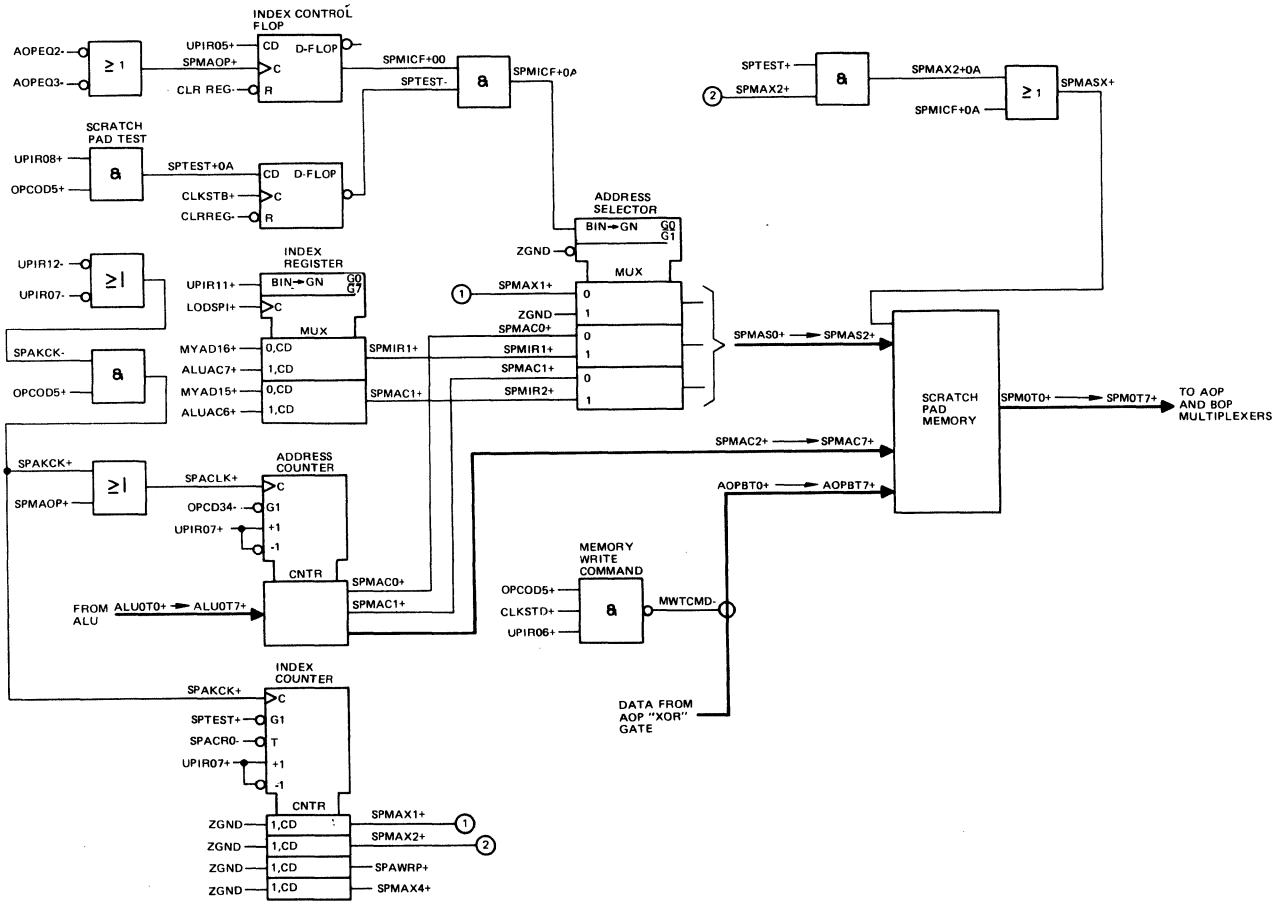


Figure 3-11 Scratch Pad Memory

3.7.2 Index Register

The MPDC may be configured with up to four devices (channels), and the index register (SPMIR1+ and SPMIR2+) specifies which channel is busy at any given time. The content of the index register is utilized to address a specific segment of the scratch pad memory and determine which channel is enabled for proper firmware command execution.

The index register may be loaded with a specific channel number from the firmware (ALUAC6+ and ALUAC7+), or it may be loaded with the channel number designated by the channel requested across the address lines (MYAD15+ and MYAD15+). The input to the index register is determined by the logic state of MPIR bit 11 (UPIR11+) and is loaded whenever the function LODSPI is active.

3.7.3 Index Control Flip-Flop

The SPM index control flip-flop (SPMICF+) determines from where the scratch pad address is to be loaded. When SPMICF is set, the scratch pad is addressed by way of the six low-order bits of the address counter and the two high-order bits are taken from the contents of the scratch pad memory index register. When SPMIFC is reset, scratch pad is accessed by all eight bits of the address counter and the counter output SPMAX1, which switches the scratch pad quadrant when it reaches a count of 256 locations. Normally, the index control flip-flop is set during functional firmware except when the firmware must address the memory independently of the channel which is active.

3.7.4 Address Selector

The address selector determines the most significant bits of the scratch pad memory address. When the index control flip-flop (SPMICF+) is set, the address selector selects the contents of the index register as the determining factor in which 64 locations of a 256-location scratch pad memory quadrant are to be addressed. These SPM locations are used for the storage of general information, and the SPM is considered to be nonindexed.

If SPMICF+ is reset, the address selector output reflects the two high order bits of the address counter and the index counter output SPMAX1+. If the index counter output is high, the scratch pad address reflects the location of a specific device channel and contains information pertinent to that device. When SPMICF+ is set, the SPM is considered to be indexed.

Firmware normally indexes the scratch pad memory, thereby dividing the memory into quadrants, one for each available channel. Each quadrant of memory has the same topology.

3.7.5 Address Counter

The address counter is an 8-bit counter which can be parallel loaded with the output of the ALU or sequentially incremented. The contents of the arithmetic logic unit (ALUOT0+ through ALUOT7+) are loaded into the address counter at scratch pad address clock time (SPACLK+) whenever an ALU or Constant command (OPCD34-) is detected.

The address counter is incremented whenever the microprogram index register bit 7 is active at SPACLK time. Therefore, the increment occurs during an increment SPM address or a memory write and increment operation.

3.8 ARITHMETIC LOGIC UNIT AND ACCUMULATOR FUNCTIONAL COMPONENTS

The arithmetic logic unit performs logic or arithmetic operations on the incoming data. The major logic elements of this area are the A-operand multiplexer, the B-operand multiplexer, the arithmetic logic unit (ALU), and the accumulator. Figure 3-12 depicts the ALU and its functional components.

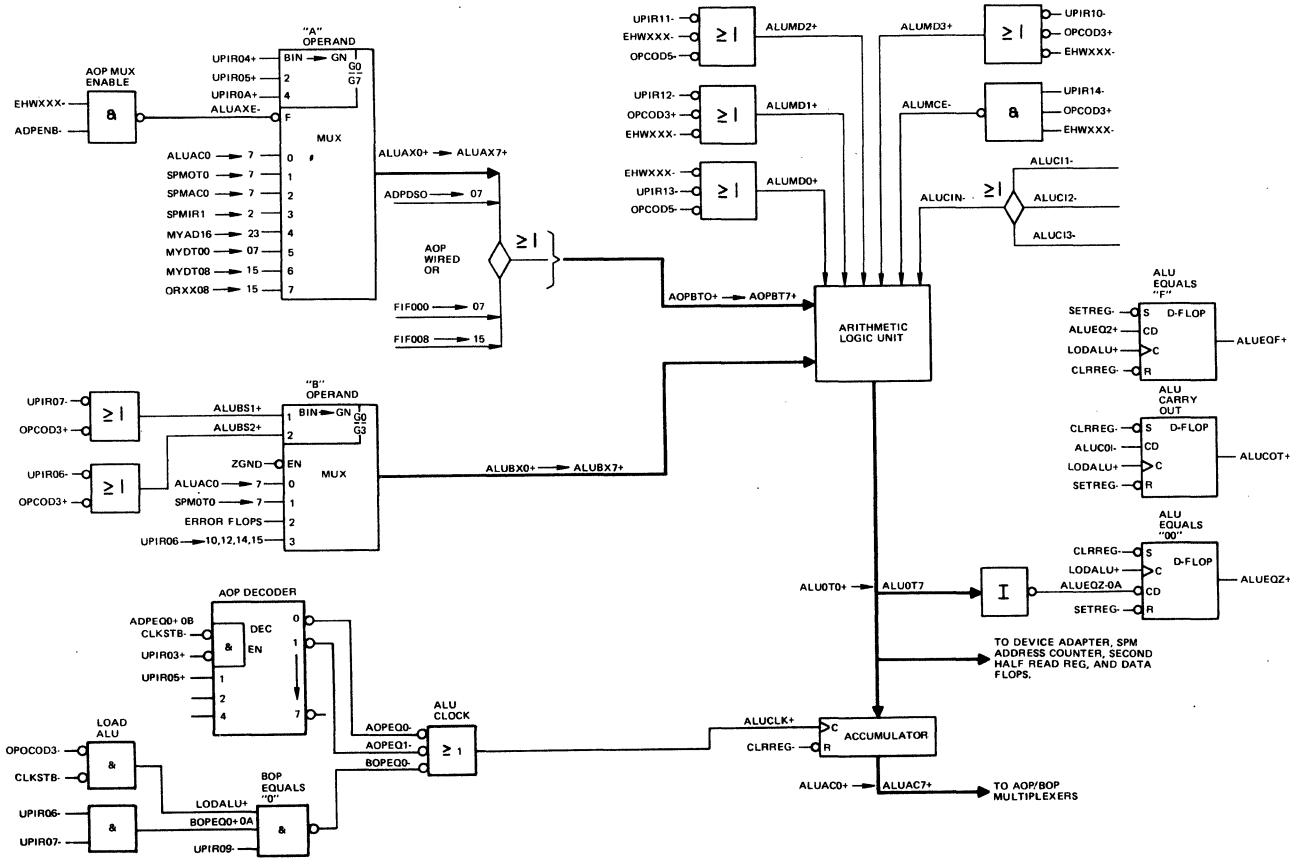


Figure 3-12 Arithmetic Logic Unit

3.8.1 A-Operand Multiplexer

The A-operand multiplexer (AOP mux) can select one of eight types of data fields according to a three-bit address defined by bits 03, 04, and 0A (15) of the MPIR. Table 3-2 shows the various bit configurations and the input that is selected by the setting of the bits.

The AOP multiplexer is a tri-state multiplexer and is enabled when function ALUAXE- is low. This signal is low when both the Enable Hardware (EHWXXX-) and the Adapter Enable (ADPENB-) functions are set. The output from the AOP multiplexer (ALUAX0+ through ALUAX7+) is gated to the AOP wired-OR logic gate.

Table 3-2 AOP Multiplexer Input Selection

MPIR BIT CONFIGURATION			SELECTED REGISTER	MNEMONIC
Address Bits 15	04	05		
0	0	0	Accumulator	ALUAC0 → 7
0	0	1	Scratch Pad Data	SPMOT0 → 7
0	1	0	Scratch Pad Address	SPMAC0 → 7
0	1	1	S.P. Index Register	SPMIR1 → 2
1	0	0	Bus Information	MYAD16 → 23
1	0	1	Bus Data	MYDT00 → 07
1	1	0	Bus Data	MYDT08 → 15
1	1	1	Offset Range	ORXX08 → 15

3.8.2 B-Operand Multiplexer

The B-operand multiplexer (BOP mux) can select one of four data fields as a B-input to the ALU. The BOP is defined during ALU commands by MPIR bits 6 and 7. When performing Constant commands, the BOP multiplexer selects the output of the MPIR bits 6 to 10, 12, 14, and 15 for the input to the ALU. These particular MPIR bits represent the data constant to be operated with during the Constant command.

The address for the data field to be loaded into the BOP mux is a two function decode (ALUBS1, ALUBS2) of the op code and MPIR bits 6 and 7. Table 3-3 shows the four possible configurations of ALUBS1 and ALUBS2 and the data input selected.

Table 3-3 BOP Multiplexer Input Selection

MPIR BIT DECODE		SELECTED DATA INPUT	MNEMONIC
Address Functions ALUBS1·ALUBS2			
0	0	Accumulator	ALUAC0 → 7
0	1	Scratch Pad Data	SPMOT0 → 7
1	0	Bus Status	SPMAC0 → 7
1	1	MPIR Constant	UPIR06 → 10,12,14,15

3.8.3 Arithmetic Logic Unit (Figure 3-12)

The ALU can perform an 8-bit arithmetic or logic operation on the data supplied by the AOP wired-OR logic and the BOP multiplexer. The information gated to the wired-OR logic consists of the AOP multiplexer output (ALUAX0-7), the adapter data in (ADPDS0-7), and the data output from the first-in-first-out (FIFOs) buffers (FIFO00-15). The type of operation to be performed is determined by the four mode signals (ALUMD0-3), the carry enable function (ALUMCE), and the carry input (ALUCIN) to the ALU. Table 3-4 shows the relationship between these functions and the operation to be accomplished by the ALU.

The mode signals, as well as carry enable and carry-in, are explicitly defined by bits of the MPIR. The firmware can set Carry Enable and specify that the carry-in is to reflect the previous ALU command's carry-out as defined by the flip-flop ALUCOT+.

The ALU data output is fed to the accumulator, the scratch pad memory address counter, the second half read register, the data flip-flops, the device adapter, and the status flip-flops. The status flip-flops include the ALU equal to Zero (ALUEQZ) flip-flop, the ALU equal to "FF" (ALUEQF) flip-flop, and the ALU carry-out (ALUCOT) flip-flop, which are set during the ALU operation by the signal LODALU. In this manner, the ALU status flip-flops are set or reset during an ALU command and remain valid for firmware interrogation until the subsequent ALU operation.

The ALUEQZ and ALUEQF status flip-flops indicate an all-Zeros ALU output and an all-Ones ALU output, respectively. The ALUCOT flip-flop indicates a carry-out of the ALU, which serves to indicate the relative magnitudes of the AOP and BOP fields during an ALU subtract operation. Table 3-5 shows the relationship between the ALU carry-in, the ALU carry-out, and the size of the AOP and BOP fields.

For diagnostic purposes, each of the ALU status flip-flops can be set independently of the ALU outputs by functions SETREG- or CLRREG-. Both SETREG and CLRREG can be enabled by firmware commands. CLRREG is also active during a Master Clear.

Table 3-4 ALU Arithmetic or Logic Operations

HEX CODE	ALUMD3+	ALUMD2+	ALUMD1+	ALUMD0+	BINARY CODE	ALUMCE- = 1 =H, LOGIC OPERATIONS	POSITIVE LOGIC	
							ALUMCE- = 0 = L, ARITHMETIC OPERATIONS	
							ALUCIN- = 1	ALUCIN- = 0
0	L	L	L	L	(0000)	$F = \overline{A}$	F = A	F = A plus 1
1	L	L	L	H	(0001)	$F = \overline{A + B}$	F = A + B	F = (A + B) plus 1
2	L	L	H	L	(0010)	$F = \overline{A}B$	$F = A + \overline{B}$	$F = (A + \overline{B})$ plus 1
3	L	L	H	H	(0011)	F = 0	F = minus 1 (2s compl)	F = ZERO
4	L	H	L	L	(0100)	$F = \overline{AB}$	$F = A$ plus $A\overline{B}$	$F = A$ plus $A\overline{B}$ plus 1
5	L	H	L	H	(0101)	$F = \overline{B}$	$F = (A + B)$ plus $A\overline{B}$	$F = (A + B)$ plus $A\overline{B}$ plus 1
6 †	L	H	H	L	(0110)	$F = A \oplus B$	F = A minus B minus 1	F = A minus B
7	L	H	H	H	(0111)	$F = A\overline{B}$	$F = A\overline{B}$ minus 1	$F = A\overline{B}$
8	H	L	L	L	(1000)	$F = \overline{A} + B$	F = A plus AB	F = A plus AB plus 1
9 ††	H	L	L	H	(1001)	$F = \overline{A \oplus B}$	F = A plus B	F = A plus B plus 1
10	H	L	H	L	(1010)	F = B	$F = (A + \overline{B})$ plus AB	$F = (A + \overline{B})$ plus AB plus 1
11	H	L	H	H	(1011)	F = AB	F = AB minus 1	F = AB
12	H	H	L	L	(1100)	F = 1	F = A plus A*	F = A plus A plus 1
13	H	H	L	H	(1101)	$F = A + \overline{B}$	F = (A + B) plus A	F = (A + B) plus A plus 1
14	H	H	H	L	(1110)	F = A + B	$F = (A + \overline{B})$ plus A	$F = (A + \overline{B})$ plus A plus 1
15	H	H	H	H	(1111)	F = A	F = A minus 1	F = A

† Subtract Mode
†† Add Mode
* Each bit is shifted to the next more significant position

NOTE

When $\overline{CE}$ is a logic ONE, $\overline{C_{in}}$ is irrelevant.

HONEYWELL PROPRIETARY AND CONFIDENTIAL

Table 3-5 ALU Carry Flip-Flop Indications

ALU CARRY-IN (ALUCIN)	ALU CARRY-OUT (ALUCOT)	AOP AND BOP RELATIONSHIP
1	0	A $\leq$ B
1	1	A > B
0	1	A $\geq$ B
0	0	A < B

3.8.4 Accumulator (ACU)

The accumulator is an 8-bit register that temporarily stores the ALU output. The operation of the ACU is determined at CLKSTB- time when an ALU operation (OPCOD3-) is to be performed or when the AOP is equal to a Zero or a One. The ACU is cleared when CLRREG- is active. The outputs of the ACU are fed only to the AOP and BOP multiplexers.

3.9 TEST MULTIPLEXER FUNCTIONAL DESCRIPTION (Figure 3-13)

Firmware intelligence is derived from the capability of testing data, status, and errors within the subsystem. Test commands specify a test item and a test condition. The next sequential command is skipped if the state of the test item equals the specified test condition.

All test items, specified by a 5-bit test parameter code, enter the data selectors whose outputs are TSTXB0 through TSTXB3. The data selectors use the three low-order bits of the MPIR 5-bit field to select one of eight inputs. The high-order two bits of the test item field are used to select the test item inputted to the test multiplexer (TSTMUX). The state of the test item, as defined by the output of TSTMUX, is compared against the test condition, and the Test Valid signal (TSTVLD) goes high if the comparison is true. A valid test is recorded in the clear microprogram instruction flip-flop (CLRUPI), which clears the MPIR during the next clock pulse (performs a NOP). If the test is not valid, the MPIR is not cleared, and the next sequential instruction is executed normally.

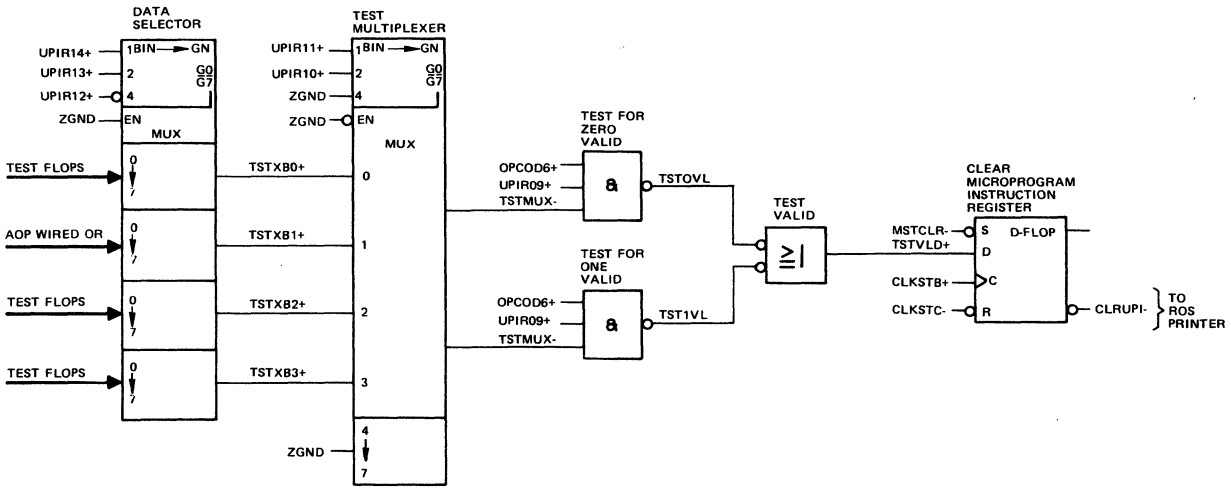


Figure 3-13 Test Multiplexer

3.10 RANGE AND OFFSET RANGE FUNCTIONALITY

The range counter is 16 bits wide and is loaded with the number of bytes to be transferred during the operation that is being set up. The offset range counter is 16 bits wide and is loaded with the number of bytes to be discarded from the beginning of a data transfer prior to the transfer of any data to main memory. The offset range counter is used only for read operations.

The range and offset range counter logic is shown in Figure 3-14. Whenever the A Operand Equals 7 (AOPEQ7+), the range counters (RNGX00 through RNGX07) can be loaded with the contents of the bus address register (BBAD16 through BBAD23). If both the range and offset range counters are to be loaded, the information is shifted through the counters until they are full.

Whenever Range Decrement (RNGDEC+10) is active, the range or offset range counter is decremented. If the offset range counter is equal to zero, the range counter is decremented; otherwise, the offset range counter is decremented.

The range decrement signal is enabled through a 5-signal OR gate as follows:

1. During a write, if the first decrement did not produce an end or range (RNGCR3-), a second decrement is generated.
2. During a read, whenever a byte is shipped from the adapter to the output bus, the range is decremented.
3. This gate allows the range and offset range counters to be loaded.
4. When doing a write and a bus cycle is generated, a decrement is executed if the hardware has not been set in the byte mode (BYTXXX-).
5. This gate enables a decrement via firmware control.

A test gate is connected to both the range and offset range counters to determine when they are empty. When the offset range is empty, the carry out (ORCAR3-) and function FRBTTT+ set the offset range flip-flop equal to zero (OFRNGZ), which notifies firmware that the next data transfer is valid data and the range counter should be decremented. Until the offset range counter equals zero, the data to be transferred is discarded.

The end-of-range gate (EORTST-) notifies firmware that the range counters are empty and that no further data transfers can be executed.

The load for the low order eight bits of the offset range counter is the ALU AOP multiplexer.

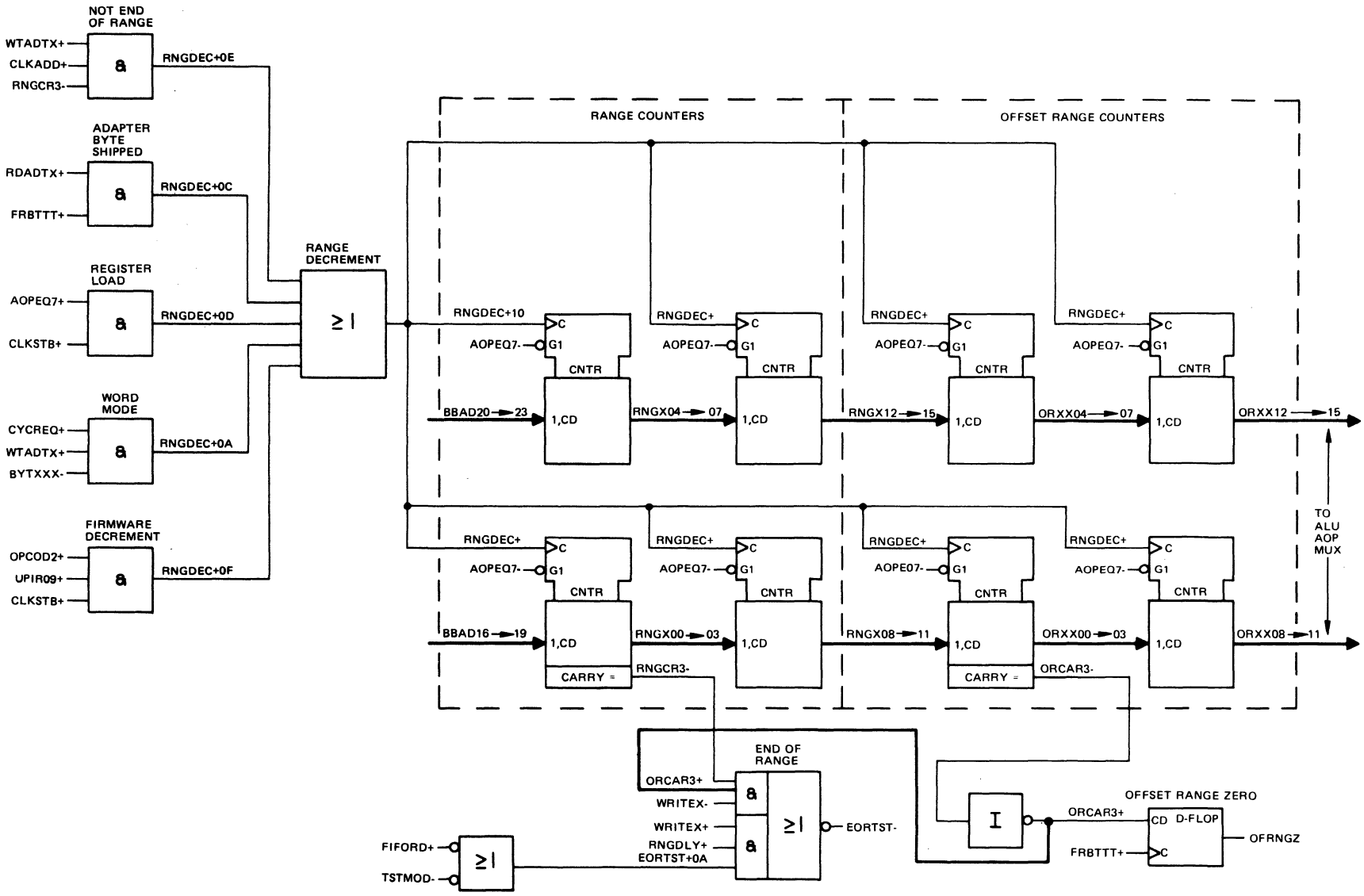


Figure 3-14 Range and Offset Range Counter

HONEYWELL PROPRIETARY AND CONFIDENTIAL

3.11 ADAPTER CONTROL LOGIC

The signals that control the transfer of information between the MPDC and the device adapter are shown in Figure 3-15 and described in the following text.

The Bus Cycle Active Test (BCATST+) signal is used in a Storage Module subsystem only, and notifies the device adapter that the MPDC is executing a bus cycle or that the MPDC has issued an Acknowledge response.

The Clock Signal (CLKSIG+) line is used in a Storage Module subsystem only, and is used as part of a test clock for that subsystem.

The Clock Strobe (CLKSTB+) signal, located on the MPDC/device adapter interface, is not used at the present time.

The Master Clear (MSTCLR-) signal is used with a Storage Module subsystem only and clears flip-flops and registers in that subsystem when initiated.

The Adapter Strobe signals (ADSTB0 through ADSTB7) are generated through a decoder by firmware MPIR bits and enable logic areas of the adapter for the transfer of data or control information from the MPDC. Table 3-6 identifies the decoded signals and depicts the adapter logic areas enabled by these signals.

The Clear Adapter signal (CLRADP-) is sent to the device adapter whenever a Master Clear (MSTCLR-) signal is received by the MPDC or when a Reset Device adapter command (RDACMD+) is issued by firmware and Clock Signal (CLKSIG-) is active.

The Enable Hardware (EHWXXX+) signal is enabled when firmware instructs the MPDC to enable its read/write data paths through a Miscellaneous command. The EHWXXX+ signal is gated to the service request logic in the device adapter.

The Adapter Enable (ADPENB-) signal is issued whenever the read/write data paths are enabled or when the device channel is ready. The ADPENB- signal is gated to the device adapter input multiplexer and is the enable gate for transfers from the device adapter to the MPDC.

The Adapter Multiplexer Select (ADPMS0+ and ADPMS1+) lines are enabled through firmware and select the information transfer that is to be delivered from the device adapter to the MPDC. The bit configurations and the adapter data input lines selected are shown in Table 3-7.

The Adapter Data In (ADPDS0 through ADPDS7) lines are used to transfer the information requested by the adapter multiplexer select lines to the MPDC's AOP wired-OR gate. (See Table 3-7.)

The Hardware Data Service Request (HDTSRQ+) signal notifies the MPDC that the device adapter has a data byte ready to be taken. If Enable Hardware (EHWXXX+) is set, the MPDC enables its hardware data service request line and sets up for the transfer.

The Arithmetic Logic Unit (ALUOT0 through ALUOT7) lines handle all information transfers from the MPDC to the device adapter.

The end-of-range test (EORTST-) gate notifies the device adapter when the range has been exhausted. This signal is generated through either of two gates, for a read or write operation. During a write operation, if the end of range is reached, Range Delay sets (RNGDLY+05) with Write Mode (WRITEX+) and End of Range Test (EORTST+0A) to indicate the end of range. For a read operation, function WRITEX- from the adapter is set with a Range Carry Out (RNGCR3+) and an Offset Range Carry Out (ORCAR3+), indicating the end of range. The End of Range Test signal mnemonic becomes ADPEOR- in the cartridge disk device adapter.

Four signals from the device adapter to the test multiplexers in the MPDC indicate the status of conditions in the device adapter. The signals are Firmware Data Service Request (FDTSRQ+), Nondata Service Request (NDTSRQ+), Device Adapter Busy (BUSYXX+), and No Hardware Transfer Request (NORQT4+). The mnemonic for the NORQT4+ signal in the device adapter is NOHTRQ+.

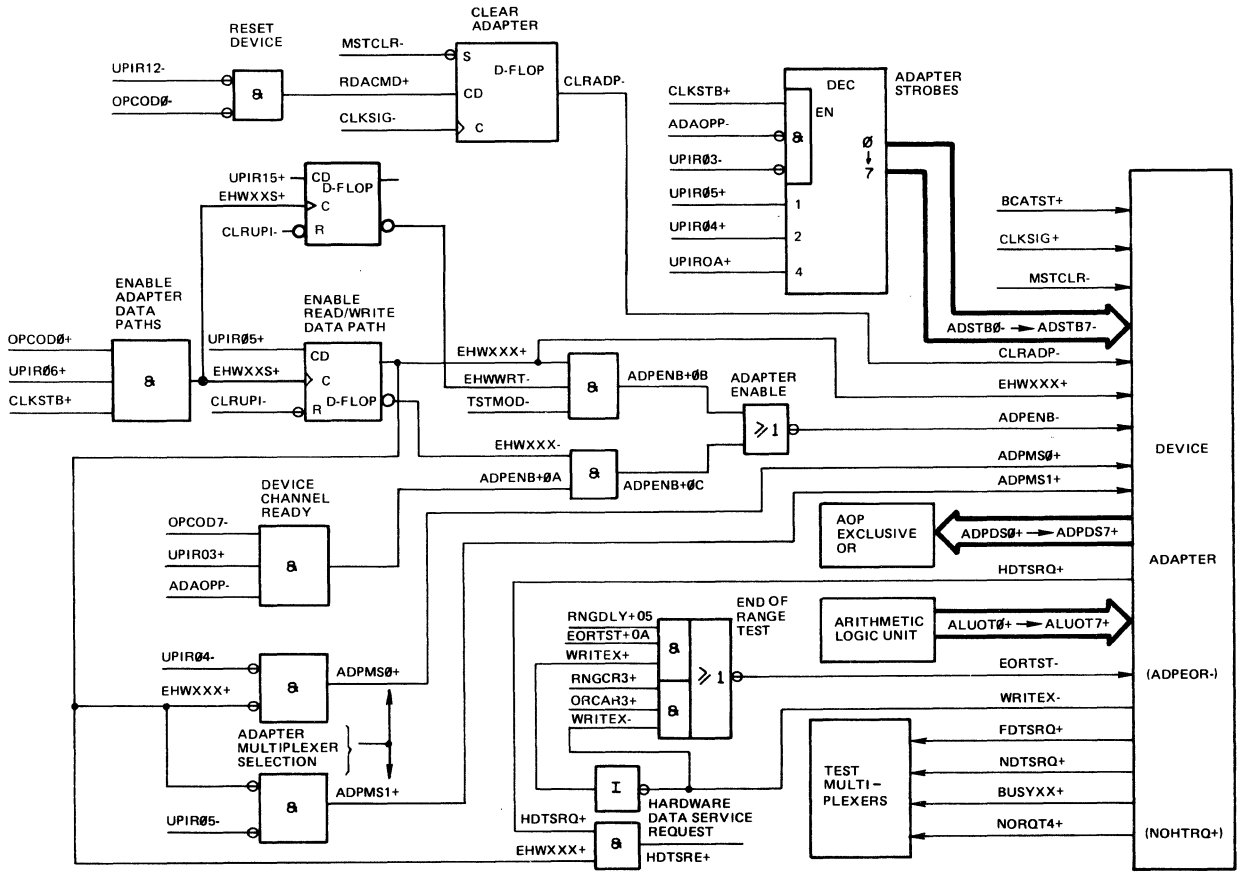


Figure 3-15 Adapter Control Logic

Table 3-6 Adapter Strobe Functionality

MPIR BITS (UPIR05+, UPIR04+, UPIR0A+) 4	2	1	ADAPTER STROBE	FUNCTION: CARTRIDGE DISK ADAPTER	FUNCTION: STORAGE MODULE ADAPTER
0	0	0	ADSTB0-	Load FIFO Buffer	Load FIFO Buffer
0	0	1	ADSTB1-	Load Data Counter	Load Device Command Register
0	1	0	ADSTB2-	Load Read/Write Command	Clear FIFO, Status, and Adapter Strobes (ALUOT0 = 1) Load Adapter Command Register (ALUOT0 = 0)
0	1	1	ADSTB3-	Load Selection Logic	Load Data Counter
1	0	0	ADSTB4-	Reset Index Count	Load Data Counter
1	0	1	ADSTB5-	Clear Status and FIFO	Load Tag Register
1	1	0	ADSTB6-	Seek Pulse	Not Used
1	1	1	ADSTB7-	Data Byte Taken	Data Byte Takne/Select FIFO Output for Transfer to the MPDC.

Table 3-7 Input Multiplexer Information Selection

INPUT SELECT (ADPMSL+, ADPMS0+) 2	1	INFORMATION: CARTRIDGE DISK ADAPTER	INFORMATION: STORAGE MODULE ADAPTER
0	0	Data In	EDAC Code*
0	1	Device Identification	EDAC Code*
1	0	Adapter Status	Adapter Status
1	1	Not Used	Device Status, Device ID, etc. as defined by the Concurrent Tag Code

*Error Detection and Correction Code.

IV THEORY OF OPERATION - CYCLE FLOW

The firmware associated with the MPDC subsystem is peculiar to the type of device adapter attached to the MPDC. Not only do the device-specific routines differ, the MPDC setup routines differ as well. For information pertaining to firmware, refer to either the Type CDM9101 Cartridge Disk Adapter Manual, Document No. 71010290-200, Order No. FL34, or the Type MSC9102 Storage Module Adapter Manual, Document No. 71010429-100, Order No. FN80, depending on the type of device adapter attached to the MPDC.

PLEASE FOLD AND TAPE –
NOTE: U. S. Postal Service will not deliver stapled forms

M&TO HARDWARE PUBLICATIONS
USER COMMENTS FORM

DOCUMENT TITLE: _____

PART NO.: _____

ORDER NO.: _____

ERRORS:

HOW DO YOU USE THIS DOCUMENT?

THEORY ☐

MAINTENANCE ☐

TROUBLESHOOTING ☐

OTHER: _____

DOES THIS MANUAL SATISFY YOUR REQUIREMENTS?

YES ☐ NO ☐

IF NOT, PLEASE EXPLAIN _____

FROM: NAME _____ DATE _____

TITLE _____

COMPANY _____

ADDRESS _____

FIRST CLASS
Permit No. 39531
Waltham, Ma.

Business Reply Mail NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

HONEYWELL INFORMATION SYSTEMS INC.
200 SMITH STREET
WALTHAM, MA. 02154

MAIL STATION 872A
HARDWARE PUBLICATIONS, BILLERICA

Honeywell

Honeywell

Honeywell Information Systems
In the U.S.A.: 200 Smith Street, MS 486, Waltham, Massachusetts 02154
In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario M2J 1W5
In Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F.

FM54, Rev. 1