
digital
RH780
MASSBUS ADAPTER
TECHNICAL DESCRIPTION
VAX11
780

EK-RH780-TD-001

RH780 Massbus Adapter Technical Description

digital equipment corporation • maynard, massachusetts

First Edition, March 1979

Printed in U.S.A.

CONTENTS

CONTENTS (Cont)

CONTENTS (Cont)

FIGURES

FIGURES (Cont)

TABLES

CHAPTER 1
INTRODUCTION

1.1 GENERAL

The RH78Ø Massbus Adapter (MBA) is the interface between the Synchronous Backplane Interconnect (SBI) and Massbus storage devices (disk and tape). The RH78Ø is used with the VAX-11/78Ø processor to transfer data between mass storage devices and main memory. The processor can accommodate up to seven MBAs. Each MBA can be used with up to eight drives.

1.1.1 Scope

This manual is intended to be used as a training resource and as a field reference guide for the RH78Ø MBA.

1.1.2 Related Documentation

Table 1-1 lists related documentation that supplements the information in this manual.

Table 1-1 Related Documentation

Title	Document Number
RH78Ø Print Set†	B-DD-RH78Ø
RPØ5/RPØ6 Device Control Logic Maintenance Manual†	EK-RPØ56-MM
RPØ5/RPØ6 Disk Drive Installation Manual*	EK-RPØ56-IN
Digital Logic Handbook*	Ø58.ØØ173.25Ø5
Memorex RPØ5/RPØ6 Operation and Maintenance Manual†	EK-RPØ5M-MM
Memorex RPØ5/RPØ6 677-Ø1/677-51 Disk Storage Drive Illustrated Parts Catalogue†	EK-RPØ5M-IP
Memorex RPØ5/RPØ6 8ØØ Disc Storage Subsystem Tester Operator's Manual†	EK-RPØ5M-OP
Memorex RPØ5/RPØ6 677-Ø1 Logic Manual†	EK-RPØ5M-TM
VAX-11/78Ø Architecture Handbook*	
VAX-11/78Ø Central Processor Technical Description†	EK-KA78Ø-TD

Table 1-1 Related Documentation (Cont)

Title	Document Number
VAX-11/780 System Installation Manual*	EK-SI780-IN
VAX-11/780 Diagnostic System User's Guide*	EK-DS780-UG
TE16/TE10W/TE10N DECmagtape Transport Maintenance Manual†	EK-TE16-MM
TE16/TE10W/TE10N DECmagtape Transport User Manual*	EK-TE16-OP
TU45A Magnetic Tape Subsystem Maintenance Manual†	EK-TU45A-TM
TM03 Magnetic Tape Formatter Technical Manual†	EK-TM03-TM
TM03 Magnetic Tape Formatter User's Manual*	EK-TM03-OP
RM03 Disk Drive Technical Manual†	ER-RM03-TM
RM03 Disk Drive Maintenance Print Set†	ER-RM03-MP

* Hard copy only.
† Microfiche and hard copy.

1.2 MASS STORAGE SUBSYSTEMS

Figure 1-1 illustrates a typical mass storage subsystem. It is beyond the scope of this manual to discuss, in detail, the various configurations (tape and disk) that can be used with the MBA to store data. Throughout this manual a Massbus device is defined as a mass storage device (drive) and its associated Drive Control Logic (DCL) or formatting interfaces. An explanation of the basic components in the subsystem is presented in subsequent paragraphs.

1.2.1 Synchronous Backplane Interconnect (SBI)

The SBI is a bidirectional information path for data exchanges between the central processor (CPU), memory, and adapters of the VAX-11/780 system. The SBI provides checked, parallel information exchanges synchronous with a common system clock.

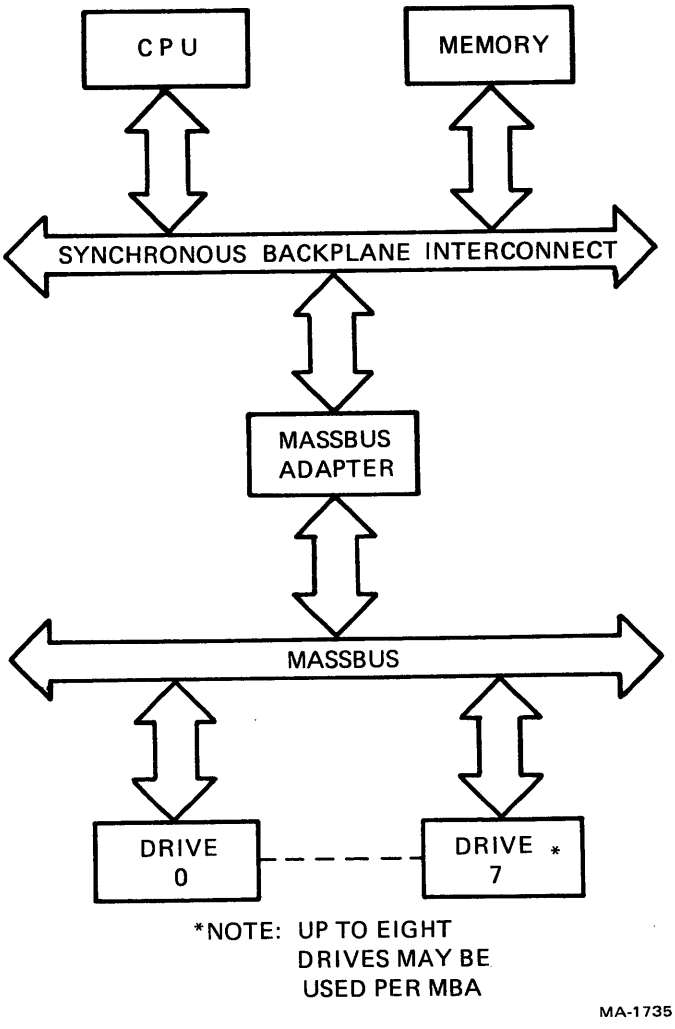


Figure 1-1 Typical Mass Storage Subsystem

A communications protocol allows the information path to be time multiplexed such that several data exchanges can be in progress simultaneously. In each clock period (or cycle) the next cycle's interconnection arbitration, or information exchange, and transfer confirmation about information exchange two cycles ago can occur in parallel.

Every 200 ns SBI signals are clocked into data latches. All checking and subsequent decision making is based on these latched signals. Error checking logic in every SBI device detects and reports single bit failures in the information path.

The SBI has the following characteristics:

- Distributed arbitration
- 200 ns bus cycle time
- 32-bit data width
- 28 bits of physical address space
- 13.3 megabyte maximum data transfer rate

The following terms are defined for SBI-specific units and operations:

a. Nexus -- a physical connection to the SBI capable of any or all of the functions described in b -- e.

b. Commander -- a nexus that transmits command and address information.

c. Responder -- a nexus that recognizes command and address information which is directed to it and requires a response.

d. Transmitter -- a nexus that drives the signal lines.

e. Receiver -- a nexus that samples and examines the signal lines.

1.2.2 Massbus

In the VAX-11/780 system the Massbus connects the drive to the MBA. The Massbus is composed of two separate, independent buses: control bus and data bus. These independent buses allow for the exchange of control information and data between the MBA and its drives.

The data bus provides a bidirectional, parallel data path (16 bits plus 1 parity bit) between the MBA and its drives. Data is transferred synchronously, using a clock generated in the drive. Only one drive can transfer data at any one time, with the data rate being drive dependent, and the data bus dedicated to a single drive for the duration of a data transfer operation. The asynchronous control bus provides the parallel control and status path (16 bits plus 1 parity bit). The control bus is used to read and write registers within the drives and to command the drives to transfer data over the data bus.

1.3 MASSBUS ADAPTER

The MBA is the interface between the SBI and the high-speed Massbus device (disk and tape). It consists of an SBI/MBA interface board, an internal registers board, a control path board, and a data path board. Figure 1-2 is a simplified block diagram of the MBA. A tristate internal bus connects the SBI interface to the other boards and provides for the passage of data between them.

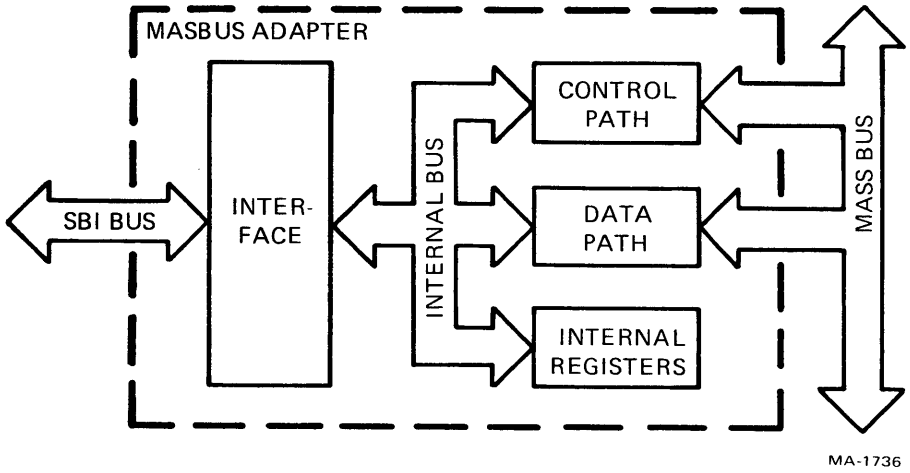


Figure 1-2 MBA Block Diagram

The MBA accepts and executes commands from the CPU and reports the necessary status changes and fault conditions to the CPU. The MBA will accept a CPU command to read or write a register within the MBA or within a drive. The MBA always monitors the data written to drive registers, thereby knowing when to begin a data transfer and what kind of a transfer it is.

Special diagnostic features are built into the hardware to allow on-line diagnosis of the MBA and Massbus drives. The following are features of the MBA.

1. The MBA handles a Massbus drive with a maximum data transfer speed of 1 us per 16 bits.

2. The Massbus data path is 16-bits wide; 18-bit data is not handled by the MBA.

3. A silo (32-byte deep data buffer) smoothes out data transfers between the SBI and the Massbus drives.

4. The MBA can be exercised, through diagnostic features, with no drives on the Massbus.

1.3.1 MBA/SBI Interface

The MBA examines the information on the SBI for every SBI bus cycle. It checks the parity of the data, decides if the MBA is the receiving nexus, and acts accordingly. The SBI interface board contains logic to accomplish preliminary SBI command/address decoding and generation of internal timing signals from the timing sources on the SBI.

1.3.2 MBA Internal Registers

There are two sets of registers in the MBA address space: internal and external. The MBA internal registers are the registers that are physically located in the MBA. The external registers are located in the Massbus drives and are drive dependent.

There are eight internal registers and a 256 X 32-bit RAM. The primary function of the internal registers is to control the MBA and monitor operating status conditions. The internal registers also control certain phases of data transfers between the SBI and the Massbus device, such as maintaining a byte count to ensure that all of the data to be transferred has been accounted for, and converting virtual addresses to physical addresses to read or write data in memory.

The eight internal registers are listed as follows:

RS = ØØ	MBA Configuration/Status Register (CSR)
RS = Ø1	MBA Control Register (CR)
RS = Ø2	MBA Status Register (SR)
RS = Ø3	MBA Virtual Address Register (VAR)
RS = Ø4	MBA Byte Count Register (BCR)
RS = Ø5	MBA Diagnostic Register (DR)
RS = Ø6	MBA Selected Map Register (SMR)
RS = Ø7	MBA Command Address Register (CAR)

NOTE

Registers Ø6 and Ø7 are read-only registers and are valid only during data transfers.

The MBA contains a 256 X 32-bit RAM (bits 21--3Ø read as Ø) that maps virtual addresses from the virtual address register into SBI physical addresses. The mapping registers allow transfers to or from contiguous or noncontiguous physical memory.

1.3.3 Control Path

The control path handles the transfer of control data to and from the Massbus devices. It contains logic to select the Massbus device and device register and to perform the register transfer and determine the data transfer function (if any) to be performed (read, write, write check). The control path also coordinates the control data function with other MBA and SBI activity.

1.3.4 Data Path

The data path controls the manner in which data is transferred to and from the Massbus device and the SBI. These circuits divide the 32-bit SBI data word into 16-bit (2 bytes) segments required as input by the Massbus and its devices when performing a write function. When performing a read from a Massbus device, the data path assembles the two 8-bit bytes from the Massbus into the 32-bit SBI format. A silo and I/O data buffer provide the means for smoothing the data transfer rate. The data path also contains a write check circuit that allows the user to verify the accuracy of a preceding data transfer function.

1.4 MBA SBI OPERATIONS

An SBI write operation is specified as a data transfer from the SBI to the MBA or Massbus device. This data transfer requires two SBI cycles. The first cycle contains the command/address; the second cycle contains the data word.

An SBI read operation is specified as a data transfer from the MBA or Massbus device to the SBI. An SBI read operation requires two SBI cycles. The first cycle is a command/address to the MBA specifying a read operation. The MBA then accesses the requested register contents. Several SBI cycles later, the MBA will arbitrate for use of the SBI and send the requested data back to the CPU.

The MBA only accepts aligned longword (32 bits) register reads and writes. The following paragraphs provide a basic description of the various functions the MBA supports. These functions will be described in more detail later in this manual.

1.4.1 Write to Internal Registers

SBI data is constantly checked by the MBA. When a valid command/address (one whose address is within the range recognized by the MBA) is decoded, it is latched in the SBI/MBA interface transceivers. The MBA decodes a section of the address range which selects registers internal to the MBA. If the function to be performed is a write, and the MBA's SBI interface is not busy, the command will be accepted. The selected register address is latched in the internal register board. The next SBI cycle will contain the data to be written. It is then passed through the internal bus to the internal register board and written into the selected register or map. Certain registers can be written when the MBA is processing a data transfer; however, most registers cannot. Any attempt to write to a register that is not allowed will set the Programming Error (PGE) bit in the status register. The MBA will not modify the intended register and the data transfer in progress will continue.

A confirmation signal informs the transmitting device that the command/address has been decoded and validated and the data has been received correctly.

1.4.2 Write To External Registers

Receipt and initial processing of the command/address for an external write function is the same as that for a write to internal registers, except the address specifies a register within a Massbus device. The MBA will not accept another SBI command until the write to external register function is complete. All command/address and data words are applied to the internal bus drivers as they are latched in the SBI interface transceivers. The address is made available to the internal registers, control paths, and data paths via the internal bus. Following receipt of the command/address (assuming the decoding and validation process is performed satisfactorily), the command/address is loaded into the control path internal bus receivers, then latched into the

control path address storage registers and applied to the Massbus control path address lines. Data is loaded into the control path lines. When the proper protocol between the MBA and the drive is exchanged, the drive has accepted the data. The MBA busy logic is then cleared and the write to external register function is completed. The MBA is now ready to accept another SBI command.

1.4.3 Read Internal Register

The command/address, after decoding and validation, is loaded into the MBA to select the internal register or map from which data is to be read. The decoding process instructs the MBA to retrieve the addressed data word and transfer it to the specified destination. The MBA then saves the destination information, issues a comand/address acknowledge to the SBI, and accesses the requested register's content. Before the MBA can transfer data to the requester, it must first arbitrate for control of the SBI. Once the MBA gains control of the SBI, data is then transferred to its destination.

1.4.4 Read External Registers

If the command/address received from the SBI is one that selects a register in a drive, the address is sent to the control path, which selects the device and the register within the device from which data is to be read. During this time a confirmation signal is sent to the SBI indicating that the command/address has been received by the MBA. After the proper Massbus protocol has taken place, the requested data is strobed into the Massbus control path receivers. Data is then transferred to the internal bus and the MBA will arbitrate for control of the SBI. When the MBA gains control of the SBI, the requested data can be transferred to its specified destination.

1.5 MASSBUS DATA TRANSFERS

In order to initiate a data transfer, specific registers within the MBA and the selected drive must be loaded (the programmer's handbook provides further details). The last register to be loaded within a device is the control register. The loading of a drive's control register with a valid command will cause both the drive and the MBA to prepare for a transfer. The drive will seize the data bus for the duration of the transfer and the MBA will prepare to move data between the device and memory. The MBA buffers the data and transfers eight bytes at a time to and from memory (this eight byte quantity is a quadword). Three SBI cycles are needed to transfer a quadword: one for the command/address, one for the first four bytes of data, and one for the last four bytes of data of the quadword. Once the specified number of bytes have been transferred, the MBA informs the drive that the transfer has terminated. At this time the drive disconnects from the data bus and the MBA may interrupt the CPU to inform it that the transfer has been completed.

1.5.1 Write To Massbus Device

The MBA constantly monitors data sent to the drives via the Massbus control path. If the MBA sees a write command to a drive, it will initialize itself in preparation for the transfer (data path cleared). The virtual address supplied by the program is translated, via the MAPs, into a physical address. Virtual addressing makes data storage appear as if all of the information transferred from memory is stored in successive fashion (contiguous pages).

Virtual address translation is transparent to the user and takes place under system control. Figure 1-3 illustrates the virtual address translation process.

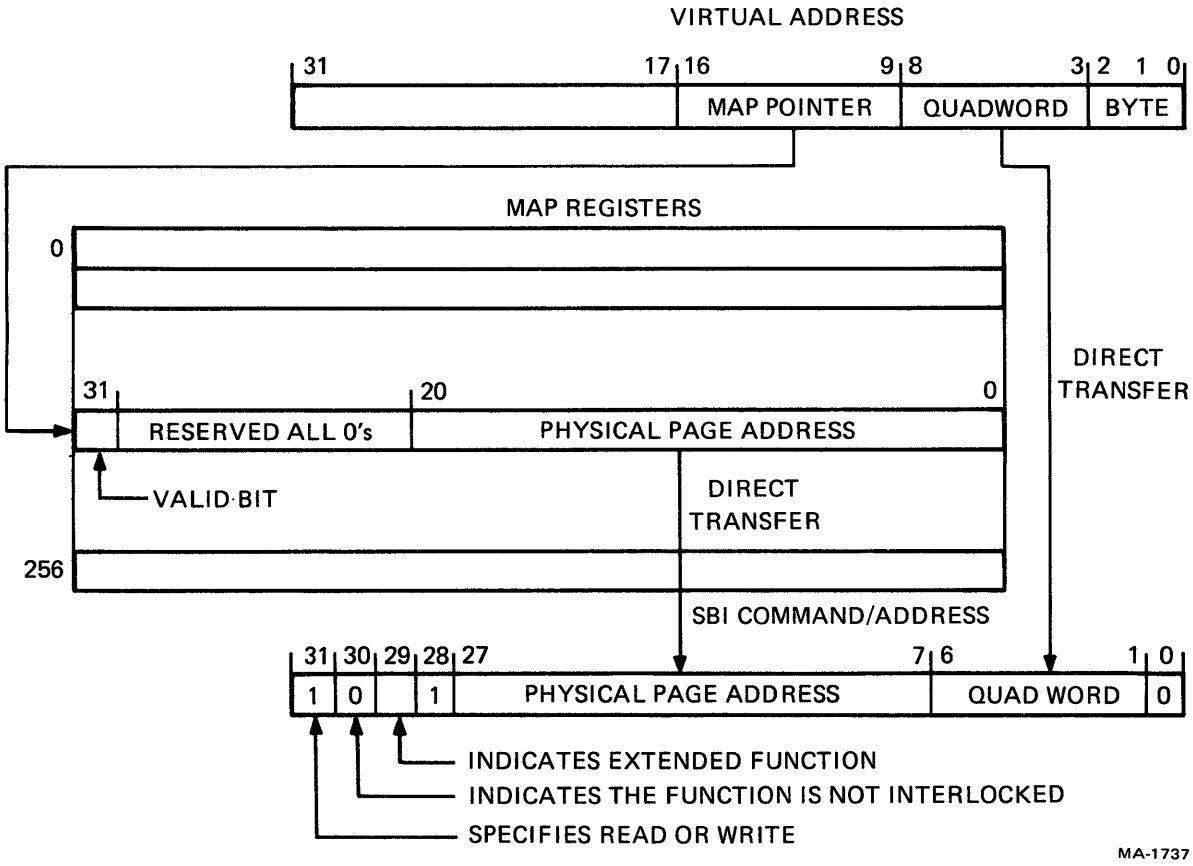


Figure 1-3 Virtual Address Translation

Bits 9 through 16 of the virtual address specify one of 256 MAP registers in the MBA. The MAP register contains the physical page address of the data and a valid bit (bit 31) that indicates the integrity of the information in this register. The valid bit must be set in order to use this register. Bits 3 through 8 of the virtual address specify the address of the quadword in the physical memory page pointed to by the MAP register. This value is directly transferred to bits 1 through 6 of the physical address. Bits Ø through 3 of the virtual address specify the next byte of the quadword to be loaded into or from the internal silo.

Virtual address translation is checked to ensure that map information is valid and that there are no parity errors. Invalid map information or parity errors will cause the transfer to be aborted.

Once the MBA sees a write command issued to the control register of a drive, it will clear its data path and begin prefetching the data from memory. The prefetched data is then loaded into the MBA's silo.

The MBA requests data by sending a command/address to memory instructing it to transfer a quadword (eight bytes) from the specified location to the MBA. Memory will respond to the command/address by issuing one of four confirmation outputs.

No Response (NR)	Command/address will be retransmitted to memory until an MBA timeout occurs.
Busy (BSY)	Command/address will be retransmitted to memory until ACK is received.
Error (ERR)	Transfer aborted.
Acknowledge (ACK)	Indicates memory has received the command/address correctly.

When memory has accessed the requested data, arbitrated, and obtained the SBI, the data will be transferred to the MBA with the proper identification and status codes. The MBA will then transfer the data through its silo onto the Massbus (two bytes at a time) and begin another SBI transfer to obtain the next eight bytes of data from memory. Eventually the byte counter within the MBA will go to zero and the transfer will be complete.

1.5.2 Write Check Data Transfer

The write check function is used to verify the integrity of data that has been written to a drive. During a write check, the data from the drive is compared with the data in memory. On the MBA, the write check function is essentially the same as a write function, except data is clocked into a write check buffer instead of the Massbus drivers. This is then compared with the Massbus data received from the drive. If a mismatch occurs, an error bit is set and the write check function will be aborted.

1.5.3 Read From Massbus Device

The command specifying the initiation of a read from a Massbus device is processed in the same manner as described in Paragraph 1.5.1. If the command/address is processed satisfactorily, the data path circuit will be cleared and data will be read from a Massbus device. Device selection and the location within the device from which data is to be read is specified by previous writes to the MBA and the drive. Data from the Massbus will then be loaded into the Massbus input data buffers and a silo input operation will be initiated. Data from the Massbus data input buffers is loaded into the silo one byte at a time. As data is being loaded into the silo from the Massbus, other data may be transferred from the silo to the output buffer registers. After the eight bytes are loaded into the output buffer register, the MBA will initiate a write to memory operation. Virtual addresses are translated into physical address as described in Paragraph 1.5.1. When the proper confirmation signals are received, the output buffer is cleared and more data will be loaded to be transferred to memory. Once the byte count register goes to Ø, the data transfer operation is terminated and the MBA will be ready to accept the next data transfer command.

1.5.4 Data Transfer Rate

The data transfer rate from the drive is determined by a clock in the Massbus device. The memory transfer rate depends on cycle arbitration time and memory cycle time. The MBA can handle transfer rates of up to 1 us/word.

MBA Specifications

Packaging	Four extended hex board slots of backplane plus one paddle card slot for cable connection
Power Requirements	+5 Vdc, 35 A, 175 W -5 Vdc, 1 A, 5 W Total wattage < 18Ø W
Operating Configuration	The minimum operating configuration consists of a CPU, a memory, and at least one Massbus device. (A special diagnostic feature enables the Massbus to be checked with no Massbus device.)

CHAPTER 2
MASSBUS AND SYNCHRONOUS BACKPLANE INTERCONNECT

2.1 MASSBUS

The Massbus provides the interface between the MBA and the Massbus drives (Figure 2-1). The total external Massbus cable can be up to 49 m (160 ft) in length; up to eight drives can be connected in a daisy-chain configuration. The Massbus consists of two sections: a data bus and a control bus. These buses are described in the following paragraphs.

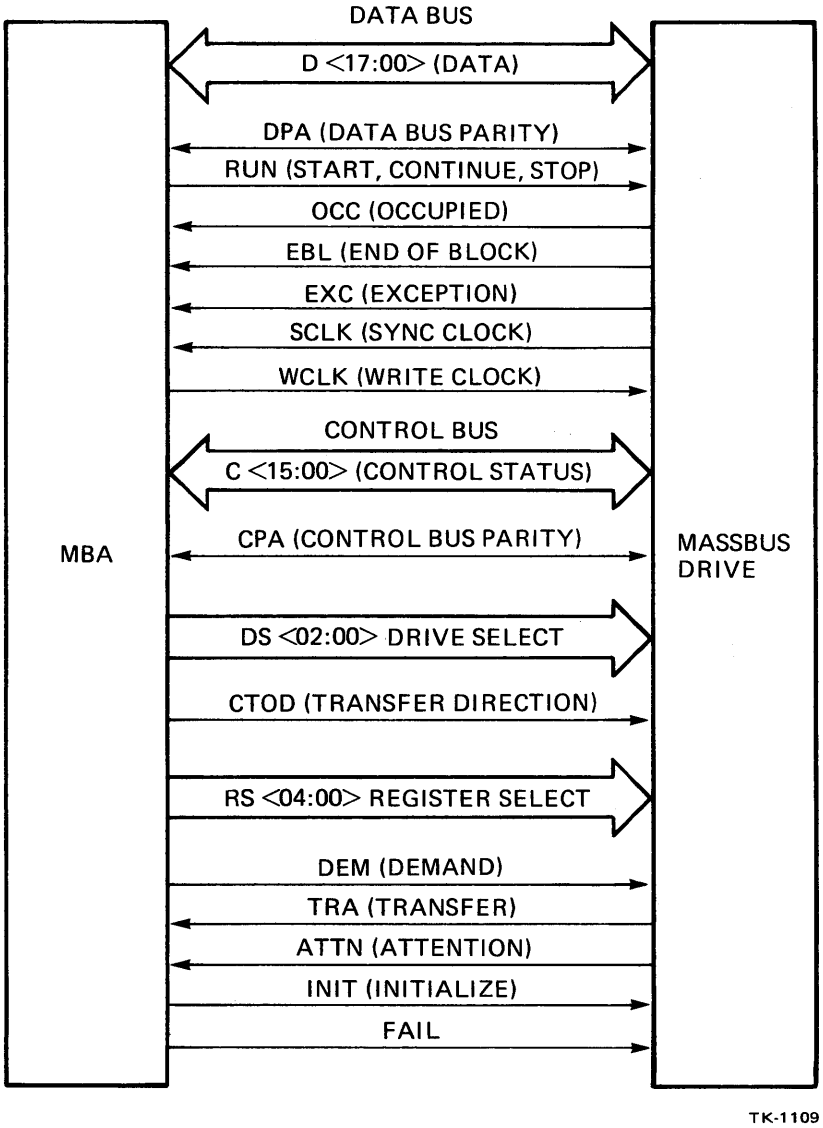


Figure 2-1 Massbus Interface

2.2 DATA BUS

The data bus section of the Massbus consists of a 17-bit (16 data bits plus parity bit) parallel data path and six control lines (Figure 2-1). The control lines are described in the following paragraphs.

2.2.1 Parallel Data Paths

The parallel data path consists of an 18-bit plus parity bus. The data path is bidirectional and employs odd parity. Data is transmitted synchronously, using a clock generated in the drive. The MBA only transfers 16 bits at a time; thus, the drive must have its 16-bit format bit set in the drive control logic. Bits 17 and 18 are always unasserted.

2.2.2 RUN

After a data transfer command has been written into the control register of a drive, the drive connects to the data bus. The MBA asserts the RUN line to initiate the function. At the end of each sector, on the trailing edge of the EBL (End of Block) pulse, RUN is strobed by the drive. If it is still asserted, the function continues for the next sector; if negated, the function is terminated.

2.2.3 Occupied (OCC)

This signal is generated by the drive to indicate "data bus busy." As soon as a valid data transfer command is written into a drive, the drive asserts OCC. Various errors can prevent a drive from executing a command. The controller will timeout in these cases, due to no assertion of OCC or of SCLK (Sync Clock), and the MXF (Missed Transfer) error will be set in the controller. OCC is negated at the trailing edge of the last EBL pulse of a transfer.

2.2.4 End of Block (EBL)

This signal is asserted by the drive for 2 us at the end of each sector (after the last SCLK pulse). For certain error conditions, where it is necessary to terminate operations immediately, EBL is asserted prior to the normal time for the last SCLK. In this case, the data transfer is terminated prior to the end of the sector.

2.2.5 Exception (EXC)

This signal is asserted by the drive or the MBA when an abnormal condition occurs during a data transfer. The drive asserts this signal to indicate an error during a data transfer command (read, write, or write check). EXC is asserted at, or prior to, assertion of EBL and is negated at the negation of EBL.

2.2.6 Sync clock (SCLK), Write Clock (WCLK)

These signals are the timing signals used to control the strobing of the data in the controller and/or the drive. During a read operation, the MBA strobes the data lines on the negation of SCLK and the drive changes the data on the assertion of SCLK. During a write operation, the controller receives an SCLK and echoes it back to the drive as WCLK. On the assertion of WCLK, the drive strobes the data lines; on the negation of WCLK, the controller changes the data on the data lines.

2.3 CONTROL BUS

The control bus section of the Massbus consists of a 17-bit (16 bits plus parity) parallel control and status data path and 14 control lines (Figure 2-1), which are described in the following paragraphs.

2.3.1 Parallel Control

The parallel control path consists of a 16-bit parallel data path designated C<15:00> and an associated parity bit (CPA). The control lines are bidirectional and employ odd parity.

2.3.2 Drive Select (DS<02:00>)

These three lines transmit a 3-bit binary code from the MBA to select a particular drive. The drive responds when the selected (unit) number in the drive corresponds to the transmitted binary code.

2.3.3 Controller to Drive (CTOD)

This signal is generated by the MBA and indicates the direction in which control and status information is to be transferred. For a controller to drive transfer, the controller asserts CTOD. For a drive to controller transfer, the controller negates this signal.

2.3.4 Register Select (RS<04:00>)

These five lines transmit a 5-bit binary code from the controller to the selected drive. The binary code selects one of the drive registers.

2.3.5 Demand (DEM)

This signal is asserted by the controller to indicate that a transfer is to take place on the control bus. For an MBA to drive transfer, DEM is asserted by the MBA when data is present and settled on the control bus. For a drive to controller transfer, DEM is asserted by the MBA to request data and is negated when the data has been strobed off the control bus. In both cases, the RS, DS, and CTOD lines are generated and allowed to settle before assertion of DEM.

2.3.6 Transfer (TRA)

This signal is asserted by the selected drive in response to DEM. For an MBA to drive transfer, TRA is asserted after the data has been strobed and is negated after DEM is negated. For a drive to controller transfer, TRA is asserted after the data has been gated onto the bus and negated after the negation of DEM is received.

2.3.7 Attention (ATTN)

This line is shared by all eight drives attached to an MBA; it may be asserted by any drive as a result of an abnormal condition or status change in the drive. An Attention Active (ATA) status bit in each drive is set whenever that drive is asserting the ATTN line. ATTN can be asserted due to any of the following conditions.

1. An error occurring while no data transfer is taking place (asserted immediately).

2. Upon completion of a data transfer command if an error occurred during the data transfer (asserted at the end of the data transfer).

3. Upon completion of a mechanical motion command (seek, recalibrate, etc.) or a search command.

4. As a result of the Medium On Line (MOL) bit changing states (except in the unload operation). In the dual MBA configuration, a change in state of MOL will cause the assertion of ATTN to both MBAs.

The ATA bit in a drive can be cleared by the following actions.

1. Asserting INIT on the Massbus (affects all eight drives).

2. Writing a 1 into the attention summary register (in the bit position for this drive). This clears the ATA bit; however, it does not clear the error.

3. Writing a valid command (with the GO bit asserted) into the control and status register if no error occurs. Note that clearing the ATA bit of one drive does not always cause the ATTN line to be negated, because other drives may also be asserting the line.

NOTE

There are three cases in which ATA is not reset when a command is written into the control and status register (with the GO bit set): 1) if there is a control bus parity error in the write, 2) if an error was previously set, or 3) if an Illegal Function (ILF) code is written.

2.3.8 Initialize (INIT)

This signal is asserted by the MBA to perform a system reset of all drives. It is asserted when a 1 is written into the INIT bit (bit Ø1 of MBA CR). When a drive receives the INIT pulse, it immediately aborts the execution of any current command and performs all actions described for the drive clear command.

NOTE

In the dual-MBA configuration, a drive will honor an INIT pulse only from the MBA that has seized the drive, or from either controller if the drive is in the unseized state.

2.3.9 FAIL

When asserted, this signal indicates that a power-fail condition has occurred in the MBA or the MBA is in the maintenance mode. While FAIL is asserted, the drive inhibits reception of the INIT and DEM signals at the drive.

2.4 COMMAND INITIATION

To initiate a command in a drive via the Massbus, the MBA (or the CPU via the MBA) writes a word into the control register. The function code and GO bit are transferred to the selected drive. If the command specified is valid and the GO bit is asserted, the selected drive executes the command.

Commands are of two types: nondata transfer commands (such as drive clear, seek, etc.) and data transfer commands (such as read, write, and write check). The command function code bits (Ø5--ØØ, including GO in the control register) are Ø1--37 for nondata transfer commands and 29--3F for data transfer commands (not all are valid functions.)

2.4.1 Nondata Transfer Commands

Nondata transfer commands only affect the state of the drive. The MBA merely writes the command word (with GO bit set) into the drive's control register. At the completion of the command execution, the drive typically asserts the ATTN line to signal its completion.

If the nondata transfer command code written into the drive is not recognized by the drive as a valid command, the drive will immediately signal an error by asserting the ATTN line. The ILF error is set.

2.4.2 Data Transfer Commands

When any data transfer command code (with the GO bit set) is written into the drive's control register, the MBA expects data transfer on the data bus to begin soon thereafter. The MBA sets its DT BUSY bit as soon as the data transfer command code is written into a drive. The drive normally responds by asserting the OCC line. The MBA asserts RUN and then data is transferred to or from the specified drive, after the proper address (sector, track, cylinder) is found.

If an error occurs in a drive during a data transfer command, the drive asserts the EXC line. This line remains asserted until the trailing edge of the last EBL pulse. The MBA always negates the

RUN line when it detects EXC asserted, so that the data transfer is terminated at the end of the sector in which the error was signaled.

2.5 READING AND WRITING REGISTERS

The process of reading or writing drive registers is accomplished via the asynchronous (control bus) portion of the Massbus (Figure 2-1). The MBA initiates the action by selecting a drive DS<Ø2:ØØ>, selecting a register RS<Ø4:ØØ> in that drive, selecting a direction of transfer (CTOD), and either reading or writing the register via the 17 bidirectional control lines (C<15:ØØ> and CPA). After a deskew delay to allow the control lines to stabilize, the MBA asserts DEM. The drive, upon receiving the DEM assertion, checks the CTOD line to ascertain whether a read or write is to occur.

If a register read operation is specified, the drive will gate the contents of the specified register onto the control bus and issue TRA. When the MBA receives TRA, it will gate the control lines onto the SBI. After a deskew delay, the MBA negates DEM. The negation of DEM causes TRA to be negated and completes the operation. The MBA will then arbitrate for the SBI and transfer the Massbus data to its destination.

NOTE

Since Massbus drive registers are 16 bits wide, the MBA appends bits 31--16 of its status register to create the longword to be sent to the requester.

If a register write operation is specified, the MBA gates the control data onto the control bus when it issues DEM. The drive will transfer the data from the control bus into the specified drive register and assert TRA, which causes DEM to be negated. The negation of DEM causes TRA to be negated to complete the operation.

The Massbus structure allows a register read operation while a data transfer (on the asynchronous data bus) is taking place. Any attempt by the MBA to write a register in a drive performing a data transfer operation (except for the maintenance and attention summary registers) will cause the drive to set the Register Modification Refused (RMR) error bit.

2.6 DATA TRANSFER

Before a data transfer takes place, the selected unit, desired sector/track address, cylinder address, bus address, and word count are specified by the program. The program then transfers the read or write data transfer command (with the GO bit asserted) to the control register. Upon receipt of the data transfer command, the drive will assert OCC, indicating that the data bus is busy. The MBA logically connects to the Massbus data bus by asserting RUN and then waits for SCLK pulses from the drive. For a write

data transfer, each WCLK pulse causes a word to be written into a data register in the drive logic; for a read data transfer, each SCLK pulse causes a word to be transferred to the Massbus. When a sector of words has been written onto or read from the disk, the disk sends an EBL pulse to the MBA. If the RUN line is still asserted at this time, the next sector of data words is transferred. If the RUN line is negated, the data transfer is terminated.

2.7 MASSBUS PHYSICAL DESCRIPTION

The Massbus consists of 56 signal lines, including data, control, status, and parity. These signal lines are routed externally to the cabinet that contains the MBA(s) via three BCØ6-R Massbus cables.

At the cabinet (containing the first MBA), the BCØ6-R cable plugs into the AD-7Ø15145 connector panel, which is mounted at the lower rear of the cabinet. This connector panel has cutouts for four receptacle housing assemblies to accommodate up to four MBAs and associated cabling. The other side of the receptacle housing assembly accepts three BCØ6-S round Massbus cables. To accommodate additional MBAs, the BCØ6-R cables plug into the 7Ø13678 cabinet to cabinet connector panel, which is mounted between the cabinet verticals on the right end of the cabinet.

Table 2-1 lists the Massbus signals and their associated pin assignments.

Table 2-1 Massbus Signal Cable Designations

Cable	Pin*		Polarity	Designation
Massbus Cable A	A	1	-	MASS DØØ
	B	2	+	
	C	3	+	MASS DØ1
	D	4	-	
	E	5	-	MASS DØ2
	F	6	+	
	H	7	+	MASS DØ3
	J	8	-	
	K	9	-	MASS DØ4
	L	1Ø	+	
	M	11	+	MASS DØ5
	N	12	-	
	P	13	-	MASS CØØ
	R	14	+	
	S	15	+	MASS CØ1
	T	16	-	
	U	17	-	MASS CØ2
	V	18	+	
	W	19	+	MASS CØ3
	X	2Ø	-	
	Y	21	-	MASS CØ4
	Z	22	+	
	AA	23	+	MASS CØ5
	BB	24	-	
	CC	25	-	MASS SCLK
	DD	26	+	
	EE	27	+	MASS RS3
	FF	28	-	
	HH	29	+	MASS ATTN
	JJ	3Ø	-	
	KK	31	-	MASS RS4
	LL	32	+	
	MM	33	-	MASS CTOD
	NN	34	+	
	PP	35	+	MASS WCLK
	RR	36	-	
	SS	37	+	MASS RUN
	TT	38	-	
	UU	39		SPARE
	VV	4Ø		GND

Table 2-1 Massbus Signal Cable Designations (Cont)

Cable	Pin*		Polarity	Designation
Massbus Cable B	A	1	-	MASS DØ6
	B	2	+	
	C	3	+	MASS DØ7
	D	4	-	
	E	5	-	MASS DØ8
	F	6	+	
	H	7	+	MASS DØ9
	J	8	-	
	K	9	-	MASS D1Ø
	L	1Ø	+	
	M	11	+	MASS D11
	N	12	-	
	P	13	-	MASS CØ6
	R	14	+	
	S	15	+	MASS CØ7
	T	16	-	
	U	17	-	MASS CØ8
	V	18	+	
	W	19	+	MASS CØ9
	X	2Ø	-	
	Y	21	-	MASS C1Ø
	Z	22	+	
	AA	23	+	MASS C11
	BB	24	-	
	CC	25	-	MASS EXC
	DD	26	+	
	EE	27	+	MASS RSØ
	FF	28	-	
	HH	29	+	MASS EBL
	JJ	3Ø	-	
	KK	31	-	MASS RS1
	LL	32	+	
	MM	33	-	MASS RS2
	NN	34	+	
	PP	35	+	MASS INIT
	RR	36	-	
	SS	37	+	MASS SP1
	TT	38	-	
	UU	39		SPARE
	VV	4Ø		GND

Table 2-1 Massbus Signal Cable Designations (Cont)

Cable	Pin*		Polarity	Designation
Massbus Cable C	A	1	-	MASS D12
	B	2	+	
	C	3	+	MASS D13
	D	4	-	
	E	5	-	MASS D14
	F	6	+	
	H	7	+	MASS D15
	J	8	-	
	K	9	-	MASS D16
	L	10	+	
	M	11	+	MASS D17
	N	12	-	
	P	13	-	MASS DPA
	R	14	+	
	S	15	+	MASS C12
	T	16	-	
	U	17	-	MASS C13
	V	18	+	
	W	19	+	MASS C14
	X	20	-	
	Y	21	-	MASS C15
	Z	22	+	
	AA	23	+	MASS CPA
	BB	24	-	
	CC	25	-	MASS OCC
	DD	26	+	
	EE	27	+	MASS DS0
	FF	28	-	
	HH	29	+	MASS TRA
	JJ	30	-	
	KK	31	-	MASS DS1
	LL	32	+	
	MM	33	-	MASS DS2
	NN	34	+	
	PP	35	+	MASS DEM
	RR	36	-	
	SS	37	+	MASS SP2
	TT	38	-	
	UU	39	H	MASS FAIL
	VV	40		GND

* Alternate pin designation schemes

NOTE

Massbus cables are to be installed per markings on the cable.

2.8 SYNCHRONOUS BACKPLANE INTERCONNECT DESCRIPTION

The SBI is the backplane of the VAX-11/780 system. It interconnects the CPU with the memory system and all adapters in the system. The following paragraphs describe all interconnect lines and their associated communication protocol.

2.8.1 Interconnect Synchronization

Six control group lines are clock signals that are used as a universal time base for all nexus connected to the SBI. All SBI clock signals are generated on the CPU clock module and provide a 200 ns clock period.

The clock signals, in conjunction with the standard nexus clock logic, provide the derived clocks within an attached nexus to synchronize SBI activity. Two clock signals (TPH and TPL) produce the basic nexus time states. The remaining four (PCLKH, PCLKL, PDCLKH, and PDCLKL) are phased clocks and help compensate for the clock distribution skew due to cable, backplane, and driver/receiver propagation delays.

2.8.1.1 Derived Time States -- The derived clocks (within the nexus) define four, 50 ns (nominal) time states in one clock period. The time states (T0, T1, T2, and T3) determine the transmit, propagate, and receive times on the SBI, with T0 representing the start of a particular clock period. Figure 2-2 illustrates the phase and timing relationships required to generate the individual derived time states. Note that T0 internal to the CPU (CPT0) is not the same as SBI T0. CPT0 corresponds to SBI T1. All nexus need a minimum of T0 and T2 for SBI transmit and receive functions.

2.8.1.2 Transmit Data -- Information to be transmitted is asserted on the SBI at T0. Immediately prior to T0 a transmitting nexus enables its transmit enable inputs to the SBI transceivers. Figure 2-3 is a basic block diagram for one SBI information path line.

2.8.1.3 Receive Data -- In the case of receive data, the nexus receiver latches are opened at T2 and latched at T3. Figure 2-4 shows the basic one-line receiver latch logic. Note that the information may be considered undefined between T2 and T3; only after T3 is information considered valid. Nexus checking, decoding, and subsequent decision making are then based on these latched signals.

2.8.1.4 Single Time States -- In single time states, the time between any T0-T1, T1-T2, T2-T3, and T3-T0 may vary from 50 ns (nominal) to an indefinitely long period of time. SBI operation and protocol will proceed normally. Nexus that implement the SBI timeout functions do so by counting SBI cycles. Memory nexus operation must be normal even though the timing may be different. Nexus that derive timing from an external source (e.g., a mass storage device) set data late and overrun error bits as appropriate. However, the SBI operation of these nexus remains normal.

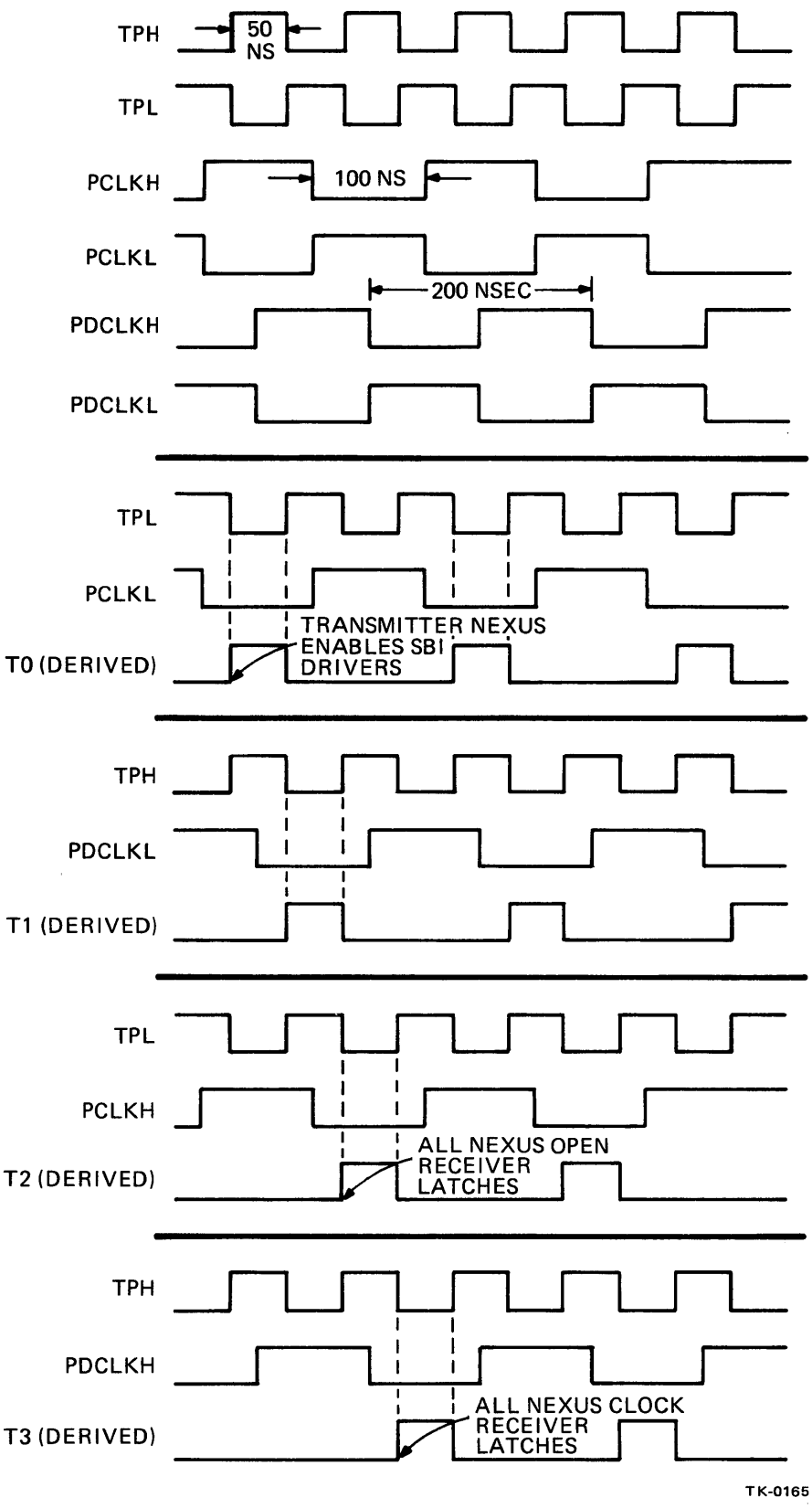


Figure 2-2 SBI Time and Phase Relationships

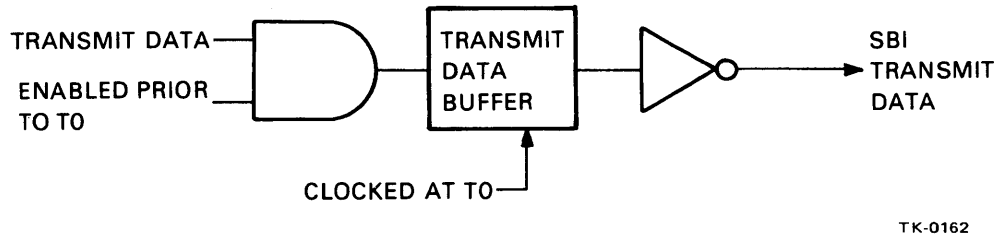


Figure 2-3 Transmit Data Path

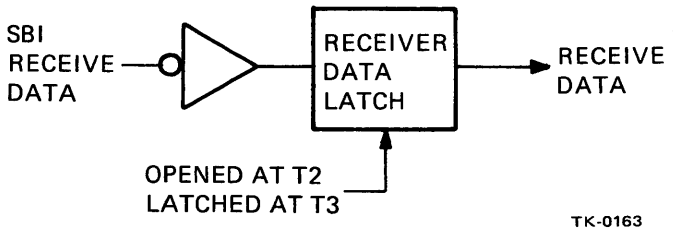


Figure 2-4 Receive Data Path

2.8.2 SBI Summary

Table 2-2 summarizes the signal fields associated with each functional group. Figure 2-5 shows the SBI configuration. The following paragraphs provide detailed descriptions of the individual group field layouts and functions.

Table 2-2 SBI Field Summary

Field	Description
ARBITRATION GROUP	
Arbitration Field (TR <15:ØØ>)	Establishes a fixed priority among nexus for access to and control of information transfer path.
INFORMATION TRANSFER GROUP	
Information Field (B<31:ØØ>)	Bidirectional lines that transfer data, command/address, and interrupt information between nexus.
Mask Field (M<3:Ø>)	Primary function: encoded to indicate a particular byte within the 32-bit information field (B<31:ØØ>).
	Secondary function: in conjunction with the tag field, indicates a particular type of read data.
Identifier Field (ID<4:Ø>)	Identifies the logical source or destination of information contained in B<31:ØØ>.
Tag Field (TAG<2:Ø>)	Defines the transmit or receive information types and the interpretation of the content of the ID and information fields.
Function Field (F<3:Ø>)	Specifies the command code, in conjunction with the tag field. This field is valid as part of the 32-bit information field only when the tag equals command/address.
Parity Field (P<1:Ø>)	Provides even parity for all information transfer path fields. P(Ø) is generated as parity for the information field. P(2) is generated as parity for the tag, ID, and mask fields.
RESPONSE GROUP	
Confirmation Field (CNF<1:Ø>)	Asserted by a receiving nexus to specify one of four response types and indicate its capability to respond to the transmitter's request.

Table 2-2 SBI Field Summary (Cont)

Field	Description
Fault Field (FAULT)	A cumulative error line to the CPU that indicates one of several errors, stored in the transmitting nexus fault register, and the associated SBI cycle in which the error occurred.
INTERRUPT REQUEST GROUP	
Request Field (REQ<7:4>)	Allows a nexus to request an interrupt to service a condition requiring CPU intervention. Each request lines represents a level of nexus request priority.
Alert Field (ALERT)	A cumulative status line that allows those nexus not equipped with an interrupt mechanism to indicate a change in power or operating conditions.
CONTROL GROUP	
Clock Field (CLOCK)	Six control lines that provide the clock signals necessary to synchronize SBI activity.
Fail Field (FAIL)	A single line from the restart nexus to provide a restart signal to the CPU to initiate a system restart operation.
Dead Field (DEAD)	A single line to the CPU to indicate an impending clock circuit or SBI terminating network power failure.
Unjam Field (UNJAM)	A single line from the CPU to attached nexus that initiates a restore operation.
Interlock Field (INTLK)	A single line that provides coordination among nexus responding to certain read/write commands to ensure exclusive access to shared data structures.

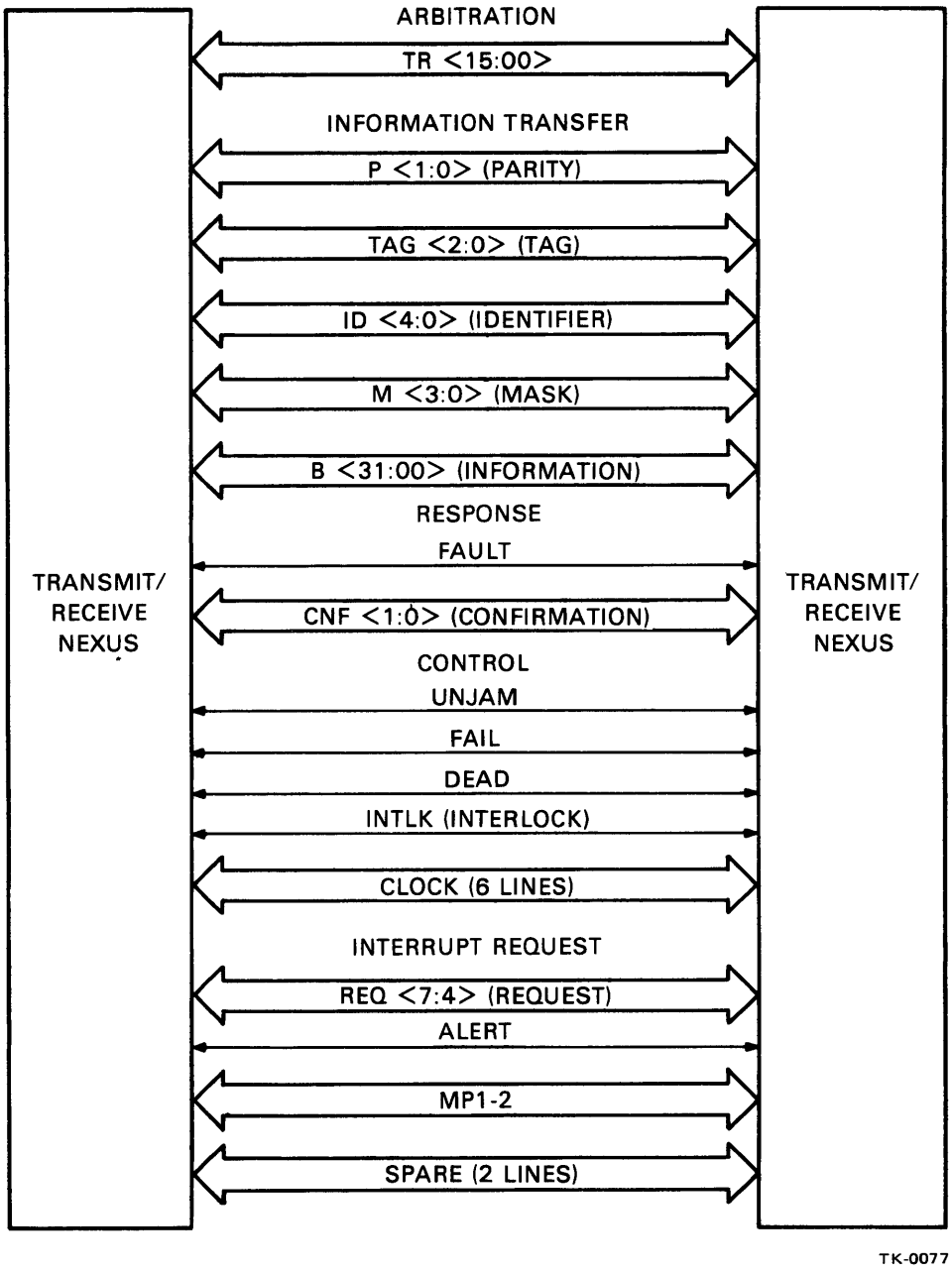


Figure 2-5 SBI Configuration

2.8.3 Arbitration Group Functions and Assignments

The arbitration lines (Transfer Request TR<15:ØØ>) allow up to 16 nexus to arbitrate for the information lines (information transfer group). One arbitration line is assigned to each nexus to establish the fixed priority access. Priority increases from TR15 to TRØØ, where TRØØ is the highest. The lowest priority level is reserved for the CPU, and it requires no actual TR signal line. The other 15 nexus are assigned TR15 through TRØ1.

The highest priority level, TRØØ, is reserved for those nexus that require more than one successive SBI cycle. TRØØ may only be used by nexus that require:

a. Two or three adjacent cycles for a write type exchange.
b. Two adjacent cycles for an extended read exchange.
c. Adjacent cycles for interrupt summary read exchanges and restore operations.

A nexus requests control of the information path by asserting its assigned TR line at TØ of an SBI cycle. At T3 of the same SBI cycle, the nexus examines (arbitrates) the state of all higher priority TR lines. If no higher TR lines are asserted, the requesting nexus assumes control of the information path at TØ of the following SBI cycle. At this TØ time state, the nexus negates its TR line and asserts command/address or data information on B<31:ØØ>. In addition, if a write type exchange is specified, the nexus asserts TRØØ to retain control of adjacent SBI cycles.

If higher priority TR lines are asserted, the requesting nexus can not gain control of the information path. The nexus keeps its TR line asserted and again examines the state of higher priority lines at T3 of the next SBI cycle. As before, if no higher TR lines are asserted, the nexus assumes information path control at TØ.

2.8.4 Information Transfer Group Description

Each information group field is described in detail in the following paragraphs. However, the information field (B<31:ØØ>) is described in the context of the other information group fields.

2.8.4.1 Parity Field -- The parity field (P<1:Ø>) provides even parity for detecting single bit errors in the information group (Figure 2-6).

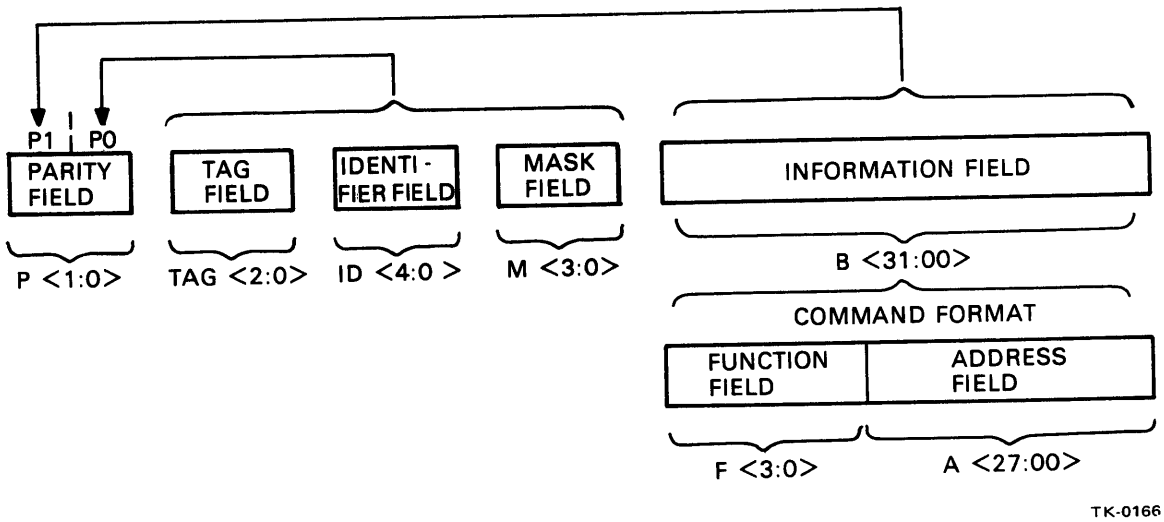


Figure 2-6 Parity Field Configuration

A transmitting nexus generates PØ as parity for TAG<2:Ø>, ID<4:Ø>, and M<3:Ø>. The P1 parity bit is generated for B<31:ØØ>. PØ and P1 are generated such that the sum of all logic one bits in the checked field, including the parity bit, is even. With no SBI transmissions, the information transfer path assumes an all zeros state; thus, P<1:Ø> should always carry even parity. Any transmission with odd parity is considered an error.

2.8.4.2 Tag Field Formats -- The tag field (TAG<2:Ø>) is asserted by a transmitting nexus to indicate the information type being transmitted on the information lines. The tag field determines the interrelation of the ID and B fields. In addition, the tag field, in conjunction with the mask field, further defines special read and write data conditions. The following paragraphs describe each information type, tag code, and associated field content.

Command/Address Tag -- A tag field content of Ø11 indicates that the content of B<31:ØØ> is a command/address word. ID<4:Ø>, asserted at this time, is a unique code identifying the logical source (commander) of the command. As shown in Figure 2-7, B<31:ØØ> is divided into a function field and an address field to specify the command and its associated address.

In a write type command, the ID field code represents the logical source and the address field specifies the logical command destination. For a read type command, the addressed nexus holds the transmitted ID for transmission with the requested data. The ID is sent with the read data to indicate destination.

The 28 bits of the SBI address field define a 268, 435, 456 longword address space, which is divided into two sections. Addresses $Ø--7FFFFFF_{16}$ are reserved for primary memory. Addresses $8ØØØØØØ_{16}--FFFFFFF_{16}$ are reserved for device control registers. Generally, primary memory begins at address Ø; the address space is dense and consists only of storage elements. The control address space is sparse with address assignments based on device type. Each nexus is assigned a 2Ø48, 32-bit longword address space for control. The addresses assigned are determined by the TR number as shown in Figure 2-8.

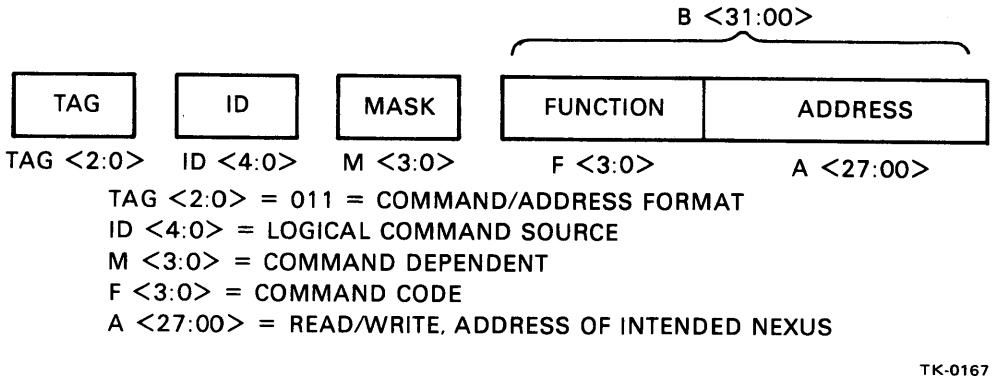


Figure 2-7 Command/Address Format

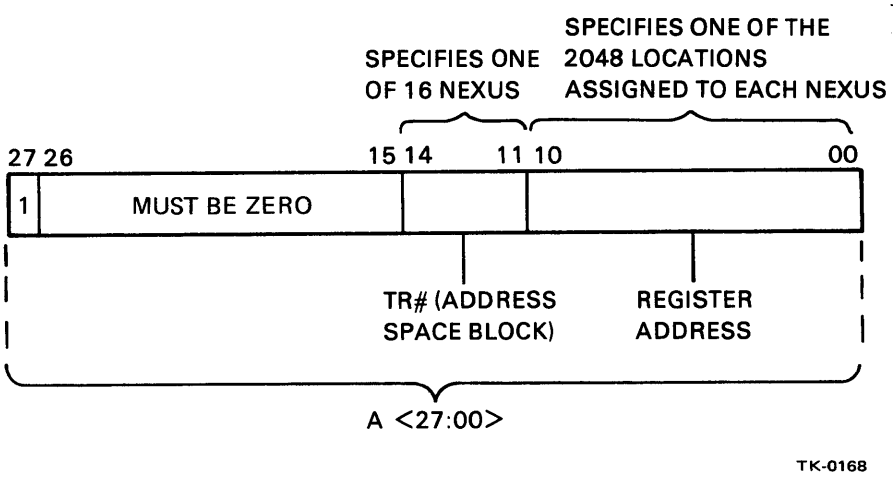


Figure 2-8 Control Address Space Assignment

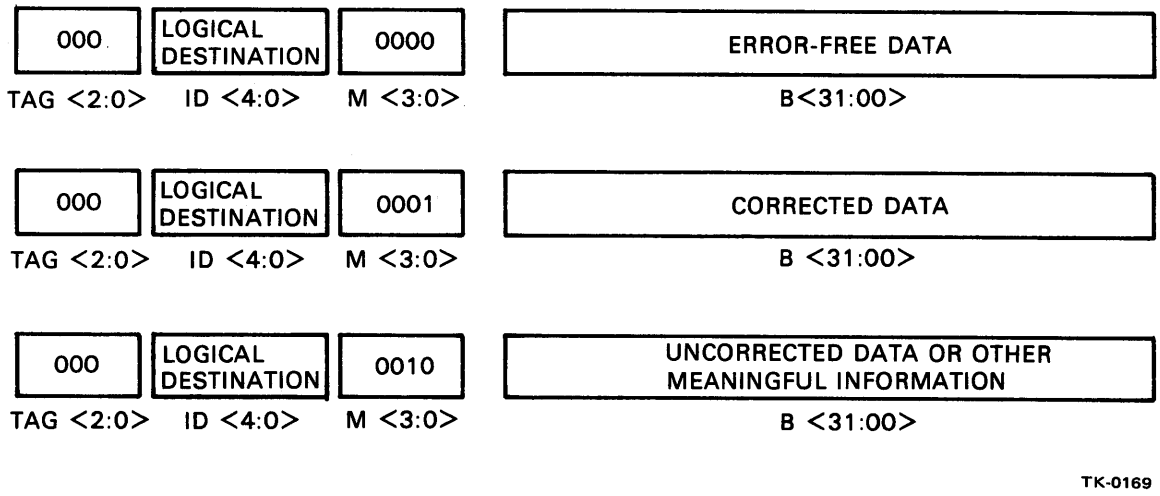


Figure 2-9 Read Data Formats

Read Data Tag -- A tag field content of ØØØ indicates that B<31:ØØ> contains data requested by a previous read type command. In this case, ID<4:Ø> is a unique code that was received with the read command and identifies the logical destination of the requested data. The retrieved data may be one of three types: read data; corrected read data; or read data substitute, where the particular type is identified by M<3:Ø>.

Read data is the normally expected error-free data having M<3:Ø> = ØØØØ (Figure 2-9). Note that this tag code is also the idle state of the tag field and that ID code Ø is reserved. No device will respond when the tag is ØØØ and the ID code is Ø.

Corrected read data is data in which an error was detected and subsequently corrected by the error correction code (ECC) logic of the device transmitting the read data. In this case, the mask field flags the corrected data with M<3:Ø> = ØØØ1.

Read data substitute represents data in which an error was detected but not corrected. In this case, B<31:ØØ> will contain the substitute data in the form of uncorrected data or other meaningful information. The mask field flags the uncorrected data with M<3:Ø> = ØØ1Ø. As with the other read data types, the ID field identifies the read commander.

Write Data Tag -- A tag field content of 1Ø1 indicates that B<31:ØØ> contains the write data for the location specified in the address field of the previous write command (Figure 2-1Ø). The write data will be asserted on B<31:ØØ> in the SBI cycle immediately following the command/address cycle. Certain command codes use M<3:Ø> to specify particular bytes within B<31:ØØ>.

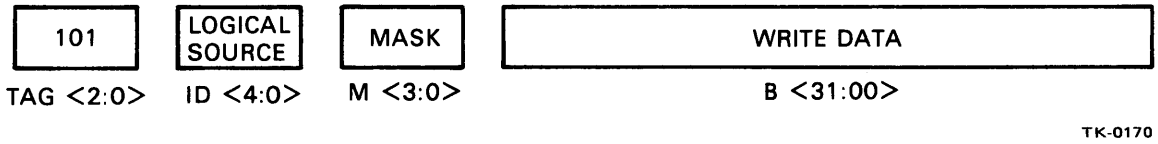


Figure 2-1Ø Write Data Format

Interrupt Summary Tag -- A tag field content of 11Ø defines B<31:ØØ> as the interrupt level mask for an interrupt summary read command. The level mask (B<Ø7:Ø4>) is used to indicate the interrupt level being serviced as the result of an interrupt request. In this case, the ID field identifies the commander, which is usually a CPU. Although unused, M<3:Ø> must be transmitted as zero.

The interrupt sequence consists of two exchanges:

a. The first exchange indicates the interrupt level being serviced.

b. The second exchange is the response, where the device requesting the interrupt identifies itself.

The interrupt summary read and response formats are illustrated in Figure 2-11. Note that the interrupt summary response encodes TAG<2:Ø> = ØØØ.

Reserved Tag Codes -- TAG<2:Ø> -- Tag code 111 is reserved for diagnostic purposes. Tag codes ØØ1, Ø1Ø, and 1ØØ are unused and reserved for future definition.

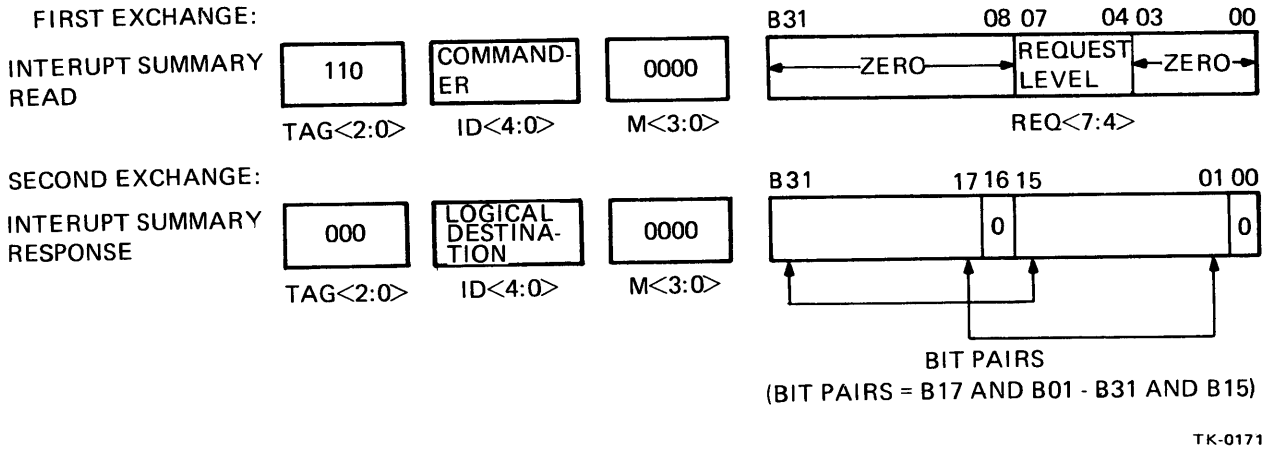


Figure 2-11 Interrupt Summary Formats

2.8.4.3 Identifier Field -- The ID field (ID<4:Ø>) contains a code that identifies the logical source or logical destination of the information contained in B<31:ØØ>. ID codes are assigned only to commander and responder nexus (i.e., those that issue and recognize command/address information). Each nexus is assigned an ID code that corresponds to the TR line that it operates. For example, a nexus assigned TRØ5 would also be assigned ID code 5.

More than one ID code may be assigned to a nexus. However, that nexus must be capable of responding to read type commands for which the read data returns in an order different from the order in which the commands were given. For write masked and extended write masked commands, the mask is transmitted in the cycle preceding the cycle for the data to which the mask applies.

Nexus using more than one code take the first code from the standard ID code assignment (Ø--15). The second code is taken from the range 17--3Ø (i.e., first ID code plus 16).

Certain ID codes are reserved: ID = 16, unit processors; ID = 31, diagnostic purposes. ID = Ø is reserved so that the idle state of the SBI (read data, destination ID = Ø) will not cause a nexus selection. Note that even though a nexus is not selected, all nexus are checking for correct SBI parity.

2.8.4.4 Mask Field -- The mask field (M<3:Ø>) has two interpretations: primary and secondary. For the primary interpretation, M<3:Ø> is encoded to specify operations on any or all data bytes appearing on B<31:ØØ>. The mask is used with the read masked, write masked, interlock read masked, interlock write masked, and extended write masked commands. As shown in Figure 2-12, each bit in the mask field corresponds to a particular byte on B<31:ØØ>.

The secondary interpretation is used when TAG<2:Ø> = ØØØ (read data). This interpretation defines the data types as specified in Table 2-3. All other mask field codes (ØØ11--1111) are reserved and are interpreted as read data substitute by the receiving nexus.

Table 2-3 Read Data Types

M<3:Ø>	Data Type
ØØØØ	Read data
ØØØ1	Corrected read data
ØØ1Ø	Read data substitute

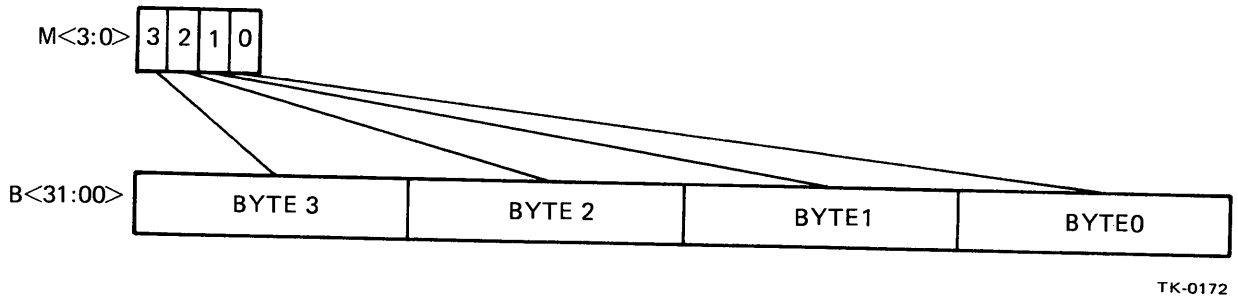


Figure 2-12 Mask Field Format

2.8.5 Response Group Description

The three response lines are divided into two fields: Confirmation (CNF<1:Ø>), and Fault (FAULT). CNF<1:Ø> informs the transmitter as to whether or not the information was received correctly and if the receiver can process the command. FAULT is a cumulative error indication of protocol or information path malfunction; it is asserted with the same timing as the confirmation field.

Either field is transmitted two cycles after each information transfer. Confirmation is delayed to allow the information path signals to propagate, be checked, and be decoded by all receivers; and to allow confirmation generation by the responder. During each cycle, every nexus in the system receives, latches, and makes decisions on the information transfer signals. Except for multiple bit transmission errors or nexus malfunction, one (or more) of the nexus receiving the information path signals will recognize an address or ID code. This nexus then asserts the appropriate response in CNF.

Any (or all) nexus may assert FAULT after detecting a protocol or information path failure.

2.8.5.1 Confirmation Codes -- Table 2-4 lists the confirmation codes and their interpretation.

Table 2-4 Confirmation Code Definitions

CNF Code	Definitions
ØØ, No Response (N/R)	The unasserted state; it indicates no response to a commander's selection.
Ø1, Acknowledge (ACK)	The positive acknowledgment to any transfer.
1Ø, Busy (BUSY)	The response to a command/address transfer that indicates successful selection of a nexus that is presently unable to execute the command.
11, Error (ERR)	The response to a command/address transfer that indicates selection of a nexus that cannot execute the command.

A BUSY (1Ø) or ERR (11) response to transfers other than command/address transfers will be considered as no response from the responder.

2.8.5.2 Response Handling -- The transmitting nexus samples the CNF and FAULT lines at T3 of the third cycle following transmission. ACK is the expected confirmation response (i.e., command will be executed, or information has been received correctly).

Should a command/address transfer receive a BUSY confirmation, the commander will repeat the transmission (after a nominal waiting period) until it is accepted or a timeout occurs.

An N/R confirmation should be treated like BUSY, except that its occurrence may be flagged in a status bit. ERR confirmation is the result of a programming error, and should abort the command and invoke the appropriate recovery routine.

Some nexus may be unable to determine, within two SBI cycles, whether a function will be completed successfully. For these cases, the nexus presumes success and responds with ACK confirmation. If it is later determined that a read type function cannot be completed, a read data transfer of all zeros is transmitted and an interrupt requested. If a write type request cannot be completed, the command is aborted and an interrupt requested. In either case, the cause of the interrupt is indicated in a configuration/status register.

2.8.5.3 Successive Cycle Confirmation -- Since write masked, extended write masked, and extended read operations consist of successive transfers, acknowledgment is more complex.

a. If the command/address transfer is confirmed with N/R or BUSY, then no notice will be taken of the data transfer confirmation and the entire sequence will be repeated.

b. If the command/address transfer receives ERR, the sequence is aborted and recovery routines are invoked.

c. If ACK is not received as confirmation for a write data command, the command is repeated.

d. Transmissions of read data are confirmed with ACK by the receiver of that data. The read data transmitter may ignore this confirmation, since only commanders execute retry sequences.

2.8.5.4 SBI Sequence Timeouts -- All commanders implement two timeout functions: interface sequence timeout and read data timeout. Both timeouts are specified as 102.4 us (or 512 SBI cycles).

The interface sequence timeout determines the maximum time allowed to complete an interface sequence. The sequence interval is defined as the time from:

a. when SBI arbitration is initiated, until ACK is received for a command/address transfer that specifies read, or

b. when SBI arbitration is initiated, until ACK is received for a command/address transfer that specifies write, and ACK is also received for each transmission of write data, or

c. when SBI arbitration is initiated, and an ERR confirmation is received for any command/address transfer.

The read data timeout is defined as the time from when an interface sequence that specifies a read command is completed, to the time that the specified read data is returned to the commander. In the case of an extended read function, both longwords must be retrieved prior to timeout (1Ø2.4 us).

If the last command/address transfer prior to an interface sequence timeout receives an N/R confirmation, it is recorded in a status bit. Certain nexus may terminate their requests for SBI control due to an unusual occurrence in those nexus. When this occurs, both timeouts are cancelled (e.g., when a nexus detects a data late error).

When a timeout occurs, the commander provides the actual address or reconstructed address for which the timeout occurred. In addition, the commander records the type of timeout received (i.e., interface sequence or read data). Either timeout will terminate a command transmission retry.

2.8.5.5 Fault Detection -- Each nexus is equipped with a 32-bit configuration and fault status register (register Ø). The fault status portion of this register contains flags that cause the assertion of the FAULT line. The fault status portion is described in Figure 2-13.

A nexus detecting one of the fault conditions will assert the FAULT signal for one cycle. FAULT then causes each nexus on the system to lock its respective configuration register. The fault status bits thus latched refer to the cycle during which the fault occurred. The CPU examines the FAULT signal and latches the signal on the leading edge of FAULT. The CPU then continues to assert FAULT until the software has examined the fault bits of all nexus and has specified the negation of FAULT. Figure 2-14 shows the timing involved.

Figure 2-15 illustrates the confirmation and fault decision flow for all responses and error conditions.

2.8.6 Interrupt Request Group Description

The interrupt request group consists of four request lines (REQ<7:4>) and an alert (ALERT) line. Request lines are assigned to some of the nexus and represent assigned CPU interrupt levels. The lines used by nexus request that the CPU service a condition requiring processor intervention. The request lines are priority encoded in an ascending order of REQ4--REQ7. A requesting nexus asserts it request lines (or line) asynchronously with respect to the SBI clock to request an interrupt. Any of the REQ lines may be asserted simultaneously by more than one nexus, and any combination of REQ lines may be asserted by the collection of requesting nexus.

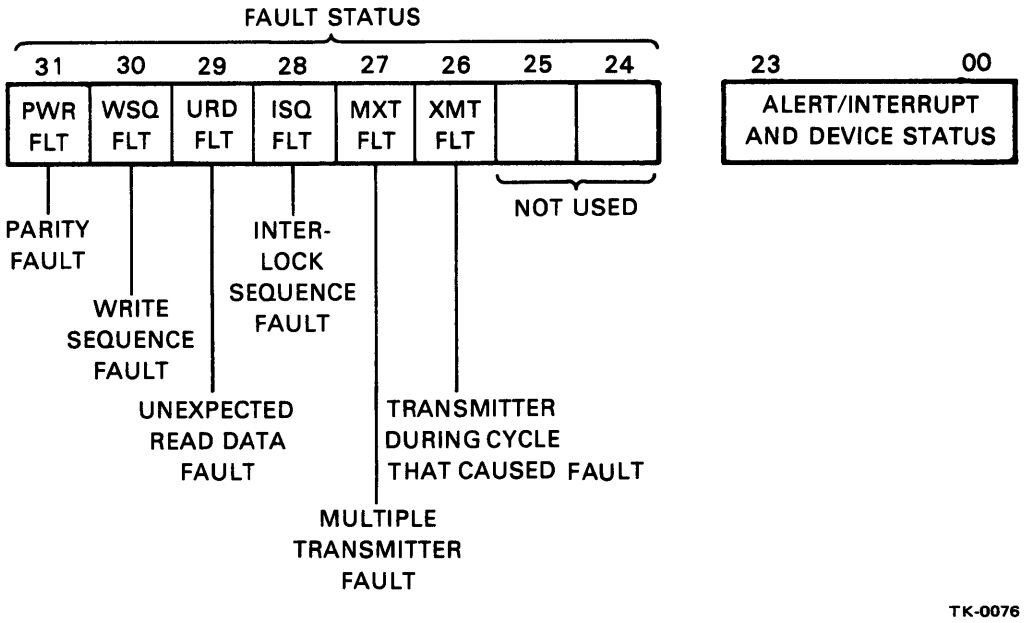


Figure 2-13 Fault Status Flags

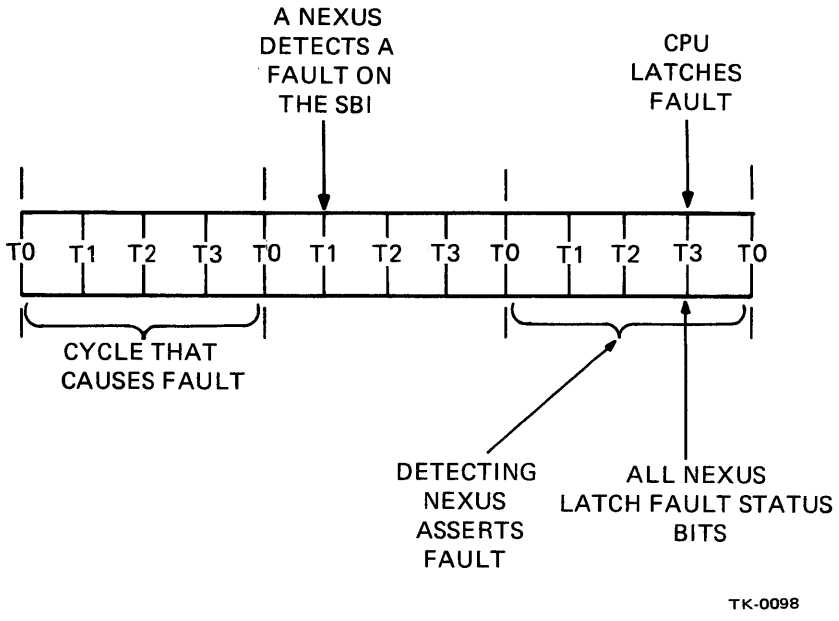
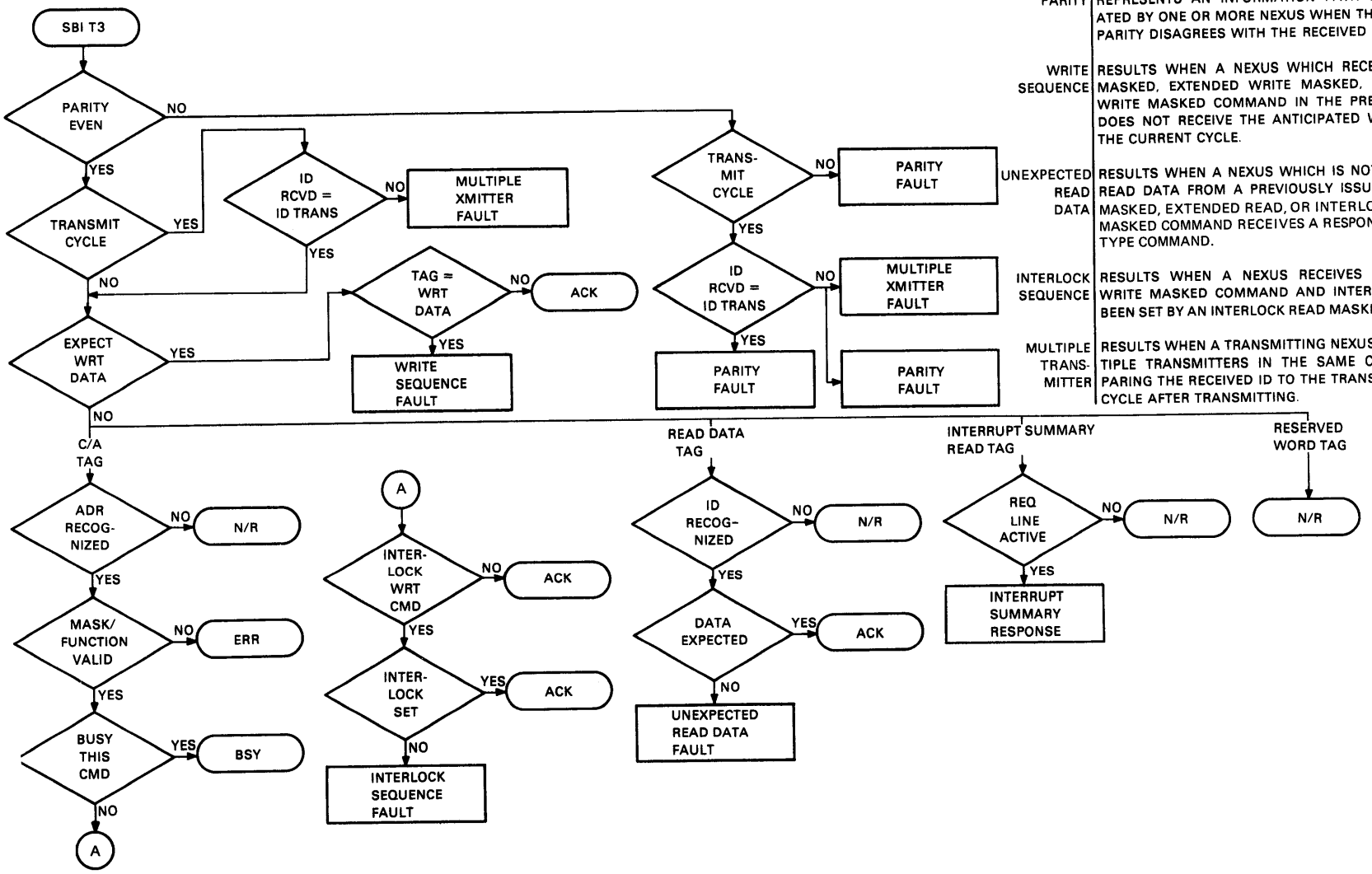


Figure 2-14 Fault Timing



SBI FAULT DEFINITIONS

FAULT	DEFINITION
PARITY	REPRESENTS AN INFORMATION PATH ERROR GENERATED BY ONE OR MORE NEXUS WHEN THE CALCULATED PARITY DISAGREES WITH THE RECEIVED PARITY.
WRITE SEQUENCE	RESULTS WHEN A NEXUS WHICH RECEIVED A WRITE MASKED, EXTENDED WRITE MASKED, OR INTERLOCK WRITE MASKED COMMAND IN THE PRECEDING CYCLE DOES NOT RECEIVE THE ANTICIPATED WRITE DATA IN THE CURRENT CYCLE.
UNEXPECTED READ DATA	RESULTS WHEN A NEXUS WHICH IS NOT WAITING FOR READ DATA FROM A PREVIOUSLY ISSUED READ MASKED, EXTENDED READ, OR INTERLOCK READ MASKED COMMAND RECEIVES A RESPONSE TO A READ TYPE COMMAND.
INTERLOCK SEQUENCE	RESULTS WHEN A NEXUS RECEIVES AN INTERLOCK WRITE MASKED COMMAND AND INTERLOCK HAS NOT BEEN SET BY AN INTERLOCK READ MASKED COMMAND.
MULTIPLE TRANS-MITTER	RESULTS WHEN A TRANSMITTING NEXUS DETECTS MULTIPLE TRANSMITTERS IN THE SAME CYCLE BY COMPARING THE RECEIVED ID TO THE TRANSMITTED ID ONE CYCLE AFTER TRANSMITTING.

TK-0086

Figure 2-15 Confirmation and Fault Decision Flow

The ALERT signal is asserted by nexus that do not implement interrupt request lines. Its purpose is to indicate to the CPU a change in the nexus power condition or operating environment. Nexus that implement the REQ lines report such changes by requesting an interrupt.

2.8.6.1 Interrupt Operation -- When a nexus requires an interrupt, it asserts its REQ line on the SBI. At a time judged appropriate, the CPU will recognize the interrupt request and issue an interrupt summary read command (TAG<2:0> = 110). The command will have a single bit set in its interrupt level mask (B<7:4>), which corresponds to the REQ line being serviced. For example, B 04 set to a logic one indicates that the REQ4 level is being serviced. Note that the remaining information path fields (i.e., B<31:08>, ID<03:00>, and M<3:0>) are transmitted as zero.

Nexus receiving the interrupt summary read command without error and asserting the REQ line specified in the interrupt level mask will assert a 2-bit code in B<31:00>. This code, which identifies the requesting nexus, is asserted with the timing of CNF<1:0>. However, the responding nexus does not assert any CNF, TR, ID, or TAG line. Nexus that detect incorrect parity will assert FAULT.

As shown in Figure 2-16, the asserted bits are in corresponding positions in the upper and lower 16 bits of B<31:00>. The bit pair uniquely identifies the nexus among those using the particular REQ line. Only 15 bit pairs in the information field are used (i.e., B31 and B15 through B17 and B01). Since only pairs of bits are asserted, parity remains correct regardless of the number of responding nexus. The two bits asserted by the requesting nexus are equal to the nexus TR number and the nexus TR number plus 16.

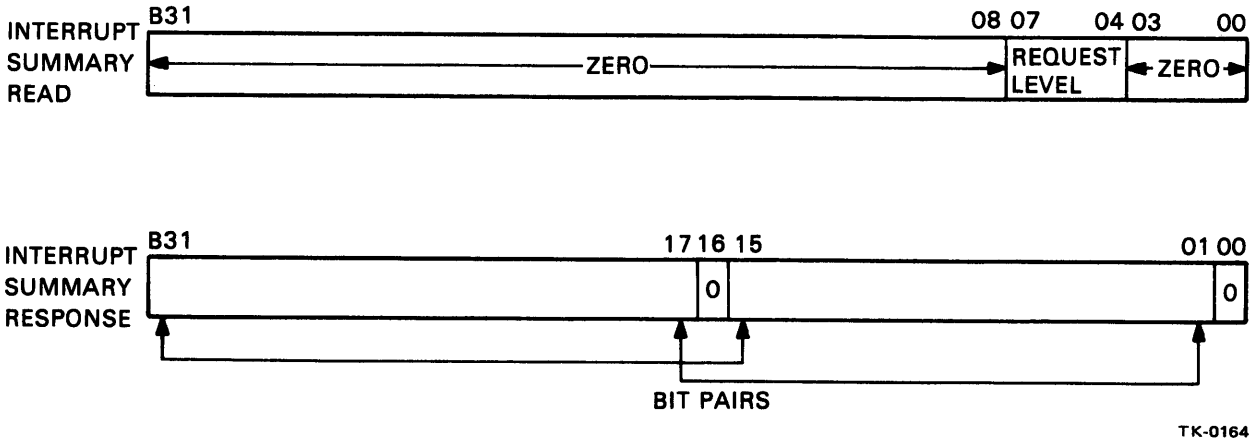


Figure 2-16 Request Level and Nexus Identification

While holding control of the SBI with TRØØ, the CPU waits two cycles after the interrupt summary read command is transmitted before latching B<31:ØØ> into an internal register. By encoding the REQ level and the bit pair received from responding nexus, the CPU generates a vector unique to that level and nexus. The vector, in turn, is used to invoke the nexus service routine. The service routine will take explicit action by writing a device register to clear the interrupt condition. Clearing the interrupt causes the nexus to negate the REQ line, provided that the nexus does not have any other outstanding interrupts at this level. The negation of REQ occurs within two cycles of the write data transmission.

Normally, the CPU will service requests in descending order, of REQ7--REQ4. Similarly, nexus are identified in descending order beginning with the nexus that asserts bits B31 and B15 and ending with the nexus that asserts bits B17 and B1. If multiple nexus are requesting interrupts on the same REQ line, multiple interrupt summary read commands are issued until all nexus have been serviced and the REQ line is no longer asserted.

Figure 2-17 is a functional timing chart for the interrupt operation.

2.8.6.2 Status Register Alert Flags -- As shown in Figure 2-18 each nexus maintains bits in its configuration register to indicate conditions that cause assertion of ALERT (or the appropriate REQ line if implemented). Power down and power up status bits are provided, but additional ALERT status bits are present if other conditions, such as overtemperature, are detectable.

The ALERT line is the logical OR of the alert status bits; it is asserted synchronously to the SBI clock. Alert status bits are cleared when written as logic one; when written as logic zero, UNJAM signal is received. Note that the UNJAM signal does not clear these status bits.

2.8.6.3 Alert Flag Operation -- A nexus asserts ALERT or an interrupt request when any of its alert status bits are set. The bits are set during the following events:

a. during power failure at the nexus when the assertion of power supply AC LO is recognized;

b. during the restoration of power when the negation of AC LO is recognized;

c. when other environmental conditions, such as overtemperature, are detected;

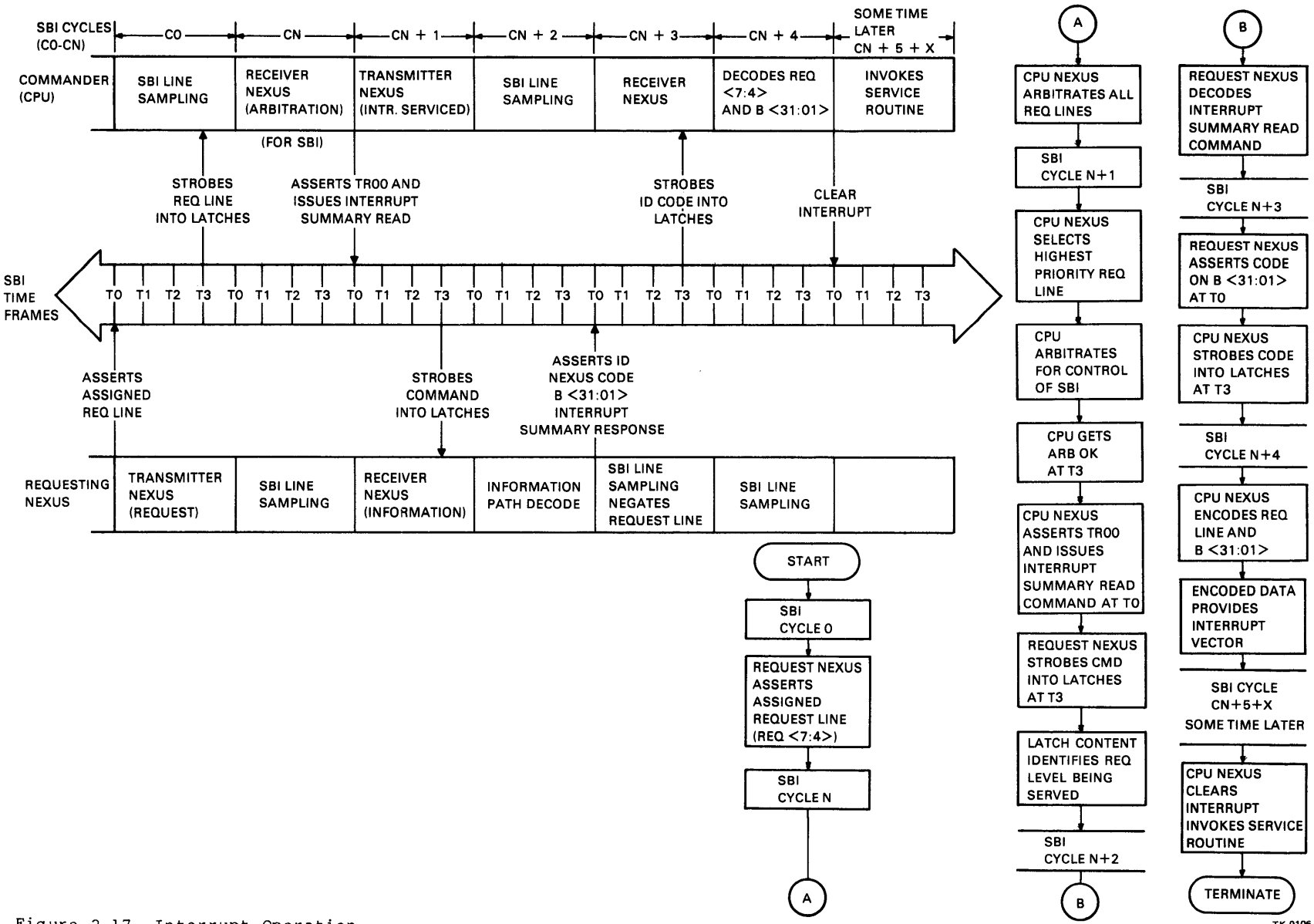


Figure 2-17 Interrupt Operation Timing and Flow

The alert status bits are only set on the transition of the event that caused them to set.

The power down status bit is set when there is a transition of the nexus AC LO from the negated to the asserted state. Setting the power down status bit clears the power up status bit; likewise, setting the power up bit clears the power down bit. The overtemperature bit is set when there is a transition from the normal to the overtemperature state.

A nexus asserting ALERT, or asserting an interrupt request due to an alert status bit set, continues to assert ALERT until:

a. all alert status bits are cleared (written with a logic one),

b. UNJAM signal is received,

c. nexus loses dc power.

The negation of ALERT (or REQ) is synchronous to the SBI clock and occurs within two cycles of the write data transmission used to clear the ALERT condition.

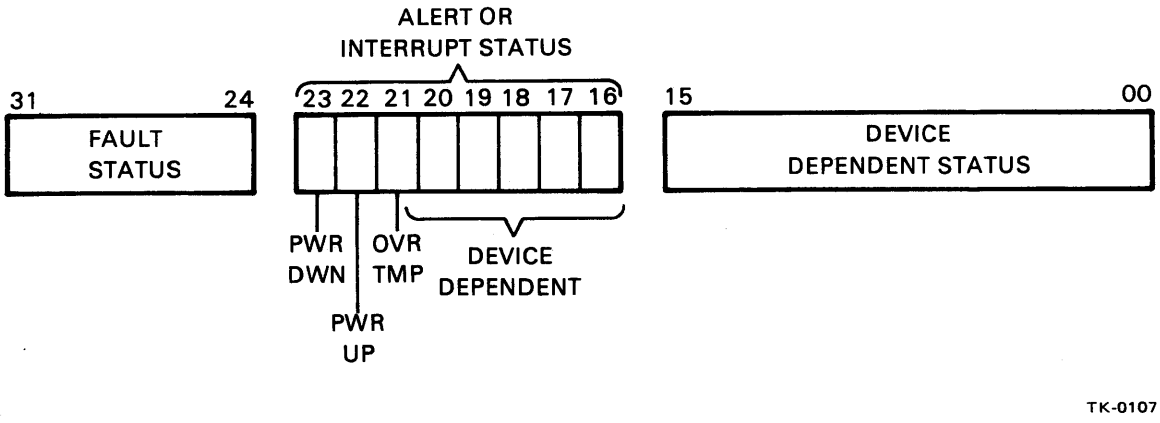
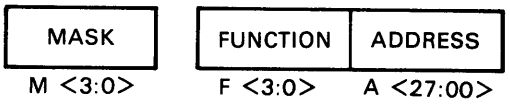


Figure 2-18 Alert Status Bits

2.8.7 Command Code Description

The operations executed over the SBI are specified in command/address format using the mask, function, and address fields. Figure 2-19 summarizes the command/address formats and lists the command codes. Several function codes are unused and reserved for future use. All nexus must respond to these reserved codes with an N/R confirmation.

2.8.7.1 Read Masked Function -- The read masked function is specified in Figure 2-2Ø.



MASK USE	FUNCTION CODE	FUNCTION DEFINITION
IGNORED	0000	RESERVED
USED	0001	READ MASKED
USED	0010	WRITE MASKED
IGNORED	0011	RESERVED
USED	0100	INTERLOCK READ MASKED
IGNORED	0101	RESERVED
IGNORED	0110	RESERVED
USED	0111	INTERLOCK WRITE MASKED
IGNORED	1000	EXTENDED READ
IGNORED	1001	RESERVED
IGNORED	1010	RESERVED
USED	1011	EXTENDED WRITE MASKED
IGNORED	1100	RESERVED
IGNORED	1101	RESERVED
IGNORED	1110	RESERVED
IGNORED	1111	RESERVED

TK-0083

Figure 2-19 SBI Command Codes

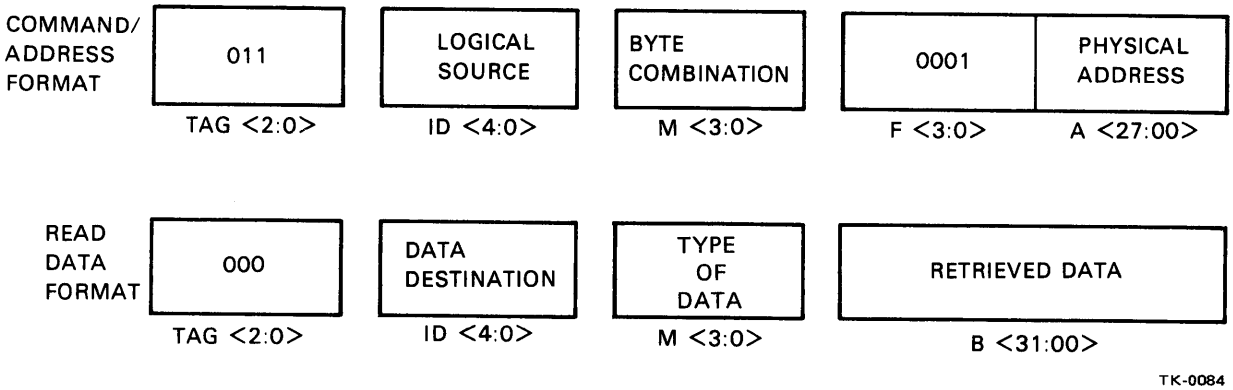


Figure 2-2Ø Read Masked Function Format

Prior to issuing the command, the commander arbitrates for SBI control. When the commander gains control of the SBI, it asserts the information transfer lines at TØ. At T3 of the same cycle, each nexus strobes the command/address information into its receiver latches for decoding. The command/address format, presented on the SBI, instructs the nexus selected by the address field, SA<27:ØØ>, to retrieve the addressed data word and transfer it to the logical destination specified in the ID field. The addressed nexus will respond to the command/address transfer with ACK (assuming no errors) two SBI cycles after the assertion of command/address.

The addressed data is retrieved in a time frame that is dependent on the nexus response time. Following the response delay, the responding nexus must arbitrate for control of the SBI. After ARB OK is true for the responder, the information fields are asserted on the SBI at TØ. TAG<2:Ø> is coded as ØØØ, specifying the read data format; and ID<4:Ø> is coded to identify the commander. The read data is asserted on B<31:ØØ> and received by the commander as read data (M<3:Ø> = ØØØØ), or as corrected read data (M<3:Ø> = ØØØ1). In the case of uncorrectable read data, the responder transmits read data substitute (M<3:Ø> = ØØ1Ø).

After the assertion of read data, the commander latches the content of B<31:ØØ> at T3 of the same SBI cycle. At TØ two cycles later, the commander confirms the successful transfer by asserting ACK.

Figure 2-21 is a functional timing chart for the read masked operation.

2.8.7.2 Extended Read Function -- The extended read function is similar to the read masked function in operation. The function format is shown in Figure 2-22.

The mask field and bit SAØØ of the received command/address word are ignored. However, the mask field must be transmitted as zero.

In an extended read, 64 bits (two data longwords) are always transmitted, and thus require two contiguous SBI data transfer cycles. In this case, F<3:Ø> instructs the nexus selected by SA<27:ØØ> to retrieve the addressed 64-bit data and transfer it to the commander (specified in the ID field) as in the read masked function.

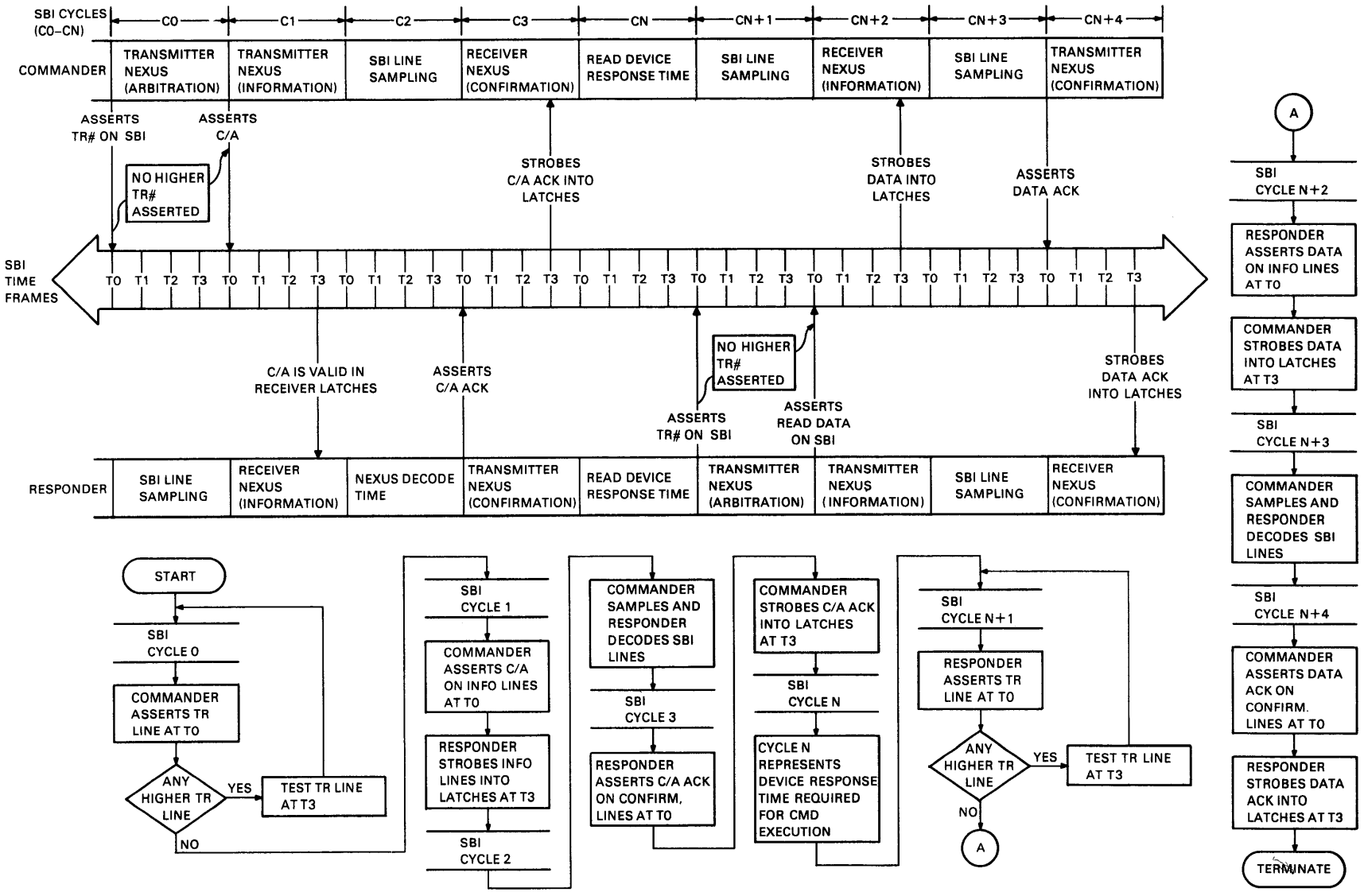


Figure 2-21 Read Masked Timing Chart

2-34

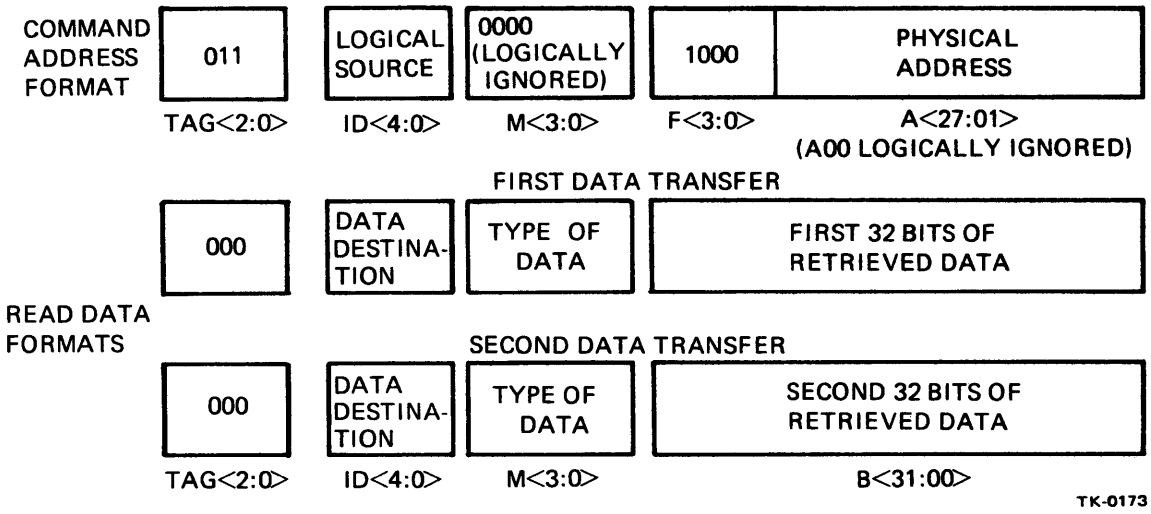


Figure 2-22 Extended Read Function Format

When the commander gains control of the SBI, it asserts the command/address information at TØ. At T3 of the same cycle, each nexus strobes the command/address information into its receiver latches for decoding. The addressed nexus confirms the command/address transfer by returning ACK two cycles after the assertion of command/address. Following the response delay and arbitration, the responder asserts the first 32-bit data longword on B<31:ØØ> (SAØØ = Ø). The other information fields are coded as in the read masked operation. The second data longword (SAØØ = 1) is asserted on B<31:ØØ> at TØ of the succeeding cycle. The mask field describing the data type will be asserted with each read data longword.

The commander latches B<31:ØØ> (first data longword) at T3 of the cycle when it was transmitted. At T3 of the next cycle, the commander again latches B<31:ØØ> (second data longword). Then, at TØ of the following cycle, the commander confirms the first data transfer with ACK. The commander confirms the second data transfer with ACK at TØ of the cycle after that.

Figure 2-23 is a functional timing chart showing the extended read operation.

2.8.7.3 Write Masked Function -- The write masked function format is shown in Figure 2-24. F<3:Ø> instructs the selected nexus to modify the bytes specified by M<3:Ø> in that storage element addressed by SA<27:ØØ> using data transmitted in the succeeding cycle.

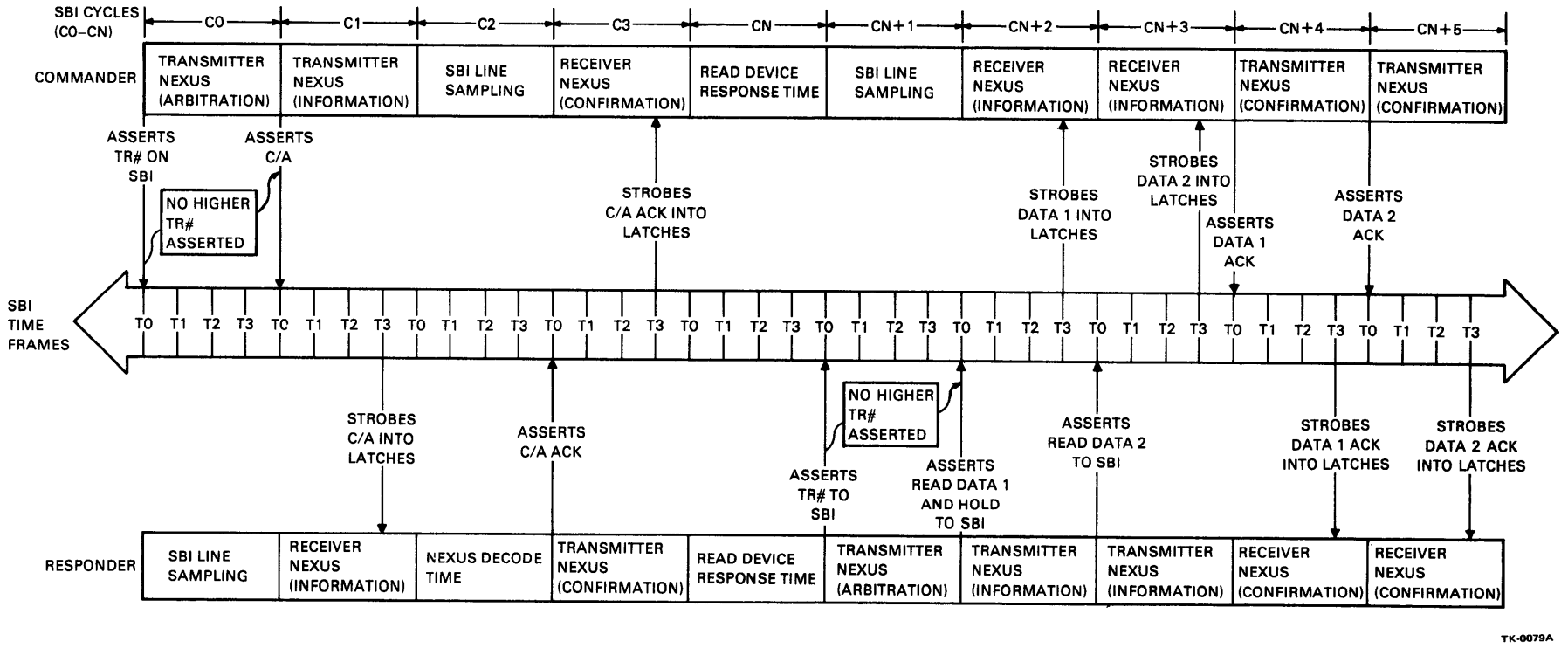


Figure 2-23 Extended Read Timing Chart and Flow (Sheet 1 of 2)

2-36

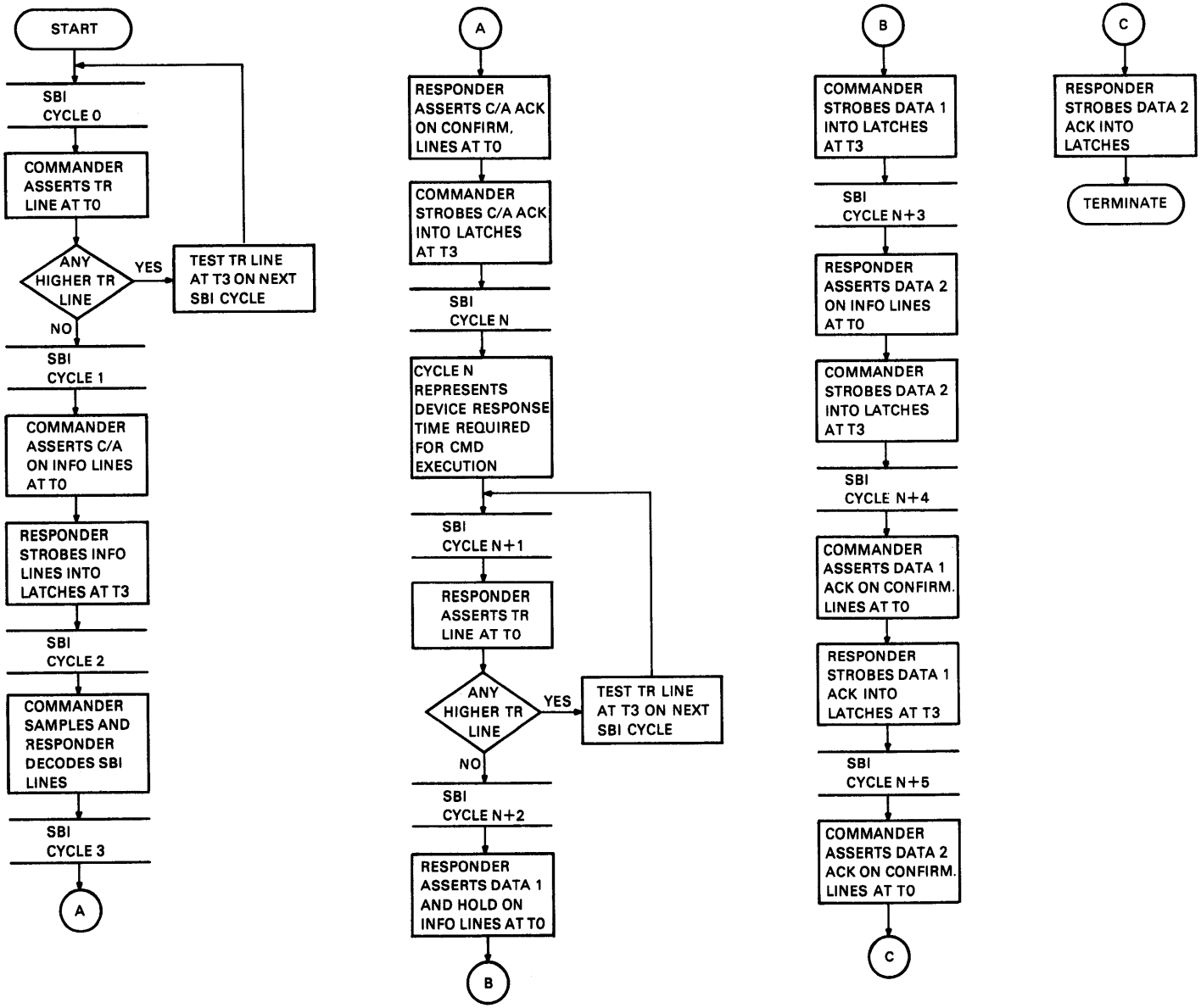


Figure 2-23 Extended Read Timing Chart and Flow (Sheet 2 of 2)

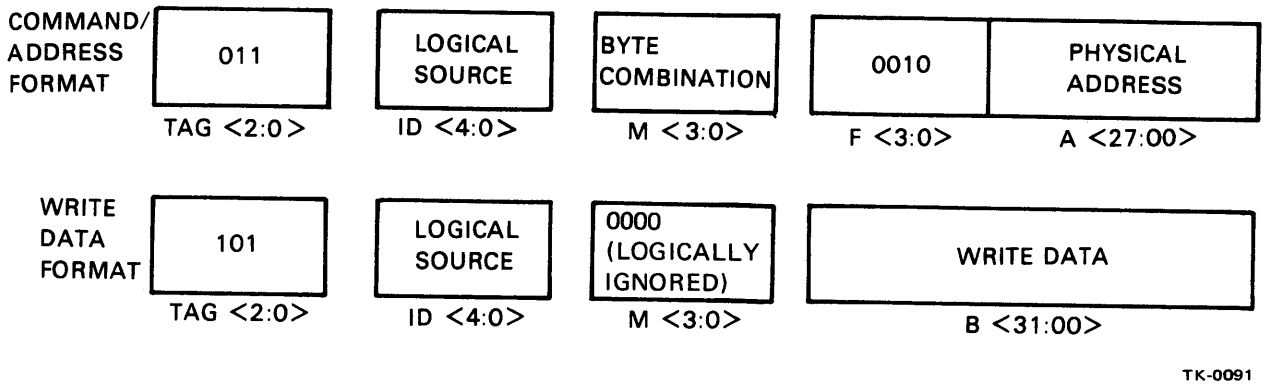
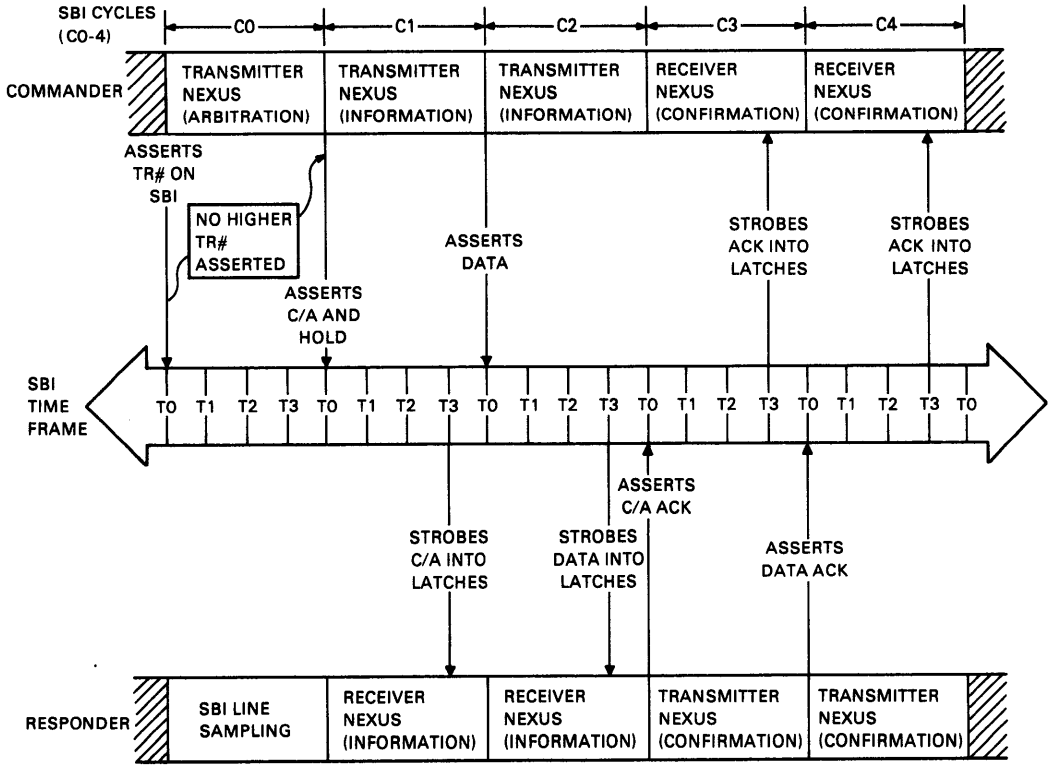


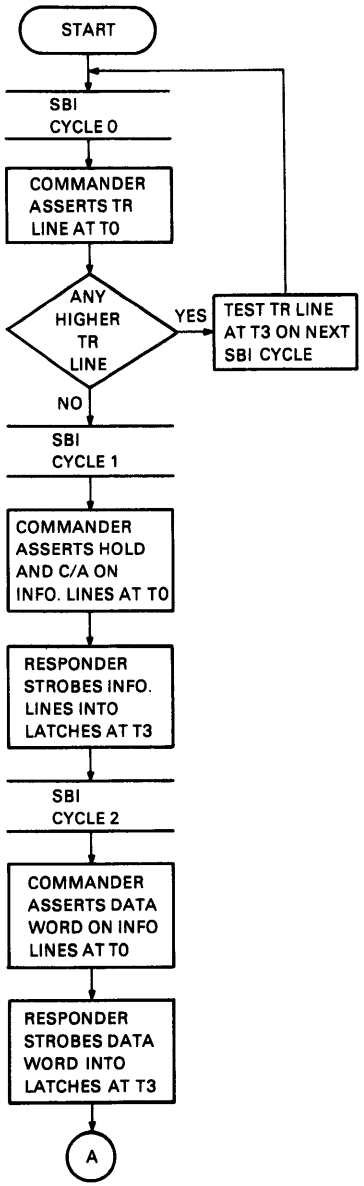
Figure 2-24 Write Masked Function Format

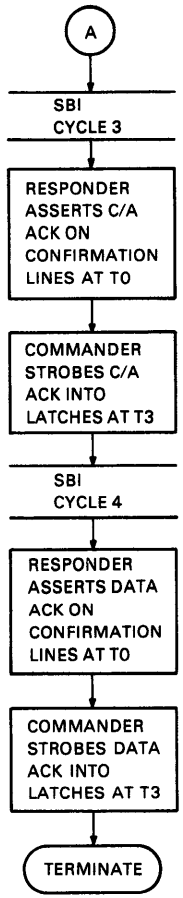
When the commander gains control of the SBI, it asserts the command/address information at TØ. The commander also asserts TRØØ at TØ to retain control during the succeeding SBI cycle. At T3 of the same cycle, each nexus strobes the command/address information into its receiver latches for decoding. At TØ of the succeeding cycle, the commander asserts data on B<31:ØØ>; at T3 of the same cycle, the selected nexus strobes the data into its receiver latches. TAG<2:Ø>, which accompanies the data, is coded 1Ø1 (write data format). The successful command/address transfer is confirmed by the receiving nexus with ACK at TØ of the succeeding cycle. The successful data transfer is confirmed by ACK at TØ, one cycle later.

Figure 2-25 is a functional timing chart for the write masked operation.

2.8.7.4 Extended Write Masked Function -- The extended write masked function format is illustrated in Figure 2-26. F<3:Ø> is coded 1Ø11 to specify the extended write masked function. In the extended write masked transfer, the number of bits written depends on the mask, but two SBI data transfer cycles are always required. When the commander gains control of the SBI it asserts the command/address information at TØ. The commander also asserts TRØØ to retain control during the succeeding SBI cycle. At T3 of the same cycle, each nexus strobes the command/address information into its latches for decoding. The mask that accompanies the command/address indicates the bytes to be written in the first data longword, corresponding to SAØØ = Ø.







FR-0080

Figure 2-25 Write Masked Timing Chart and Flow

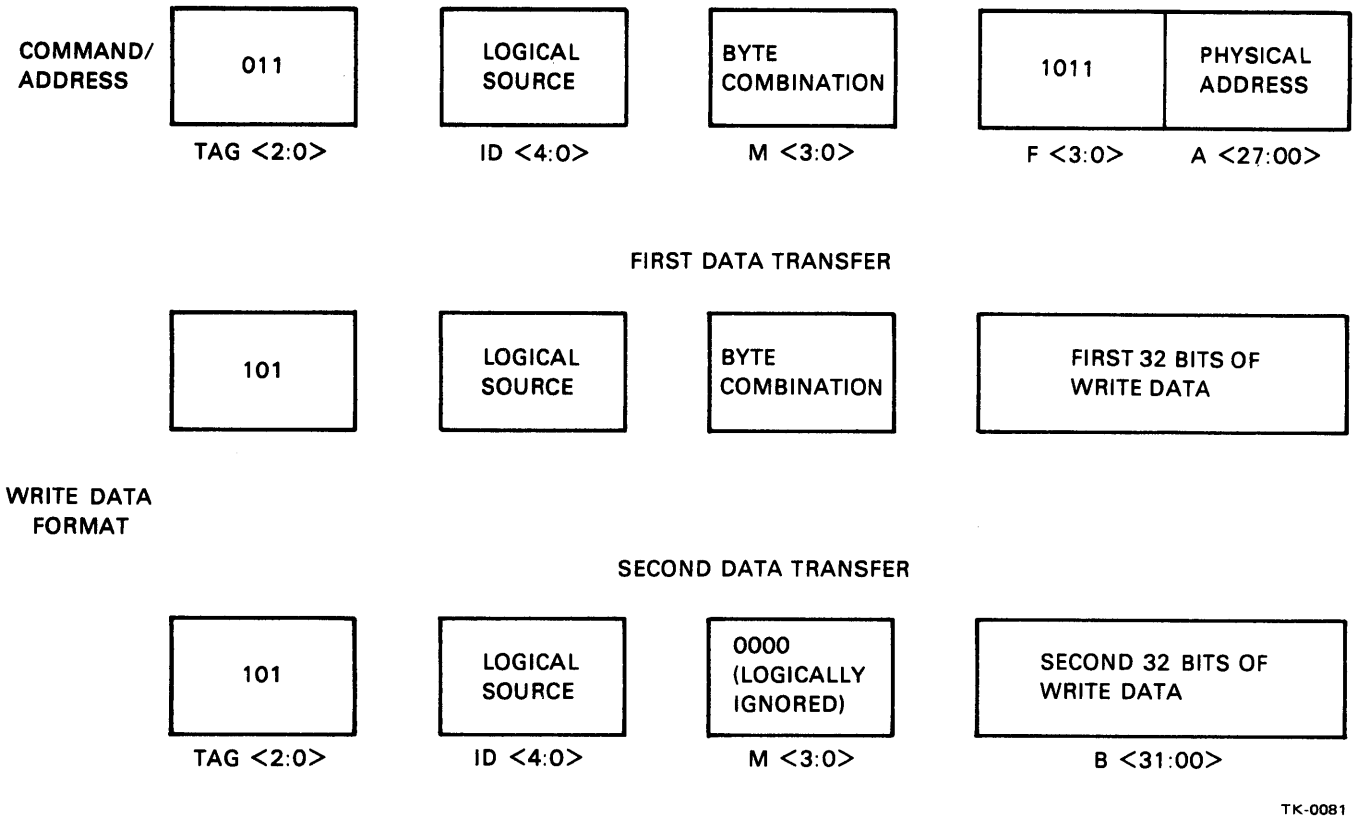
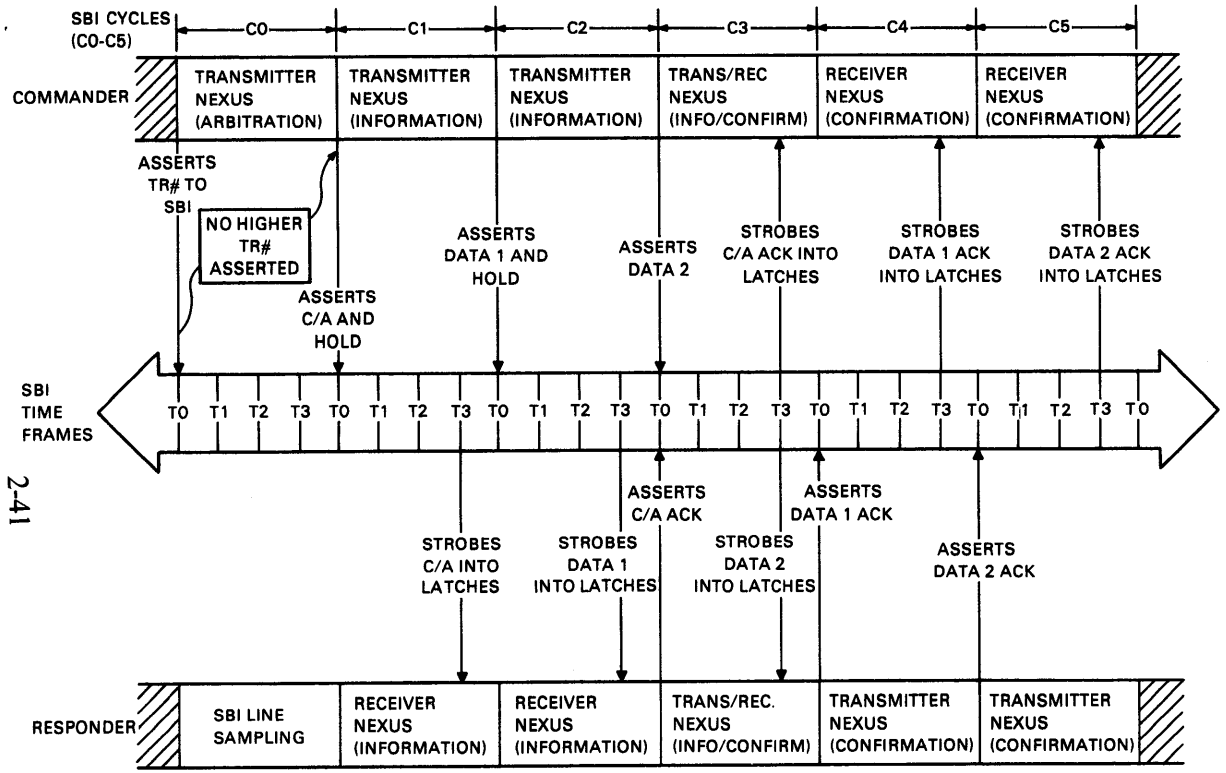


Figure 2-26 Extended Write Masked Function Format

At TØ of the succeeding cycle, the commander asserts data on B<31:ØØ> and codes TAG<2:Ø> as 1Ø1 (write data format). At T3 of the same cycle, the receiver nexus strobes the data into its latches. In addition, the commander holds TRØØ asserted to retain SBI control for the second data longword transfer. Note that the mask that accompanies the first data word indicates the bytes to be written in the second data word. At the end of this cycle, the commander negates TRØØ.

At TØ of the succeeding cycle, the second data word is asserted on B<31:ØØ>, and TAG<2:Ø> is coded 1Ø1. At the same time (TØ) the receiver nexus confirms the command/address transfer with ACK, if there is no error. At T3 of the same cycle, the receiver nexus strobes the data into its latches. The mask that accompanies the second data longword is ignored by the receiver nexus. During the two succeeding cycles, the receiver nexus confirms the two data transfers with an ACK in each cycle.

Figure 2-27 is a functional timing chart for the extended write masked operation.



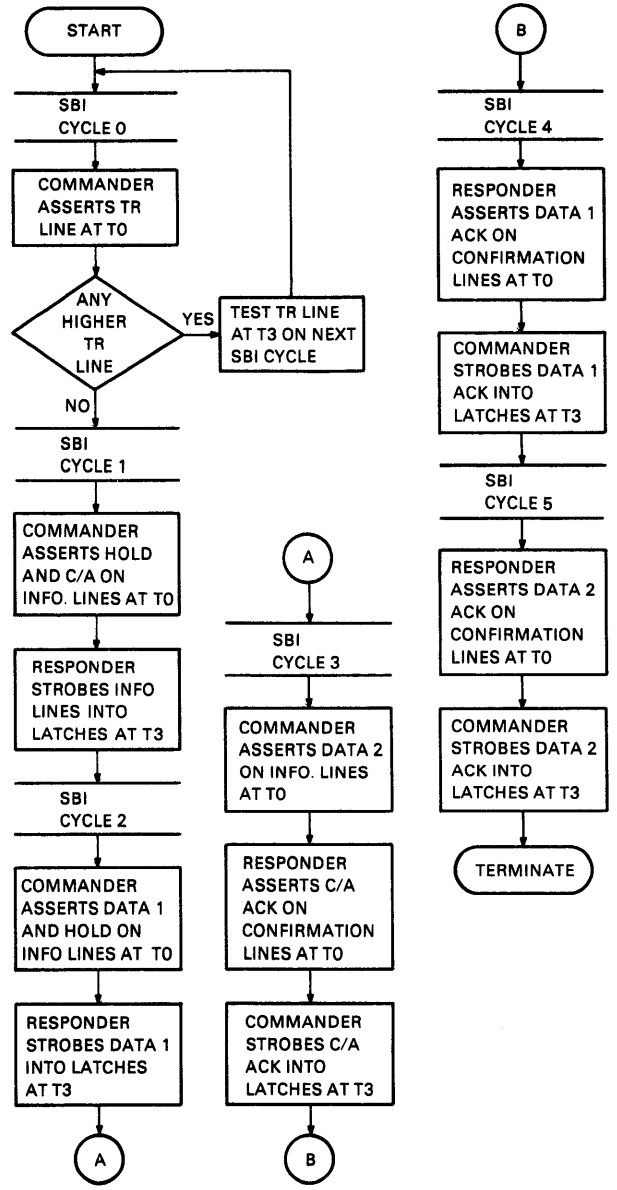


Figure 2-27 Extended Write Masked Timing Chart and Flow

2.8.7.5 Interlock Function Description -- The interlock function is used to provide coordination between memory nexus to ensure exclusive access to shared data structures. When an interlock sequence is addressed to the UBA, it indicates that a Data-In-Pause/Data-Out (DATIP/DATO) sequence is required on the Unibus. The UBA will initiate an interlock sequence when a Unibus device has initiated a DATIP/DATO sequence. The interlock functions operate like the read and write functions with the additional responsibility of setting and clearing the receiver nexus interlock flip-flop. This flip-flop controls the assertion/negation of the receiver's interlock line. However, not all nexus implement the interlock function. Those nexus that do not will respond to the interlock read and write masked functions exactly as they do to read and write masked functions.

All memory nexus implement the interlock functions and cooperate through the use of this signal. The interlock line is asserted by the commander nexus which issued the interlock read masked function for that SBI cycle following the command/address transfer. The interlock flip-flop is set in the receiving nexus memory. When the memory nexus confirms the interlock read function, it asserts the interlock signal in the same cycle as ACK. With interlock asserted, the nexus responds with a BUSY confirmation to subsequent interlock read masked commands only.

Interlock Read Masked Function Operation -- The interlock read masked function format is the same as that shown in Figure 2-20 except that F<3:0> is coded 0100. F<3:0> causes the nexus selected by SA<27:00> to retrieve and transfer the addressed data exactly as in the read masked operation. In addition, this function causes the selected nexus to set its interlock flip-flop. With the interlock flip-flop set, the nexus will assert the SBI interlock line at T0 of the ACK confirmation cycle.

The interlock flip-flop is cleared on receipt of an interlock write masked function. Interlock read masked and interlock write masked functions are always paired by commanders. If the flip-flop remains set for more than 102.4 us, the memory assumes that the commander has had a catastrophic error. In this case, the nexus will clear the flip-flop at T0 of the next cycle.

Interlock Write Masked Function Operation -- The interlock write masked function format is the same as that illustrated in Figure 2-22, except that F<3:0> is coded 0111, specifying the interlock write masked function. F<3:0> instructs the nexus selected by SA<27:00> to modify the bytes specified by M<3:0> in the addressed storage element using data transmitted in the succeeding cycle with TAG<2:0> = 101. In addition, the write data clears the interlock flip-flop set by the previous interlock read masked function.

2.8.8 Control Group

The control group functions synchronize system activities and provide specialized system communications. The clock functions provide SBI activity synchronization and are described in Paragraph 2.8.1. The interlock control, also one of the system commmunication functions, is described in Paragraph 2.8.7. The remaining control lines are described in the following paragraphs.

2.8.8.1 DEAD Function -- The DEAD signal indicates a dc power failure to the clock circuits or bus terminating networks. Nexus will not assert any SBI signal while DEAD is asserted. Thus, nexus prevent invalid data from being received while the SBI is in an unstable state.

The assertion of the power supply DC LO to the clock circuits or terminating networks causes the assertion of DEAD. DEAD is asserted asynchronously to the SBI clock and occurs at least 2 ~s before the clock becomes inoperative. With power restart, the clock will be operational for at least 2 ~s before DC LO is negated. The negation of DC LO negates DEAD.

2.8.8.2 FAIL Function -- A nexus enables the fail (FAIL) signal asynchronously to the SBI clock, when the power supply AC LO signal is asserted on that nexus. The assertion of FAIL inhibits the CPU from initiating a power-up service routine. FAIL is negated asynchronously with respect to the SBI clock when all nexus that are required for the power-up operation have detected the negation of AC LO. The CPU samples the FAIL line following the power-down routine (assertion of FAIL) to determine if the power-up routine should be initiated.

2.8.8.3 UNJAM Function -- The unjam function restores (initializes) the system to a known, well-defined state. The UNJAM signal is asserted only by the CPU or console, and is detected by all nexus connected to the SBI. The console asserts UNJAM only when a console key (or sequence) is selected. The duration of the UNJAM pulse is a minimum of 15 SBI cycles and is negated at TØ.

When the console intends to assert UNJAM, the CPU will assert TRØØ for a minimum of 15 SBI cycles. The CPU will continue to assert TRØØ for the duration of UNJAM and for a minimum of 15 SBI cycles after the negation of UNJAM. This use of TRØØ ensures that the SBI is inactive preceding, during, and after the UNJAM operation.

Each nexus receives UNJAM at T3 and begins a restore sequence. Any current operation of short duration will not be aborted if that operation might leave the nexus in an undefined state. Nexus will not perform operations using the SBI during the assertion of UNJAM. In addition, the nexus must be in an idle state, with respect to SBI activity, at the conclusion of the UNJAM pulse.

While UNJAM is asserted, nexus will not assert FAULT. However, a nexus asserting FAULT prior to UNJAM must continue to do so to preserve the content of the nexus configuration/fault status registers. The restore sequence (UNJAM asserted) should not cause a nexus to pass through any states that will assert any SBI lines. All read commands issued before the UNJAM are canceled.

In the event of a power failure during UNJAM, some nexus will assert FAIL and/or DEAD. The restore sequence should cause the nexus to negate ALERT or interrupt requests, but should not clear any device status bits.

CHAPTER 3
PROGRAMMING DEFINITIONS AND SPECIFICATIONS

3.1 GENERAL

This chapter describes some of the Massbus signals, clearing methods, and interrupt conditions; and each bit of the registers in the MBA.

3.2 DEFINITIONS

Some of the Massbus signals that are used in generating status information are described in this paragraph.

- Attention (ATTN) -- The ATTN line is a shared line that connects from all drives in common to the MBA. Each drive asserts ATTN (and sets its own ATA bit) whenever it has an error condition (ERR asserted), has just finished executing any movement command, or a change in power condition occurs.

  The logical expressions for these statements are:

  $ATTN = ATA_{\emptyset} + ATA_1 + \ldots + ATA_7$

  ATA_i = ERR_i + completion of a movement command + change in power conditions.

  (i represents the unit select code of a drive, Ø--7)

- Exception (EXC) -- The EXC line connects from the MBA to the drive that is performing a data transfer. It is asserted by the drive if an error occurs during the transfer. This line is used to distinguish errors in the drive performing a data transfer from errors signaled by the ATTN line. (A drive that is performing a data transfer never asserts ATTN while the data transfer is underway.)

- End of Block (EBL) -- The EBL line is pulsed by the drive performing a data transfer at the end of each sector.

- Clearing Methods

  Bit ØØ of the MBA control register is the Initialization (INIT) bit. The setting (writing a 1) of this bit will:

  Clear status bits in the MBA configuration register

  Clear abort data transfer on interrupt enable bits in the MBA control register

  Clear MBA status register

  Clear MBA byte count register

Clear the control and status bits of the diagnostic register

Cancel all pending commands except read data pending

Abort data transfers

Assert Massbus INIT.

3.3 PROGRAMMING NOTES

This paragraph describes miscellaneous features of the RH78Ø. The tables in subsequent paragraphs describe the other bits of the MBA. The DT Abort (Data Transfer Abort) bit is located in the MBA and is associated only with error conditions during data transfers and error conditions in the MBA. A drive clear command does not affect the DT Abort bit in the MBA. When DT END clears the DT BUSY bit in the MBA, it indicates that the MBA is ready for another data transfer command. To successfully initiate a data transfer command, DT BUSY must not be asserted and the appropriate drive ready bit must be asserted. A nondata transfer command can be issued to a drive any time a drive ready bit is asserted, regardless of the state of the DT BUSY bit in the MBA.

When a data transfer command is successfully initiated, DT BUSY is asserted and the drive ready bit is negated. When a nondata transfer command is successfully initiated, the drive ready bit is negated but the DT BUSY bit is not asserted.

If any command other than INIT is issued to a drive that has an error indicator asserted, the command will be ignored by the drive.

If a data transfer command is issued to a drive that has an error indicator asserted, the drive does not execute the command, the Missed Transfer Error (MXF, bit Ø8 of the status register) occurs in the MBA.

3.4 INTERRUPT CONDITIONS

The MBA generates an interrupt to the CPU due to the following conditions:

1. upon termination of a data transfer, if the IE bit is set when the MBA become ready;

2. upon assertion of ATTN line or the occurrence of an MBA error, while the MBA is not busy and the IE bit is set;

3. upon power up.

3.5 TERMINATION OF DATA TRANSFERS

A data transfer that has been initiated successfully may terminate in the following ways.

1. Normal Termination -- Byte count overflows to Ø and the MBA becomes ready (DT BUSY cleared).

2. MBA Error -- An error occurs in the MBA status register.

Bit	Error Indication
19	PGE (Program Error)
18	NED (Non-Existing Device)
17	MCPE (Massbus Control Parity Error)
12	DT ABORT (Data Transfer Aborted)
11	DLT (Data Late)
1Ø	WCK UP ERR (Write Check Upper Error)
Ø9	WCK LWR ERR (Write Check Lower Error)
Ø8	MXF (Missed Transfer Error)
Ø7	MBEXC (Massbus Exception)
Ø6	MDPE (Massbus Data Parity Error)
Ø5	MAPPE (Page Frame MAP Parity Error)
Ø4	INVMAP (Invalid MAP)
Ø3	ERR CONF (Error Confirmation)
Ø2	RDS (Read Data Substitute)
Ø1	IS TIMEOUT (Interface Sequence Timeout)
ØØ	RD TIMEOUT (Read Data Timeout)

3. Drive Error -- An error occurs in the drive. The drive sets the appropriate error bit in the drive.

4. Program-Caused Abort -- By causing INIT to be asserted, the MBA aborts all operations and all status and error information is lost.

3.6 MBA REGISTERS

The RH78Ø contains 8 registers and 256 MAP registers. All registers in the MBA or in the drives can only be accessed by longword references and must be longword aligned (on byte Ø of a given register). An attempt to MOVB or MOVW to or from an MBA subsystem register will result in a machine check trap. This will also occur when an attempt is made to MOVL to or from a register if either bit Ø or bit 1 of the address is a 1. An explanation of these registers and their functions is provided in subsequent paragraphs. Table 3-1 lists the various RH78Ø registers.

Table 3-1 MBA Registers

Massbus Address (Hex)	Register	Mnemonic	Type
00	Configuration/Status	CSR	Read/write
01	Control	CR	Read/write
02	Status	SR	Read/write
03	Virtual Address	VAR	Read/write
04	Byte Counter	BCR	Read/write
05	Diagnostic	DR	Read/write
06	Selected Map	SMR	Read only
07	Command Address	CAR	Read only

The eight internal MBA registers contain various configuration, status, and control information. The addresses of these registers can be derived from Tables 3-2 and 3-3. Table 3-2 shows the MBA's base address in respect to its various TR levels. Table 3-3 shows the byte offset for the registers in the MBA. This byte offset is added to the MBA's base address to produce a particular register's address.

Example

The address of the diagnostic register for an MBA with a TR of 8:

Base address from Table 3-2	20010000
Byte offset for the DR from Table 3-3	+ 14
DR address	20010014

Example

The address of the control register for an MBA with a TR of C:

Base address from Table 3-2	20018000
Byte offset for the CR from Table 3-3	+ 04
CR address	20018004

Table 3-2 MBA Base Addresses

MBA TR Level	Console Physical Base Address
1	20002000
2	20004000
3	20006000
4	20008000
5	2000A000
6	2000C000
7	2000E000
8	20010000
9	20012000
A	20014000
B	20016000
C	20018000
D	2001A000
E	2001C000
F	2001E000
10	20020000

Table 3-3 Register Byte Offsets

Register	Offset from Console Base Address
Configuration/Status Register (CSR)	ØØ
Control Register (CR)	Ø4
Status Register (SR)	Ø8
Virtual Address Register (VAR)	ØC
Byte Counter Register (BCR)	1Ø
Diagnostic Register (DR)	14
Selected MAP Register (SMR)	18
Command/Address Register (CAR)	1A

3.6.1 Configuration/Status Register (CSR)

The configuration/status register is a read/write MBA register that contains fault status, interrupt status, adapter dependent status, and adapter code bits. Figure 3-1 illustrates these bits and Table 3-4 provides an explanation for the various bits in this register.

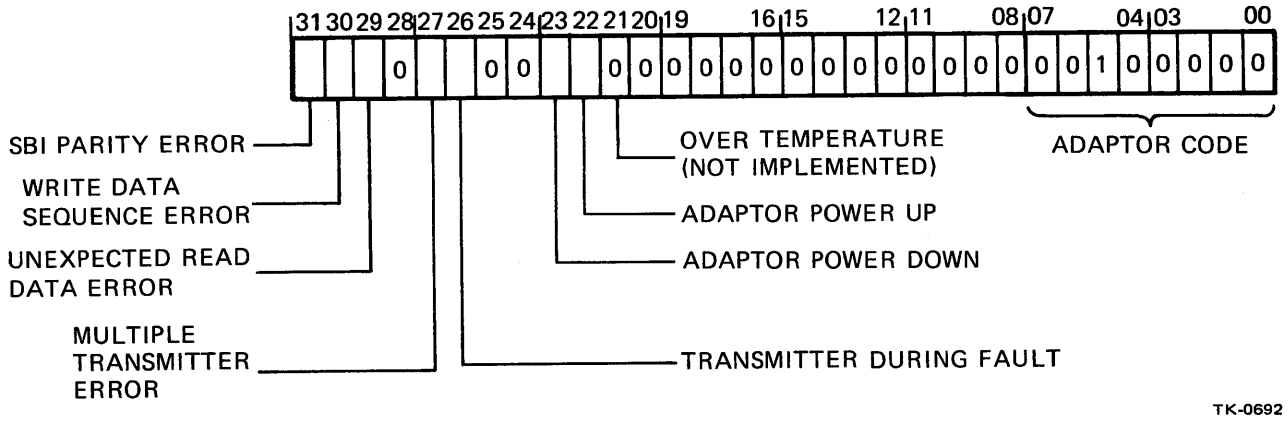


Figure 3-1 Configuration/Status Register (CSR)

Table 3-4 Configuration/Status Register (CSR) Bit Assignments

Bit	Set By/Cleared By	Remarks
31 PE SBI Parity Error (Fault A)	Set when an SBI parity error is detected. Cleared by power fail or the deassertion of fault.	The setting of this bit will cause fault to be asserted on the SBI for one cycle.
30 WS Write Data Sequence (Fault B)	Set when no write data is received (neither tag = write data nor ID = write command ID) following a write command. Cleared by power fail or the deassertion of fault.	The setting of this bit will cause fault to be asserted on the SBI for one cycle.
29 URD Unexpected Read Data (Fault C)	Set when read data is received and not expected. Cleared by power fail or the deassertion of fault.	The setting of this bit will cause fault to be asserted on the SBI for one cycle.
28 0		Reserved for future use.
27 MT Multiple Transmitter (Fault D)	Set when the ID on the SBI does not agree with the ID transmitted by the MBA while the MBA is transmitting data on the SBI. Cleared by power fail or the deassertion of fault on the SBI.	The setting of this bit will cause fault to be asserted on the SBI. (The fault signal will be asserted at the normal confirmation time for one cycle if the MBA detects one of the fault conditions. The negation of the fault signal on the SBI will clear all the fault status bits. Fault = Fault A + Fault B + Fault C + Fault D.
26 XMTFLT Transmit Fault	Set when the SBI fault is detected at the 2nd cycle after the MBA transmits information on to the SBI. Cleared by power fail or the deassertion of fault.	

Table 3-4 Configuration/Status Register (CSR) Bit Assignments (Cont)

Bit	Set By/Cleared By	Remarks
25:24 All Ø's		Reserved for future use.
23 PD Adapter Power Down	Set when the MBA power goes down. Cleared when power goes up.	The setting of this bit will cause interrupt to the CPU if IE is set.
22 PU Adapter Power Up	Set when the MBA power goes up. Reset when power goes down. Cleared by assertion of INIT, UNJAM, DC LO, or writing a 1 into this bit.	The setting of this bit will also set the IE bit and interrupt the CPU.
21 OT Over Temperature	Always Ø.	
2Ø:Ø8 All Ø's		Reserved for future use.
Ø7:ØØ Adapter Code		Each adapter is assigned a unique code identifying it. The MBA adapter code is: Bit <Ø7:ØØ> = ØØ1ØØØØØ

3.6.2 Control Register (CR)

The control register is a read/write register that contains the control bits: Interrupt Enable, Abort, and Initialization. This register can put the MBA in the maintenance mode. Figure 3-2 illustrates this register's bits; Table 3-5 provides an explanation of the bits.

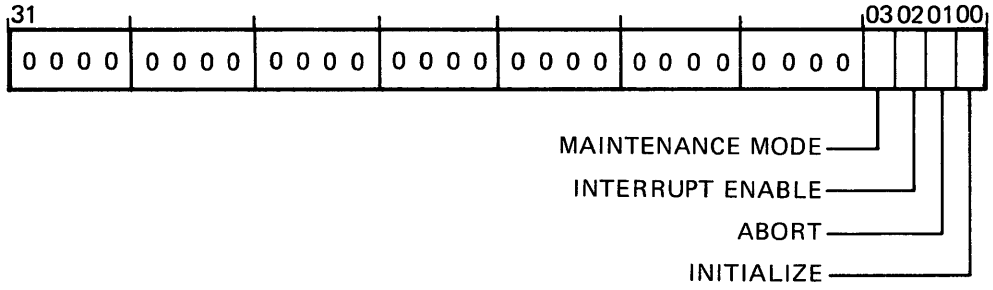


Figure 3-2 Control Register (CR)

Table 3-5 Control Register (CR) Bit Assignments

Bit	Set By/ Cleared By	Remarks
31:04 All 0's		Reserved for future use.
03 MM Maintenance Mode		The setting of this bit will put the MBA in the maintenance mode, which will allow the diagnostic programmer to exercise and examine the Massbus operations without a Massbus device. When this bit is set, the MBA will block RUN, DEM, and assert FAIL to the Massbus so that all the devices on the Massbus will detach from the Massbus. This bit can only be set if a data transfer is not in progress.

Table 3-5 Control Register (CR) Bit Assignments (Cont)

Bit	Set By/ Cleared By	Remarks
Ø2 IE Interrupt Enable	Set by writing a 1 or power up. Cleared by writing Ø or INIT.	Allows the MBA to interrupt CPU when certain conditions occur.
Ø1 ABORT Abort Data Transfer	Set by writing 1; cleared by writing Ø, INIT, or UNJAM.	The setting of this bit will initiate the data transfer abort sequences that will stop sending commands and addresses, and stop the byte counter. It will also negate RUN, assert EXC to Massbus, wait for EBL, set ABORT to 1 at trailing edge of EBL. Interrupt CPU if IE bit is 1.
ØØ INIT Initialization	This bit is self-clearing (always reads as Ø).	The setting of this bit will: clear status bits in the MBA configuration register, clear ABORT and IE in the MBA control register, clear MBA status register, clear MBA byte count register, clear control and status bits of the diagnostic registers. It will also Cancel all pending commands except read data pending abort data transfer. Asserts Massbus INIT.

3.6.3 Status Register (SR)

The status register is a read/write register that contains MBA status information such as: error indications, timeouts, and busy indicators. Figure 3-3 illustrates the bit assignments and Table 3-6 describes the functions of the bits in the status register. All interrupts will occur immediately if there is no data transfer in progress. The interrupt will be postponed until the data transfer has terminated.

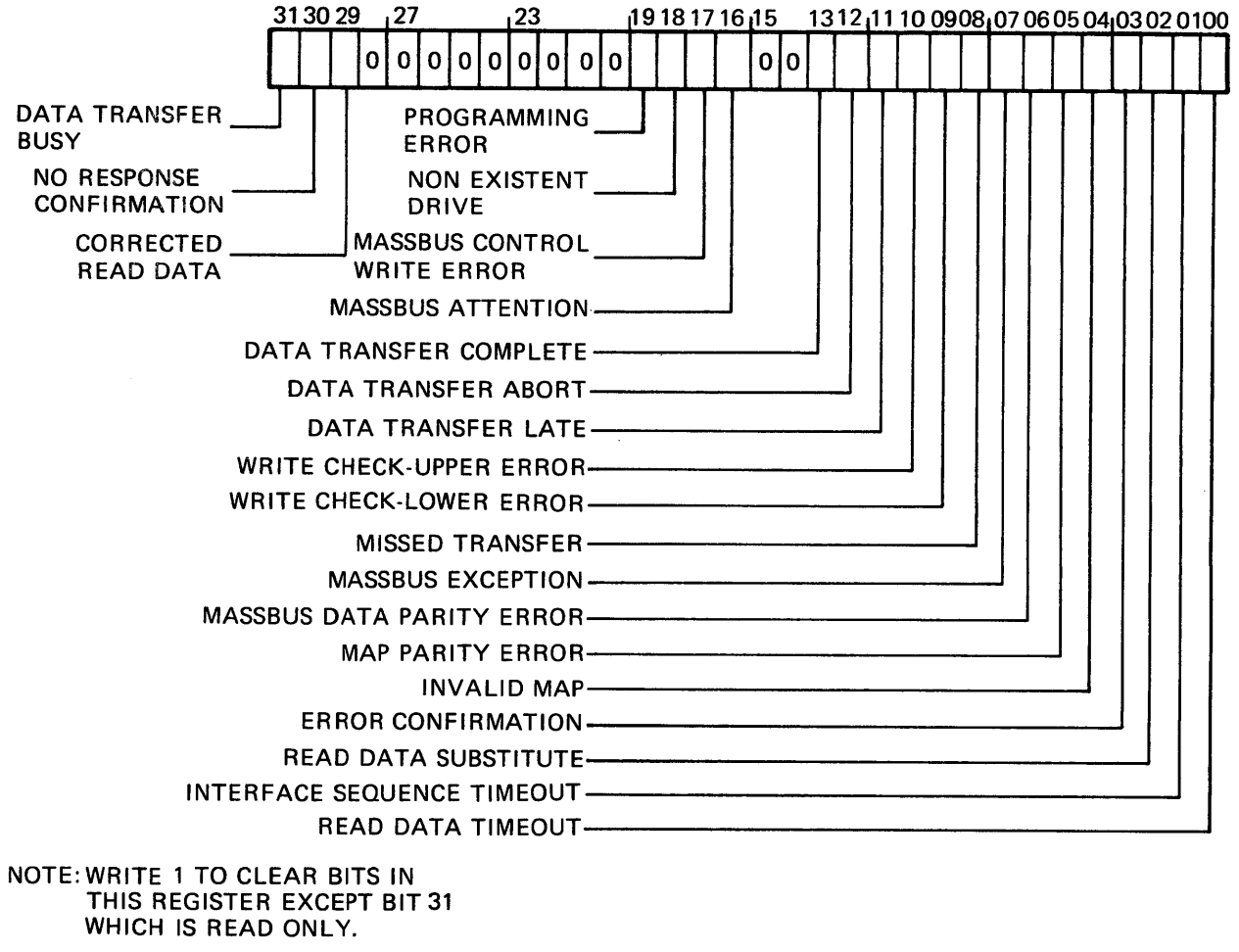


Figure 3-3 Status Register (SR)

Table 3-6 Status Register (SR) Bit Assignments

Bit	Set By/Cleared By	Remarks
31 DTBUSY Data Transfer Busy	Set when a data transfer command is received. Cleared when a data transfer is aborted.	Read only.
30 NRCONF No Response Confirmation	Set when the MBA receives a no-response confirmation for the read command or write command and write data sent to the SBI. Cleared by writing a 1 to this bit or INIT.	The setting of this bit will cause retry of the command.
29 CRD Corrected Read Data	Set when tag of read data received from memory is CRD. Cleared by writing a 1 to this bit or INIT.	
28:20 All 0's		Reserved for future use.
19 PGE	Set when one or more of the following conditions exists. Program tries to initiate a data transfer when the MBA is currently performing one. Program tries to load MAP, VAR, or byte counter while the MBA is performing a data transfer operation. Program tries to set MBA maintenance mode during a data transfer operation. Program tries to initiate a nonacceptable data transfer command. Cleared by writing a 1 to this bit.	The setting of this bit will cause an interrupt to the CPU if IE bit in the control register is set.
18 NED Nonexisting Drive	Set when drive fails to assert TRA within 1.5 us after assertion of DEM. Cleared by writing a 1 to this bit.	The setting of this bit will send zero read data back to the SBI, and interrupt the CPU if IE bit in the control register is set.

Table 3-6 Status Register (SR) Bit Assignments (Cont)

Bit	Set By/Cleared By	Remarks
17 MCPE Massbus Control Parity Error	Set when Massbus control parity occurs. Cleared by writing a 1 to this bit.	The setting of this bit will cause an interrupt to CPU if the IE bit in the control register is set.
16 ATTN Attention from Massbus	Set when the attention line in the Massbus is asserted.	The setting of this bit will cause an interrupt to the CPU if the IE bit in the control register is set.
15:14 All Ø's		Reserved for future use.
13 DT COMP Data Transfer Completed	Set when data transfer is completed. Cleared by writing a 1 to this bit.	The setting of this bit will cause an interrupt to the CPU if the IE bit in the control register is set.
12 DTABT Data Transfer Aborted	Set with the trailing edge of EBL when the data transfer has been aborted. Cleared by writing a 1 to this bit or INIT.	The setting of this bit will cause an interrupt to the CPU if the IE bit in the control register is set.
11 DLT Data Late	This bit is set: 1) for either a write data transfer or a write check data transfer providing the data buffer is empty when WCLK is sent to the Massbus, 2) for a read data transfer providing the data buffer is full when SCLK is received from the Massbus.	The setting of this bit will cause the data transfer to be aborted.
1Ø WCK UP ERR Write Check Upper Error	Set when a compare error is detected in the upper byte while the MBA is performing a write check operation. Cleared by writing a 1 to this bit or INIT.	The setting of this bit will cause the data transfer to be aborted.

Table 3-6 Status Register (SR) Bit Assignments (Cont)

Bit	Set By/Cleared By	Remarks
Ø9 WCK LWR ERR Write Check Lower Error	Set when a compare error is detected in the lower byte while the MBA is performing a write check operation. Cleared by writing a 1 to this bit or INIT.	The setting of this bit will cause the data transfer to be aborted.
Ø8 MXF Missed Transfer Error	Set when no OCC or SCLK is received within 5Ø us after data transfer busy is set. Cleared by writing a 1 to this bit or INIT.	The setting of this bit will cause an interrupt to the CPU if the IE bit in the control register is set.
Ø7 MBEXC Massbus Exception	Set when EXC is received from Massbus. Cleared by writing a 1 to this bit or INIT.	The setting of this bit will cause the data transfer to be aborted.
Ø6 MDPE Massbus Data Parity Error	Set when a Massbus data parity error is detected during a read data transfer operation. Cleared by writing a 1 to this bit or INIT.	The setting of this bit will cause the data transfer to be aborted.
Ø5 MAPPE Page Frame Map Parity Error	Set when a parity error is detected on the data read from the map during a data transfer. Cleared by writing a 1 to this bit or INIT.	The setting of this bit will cause the data transfer to be aborted.
Ø4 INVMAP Invalid Map	Set when the valid bit of the next page frame number is zero and the byte counter is not zero. Cleared by writing a 1 to this bit or INIT.	The setting of this bit will cause the data transfer to be aborted.

Table 3-6 Status Register (SR) Bit Assignments (Cont)

Bit	Set By/Cleared By	Remarks
Ø3 ERR CONF Error Confirmation	Set when the MBA receives error confirmation for a read or write command. Cleared by writing a 1 to this bit or INIT.	The setting of this bit will cause the data transfer to be aborted.
Ø2 RDS Read Data Substitute	Set when the TAG of the read data received from memory is read data substitute. Cleared by writing a 1 to this bit or INIT.	The setting of this bit will cause the data transfer to be aborted.
Ø1 IS TIMEOUT Interface Sequence Timeout	Set when an interface sequence timeout occurs. An interface sequence timeout is defined as the time from when arbitration for the SBI is begun until: 1) ACK is received for a command address transfer that specifies read; or 2) ACK is received for a command/address transfer that specifies write and write data; or 3) ERR confirmation is received for any command/address transfer. The maximum timeout is 1Ø2.4 us. Cleared by writing a 1 to this bit or INIT.	The setting of this bit will cause the data transfer to be aborted.
ØØ RD TIMEOUT Read Data Timeout	Set when a read data timeout occurs. A read data timeout is defined as the time from when an interface sequence that specifies a read command is completed to the time that the specified read data is returned to the commander. The maximum timeout is 1Ø2.4 ~s. Cleared by writing a 1 to this bit or INIT.	The setting of this bit will cause the data transfer to be aborted.

3.6.4 Virtual Address Register (VAR)

Before a data transfer is initiated, the program must load an initial virtual address (pointing to the first byte to be transferred) into this register. Figure 3-4 illustrates the bit assignments for the virtual address register. Bits Ø9 through 16 select one of the 256 MAP registers. The contents of the selected MAP register and the value of bits ØØ through Ø8 are used to assemble a physical SBI address to be sent to memory. Bits ØØ through Ø8 indicate the byte offset into the page of the current data byte. The virtual address register may not be written into during a data transfer. An attempt to do so will set PGE, but the virtual address register will not be modified and the data transfer will continue.

NOTE

The MBA virtual address register is incremented by eight after every memory read or write and will not point to the next byte to be transferred if the transfer does not end on a quadword boundary (it will point eight bytes ahead). When a write check error occurs, the virtual address register will not point to the failing data in memory due to the preloading of the silo data buffer. The virtual address of the bad data may be found by determining the number of bytes actually compared on the Massbus (the difference between bits 16 to 31 of the byte count register and their initial value) and adding that difference to the initial virtual address.

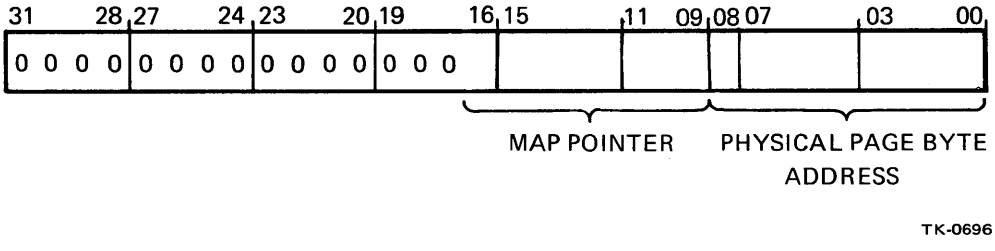


Figure 3-4 Virtual Address Register (VAR)

3.6.5 Byte Count Register (BCR)

The program loads the 2's complement of the number of bytes for the data transfer to bits 15 through ØØ of this register. The MBA hardware will load these 16 bits into bits 31 through 16 and bits 15 through ØØ of the byte count register. Bits 31 through 16 serve as the byte counter for the number of bytes transferred through the Massbus and bits 15 through ØØ serve as the byte counter for the number of bytes transferred through the SBI interface. The starting byte count with 16 bits of zeros is the maximum number of bytes of a data transfer. Figure 3-5 illustrates the byte count register's bit assignments. The byte count register may not be modified during a data transfer. An attempt to do so will be ignored and PGE will be set.

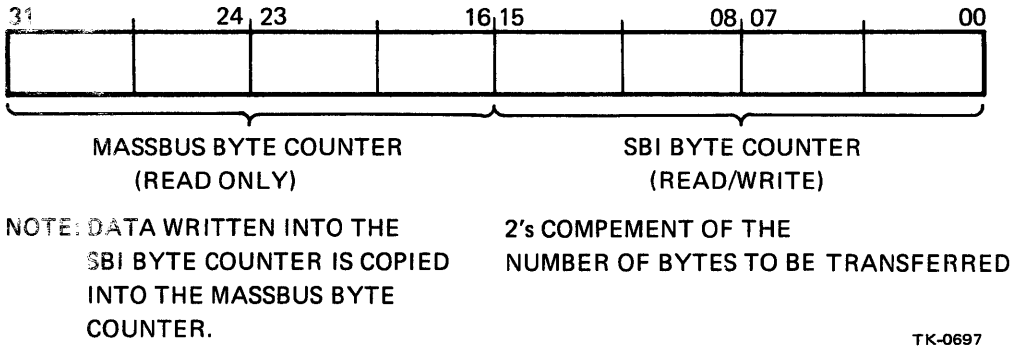


Figure 3-5 Byte Counter Register (BCR)

3.6.6 Diagnostic Register (DR)

The diagnostic register is a read/write register that contains MBA diagnostic information. This register allows diagnostics to be run without any drives on the Massbus. Figure 3-6 illustrates the bit assignments, and Table 3-7 describes the function of the various bits in this register. The diagnostic register may not be written unless the MBA is in the maintenance mode. An attempt to write the diagnostic register when not in the maintenance mode will be ignored. Caution should be exercised when reading this register in the maintenance mode. The data path used to read bits ØØ through Ø7 may inject invalid data into the silo if the MBA has just read data from memory. It is advisable to wait 2Ø us from the initiation of a transfer or the deassertion of SCLK before reading or modifying this resister.

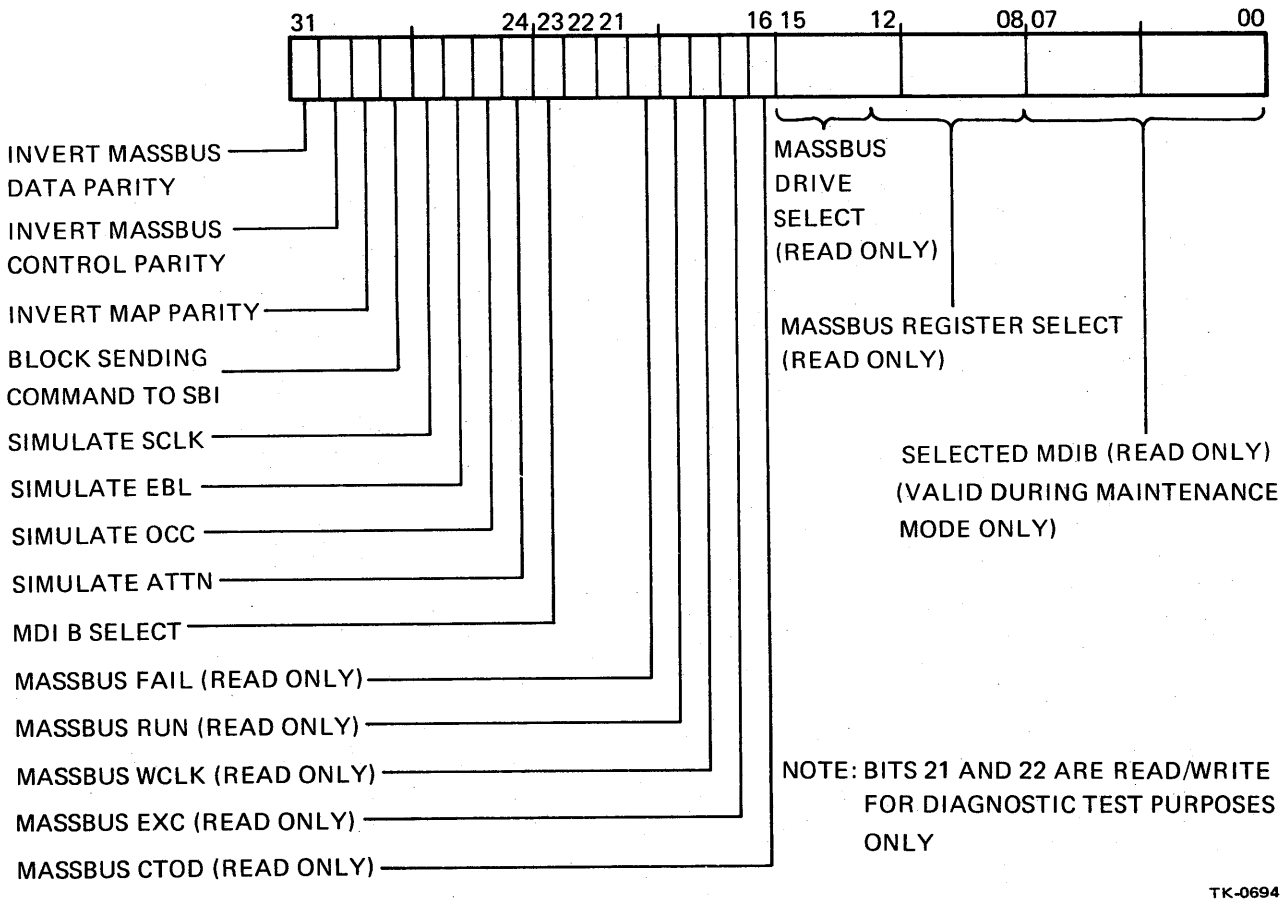


Figure 3-6 Diagnostic Register (DR)

Table 3-7 Diagnostic Register (DR) Bit Assignments

Bit		Description
31	IMDPG	Invert Massbus Data Parity Generator.
30	IMCPG	Invert Massbus Control Parity Generator.
29	IMAPP	Invert MAP Parity.
28	BLKSCOM	Block Sending Command to the SBI. During a data transfer, the setting of this bit will eventually cause a DLT bit set and a DT ABORT.
27	SIMSCLK	Simulate SCLK. When the MM bit is set, writing a 1 to this bit will simulate the assertion of SCLK; writing a 0 to this bit will simulate the deassertion of SCLK.

Table 3-7 Diagnostic Register (DR) Bit Assignments (Cont)

Bit		Description
26	SIMEBL	Simulate EBL. When MM bit is set, writing a 1 to this bit will simulate the assertion of EBL; writing a Ø to this bit will simulate the deassertion of EBL.
25	SIMOCC	Simulate OCC. When MM bit is set, writing a 1 to this bit will simulate the assertion of OCC; writing a Ø to this bit will simulate the deassertion of OCC.
24	SIMATTN	Simulate ATTN. When MM bit is set, writing a 1 to this bit will simulate assertion of ATTN; writing a Ø to this bit will simulate the deassertion of ATTN.
23	MPIB SEL	Maintenance Massbus Data Input Buffer Select. When this bit is set to a 1, the upper eight bits (B<15:Ø8>) of the MDIB will be sent out from bits Ø7 through ØØ of the diagnostic register if the diagnostic register is read. When the bit is Ø, the lower eight bits (B<Ø7:ØØ>) of the MDIB will be sent out from bits Ø7 through ØØ of the diagnostic register if it is read.
22:21	MAINT ONLY	Read/write with no effect. Used to test the writability of these bits.
2Ø	MFAIL	Massbus Fail (read only). Fail is asserted when MM is set.
19	MRUN	Massbus Run (read only).
18	MWCLK	Massbus WCLK (read only).
17	MEXC	Massbus EXC (read only).
16	NCTOD	Massbus METOD (read only).
15:13	MDS	Massbus Device Select (read only).
12:8	MRS	Massbus Register Select (read only).
7:Ø	U/L MDIB	Maintenance Upper/Lower MDIB.

3.6.7 Selected MAP Register (SMR)

This register is read-only and has the same format as a MAP register (Paragraph 3.6.9) but is valid only when DT BUSY is set. This is the contents of the MAP register pointed to by bits 16 through Ø9 of the virtual address register. Figure 3-7 illustrates the bit assignments of the selected MAP register.

3.6.8 Command Address Register (CAR)

This register is read-only and valid only when DT BUSY is set. It contains the value of bits 31 through ØØ of the SBI during the command/address part of the MBA's next data transfer.

3.6.9 MAP Registers

The MBA contains 256 MAP registers, each of which may be selected by address bits ØØ to Ø7. Bit 31 of the MAP registers is a valid bit, bits 3Ø through 21 are all Ø's (intended for future use), and bits 2Ø through ØØ represent the physical page frame number. Bits Ø9 and Ø8 are 1 and Ø, respectively. MAP registers can only be written when there is no data transfer operation in progress. A write to a MAP register while a data transfer is in progress will be ignored and cause the setting of PGE and interrupt the CPU at the end of a transfer if IE is set. Figure 3-7 illustrates the bit assignment of the MAP registers.

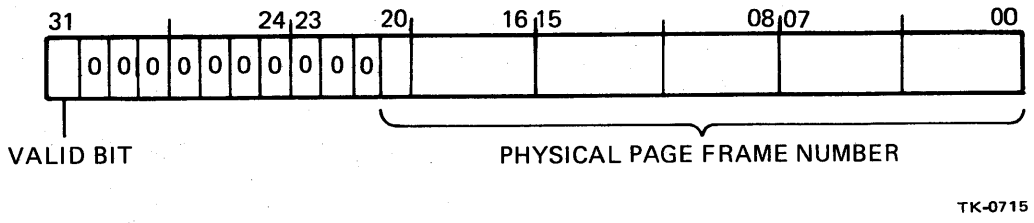


Figure 3-7 MAP Registers

3.7 DRIVE REGISTER CALCULATIONS

The registers within a drive contain various drive information. The address of a particular register in a drive is dependent on the drive's unit number and the assigned TR level of the MBA that controls the drive(s). To calculate the address of a register, follow the procedure outlined below.

1. Obtain the base address of the MBA from Table 3-2.

2. Locate the desired register number and drive number for the drive register from Table 3-8. The intersection of the row and column will be the register offset.

3. Add the MBA base address to the register offset to obtain the register address.

Example
To determine the address of register number Ø2 in drive number Ø4 with an MBA with a TR level of Ø8:

MBA base address	20010000
Intersection of register and drive number	+ 608
Register Ø2 in drive number Ø4	20010608

Since Massbus drive registers are only 16 bits wide, the following convention has been adopted.

1. On writes only, the low 16 bits of the longword will be written.

2. On reads, the drive register will supply the low 16 bits and the MBA's Status Register (SR) will supply the upper 16 bits (B<31:16> of the SR) when a drive register is read.

Table 3-8 Drive Address Conversion

MBA Base Address + (Value from table below) = register address

Register	Drive Number							
	Ø	1	2	3	4	5	6	7
ØØ	4ØØ	48Ø	5ØØ	58Ø	6ØØ	68Ø	7ØØ	78Ø
Ø1	4Ø4	484	5Ø4	584	6Ø4	684	7Ø4	784
Ø2	4Ø8	488	5Ø8	588	6Ø8	688	7Ø8	788
Ø3	4ØC	48C	5ØC	58C	6ØC	68C	7ØC	78C
Ø4	41Ø	49Ø	51Ø	59Ø	61Ø	69Ø	71Ø	79Ø
Ø5	414	494	514	594	614	694	714	794
Ø6	418	498	518	598	618	698	718	798
Ø7	41C	49C	51C	59C	61C	69C	71C	79C
Ø8	42Ø	4AØ	52Ø	5AØ	62Ø	6AØ	72Ø	7AØ
Ø9	424	4A4	524	5A4	624	6A4	624	6A4
ØA	428	4A8	528	5A8	628	6A8	728	7A8
ØB	42C	4AC	52C	5AC	62C	6AC	72C	7AC
ØC	43Ø	4BØ	53Ø	5BØ	63Ø	6BØ	73Ø	7BØ
ØD	434	4B4	534	5B4	634	6B4	734	7B4
ØE	438	4B8	538	5B8	638	6B8	738	7B8
ØF	43C	4BC	53C	5BC	63C	6BC	73C	7BC

CHAPTER 4
MBA FUNCTIONAL/LOGIC DESCRIPTION

4.1 GENERAL

This chapter provides a functional description of the RH78Ø Massbus Adapter. Logic descriptions are included in subsequent paragraphs, where necessary, to clarify operation of the MBA. The chapter is divided into four sections: MBA/SBI interface, internal registers, data paths, and control paths. The reader should be familiar with the material in the preceding chapters and the timing diagrams in the RH78Ø Field Maintenance Print Set (REV B or later). The print set will be referenced throughout this discussion. Figure 4-1 is a block diagram of the RH78Ø Massbus Adapter.

4.2 MBA/SBI INTERFACE

The MBA/SBI interface accepts information from the SBI when it recognizes itself as the intended receiver. Figure 4-2 is a block diagram of the interface. The various components of the interface are discussed in the following paragraphs.

4.2.1 SBI Decoding and Validating (Overview)

CPU transfers to or from MBA (or drive) registers require one or two successive SBI transfer cycles. The first cycle always contains the command/address. The second contains the data word if the command was a write.

The command/address placed on the SBI data lines by the CPU at TØ is loaded into the SBI/MBA transceivers between T2 and T3 and latched at T3. The time lapse between assertion and latching of the input in the transceivers provides deskew and settling time to compensate for any propagation delays on the SBI.

The parity of the latched information is then checked and the TAG field is decoded. If the tag is determined to be a command/address, the SBI address B<14:11> is compared with the preselected MBA device address to determine if the MBA is the intended recipient of the SBI command. If the address is not the MBA's, the information will not be processed, nor will the MBA issue a confirmation (no response). In this case, the interface circuits will continue to check the SBI data lines during the following SBI cycles until a valid command/address is latched.

The MBA only recognizes the following SBI commands: write masked, interlocked write masked (entire longword only), read masked, interlocked read masked, and interrupt summary read.

If the MBA recognizes the address and is not performing another SBI data transfer operation (MBA BUSY is not asserted) and the mask and function bits indicate a valid command, the MBA will issue an ACK confirmation to the CPU verifying that the command/address has been received correctly. Confirmation occurs two cycles after receipt of the command/address.

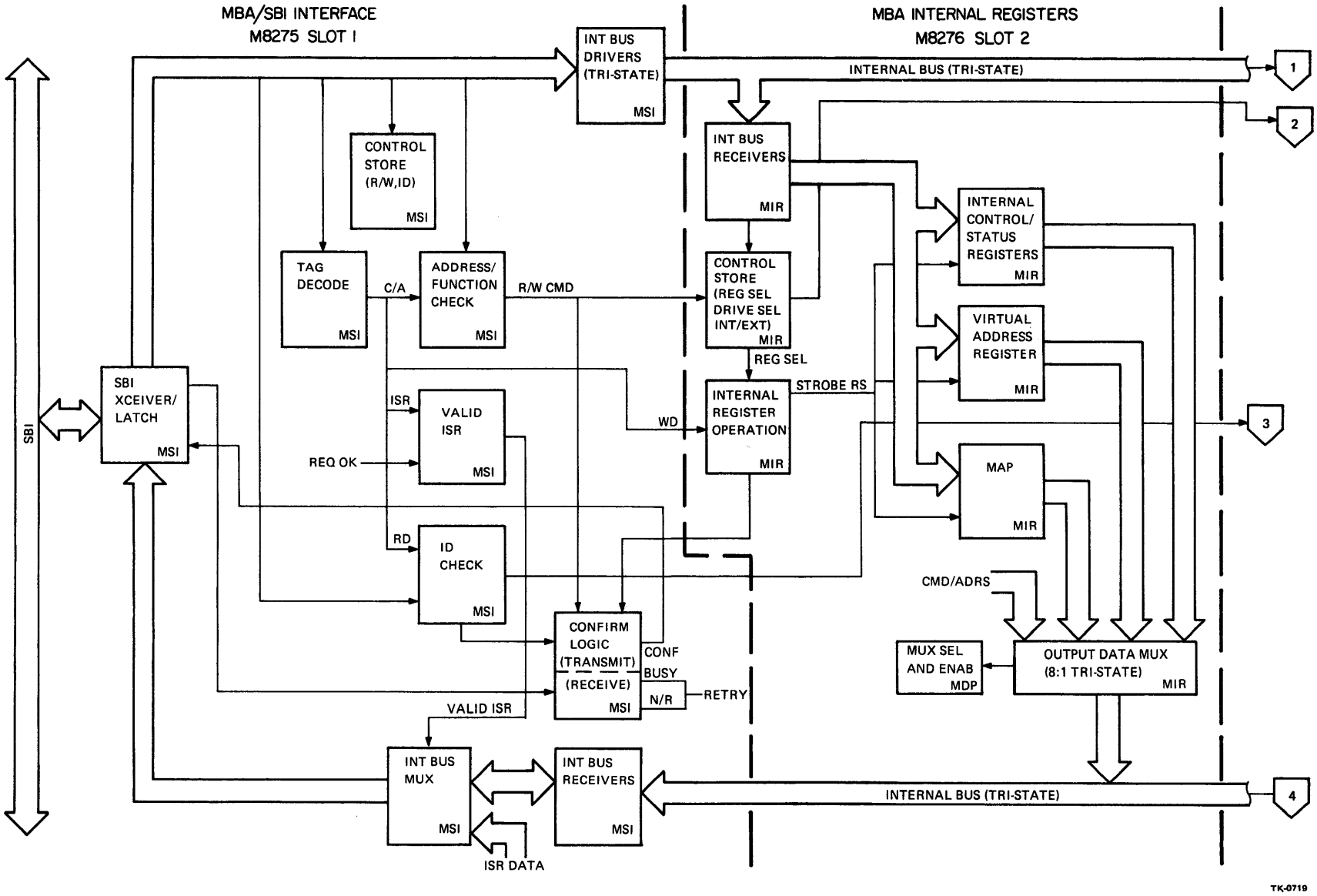


Figure 4-1 MBA Block Diagram
(Sheet 1 of 2)

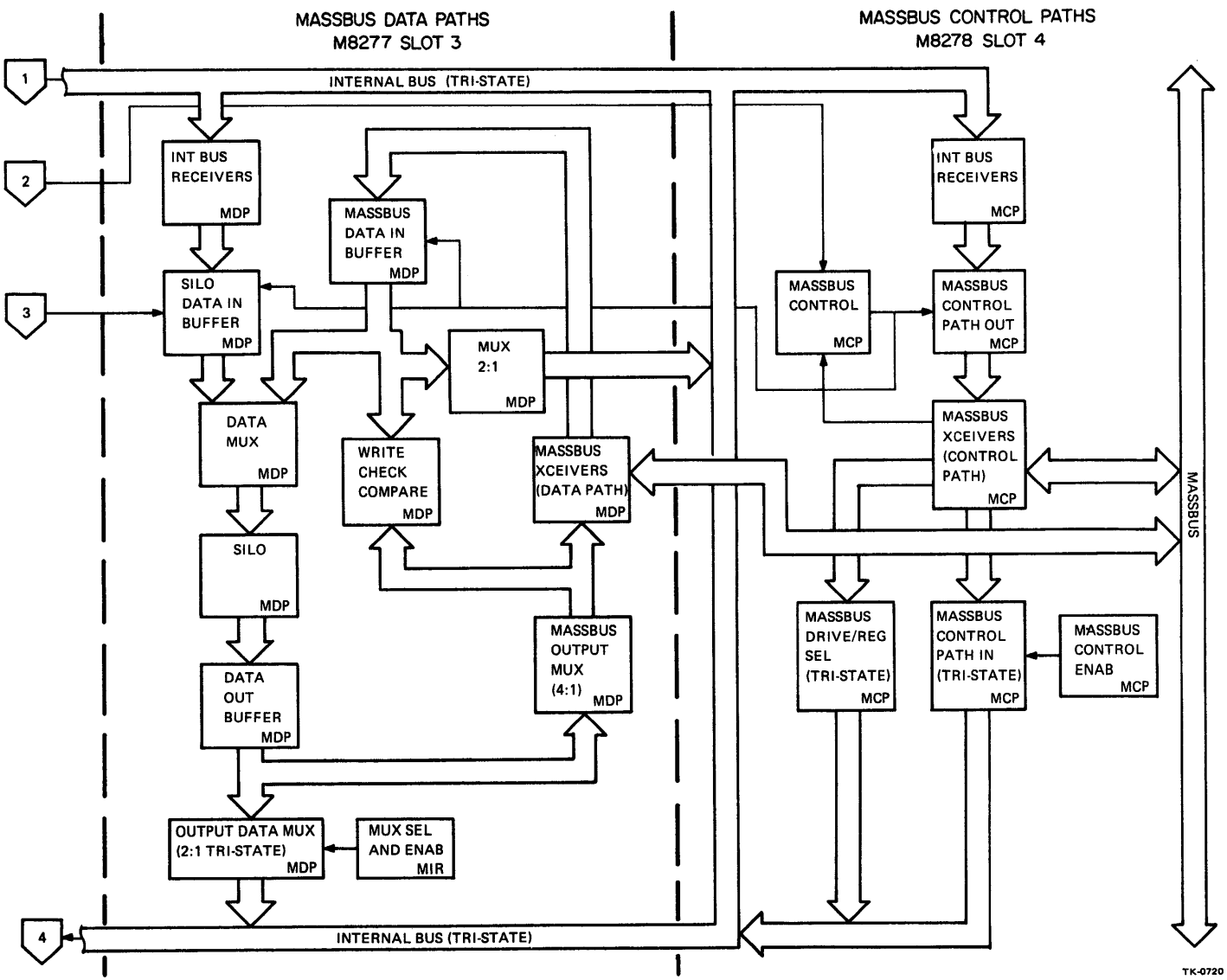


Figure 4-1 MBA Block Diagram
(Sheet 2 of 2)

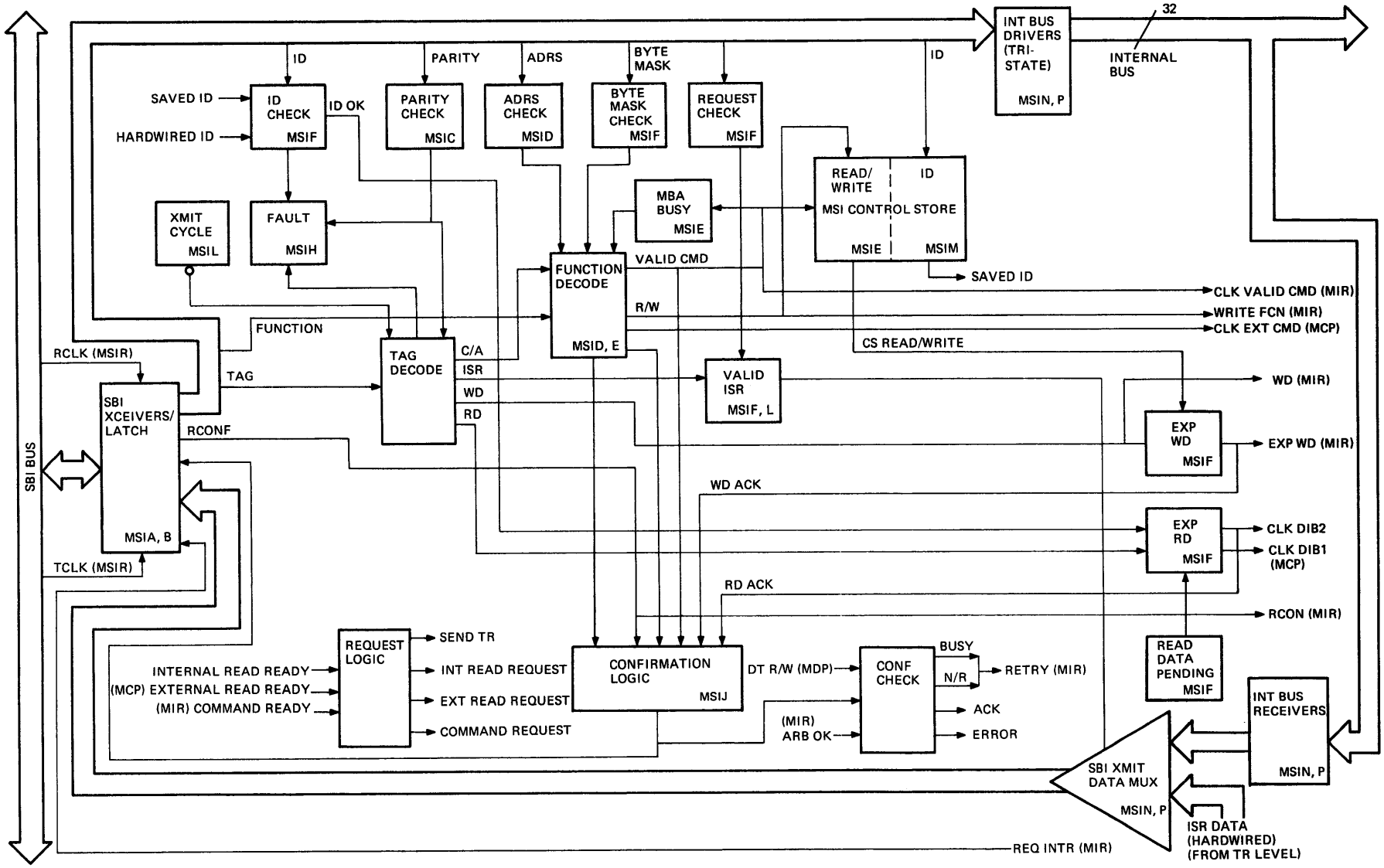


Figure 4-2 MBA/SBI Interface

If the MBA is busy, a BUSY confirmation signal is sent to the CPU in response to the valid command/address. The CPU will then rearbitrate for use of the SBI and repeat the transmission until the command/address is accepted or until the CPU times out.

It should be noted that there are two busy indications in the MBA, DT BUSY and MBA BUSY. DT BUSY indicates that a data transfer is in progress to or from the storage media. While DT BUSY is set, the MBA registers may be read and some written. MBA BUSY indicates that the MBA's SBI interface is in use. While MBA BUSY is asserted, the MBA will respond with a BUSY confirmation to all commands issued by the CPU. DT BUSY remains asserted for the duration of a data transfer (typically 10--100 ms). During a data transfer, MBA BUSY will be set or reset as the MBA uses the SBI to transfer 8 data bytes to or from memory in an 8-byte burst. Typically, MBA BUSY will be set for approximately 1 us and will remain clear until enough data has accumulated to cause another transfer (approximately 10 us). While MBA BUSY is clear, the CPU may access MBA registers.

If the MBA recognizes the address but the command is invalid, an error confirmation will be issued two cycles after receipt of the command/address causing the CPU to abort the attempted transfer.

After receipt of the command/address, the appropriate MBA BUSY and/or the expected write data flip-flop is set, depending on the read/write function to be performed. These functions will be described later.

Note that the parity checking, decoding, and validation functions described in the following paragraphs are performed in parallel by the SBI interface.

4.2.2 Timing (MSIR)

All data transfers between the SBI and the MBA are synchronized by six timing signals: BUS SBI TP H, BUS SBI TP L, BUS SBI PCLK H, BUS SBI PCLK L, BUS SBI PDCLK H, and BUS SBI PDCLK L. These timing signals are generated by the CPU. The MBA uses these timing signals to derive four SBI time states: T0, T1, T2, and T3. These derived time states define an SBI cycle (200 ns). The major MBA timing signals and their relationship to the SBI timing is illustrated in Figure 4-3. These signals are used to synchronize MBA operations with the SBI.

4.2.3 SBI Control and Data Transceivers (MSIA, MSIB)

Information transferred to and from the MBA is loaded and latched in the control and data transceivers (Figure 4-2) by TCLKA and TCLKB (T0, transmit) or RCLKA and RCLKB (T2, receive).

The first longword transferred in any SBI/MBA operation is the command address. When transmitting a command/address or data longword, the information is loaded in the control and data transceivers and transmitted to the SBI at the appropriate time.

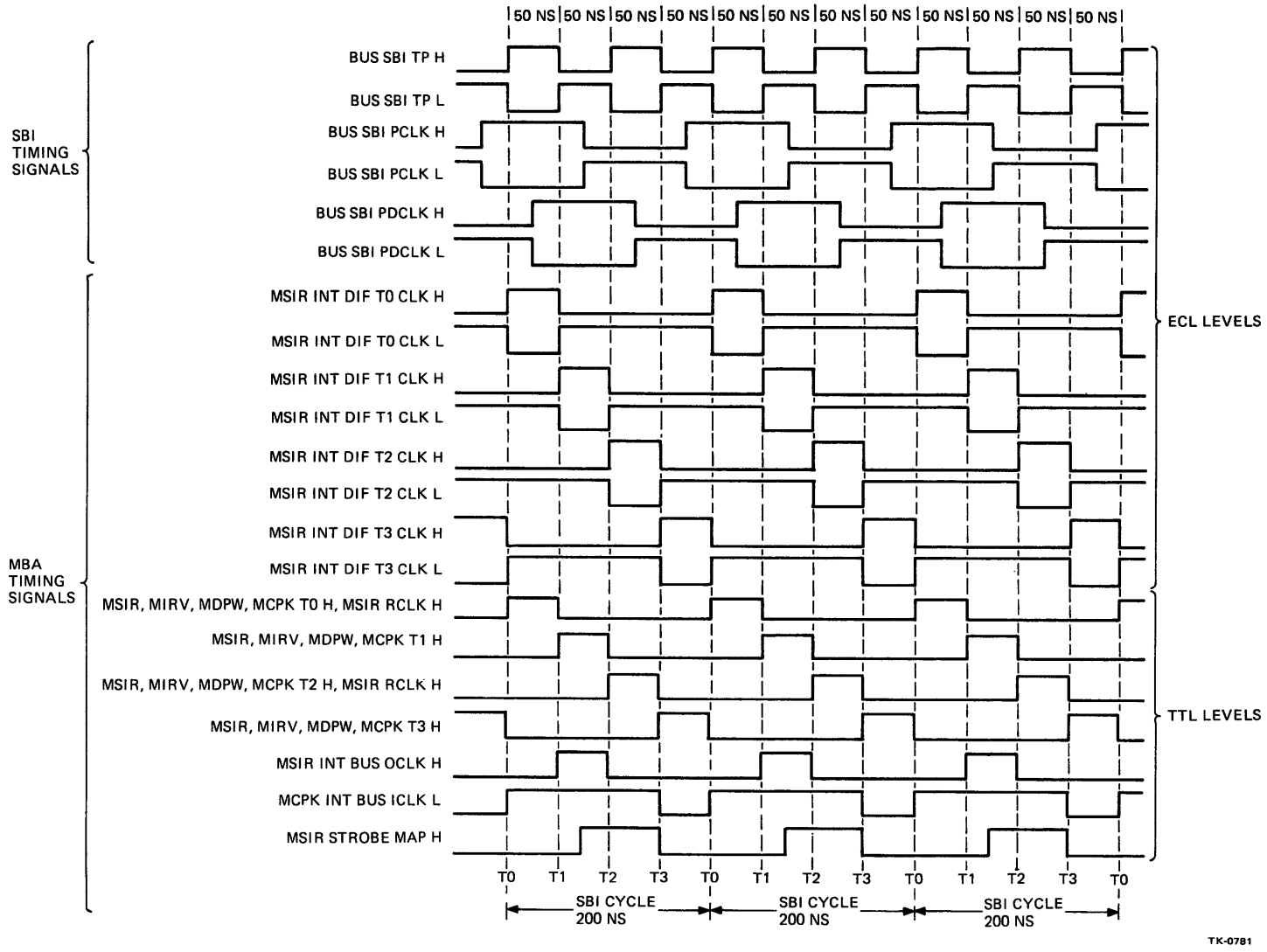


Figure 4-3 Internal Timing

When a command/address is received, it is checked for parity errors and decoded to identify the format of the information transferred, establish the function to be performed (read, write or interrupt summary read), and identify the source and destination of the data.

4.2.4 Parity Error Checking (MSIC)

The parity check provides a means of detecting single-bit errors in incoming information. Figure 4-4 shows how the parity bits are decoded. Each control field is parity checked to produce a single parity bit; each byte of the address is parity checked to produce two parity bits. These bits are configured so that the contents of the control and data fields, plus their parity bits, will always have an even number of bits (even parity). Any control field or data longword containing an odd number of bits is assumed to be in error.

If there are no parity errors, CNTRL PAR OK H and DATA PAR OK H will be generated. These signals, together with other validation signals, verify the incoming information and enable other interface circuits to continue processing the command/address or data.

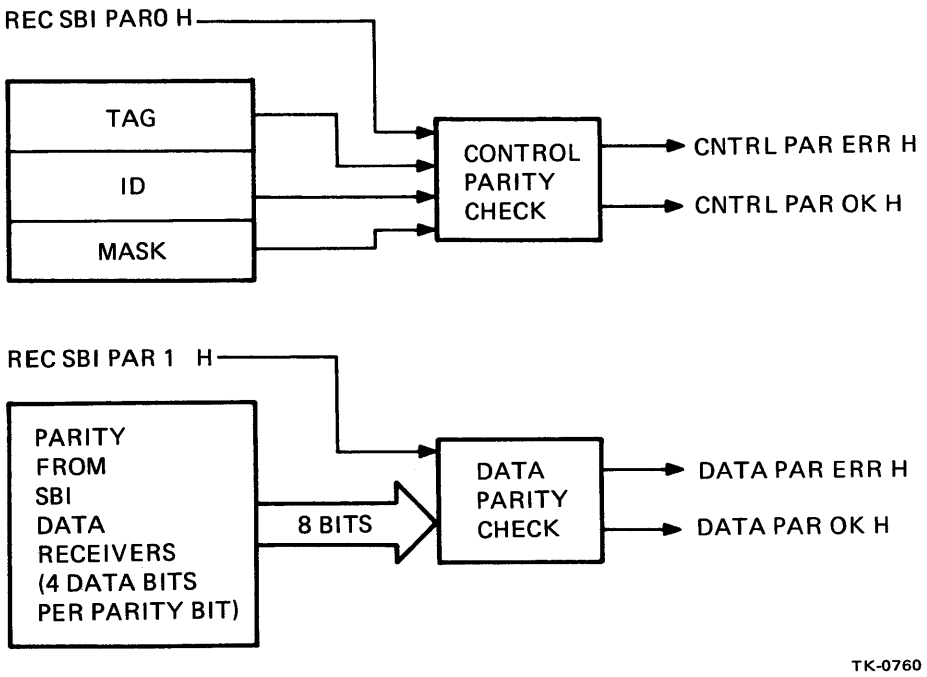


Figure 4-4 Parity Check (MSIC)

If a parity error is detected, the parity check circuit produces a fault indication CNTRL PAR ERR H or DATA PAR ERR H, depending on the source of the error. Either signal will inhibit the tag and function decoding (preventing the MBA from recognizing the command) and set the configuration/status register parity fault flag causing assertion of the fault line to the CPU. The CPU continues to assert fault until the program examines the fault and negates the condition. (Refer to Translation Buffer, Cache and SBI Control Technical Description, EK-MM78Ø-TD-ØØ1.)

4.2.5 Tag Decoding (MSID)

If no parity errors are detected, the MBA decodes the tag field.

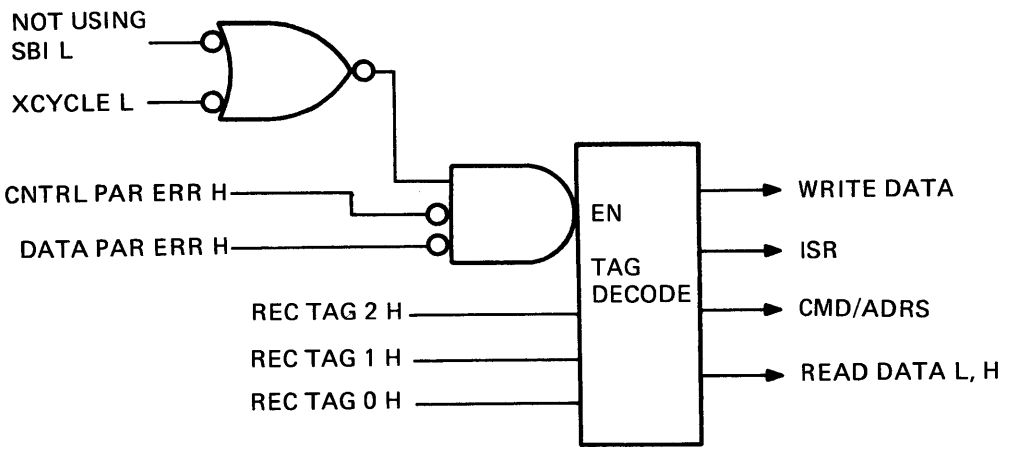
The tag field identifies the type of information transferred over the SBI and the relationship of the ID and data fields. The tag field also functions in conjunction with the mask field to define special read and write conditions.

The decoding of the tag field is accomplished by a 3-to-8 line decoder (Figure 4-5). The decoder is enabled provided an MBA transfer is not in progress (NOT USING SBI L and XCYCLE L are not asserted), and there are no parity errors. Tag bits <2:Ø> are decoded to produce one of four outputs: READ DATA, WRITE DATA, CMD/ADRS, and ISR (Interrupt Summary Read). The tag is decoded (provided a parity error did not occur) during every SBI cycle. When a command/address is received, the tag decoder generates CMD/ADRS H. This signal is sent to the confirmation circuit. Further decoding and validation checks are then performed. If the decoded tag indicates the read data, write data, or an ISR, the appropriate output will set the corresponding flip-flop preparing the MBA for execution of the first data longword during the next SBI cycle.

4.2.6 Address Validation (MSID)

Address validation determines if the address received from the SBI corresponds to that assigned to the MBA. Figure 4-6 is a simplified block diagram of the logic. As seen in this figure, address validation is accomplished by comparing address bits <14:11> with the preselected TR level (TR SELA:SELD). TR SELA:SELD corresponds to the ID code identifying the MBA and represents the arbitration level.

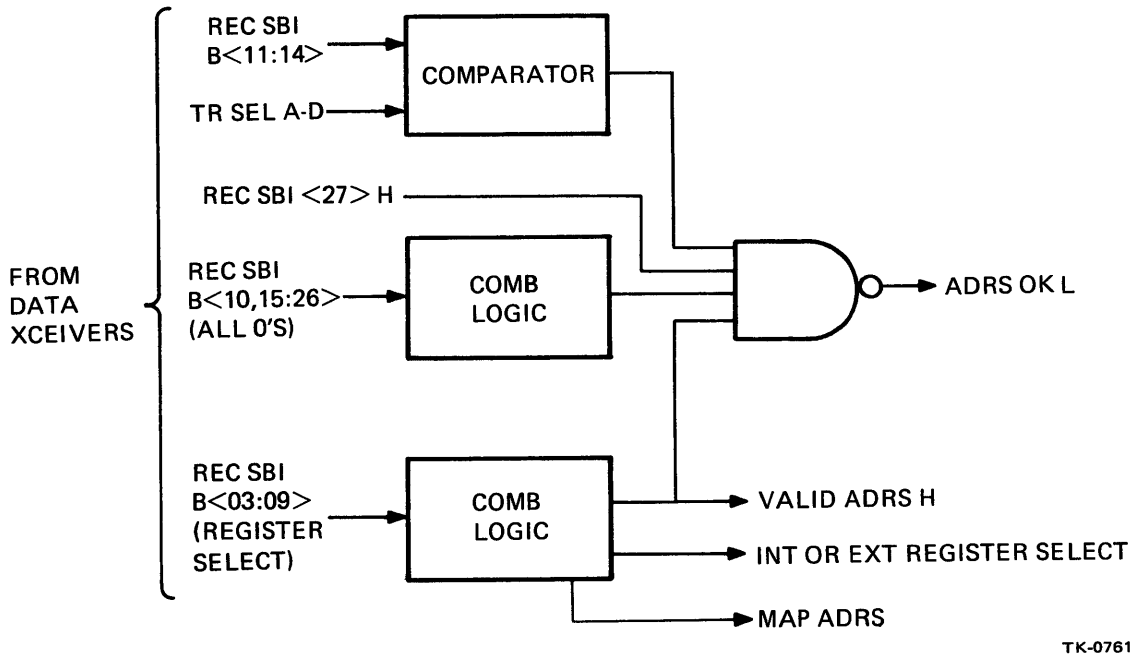
If address bits <14:11> and TR SELA:SELD do not compare (≠), the address is invalid and the MBA will ignore the command. If the address bits and the TR bits are equal, the comparator produces a single output (H). This output produces ADRS OK H when ANDed with REC SBI B 27 H, REC SBI B<1Ø,27:15>H (which must be all zeros), and VALID ADRS H. VALID ARDS H is generated by decoding the address register selection bits, REC SBI B<Ø9:Ø3>H, which select the internal or external register addressed by this command. ADRS OK H is sent to the configuration logic along with CMD/ADRS H described in Paragraph 4.2.3.



	TAG		
DATA TYPE	2	1	0
ISR	1	1	0
WRITE DATA	1	0	1
CMD/ADRS	0	1	1
READ DATA	0	0	0

TK-0757

Figure 4-5 Tag Decoding (MSID)



TK-0761

Figure 4-6 Address Validation Simplified Block Diagram

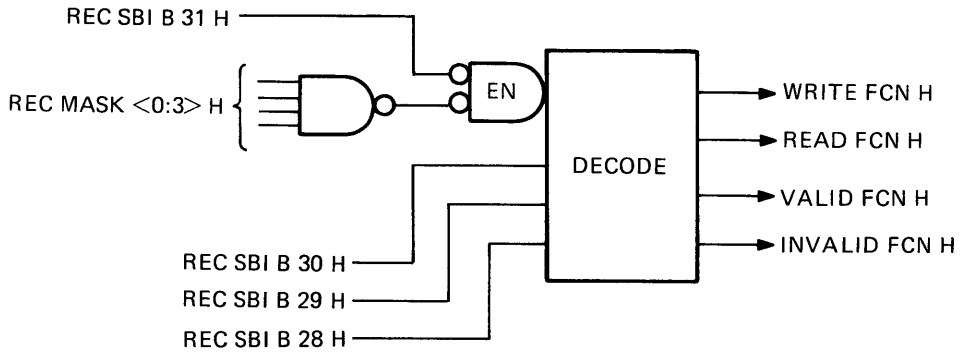
4.2.7 Function Decoding (MSID)

Since the MBA will only respond to longword reads or writes, the mask bits of the command/address from the SBI must be all 1's in order to enable the decoder that generates WRITE FCN H or READ FCN H, and VALID FCN H or INVALID FCN H. Function bits <31:28> of the command/address format specify the type of read or write command. Figure 4-7 illustrates the function decoding process.

If the mask or function bits <31:28> of a command address are not valid the decoder generates INVALID FCN H, which returns an error confirmation.

4.2.8 MBA BUSY Generation Logic (MSIE)

MBA BUSY is asserted when the MBA/SBI interface is processing an SBI command. The busy flip-flop is set upon the successful completion of the command/address decoding and validation process. If MBA BUSY is already asserted, a confirmation signal (SEND BUSY CNF H) is returned and the function is not decoded. Write to internal register functions complete immediately and do not set MBA BUSY. MBA BUSY generation logic is illustrated in Figure 4-8.



DECODED FUNCTION BITS				
	B31	B30	B29	B28
WRITE FCN	0	1	1	1
WRITE FCN	0	0	1	0
READ FCN	0	0	1	1
READ FCN	0	1	0	0

TK-0762

Figure 4-7 Functioning Decoding (MSID)

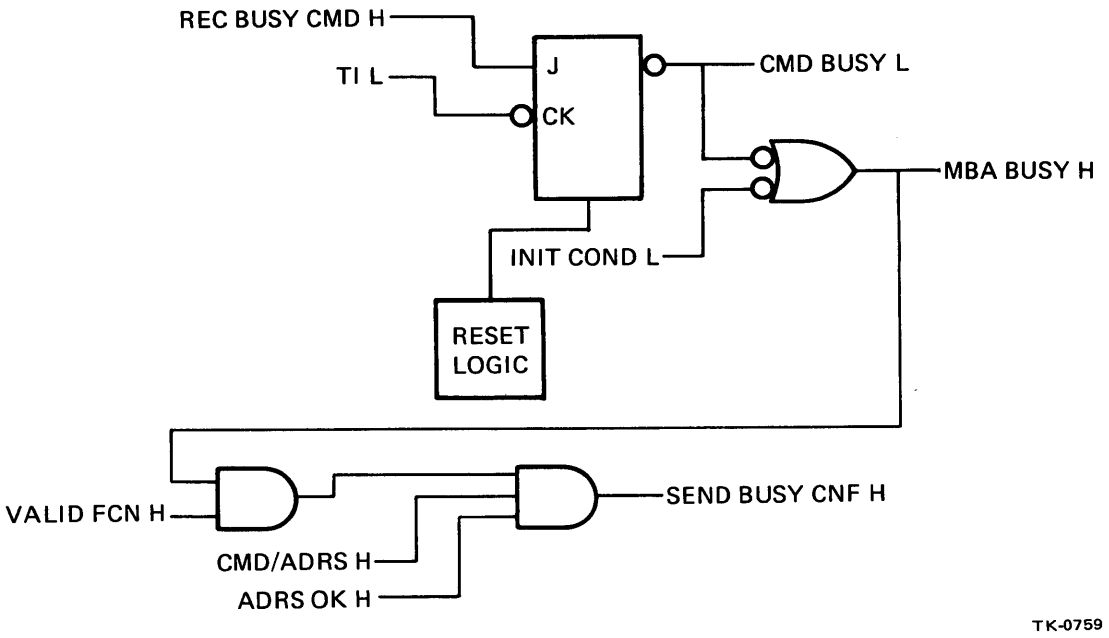


Figure 4-8 MBA BUSY Generation

When a valid command/address is decoded, ACK is sent to the SBI, followed by the assertion of MBA BUSY. When a read has occurred, MBA BUSY is cleared after the requested data is put on the SBI or if the SBI was not granted within a specific time period. When a write has occurred to a drive register, MBA BUSY is cleared 250 ns after TRA is received. If TRA is not received after a specific time period, NED is set and MBA BUSY is cleared.

MBA BUSY is not asserted when there is a parity error and we are expecting write data (EXP WP H and PAR ERR H asserted) or a write data error (WD ERR H) occurs. WDERR H is asserted when we are expecting write data, but do not receive it. SEND BUSY CNF H is asserted when MBA BUSY is asserted and the received command/address has indicated a valid function is to be performed.

4.2.9 Confirmation Logic (MSIJ, MSIM)

Confirmation logic in the MBA informs the CPU whether or not an information transfer has been received correctly and, in the case of a received command/address, if the MBA can process the command. CNFØ and CNF1 are encoded or decoded (depending upon whether the function is a read or write) to specify one of four responses:

CNFØ	CNF1	Response
L	L	No Response (N/R)
L	H	Acknowledge (ACK)
H	L	Busy (BUSY)
H	H	Error (ERR)

ACK is the anticipated response to a successful data transfer. ERR will cause the data transfer to be aborted, and N/R or BUSY will cause a retry of the data transfer.

When the MBA is the receiving nexus it generates the response. When the data transfer is from a Massbus storage device to memory, memory generates the response. During a data transfer, synchronization of the confirmation codes with the command/address or data word is accomplished by the cycle flip-flop.

4.2.9.1 Response Generation (MSIM) -- When the MBA receives data (read or write) it generates a response to the transmitting nexus. Acknowledge will be generated (SEND ACK H asserted) when information is received correctly. To acknowledge a received command/address there must be no errors (CMD/ADRS H, ADRS OK H asserted), the MBA cannot be processing another command (MBA BUSY cleared), and the command address must specify a valid function (VALID FCN H asserted). Acknowledgment of write data occurs when the function specified was a write and was successful (received write data and CS WRITE FCN H and WRITE DATA H are asserted). The MBA will acknowledge read data if the read data was received correctly (ID OK H, READ DATA H, and READ DATA PEND H are asserted). When the received information is not valid, an error confirmation will be sent to the transmitting nexus. SEND ERR CNF H will be asserted when the command/address specifies an invalid function. XMIT MBA FAULT will also be asserted if there is an error in the write data (WD ERR H), if the MBA receives unexpected read data (UNEXPECTED RD H), if a parity error occurs (PAR ERR H), or if the transmited and received IDs are not equal (XMIT/REC ID EQ L not asserted). A BUSY confirmation will be generated when MBA BUSY is asserted indicating that the MBA is processing a data transfer command. Figure 4-9 illustrates the logic used in response generation.

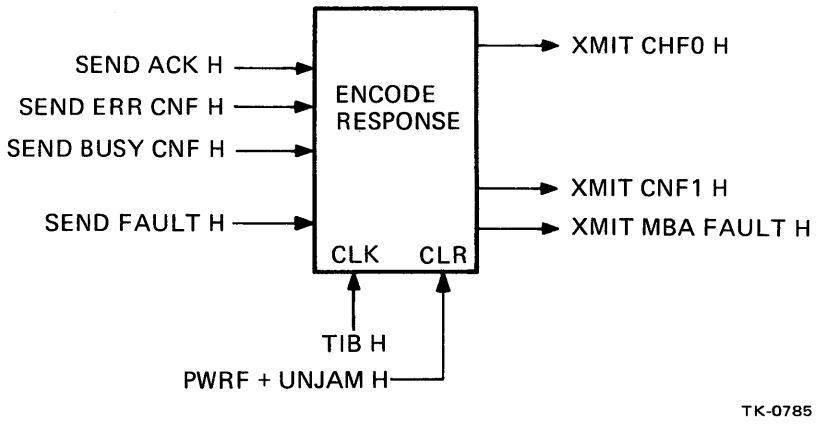


Figure 4-9 Response Generation (MSIM)

4.2.9.2 Confirmation Check (MSIJ) -- When memory generates a response to the MBA for received data, the MBA must be able to interpret the response in order to react to the transfer. If the data transfer has occurred successfully, WD2 ACK H will be asserted. If an error in received data was detected by memory, SET ERROR CNF L will be asserted. The MBA will retry to transfer (SET RETRY L) data if any one of the following conditions exist:

a. N/R received from memory
b. No ACK received for the first longword
c. Memory is BUSY
d. No ACK received for the second longword.

The MBA will retry until a timeout occurs or data late is received from memory.

CLR C/A XCYC L is asserted to clear transfer status within the MBA. This will occur when a data transfer retry occurs or an error occurs in the data transfer. One of the following conditions will assert CLR C/A XCYC L:

a. ERR occurs during the transfer
b. Memory is BUSY
c. N/R received from memory
d. No ACK received for the first longword.

SET N/R CNF L indicates that no response to the data memory requested has been received by the MBA. SET N/R CNFL is asserted when any of the following conditions exists:

a. No ACK received for the first longword
b. CPU did not ACK requested data
c. N/R received from memory
d. No ACK received for the second longword.

REQ COMPLETE L will be asserted when the data transfer requested is complete or when a nonrecoverable error occurs. Figure 4-1Ø is a simplified diagram of the logic that performs the confirmation check.

4.2.1Ø Interface Fault Assertion

Faults that occur during the SBI decoding and validation process, such as parity and unexpected read data, set a corresponding interface fault flip-flop. This output sets the appropriate bit in the configuration/status register and asserts the SBI fault line. The fault line will remain asserted until the program examines the fault status bits and negates the fault. Paragraph 3.6.1 provides an explanation of the various faults.

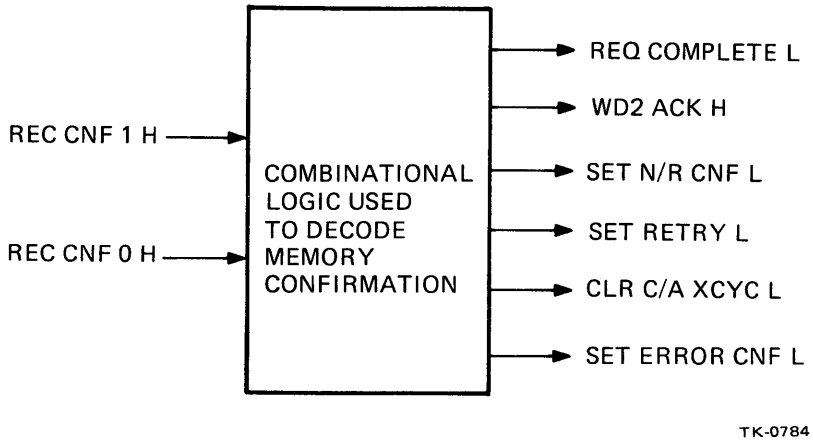


Figure 4-1Ø Confirmation Check

4.2.11 Request Logic (MSIK)

When requesting control of the SBI to respond to an internal read, external read, or to transfer data the request flip-flop produces CURRENT REQ H (indicating that an SBI operation is to occur), which sets the TR send flip-flop. Its output, SEND TR H, is applied to internal register arbitration logic, which causes the assertion of the MBA's assigned TR level at TØ of the following SBI cycle. If the MBA's arbitration line represents the highest priority, the internal register arbitration logic issues ARB OK, allowing the MBA to assume control of the SBI. The combination of SEND TR, ARB OK, and the function requested may initiate many functions such as setting the read data pending flip-flop, starting the timeout operation, and control and transfer functions within the internal registers, control paths, and data paths. Figure 4-11 is a simplified diagram of the logic used to accomplish this.

4.2.12 Timeout Logic (MSIH)

Interface sequence timeouts and read data timeouts on the MBA are accomplished by the logic shown in Figure 4-12. The counter consists of two 4-bit counters in cascade.

An interface sequence timeout (SET IS TIMEOUT L) can occur when the MBA is arbitrating for control of the SBI. The counter will wait up to 512 SBI cycles after an attempt to get the bus (SEND TR H) before the timeout occurs. A read data timeout can occur only during a read data transfer. If the read data is not received within 512 SBI cycles after memory has accepted the read command (RDPD1 H), SET RD TIMEOUT L is asserted.

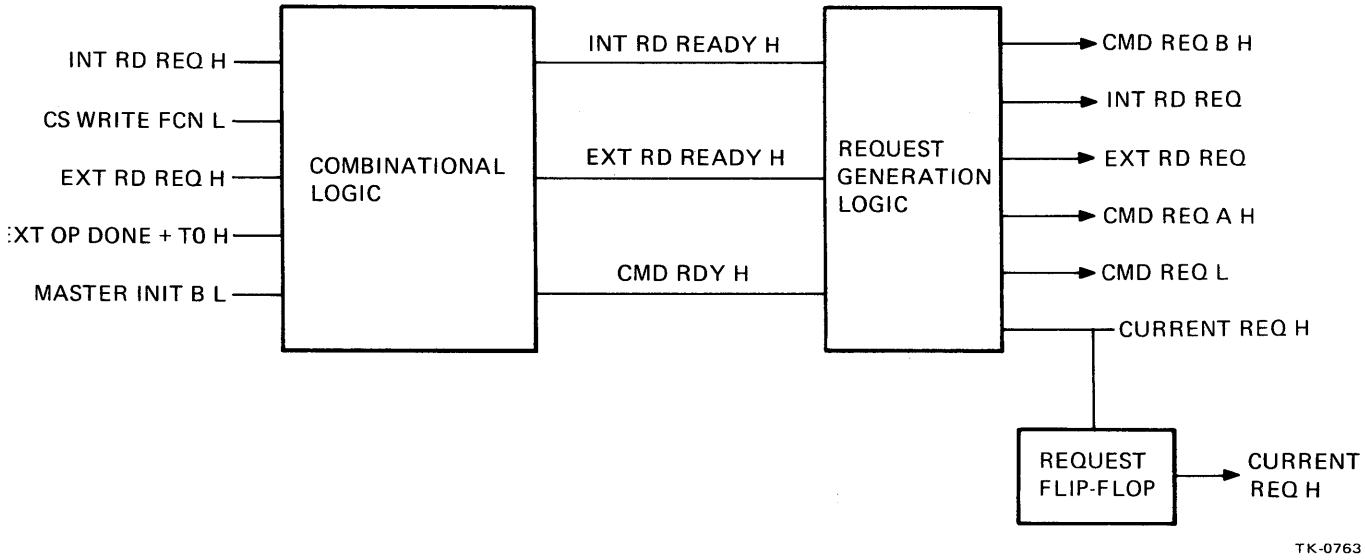


Figure 4-11 Request Logic (MSIK)

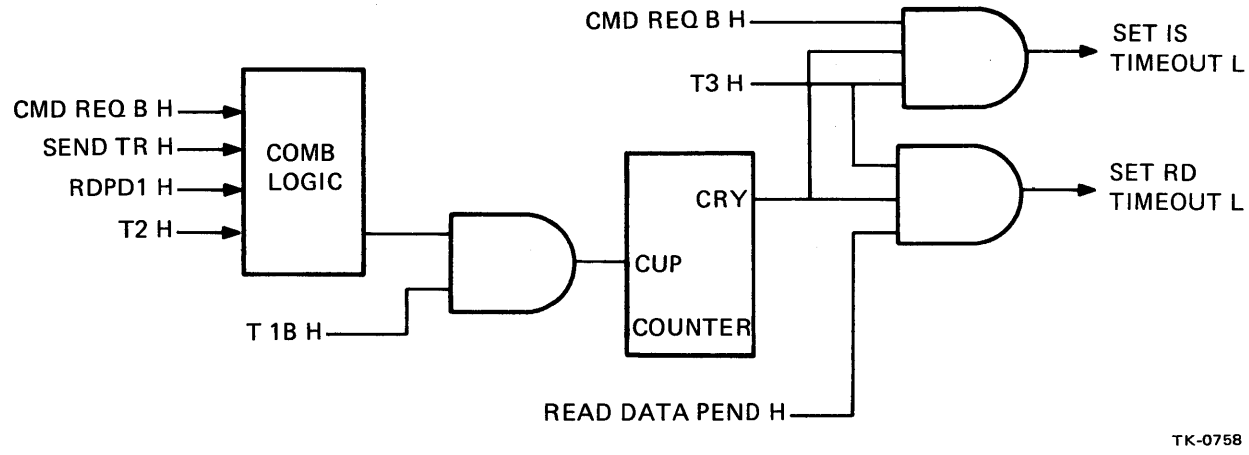


Figure 4-12 Timeout Logic (MSIH)

4.2.13 Command/Address Generation (MSIM)

The command/address format that consists of a parity, tag ID, function, and address fields is used to process data transfers from the MBA to a receiving nexus (memory) via the SBI.

During the command/address, the mask field is encoded to specify which bytes of data (read or write) are valid. Figure 4-13 is the logic associated with the mask bits.

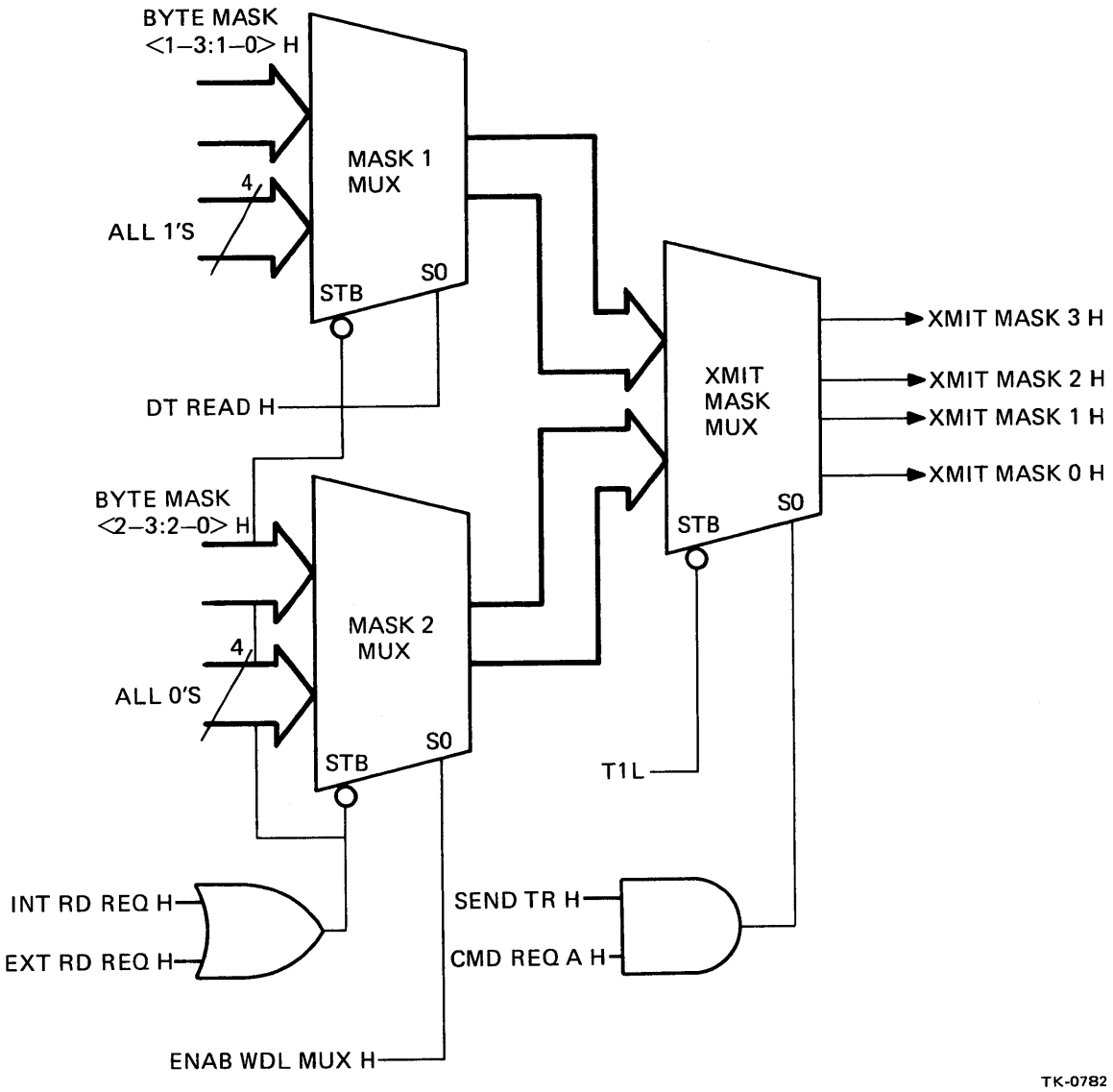
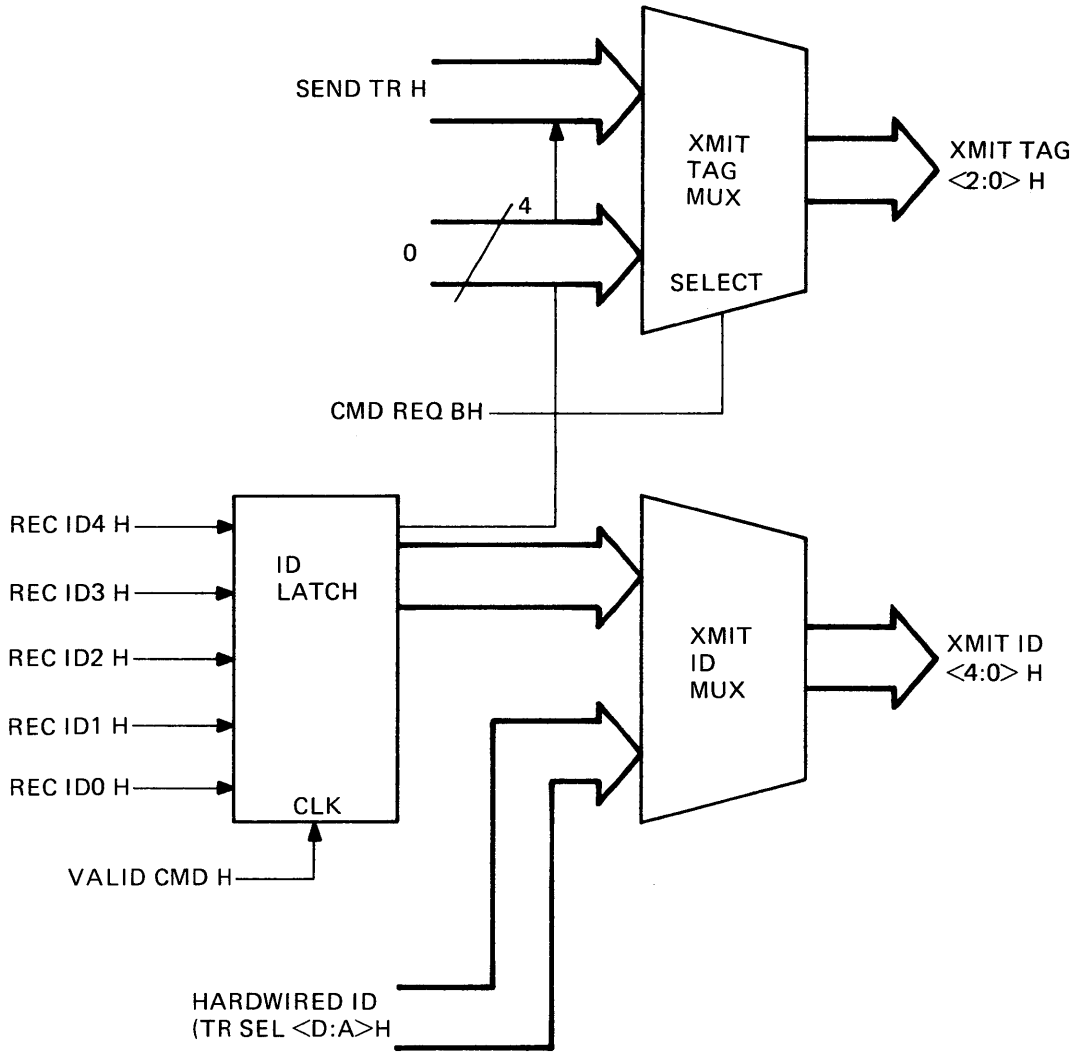


Figure 4-13 Mask Generation (MSIM)

When INT RD REQ L or EXT RD REQ L is asserted it indicates a read of an internal or external MBA register. This will cause the mask bits to be all zeros. During a data transfer, where the MBA will send the read command to memory (indicating a write to device DT READ H cleared; SEND TR H and CMD REQ A H asserted), the mask bits will be all 1's. During a write to memory indicating a read from device, mask bits BYTE MASK 1<3:Ø>H will be selected. These mask bits indicate which bytes in the first transferred quadword have valid data. In this case DT READ H, SEND TR H, and CMD REQ A H will be asserted. ENAB WDI MUX H indicates transfer of the first data quadword. When this is asserted, BYTE MASK 2<3:Ø>H is selected, which indicates which bytes of the second quadword are valid. The mask bits of the second quadword will be all zeros. The mask bits are latched through multiplexer at T1 and applied to the MBA/SBI control transceivers.

The ID field is the identifier for the MBA. The logic associated with the ID field is illustrated in Figure 4-14. The ID field can be derived from either of two sources, depending on the operation to be performed. The first source is the hardwired MBA ID (TR SEL A-D H), which is representative of the MBA priority levels assigned at system build time. This is always available at the ID select multiplexer input. The hardwired ID is used by the MBA to talk to memory. The second ID source is the received ID from the SBI/MBA control transceivers. This is loaded into the saved ID latch every time the MBA receives a command/address from the SBI and is applied to the ID select multiplexer. The saved ID is used to return data to the CPU. ID selection is determined by the state of CMD REQ. When requesting memory data, CMD REQ is high and the hardwired ID (TR SEL A-D H) is transmitted, via the SBI/MBA transceivers. When the MBA is excepting read data from memory, the received ID is compared with that of the MBA. If the received ID and the MBA ID compare, the read data has arrived, assuming that all other decoding and validation checks are performed satisfactorily. The MBA also monitors the ID on the SBI while it is asserting data on the SBI. If the received ID and MBA ID are not equal, the ID equal comparator issues a fault indication setting the appropriate configuration/status register fault bit and asserting the SBI fault line. The fault is asserted only if the MBA is putting data on the SBI and the ID on the SBI is not the same as the one the MBA asserted. As in other fault situations, the fault can be cleared by INIT or deassertion of the fault signal.

The tag field is determined by the state of SEND TR H and select input CMD REQ B. When TR H is high and CMD REQ is low (the MBA is responding to a CPU read), the encoded output of the tag multiplexer represents a read data format.



SEND TR	CMD REQ	OPERATION
0	0	NONE
0	H	READ DATA
H	0	WRITE DATA
H	H	COMMAND/ADDRESS

Figure 4-14 Tag and ID Generation (MSIM)

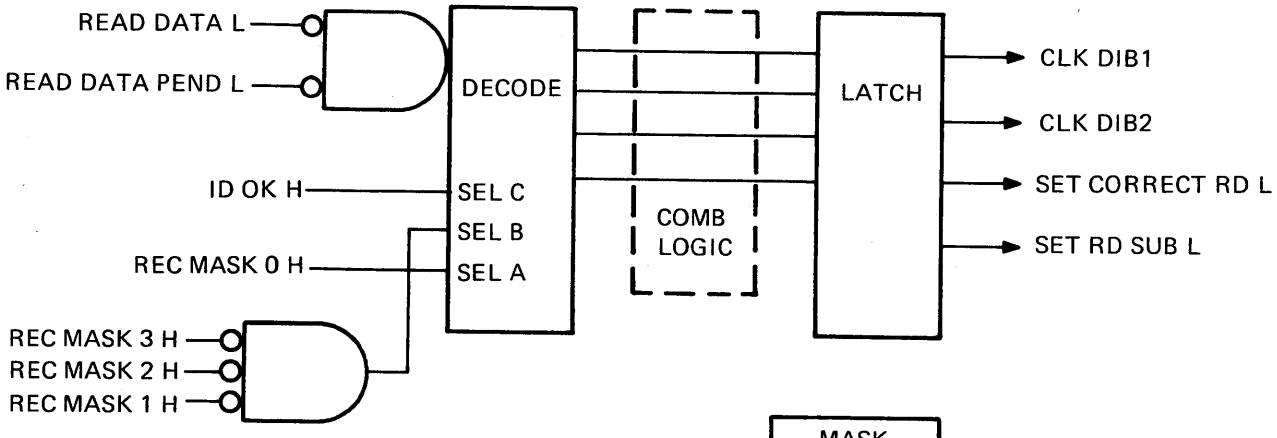
Conversely (during a data transfer), the tag field represents a command/address when SEND TR and CMD REQ are both high. When SEND TR H is low and CMD REQ is high, the tag field represents a write data format. The tag field remains latched in the tag multiplexer until receipt of T1. At this time the tag field is available to be loaded into the control transceivers for subsequent SBI transmission. The logic associated with the tag field is shown in Figure 4-14.

4.2.14 Mask Decoding

As seen in Figure 4-15, when the SBI TAG indicates read data and the ID is that of the MBA, the mask field received from the SBI specifies the type of read data the MBA has received. There are three data types: Read Data Substitute, Corrected Read Data, and Normal Read Data. RD SUBSTITUTE is data in which an uncorrectable error has occurred. CORRECTED DATA is data in which an error has occurred but has been corrected. NORMAL DATA is data in which there has been no error.

4.2.15 Internal Bus and Interrupt Summary Read Logic

INT BUS 0 CLK H latches data from the internal bus into the MBA/SBI interface receivers. This occurs independently of the decoding and validation process described previously. Multiplexers select the data to be transferred to the SBI, either internal bus data or Interrupt Summary Response data (ISR RESP<15:00>H). Data is transmitted to the SBI only if T/R ENAB A L is asserted for the internal bus data or T/R ENAB B L is asserted for the ISR data. Figure 4-16 is a simplified diagram of the logic associated with the functions.



	MASK			
READ DATA TYPE	3	2	1	0
RD SUBSTITUTE	X	X	1	X
CORRECTED DATA	0	0	0	1
NORMAL DATA	0	0	0	0

TK-0764

Figure 4-15 Mask Decoding

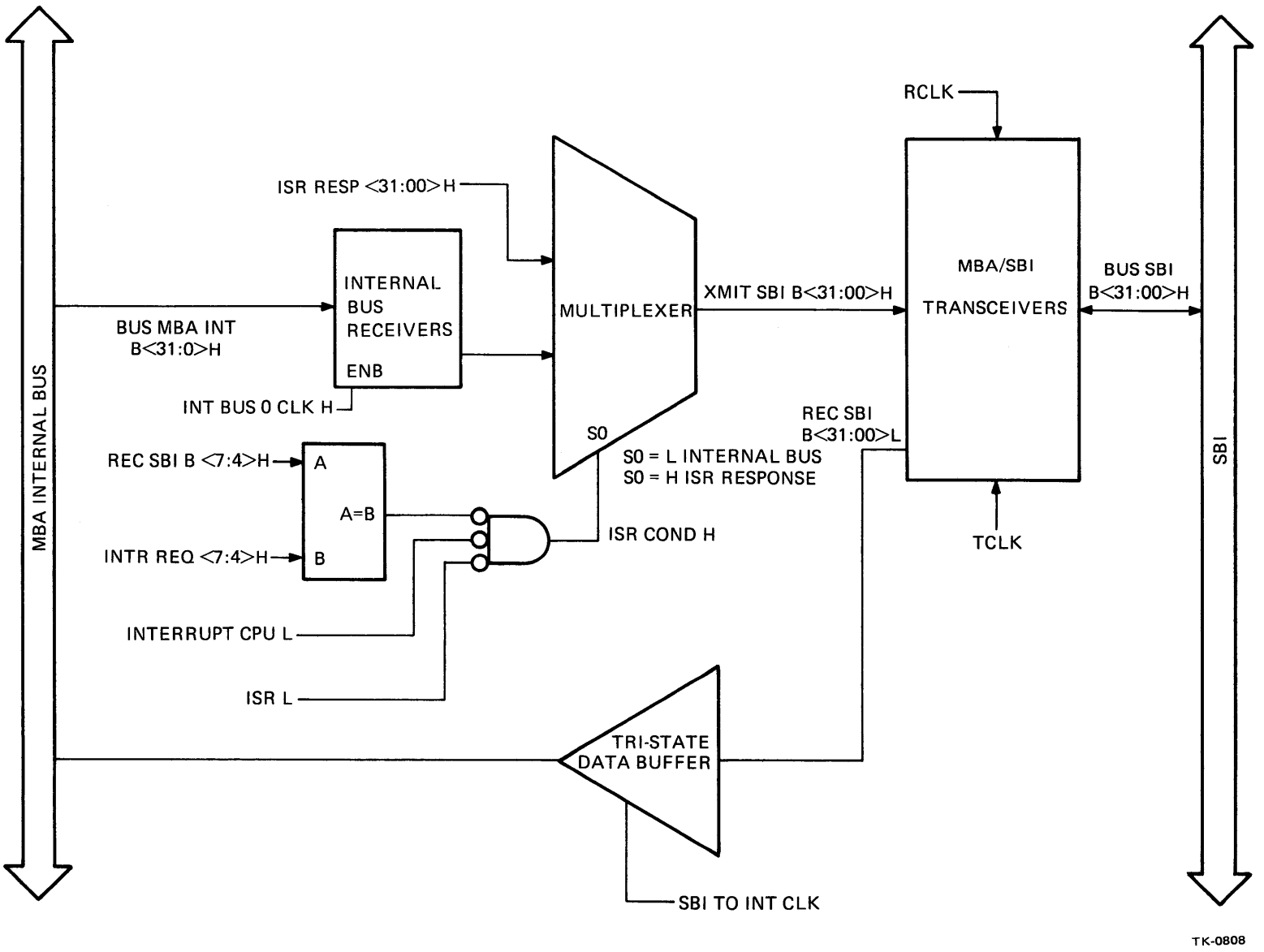


Figure 4-16 Internal Bus Drive and Interrupt Summary Read Data Logic

4.3 MBA INTERNAL REGISTERS

The MBA internal registers store various control and status information. A description of each register and their function is provided in Paragraphs 3.7 through 3.7.9 of this manual. A functional logic description is provided in the subsequent paragraphs. Figure 4-17 is a detailed block diagram of the MBA internal registers.

4.3.1 Internal Bus Receivers

Information received from the SBI is processed by the interface logic and latched into the internal bus receivers on TØ following their assertion on the SBI by INT BUS ICLK L. Figure 4-18 illustrates the internal bus receivers.

4.3.2 Internal Register Control

Register control logic decodes the command/address to select one of the internal registers to perform the specified read or write function. The logic used to accomplish this is shown in Figure 4-19.

Selection of the control path output registers, MAP registers, or one of the other six internal registers is accomplished by decoding bits CS B<Ø9:Ø8> to set a corresponding flip-flop. When an internal register operation is selected, bits CS B<ØØ:Ø2> are applied to decoding logic that selects the appropriate register.

The output of the decoding logic is ANDed with CLK IR and T2 enabling the appropriate register clock.

It should be noted that only the control register and diagnostic register (if enabled) can be read or written during the time the MBA is processing a data transfer. If an attempt is made to write any register other than these during a data transfer operation, the PGE (programming error) bit in the status register will be set. If an attempt to write the MAP register is made, the MAP register will not be modified and the data transfer in progress will continue unaffected.

4.3.3 Configuration, Control, Status, Diagnostic, and Command/Address Registers

The logic associated with the configuration, control, status, and diagnostic registers is not very complex. Each bit, which is implemented within one of these registers, is associated with its own flip-flop. The control logic for these registers is composed of simple gating networks and is self-explanatory upon examination of the print sets. Figure 4-17 is a functional block diagram of these registers. The logic and operation of the virtual address, byte counter, and MAP registers are not as obvious and are described in the following paragraphs.

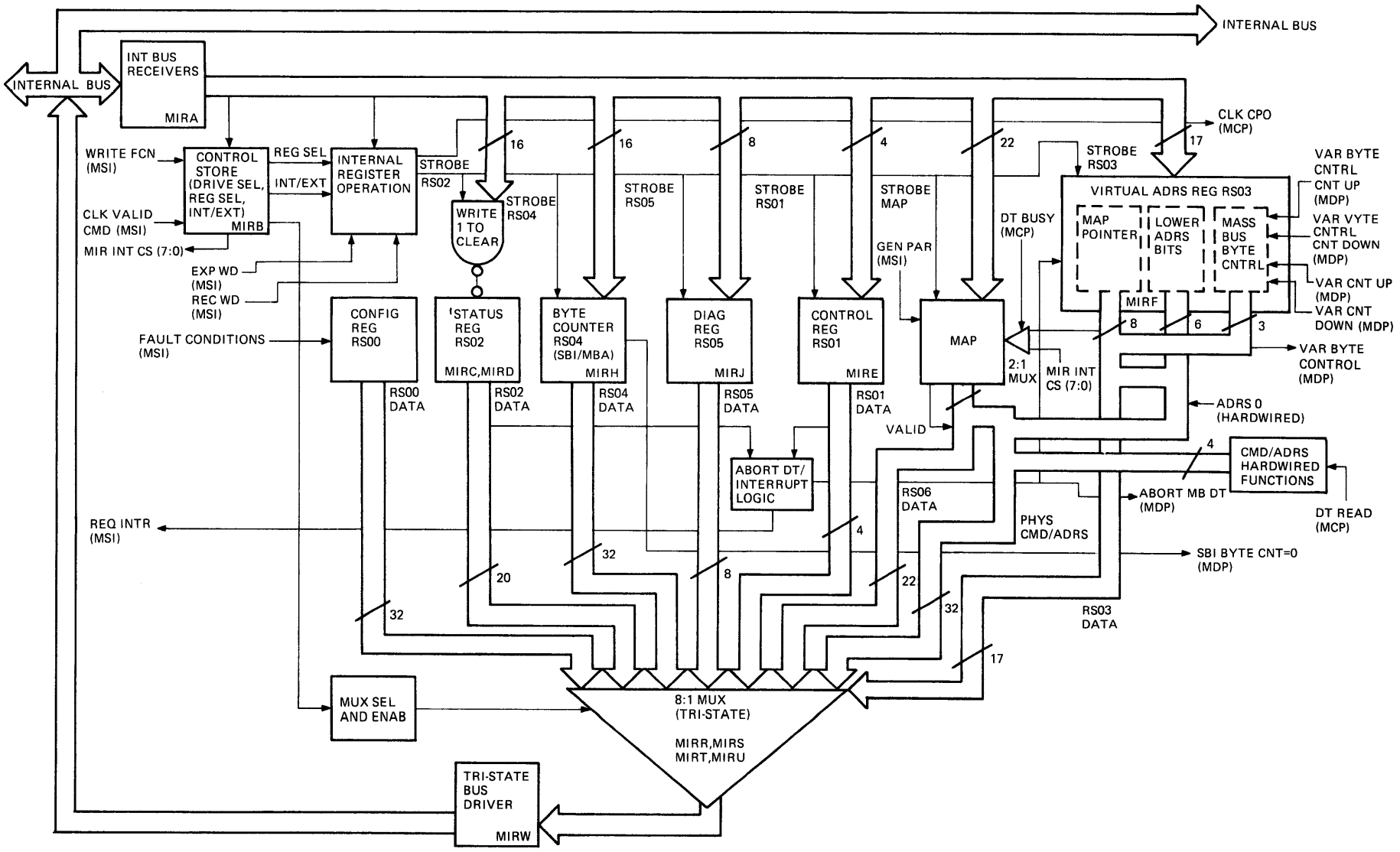


Figure 4-17 MBA Internal Registers

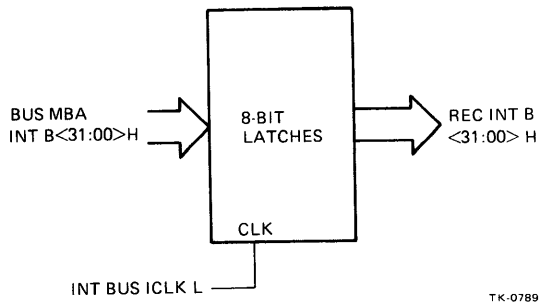


Figure 4-18 Internal
Bus Receivers

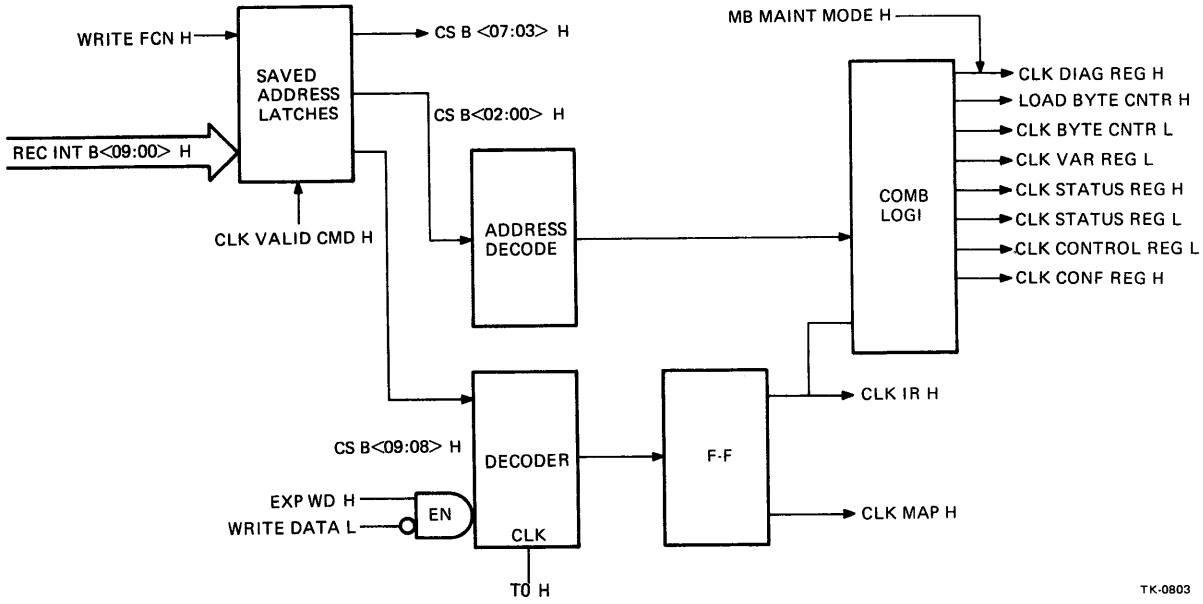


Figure 4-19 Internal
Register Control

4.3.3.1 Virtual Address Register -- The virtual address register (VAR) (Figure 4-2Ø) consists of five synchronous, up/down, 4-bit binary counters. Its function is to convert the virtual address, received from the SBI upon initiation of a data transfer operation, to a physical address pointing to the location of the first byte of data to be transferred.

Bits VAR BIT<16:Ø9>H are the map pointer bits that select the page frame of one of 256 map locations. The map pointer bits are incremented by one each time the physical byte address crosses a page boundary during a data transfer.

Bits VAR BIT<Ø8:Ø3> are the physical byte address bits specifying the location of the quadword address within the page. These bits are incremented each time VAR CNT H is received from the Massbus data paths indicating that a successful quadword data transfer to or from memory operation has been completed. Bits VAR<Ø2:ØØ>H are used to pack and unpack the quadword into bytes.

Note that the map pointer bits are applied to two data select multiplexers. The multiplexers select either VAR BIT<16:Ø8>H or CS B<Ø7:ØØ>H.

The purpose of the multiplexer is to provide a means of selecting the source of map register information, as required, to support the transfer operation to be performed. When processing a data transfer, the data busy flip-flop is set and the map pointer bits from the virtual address register are selected by the multiplexers to be sent to the maps. When the data transfer busy flip-flop is not set, the register select bits are sent to the map, allowing them to be read or written directly.

4.3.3.2 MAP Registers -- The 256 MAP registers map virtual page addresses from the virtual address register into SBI physical page addresses. This allows for data transfer to and from contiguous virtual (noncontiguous physical) memory locations. Virtual address translation (mapping) is described in Paragraph 1.5.1 of this manual.

A write to MAP register operation is performed when the write enable (WE) input is low. CS B<Ø9:Ø8>H, when asserted, produce CLK MAP H (MIRB). If the MBA is not busy processing a data transfer (DT BUSY L cleared), STROBE MAP H and CLK MAP H generate the two MAP register enabling signals, MAP WE1L and MAP WE2L. The data REC INT B<31:ØØ>H at the DI input of the RAM is loaded into the location specified by the address bits RAM ADRS<Ø7:ØØ> from the virtual address register. The parity bit for this data is also stored in the MAP registers. This information will be placed on the internal bus if selected by the internal bus output multiplexers. The MAP register can be read when MAP WE1L and MAP WE2L are not asserted. MAP registers are read to check the validity of the information stored in them. Figure 4-21 illustrates the MAP register operation.

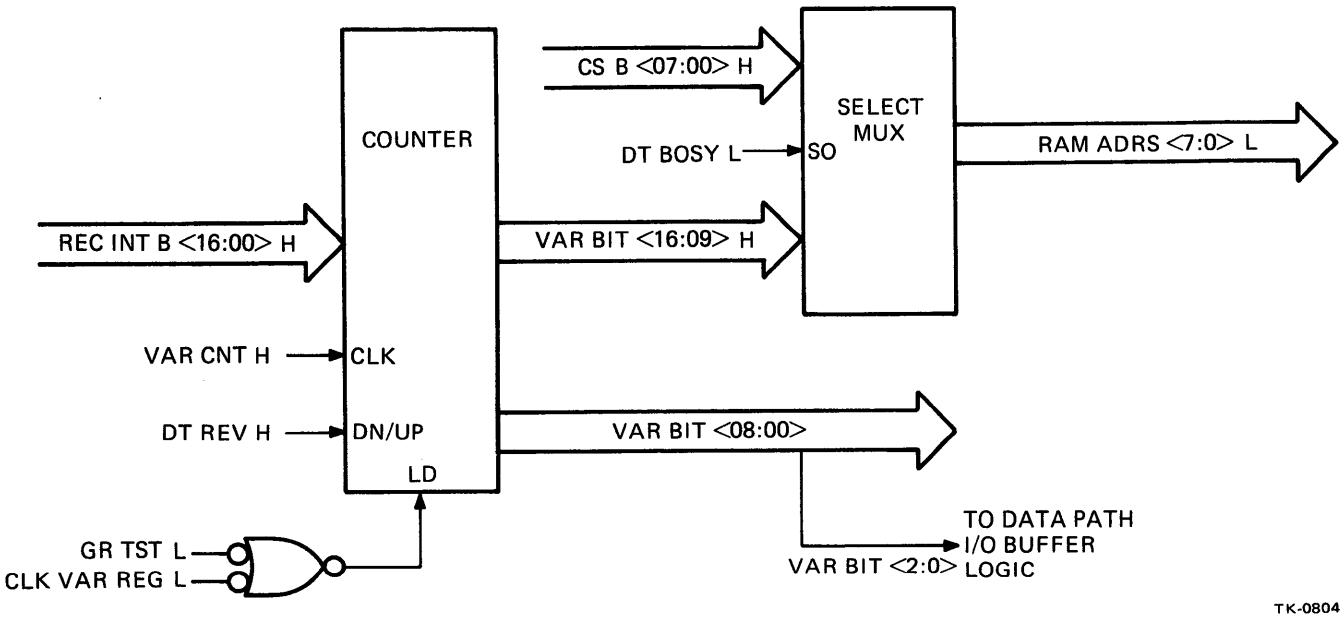


Figure 4-20 Virtual Address Register

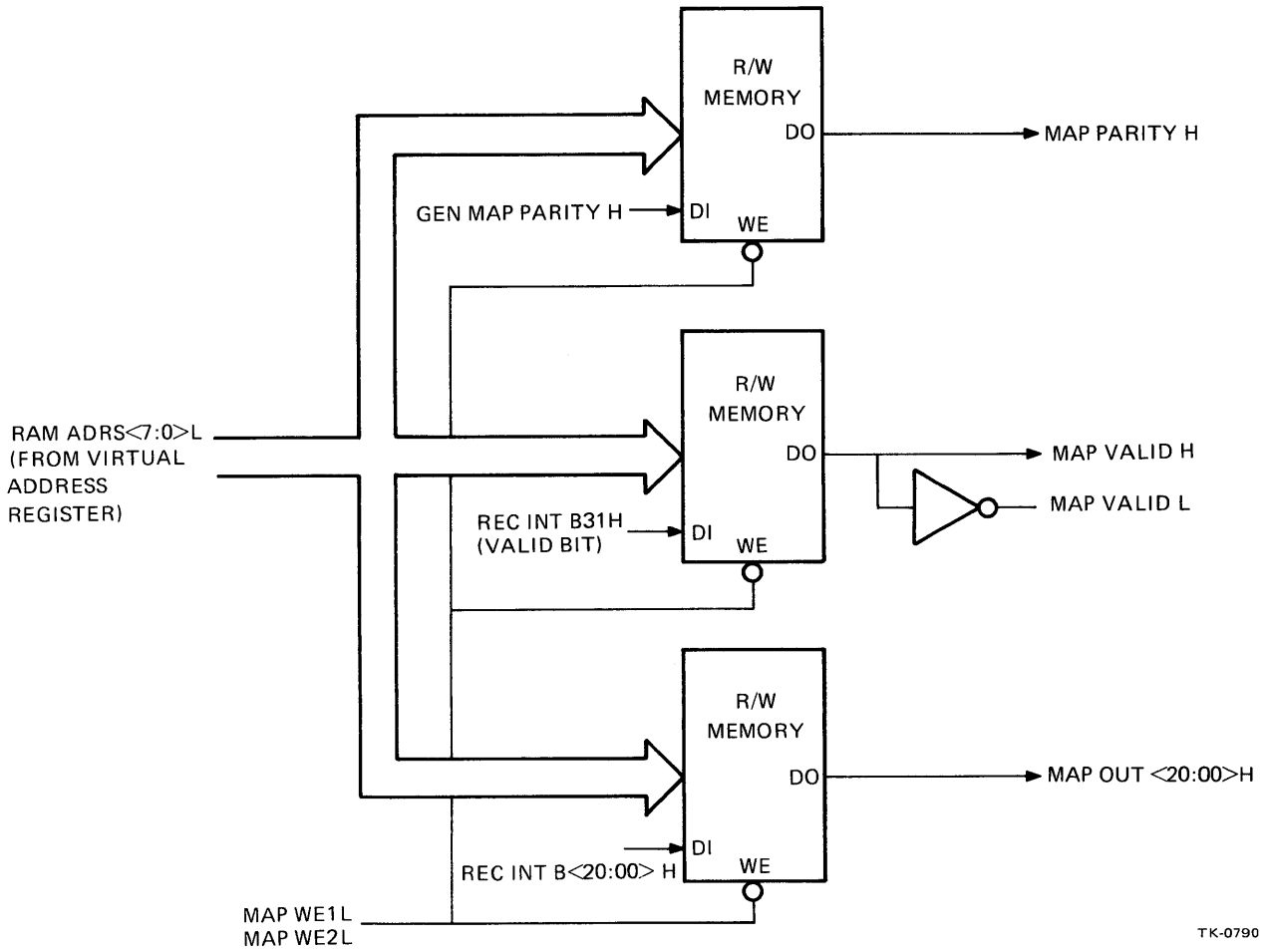


Figure 4-21 MAP Registers

4.3.3.3 Command/Address Generation -- CMD REQ L indicates that the MBA is to perform an SBI data transfer operation (read or write to memory). When this signal is asserted, OUT MUX SEL<Ø2:ØØ>H are asserted (MIRP). These multiplexer select lines indicate that the D7 input of the internal bus output multiplexers will be selected. DT READ H, MAP OUT<2Ø:ØØ>H, and VAR BIT<Ø8:ØØ>H are at the D7 input of the multiplexers. The internal bus output multiplexers generate MBA INT B<31:ØØ>L. A command/address must be generated by the MBA to memory in response to a data transfer command. If the data transfer is a read, the MBA must generate an extended write command to memory. If the data transfer is not a read (i.e., write or write check), the MBA must generate an extended read command to memory. The format of the command address is shown in Figure 4-22.

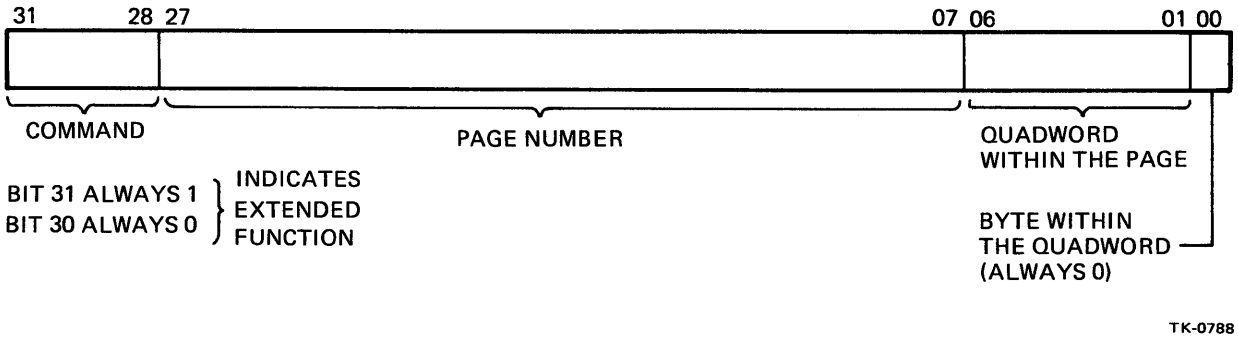


Figure 4-22 Command/Address Format

When DT READ H indicating a data transfer read is asserted, MBA INT B<29:28>L are not asserted (1). When DT READ H is not asserted indicating a data transfer write or write check, MBA INT B<29:28>L are asserted (Ø).

The page number specified by MBA INT B<27:Ø7>L is generated from the MBA MAP register bits MAP OUT<2Ø:ØØ>H (MIRK,L,M,N). The address of the quadword within the page specified by MBA INT B<Ø6:Ø1>L is generated from the virtual address register bits VAR BIT<Ø8:Ø3>H (MIRF). Only quadwords can be transferred; therefore, bit MBA INT BØØ L must always be zero for a command/address generated by the MBA.

4.3.3.4 Byte Counter Register -- The byte counter register (BCR), shown in Figure 4-23, is a 32-bit register consisting of eight 4-bit synchronous binary counters. The byte count register maintains a record of the number of bytes of data to be transferred to and from the SBI and the Massbus.

The byte counter register is divided into two 16-bit byte counters. Bits BYTE CNT<31:16>H are used as the Massbus byte counter and bits BYTE CNT<15:ØØ>H are used as the SBI byte counter.

The SBI byte counter maintains a record of the number of bytes to be received from the SBI. When performing a write (write check) to Massbus device, the SBI byte counter is incremented each time a byte of data is loaded into the silo from the data input buffer. When the last byte of data is transferred from the SBI to the silo, the SBI counter carry output sets the SBI BYTE CNTR=Ø L flip-flop and no further data is loaded.

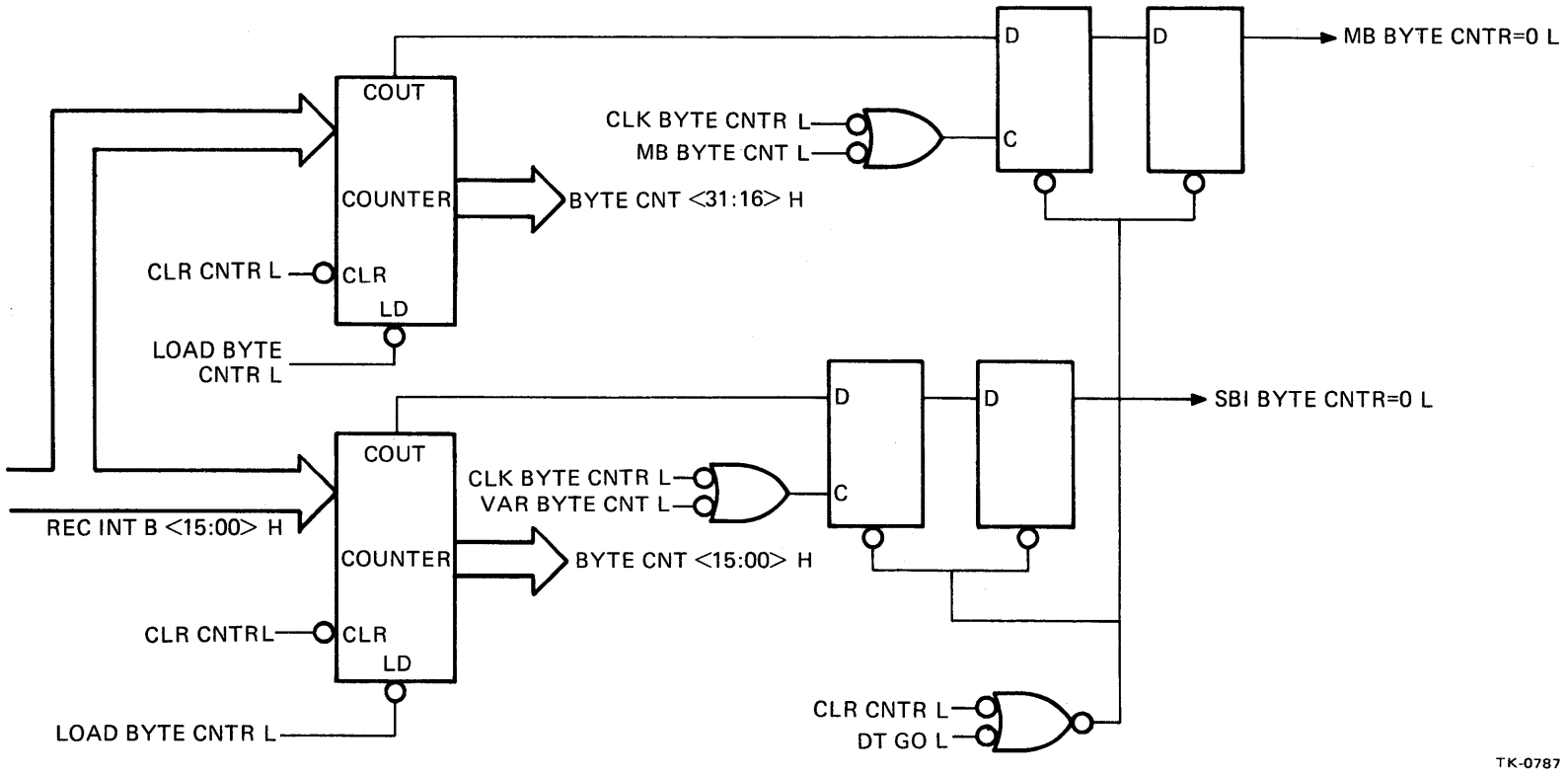


Figure 4-23 Byte Counter Register

Each time it is written the SBI byte counter is loaded automatically into the Massbus byte counter. As each word of data is transferred to the Massbus device, the Massbus byte counter is decremented twice until the counter equals Ø, indicating that the last byte of data in the silo has been transferred to the Massbus device. At this time the carry output from the Massbus counter sets the Massbus counter = Ø flip-flop, RUN is negated, and the transfer will be completed when EBL is asserted.

When performing a read from Massbus device, the operation of the byte counter is reversed. The Massbus byte counter maintains a record of the number of bytes to be received from the Massbus device and loaded into the silo, and the SBI byte counter maintains a record of the number of bytes transferred from the silo to the data output buffers. The Massbus byte counter goes to zero as data is transferred into the silo. When the data is transferred to the SBI, the SBI byte counter goes to zero. When EBL is received and the SBI byte counter is zero, the data transfer is complete. To read the byte counter register, CS BØ9 L and CS BØ2 H must be asserted and CS B<Ø1:ØØ>H must be clear.

4.3.4 Output Data Multiplexers -- The output multiplexer consists of 32 8-way multiplexers connected in parallel so that one bit of each of the internal register outputs or control signals is applied to each of the multiplexers (Figure 4-24). Selection of the register or control outputs to be asserted on the internal bus is accomplished by ANDing control store selection bits CS<Ø9,Ø2:ØØ>H with INT RED REQ (internal read), EXT RD (external read), or CMD REQ (command request), depending on the function to be performed, to produce OUT MUX SEL<Ø2:ØØ>H. The output combinations and corresponding data source selected are provided in Table 4-1.

Once data from the output multiplexers is selected, the internal bus drivers are enabled by ANDing the internal read or data transfer command request input with SEND TR. Data is transferred to the internal bus upon receipt of INT BUS O CLK.

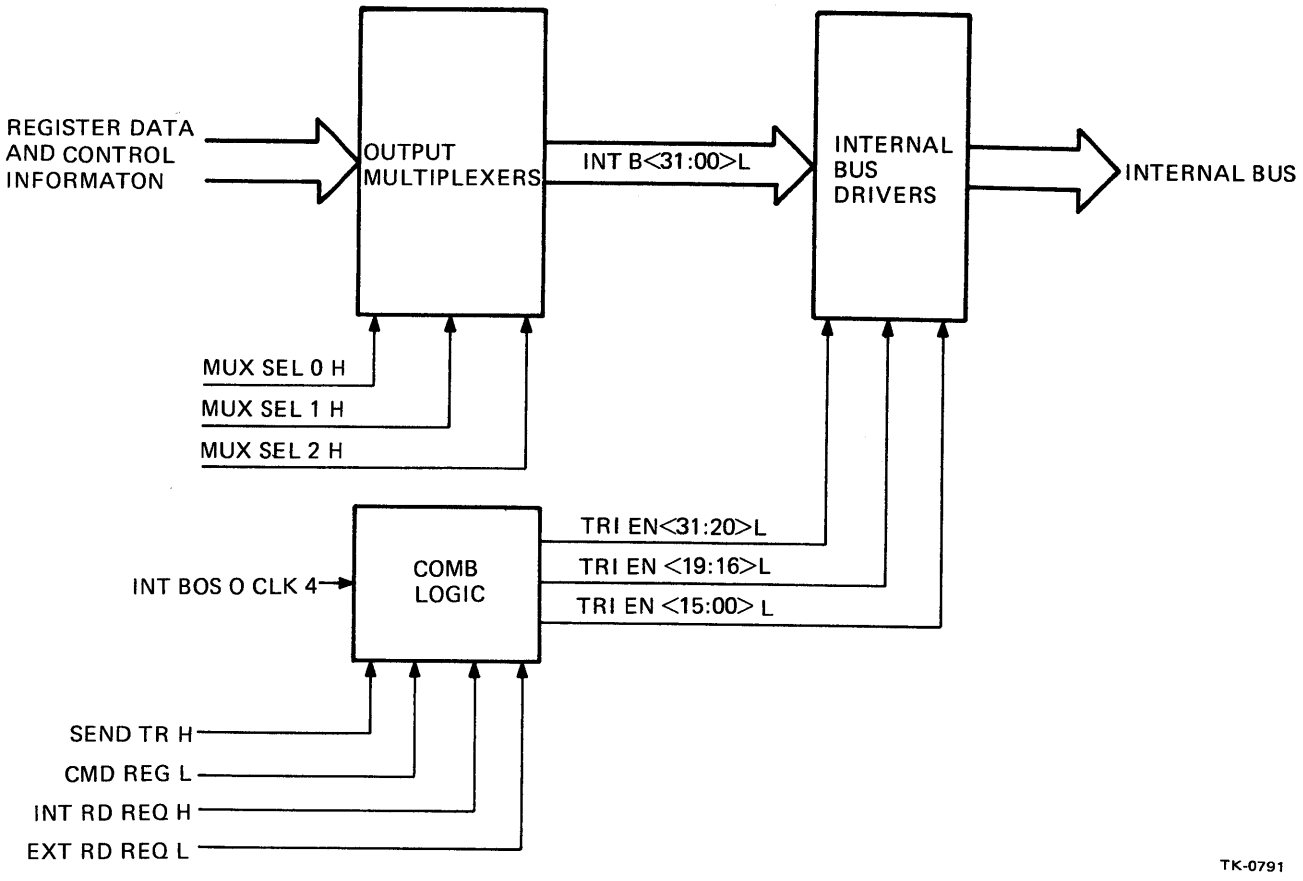


Figure 4-24 Output Multiplexers and Select/Enable Logic

Table 4-1 Internal Bus Output Multiplexer

CMD REQ L	EXT RD REQ L	INT RD REQ H	CS BØ9 L	CS 2	Bits 1	Bits Ø	MUX SEL 2	MUX SEL 1	MUX SEL Ø	Data Source Register
L	X	X	X	X	X	X	H	H	H	Command/Address
H	H	H	L	X	X	X	H	H	L	MAP
H	H	H	H	H	L	H	H	L	H	Diagnostic
H	H	H	H	H	L	L	H	L	L	Byte Counter
H	H	H	H	L	H	H	L	H	H	Virtual Address
H	H	H	H	L	H	L	L	H	L	Status
H	H	H	H	L	L	H	L	L	H	Control
H	H	H	H	L	L	L	L	L	L	Configuration
H	L	L	H	X	X	X	L	H	L	Status*

X = Don't care.
* = Only upper 16 bits enabled, lower 16 bits from Massbus control path.

4.4 MBA DATA PATHS

The MBA data paths transfer data between the SBI and the Massbus device. The data paths contain the silo that smooths out the transfers between the 32-data bit SBI and the 16-data bit Massbus. Figure 4-25 is a block diagram of the data paths.

4.4.1 Command Condition (MDPA)

COMMAND CONDITION H is asserted when the MBA data paths want to use the SBI to read from or write data to memory. This will occur during a data transfer to or from a mass storage device. For a read from device or a write (write check) to device, the following conditions must exist:

a. MBA BUSY H asserted, the MBA is processing the data transfer;

b. BLOCK SEND CMD L cleared, enabling the MBA to send a command to memory;

c. CMD REQ L cleared, is not yet processing this command.

Other conditions must also exist for read from device or write (write check) to device. These conditions are described in the following paragraphs.

Read from Device

CMD CONDITION H will be asserted if any one of the following conditions exist:

a. FULL FWD, the data transfer is a read (DT READ H) in the forward direction (DT FWD H) and DOB7 is full (BYTE MASK 2-3 H);

b. FULL REV, the data is a read (DT READ H) in the reverse direction (DT REV H) and DOBØ is full (BYTE MASK 1-Ø H);

c. DT END FWD, the data transfer is a read (DT READ H) in the forward direction. DT FWD H filled some of the data output buffers but the SBI byte counter is now Ø (SBI BYTE CNTR L);

d. DT END REV, the data transfer is a read (DT READ H) in the reverse direction. DT REV H filled some number of data output buffers but the SBI byte counter is now Ø (SBI BYTE CNTR L).

If a Massbus exception or parity error occurs (MB EXC + MDPE H), SILO Ø READY L is not asserted, indicating that the MBA will empty its silo to transfer all data prior to the error.

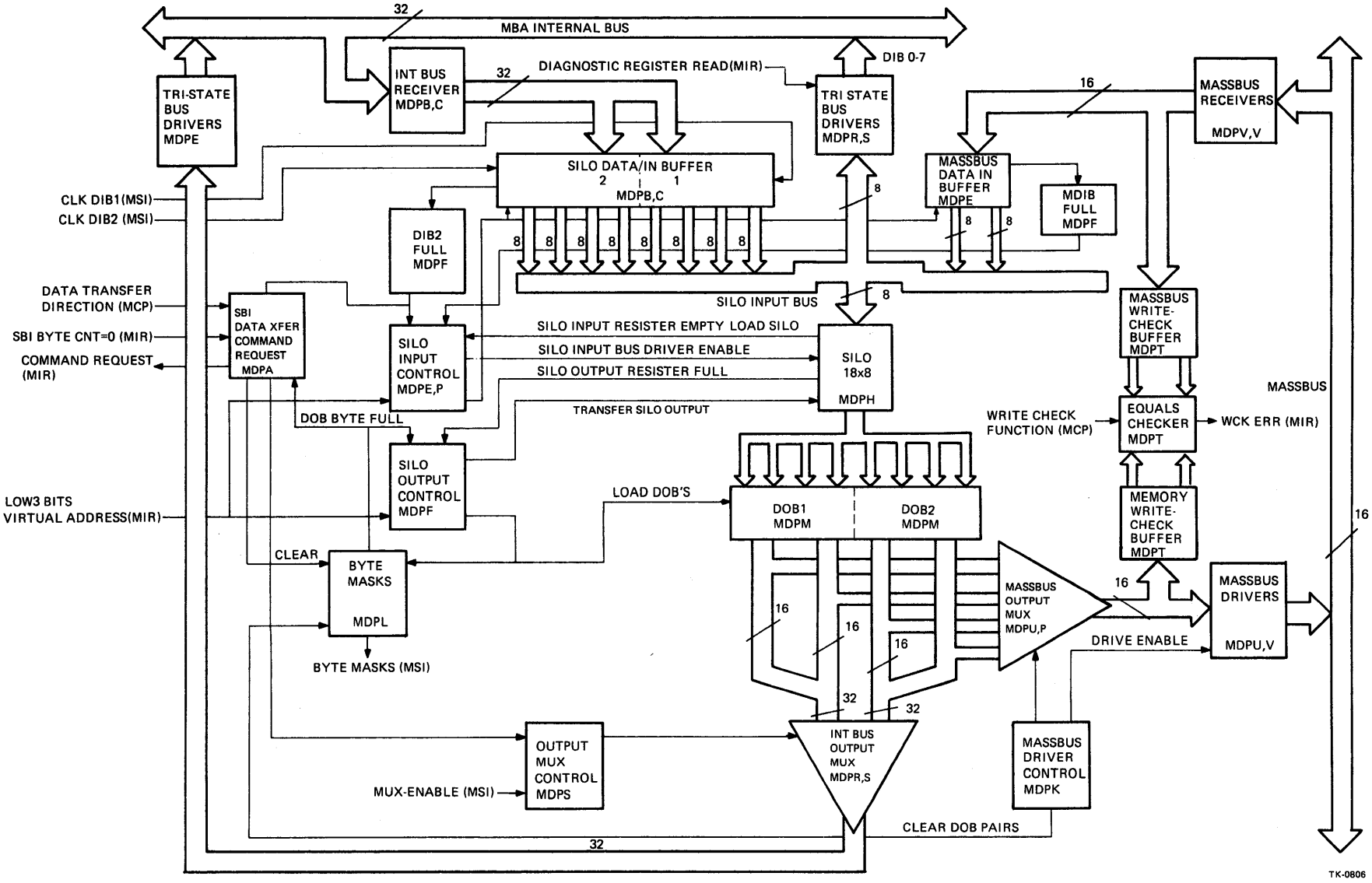


Figure 4-25 MBA Data Paths

Write (Write Check) to Device
CMD CONDITION H will be asserted if the following conditions exist:

a. DT READ L cleared, the data transfer is not a read from device;

b. DIB2 FULL L cleared, the second data input buffer is not full;

c. READ DATA PEND L cleared, the MBA is not waiting for data from memory;

d. SBI BYTE CNT=Ø L cleared, the MBA has more data to be transferred.

4.4.2 Internal Bus Receivers/Buffers

Data from the internal bus is loaded into four 8-bit latches upon receipt of INT BUS ICLK from the control path clock circuits.

The MBA input buffer consists of two sets of 32-bit latches. Their function is to control the manner in which the two 32-bit data words from memory are loaded into the silo for subsequent transfer to the Massbus. CLK DIB1 latches the first SBI data word into the first data input buffer (DIB1). CLK DIB2 latches the second data word into the second data input buffer (DIB2). CLK DIB2 H also increments a 4-bit counter.

The counter is initialized to four and counts down at CLK DIB2 H to zero, asserting DT WRITE READY indicating that all data path buffers are full, and the MBA is ready to begin a write/write check. Figure 4-26 is a logic diagram of the internal bus receivers and buffers.

4.4.3 Data Input Buffer Enable (MDPD)

The output of the data input buffer select is used to enable the eight data input buffers (Table 4-2). Virtual address register bits VAR BIT<Ø2:ØØ>H point to the data buffer where the data is stored. The virtual address register is incremented by VAR BYTE CNT L (MDPA). When the function is a write (write check) and the fill silo operation is initiated (LOAD SILO H), the data input buffers are enabled sequentially onto the silo data input bus. The data input buffers are disabled at T1 while VAR BITS <2:Ø> are changing. Figure 4-27 is a diagram of the data input buffer enable logic.

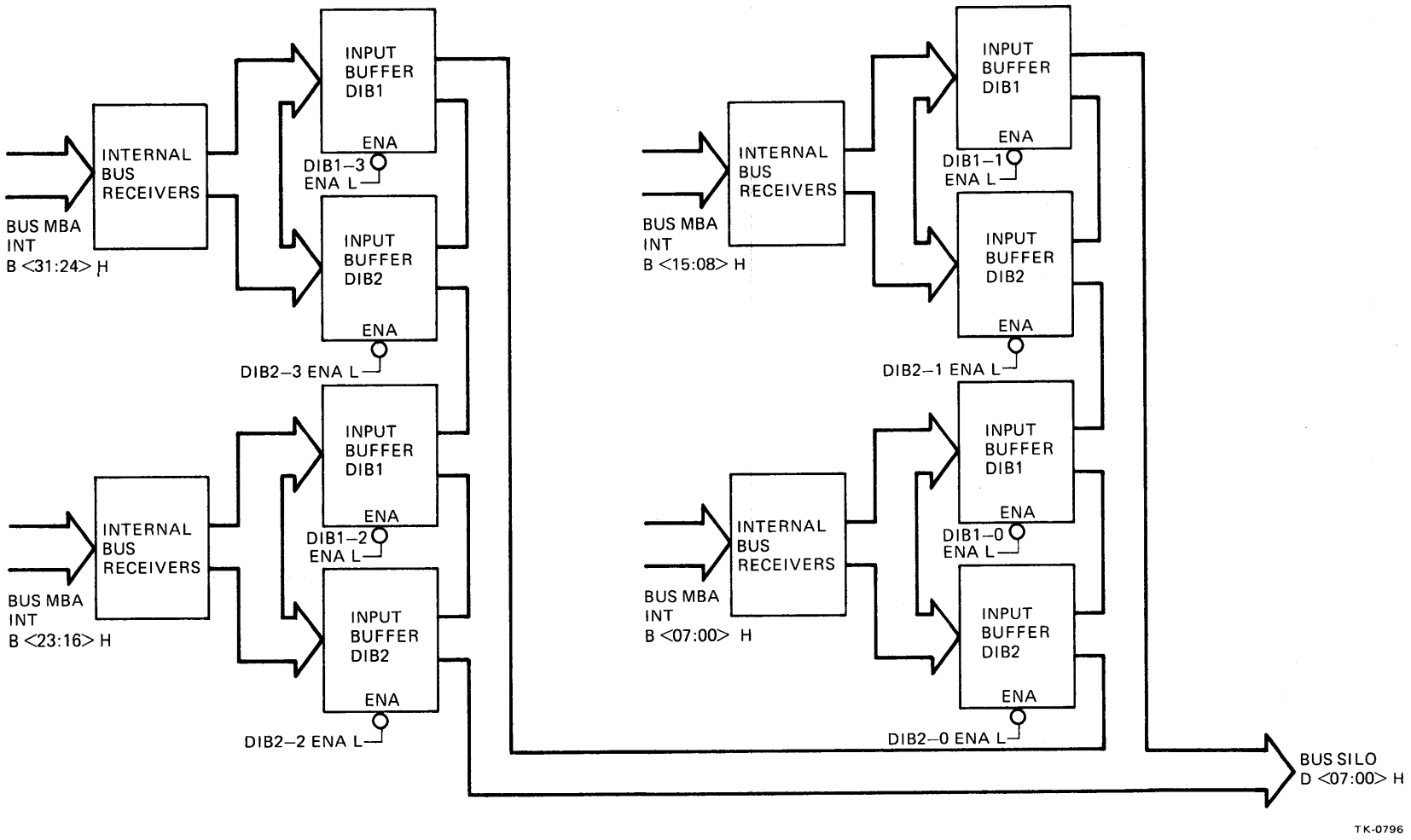


Figure 4-26 Internal Bus Receivers/Buffers

Table 4-2 Data Input Buffer Enable

VAR BIT Ø2 H	VAR BIT Ø1 H	VAR BIT ØØ H	DT READ H	T1 L	Input Buffer Enabled
*	*	*	*	L	none
*	*	*	H	*	none
L	L	L	L	H	DIB 1-Ø ENA L
L	L	H	L	H	DIB 1-1 ENA L
L	H	L	L	H	DIB 1-2 ENA L
L	H	H	L	H	DIB 1-3 ENA L
H	L	L	L	H	DIB 2-Ø ENA L
H	L	H	L	H	DIB 2-1 ENA L
H	H	L	L	H	DIB 2-2 ENA L
H	H	H	L	H	DIB 2-3 ENA L

NOTE

VAR BIT <Ø2:ØØ> H are decoded after they have been clocked through a latch by SET SBI SEL H.

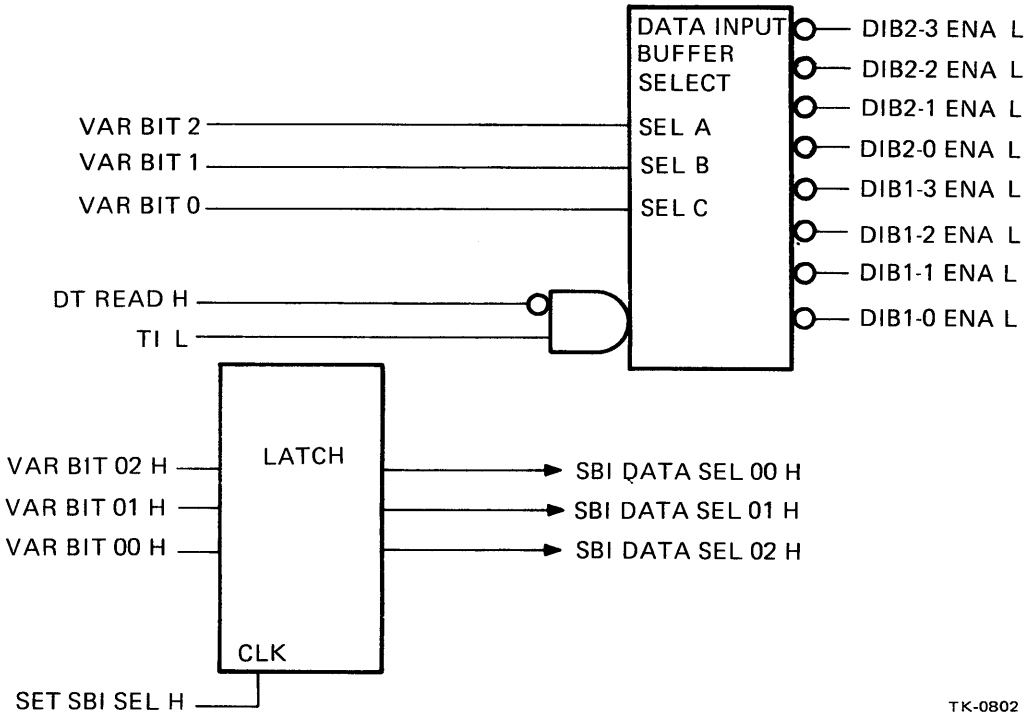


Figure 4-27 Data Input Buffer Enable

4.4.4 Silo and Control Logic (MDPF, MDPH)

The MBA silo is 16-bytes deep by 1-byte wide. Its function is to provide a source of interim storage for SBI or Massbus data, permitting regulation of the data transfer rate. Figure 4-28 is a simplified diagram of the silo logic.

When transferring data from the SBI to the Massbus device, the input buffer and enable circuits divide the two 32-bit data words into 8-bit bytes. LOAD SILO H loads data into the silo. The data will be loaded in at TØ if all the following conditions exist prior to TØ:

a. SBI BYTE CNTR = Ø L cleared, the SBI byte count register must indicate more bytes to be transferred;

b. DIB2 FULL H asserted, second data input buffer must be full;

c. DT READ L cleared, the data transfer must be a write (write check);

d. SILO FULL L cleared, the silo cannot be full.

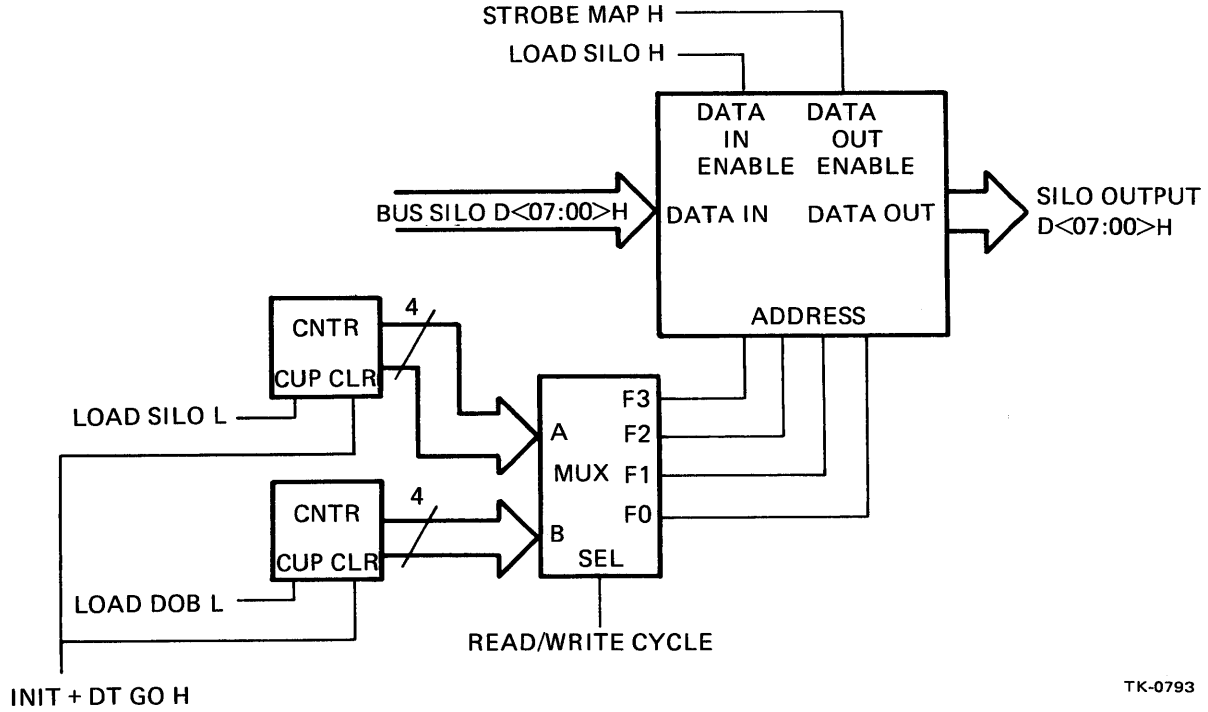


Figure 4-28 MBA Silo

As mentioned previously, the byte counter maintains a record of the number of bytes to be received from the SBI and the Massbus byte counter maintains a record of the number of bytes to be transferred to the Massbus device.

The silo loading function process will continue until the silo is full or until the SBI byte counter goes to zero indicating that all of the bytes required to complete the data transfer have been loaded into the silo.

If the silo is filled and data remains to be transferred from the SBI to the Massbus, the silo loading process will stop until a portion of the silo data is transferred to the Massbus. When this occurs, the silo loading process will continue.

When transferring data from the Massbus device to the SBI (data transfer read), data is loaded, via the Massbus receivers, into the Massbus data input buffers (MDPE). If a parity error is detected in the Massbus input data, the MBA PE bit in the status register is set and the data transfer operation will be aborted.

The following conditions must exist to enable the load silo function:

a. DT READ H asserted, the data transfer must be a read;

b. MDIB READY H asserted, the Massbus data input buffers must be ready to accept Massbus data;

c. SILO FULL L cleared, the silo cannot be full.

If there are no parity errors, SCLK from the Massbus device is received, and the data input buffer ready flip-flop has not been set previously, the ready and data input buffer full flip-flop will be set upon receipt of data from the Massbus.

If SCLK were received and the ready flip-flop was already set, a data overrun would occur. The DLT bit in the status register will be set, RUN will be cleared, and the transfer operation will be terminated.

Setting the ready and data input buffer full flip-flops initiates the silo input operation. Actual silo loading operations are the same as described for the write to Massbus device operation. Unlike the write to Massbus device operation, however, the data transfer begins immediately.

During a DT READ, each time a byte of data is loaded into the silo the MBA byte counter is bumped. After the data word has been transferred to the silo, the ready and data input full flip-flops are reset and the data input buffer is ready to accept the next Massbus device data word.

Each time a byte of data is transferred from the silo to the data output buffer, the SBI byte counter is decremented. The process of loading the silo and transferring bytes of data, via the output buffer, to the internal bus continues until the last byte has been transferred from the silo. At this time both the MB and SBI byte counters will equal Ø and the data transfer operation will be terminated upon receipt of EBL from the Massbus device.

4.4.5 Data Output Buffer and Control Logic (MDPJ, MDPM)

The data output buffer consists of eight 8-bit latches that receive silo data (SILO OUTPUT D<7:Ø>H). Virtual address register bits VAR BIT<Ø2:ØØ>H are decoded to produce LOAD DOB<7:Ø>H. These signals are used to enable one of the eight data output latches (Figure 4-29).

As each data output buffer is enabled, LOAD DOB<7:Ø>H set eight corresponding flip-flops (MDPL) that generate BYTE MASK<1-3:1-Ø>H, BYTE MASK<2-3:2-Ø>H, and DOB<7:Ø>FULL L. DOB SELC H, DOB SELB H, and DOB SELA H are decoded to produce NEXT DOB FULL L each time a byte of data is loaded into the output buffer register. If the next data output buffer already has valid data (byte mask set), the silo output logic will not load the next byte until the byte mask is cleared.

LOAD DOB H from the byte mask selection logic is also used to bump the virtual address register each time a byte of data is loaded into the output buffer to select the next storage location to or from which data is to be written or read.

4.4.6 Internal Bus Output Multiplexers (MDPR, MDPS)

The internal bus output multiplexers consist of eight 2-to-1 multiplexers and corresponding line drivers. Figure 4-3Ø illustrates the logic associated with the internal bus drivers.

When a read from Massbus device occurs, data from the data output buffer is applied to the internal bus output multiplexers, as well as the Massbus multiplexers. If data is to be asserted on the internal bus DT READ L, CMD REQ L and ENAB WD2 MUX or ENAB WD1 MUX H must be asserted. Selection of the data word to be asserted is determined by ENAB WD2 MUX H. When this signal is asserted, DOB D<37:ØØ>H are selected; when the signal is not asserted, DOB D<77:4Ø>H are selected. These words are strobed onto the internal bus by INT BUS OCLK H.

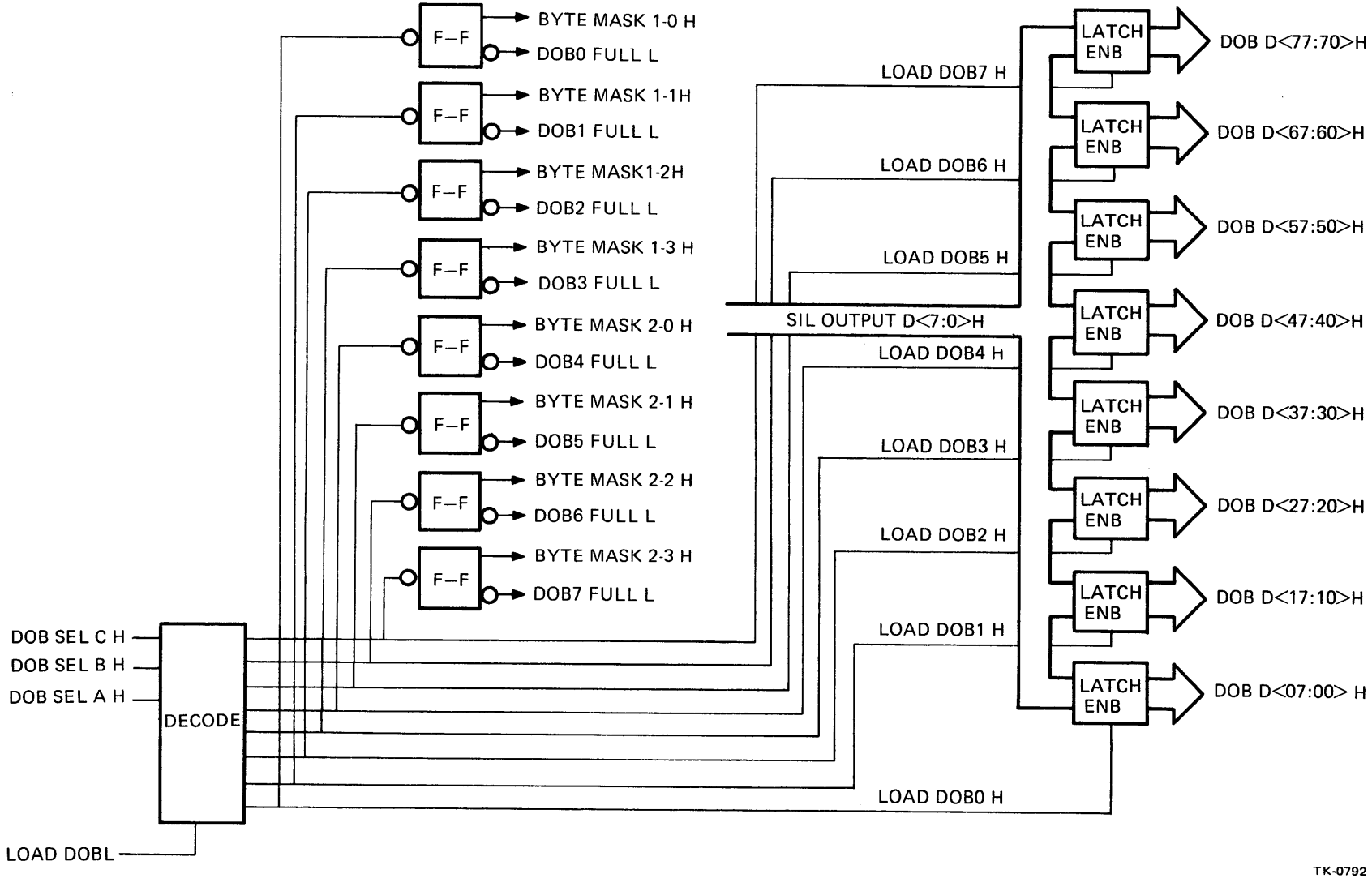


Figure 4-29 Data Output Buffer and Control Logic

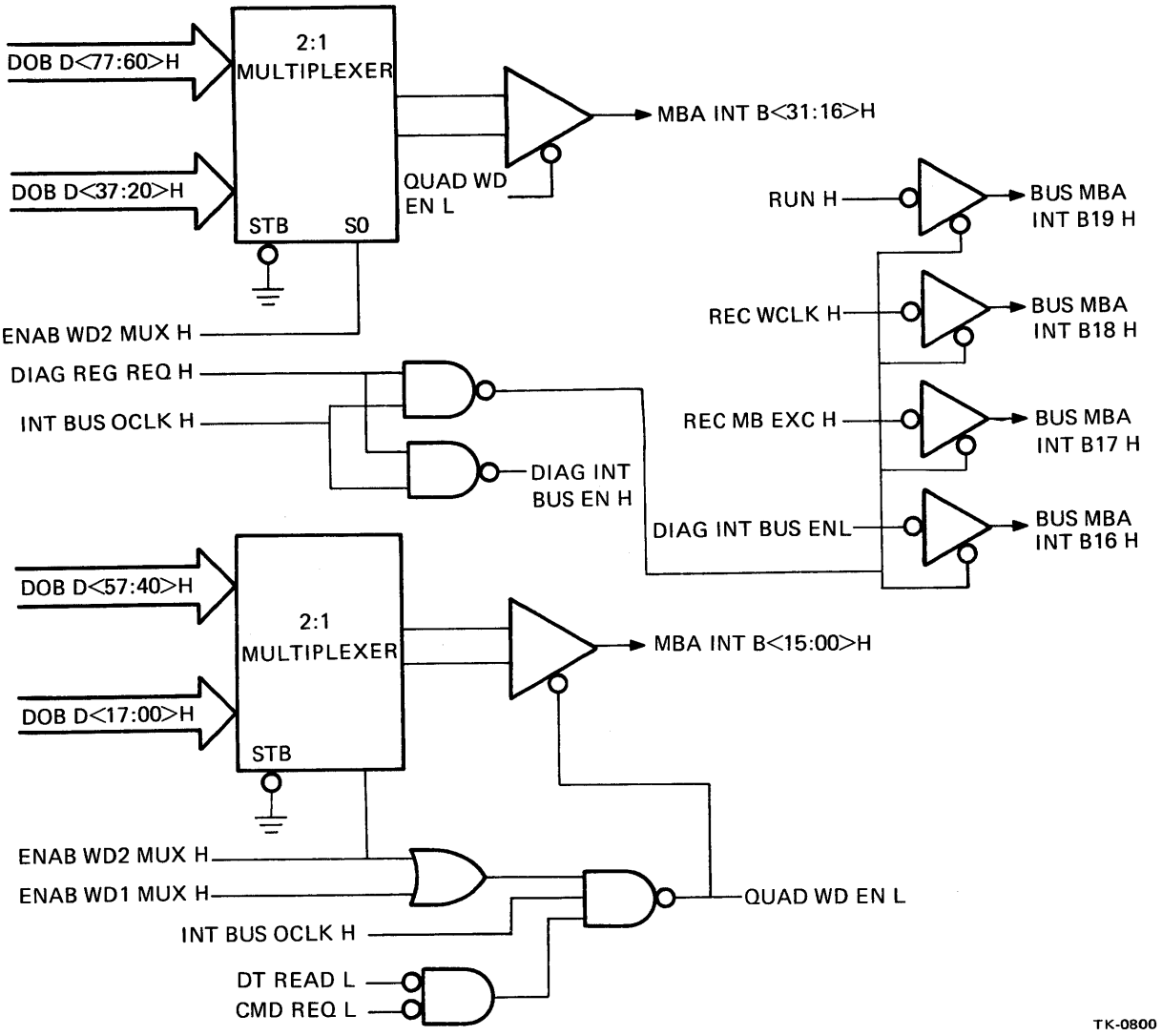


Figure 4-30 Internal Bus Output Multiplexers

4.4.7 Massbus Output Multiplexers and Parity Generator

Figure 4-31 illustrates the Massbus output multiplexers and parity generator.

For a write to Massbus device, data from the output buffer is loaded into the Massbus data multiplexer. The Massbus data multiplexer consists of eight dual 4-to-1 line multiplexers that divide the 32-bit words into two 8-bit bytes (16 bits) to comply with the input requirements of the Massbus device. Byte selection is determined by DOB BYTE CNTRL 1 and DOB BYTE CNTRL 2, developed by the data output select logic in response to SCLK from the Massbus device. The output from the MBA multiplexer is checked to ensure that the byte transmitted to the Massbus device contains an odd number of bits. If they do not, the parity generator produces XMIT MB DPA L to ensure odd parity.

If DT WRITE is asserted and the RUN flip-flop is set, data from the multiplexer is transmitted, via the Massbus drives, to the Massbus.

4.4.8 Write Check Logic

Write check logic (Figure 4-32) in the MBA verifies the integrity of write data by comparing the data from the Massbus output multiplexers with that of the data on the Massbus. DT WRITE CHK H must be asserted in order for the write check operation to be valid. A write check operation is performed identical to a write operation except for the following additional processes. Data from the Massbus output multiplexers is stored in the write check buffer memory. Data received from the Massbus is stored in the Massbus write check buffer memory. This received data should be identical to the data out of the Massbus output multiplexers. The data in the two write check memories is compared to test its validity (each compared bit must be equal). If an inequality exists in data bits <15:08>, WCK UPPER ERR H is asserted, if an inequality exists in data bits <07:00>, WCK LWR ERR H is asserted. The assertion of either one or both of these signals causes the data transfer to be aborted.

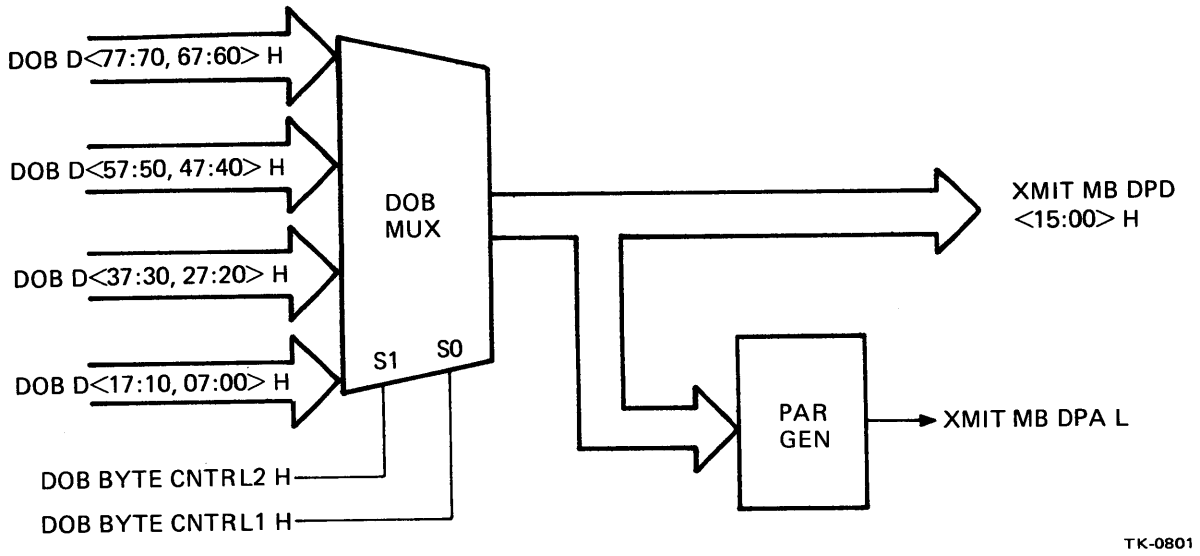


Figure 4-31 Massbus Output Multiplexers and Parity Generator

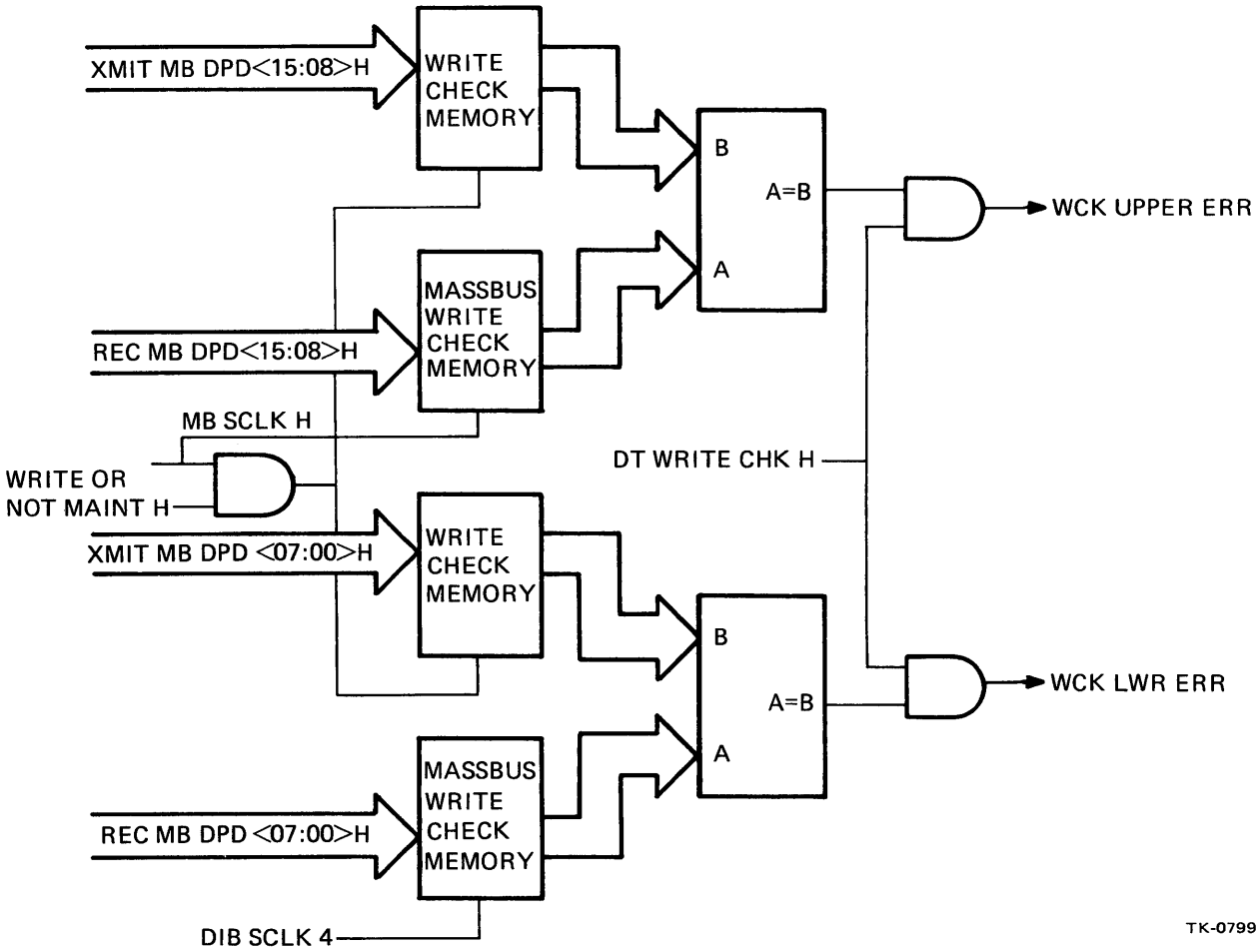


Figure 4-32 Write Check Logic

4.5 MBA CONTROL PATH

A control path data transfer is initiated when the command/address, decoded by the interface logic, specifies a read or write external register operation. Figure 4-33 is a block diagram of the MBA control path logic. The control logic has the following major functions:

- derive timing signals from the SBI and Massbus clocks
- decode and store function codes for data transfers
- decode external register addresses
- transfer data to and from external registers
- initiate Massbus control path cycles
- set certain status and error conditions in the MBA
- synchronize SBI and Massbus operations.

The following paragraphs describe the control logic used to perform these functions.

4.5.1 Internal Bus Receivers (MCPA)

The control path internal bus receivers consist of two 8-bit registers. Internal bus data (bits BUS MBA INT B<15:ØØ>H) is latched into the receivers, to be processed by the control path logic upon receipt of INT BUS ICLK L from the control path clock.

4.5.2 Data Transfer Control Logic (MCPB)

The data transfer control logic is illustrated in Figure 4-34. When a control path data transfer is to take place, the MBA asserts a drive address on the three Massbus drive select lines (DS<Ø2:ØØ>) and a register address on the five Massbus register select lines (RS<Ø4:ØØ>). Data transfer commands are only monitored when the selected register is the control register (RØ) of a drive. If a write to external register function is to be performed, the MBA also asserts the controller to drive line CTOD (transfer direction). Assertion of the RS, DS, and CTOD lines is accomplished in the following manner. The command/address bits REC INT B<Ø7:ØØ>H and WRITE FCN H from the internal bus and interface logic are latched into the control store register. DT GO is a 5Ø ns pulse that intializes the internal logic on the MBA at the start of a data transfer operation. EXT CMD H clocks the information through these registers to the Massbus transceivers for assertion on the Massbus. The register select lines of the control store output (EXT CS BIT<4:Ø>L) are ANDed with REC INT BØØ H (go bit) to enable a 3-to-8 line decoder. Bits REC INT B<Ø5:Ø3>H are decoded to select the function to be performed. Bits REC INT B(Ø2:Ø1>H indicate the direction of the indicated function (Table 4-3).

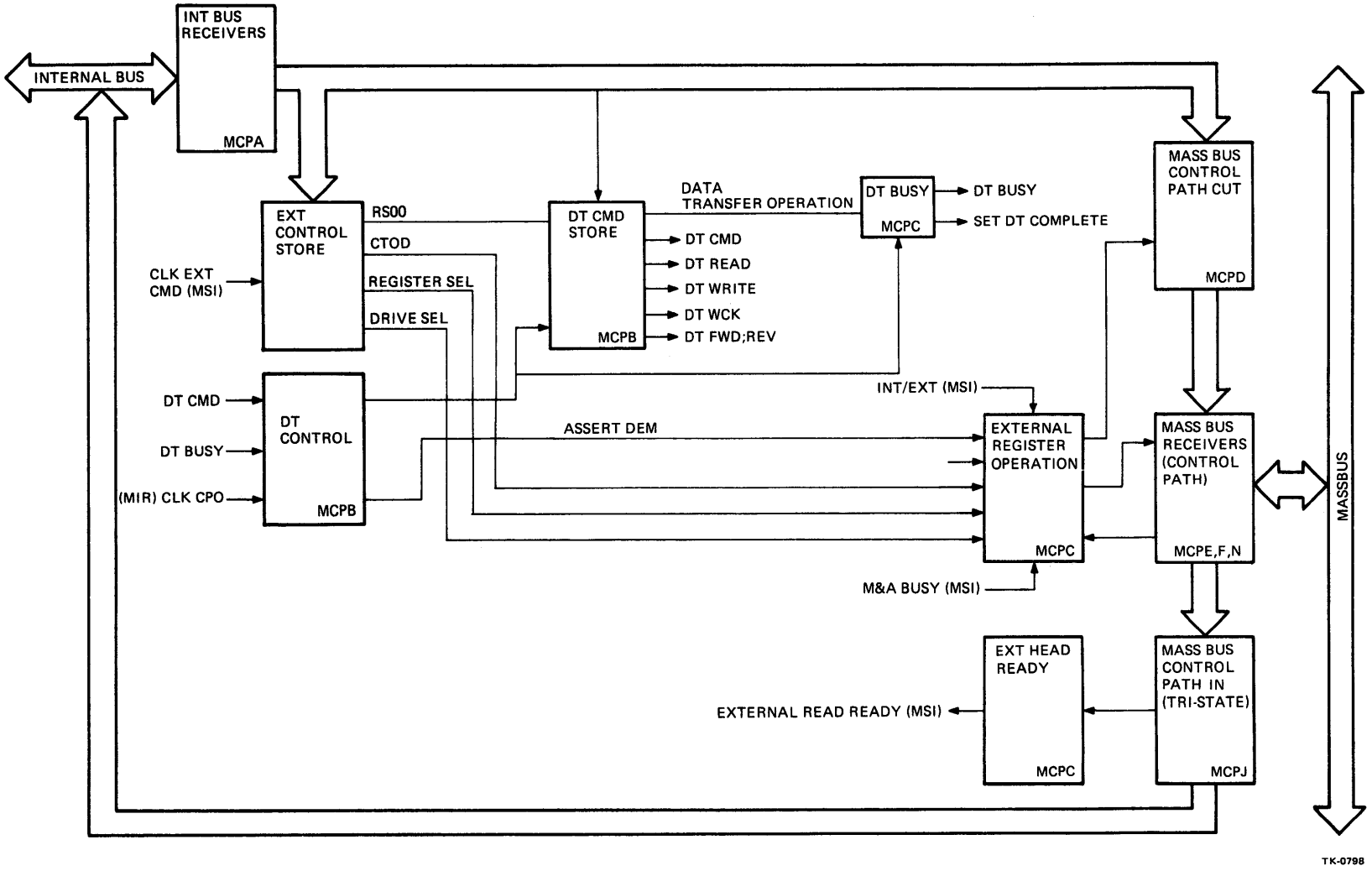


Figure 4-33 MBA Control Paths

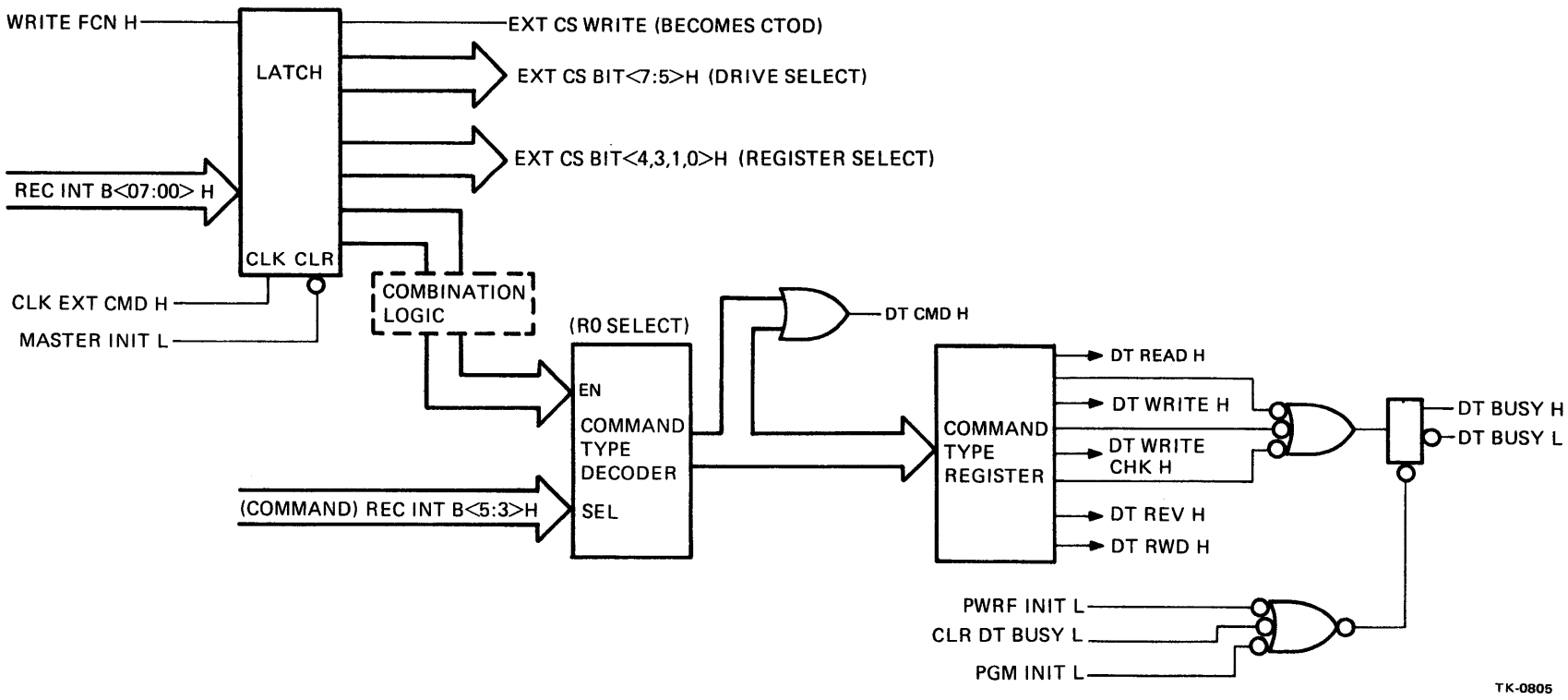


Figure 4-34 Data Transfer Control Logic

Table 4-3 Function and Direction Select

Function	REC INT Bits Ø5	Ø4	Ø3	Ø2	Ø1
Read Reverse	1	1	1	1	1
Read Forward	1	1	1	X Ø	Ø X
Write Reverse	1	1	Ø	1	1
Write Forward	1	1	Ø	X Ø	Ø X
Write Check Reverse	1	Ø	1	1	1
Write Check Forward	1	Ø	1	X Ø	Ø X

X = Don't care.

The decoded output is stored in the corresponding function flip-flop. REC INT B<Ø7:ØØ>H also produce EXT CS BIT <Ø7:ØØ>H when clocked through the internal bus latches. EXT CS BIT <Ø7:Ø5>H select the drive (8-Ø) to which data is written or from which data is read. EXT CS BIT <Ø4:ØØ>H indicate the source or destination register (ØF-ØØ).

Regardless of the selected valid function, an output from the function select flip-flops sets the DT BUSY flip-flop. If DT BUSY is already asserted, the MBA will not accept the new command. When this flip-flop is set, the MBA is processing a data transfer and cannot accept a new transfer command until the current transfer is complete.

The decoder output also produces DT CMD H, which indicates that the MBA has received a data transfer command. If CLK CPO H is asserted, indicating a write to external register function, and DT CMD H is asserted, ASSERT DEM H will be generated if the MBA is not busy with a data transfer (DT BUSY H cleared). ASSERT DEM H triggers a one-shot that sets the DEM flip-flop (MCPC). Its output, XMIT MB DEM H, when asserted on the Massbus demand line, indicates that the transfer of control path data to or from the Massbus device is to take place.

If a write to external register function is to be performed, the control data (REC INT B<15:00>) is stored in a temporary latch (MCPD) to be transferred, via the control path data output logic, to the Massbus device. Parity checking is performed to ensure that this data contains an odd number of bits (odd parity). If the content of the data word represents an even number of bits, the parity generation logic produces XMIT CNTRL PAR H, which is asserted on the Massbus CPA line (Figure 4-35).

Upon receipt of DEM, the addressed Massbus device will load the control path data into the selected register.

It should be noted that there is a 250 ns delay from the assertion of control data on the data lines to the assertion of DEM to compensate for skew in the MBA, Massbus cables, drivers, and receivers.

Once the control path data has been loaded into the selected register, the Massbus drive will issue TRA 250 ns later. DEM is negated and the DONE one-shot is triggered to indicate that the write to external register function has been completed. If after assertion of DEM, TRA is not returned to the MBA within 1.5 us, the NED (nonexistant device) flip-flop is set, DEM is negated, and the write to external register function is terminated.

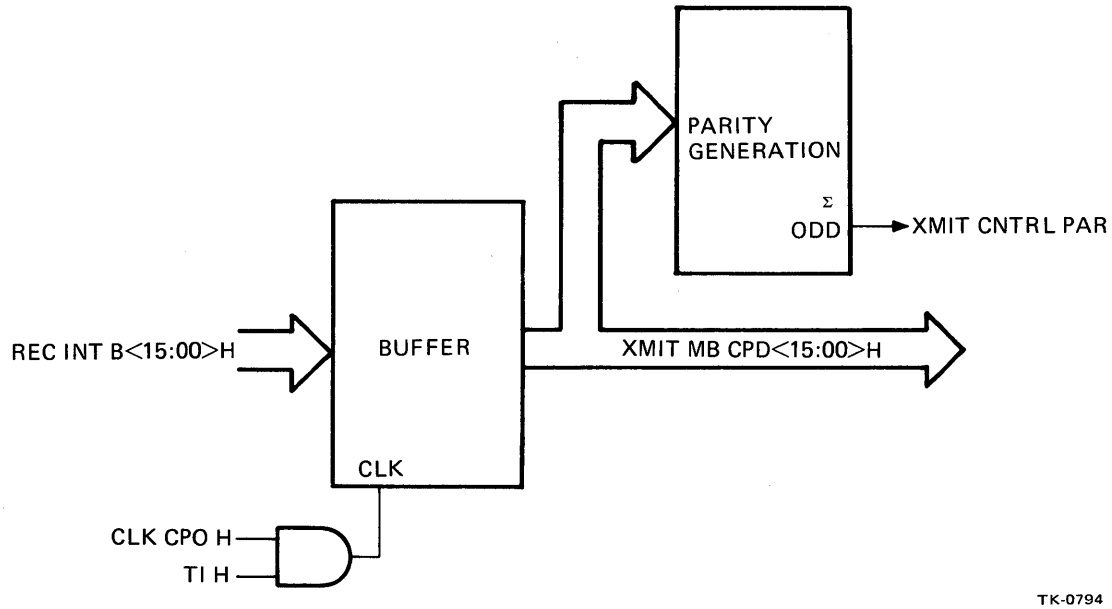


Figure 4-35 Internal Bus Data Register and Massbus Parity Generator

If the attention summary register is the destination of the write operation, TRA will be ignored and NED will not be set. A 1.5 us timeout will occur followed by the 250 ns delay and DEM will be negated. When the attention summary register is read, no timeout will occur.

When a read from external register function is executed, assertion of the device and register select address and DEM on the Massbus lines is accomplished in the same manner as described previously for the write function; however, CTOD is not asserted. Upon receipt of DEM, the device will load the contents of the selected register on the C<15:00> lines for transmission to the MBA and assert TRA. This data is applied, via the MBA control path receivers, to data latches. DESKEW DATA L (MCPC) latches the data in. Parity is checked to ensure that there were no transmission errors. If a parity error is detected, the MCPE parity error bit in the status register will be set (Figure 4-36). If no parity errors are detected, the latches are clocked by DESKEW DATA L. The output from the data latches is placed on the internal bus upon receipt of SEND TR H from the interface logic.

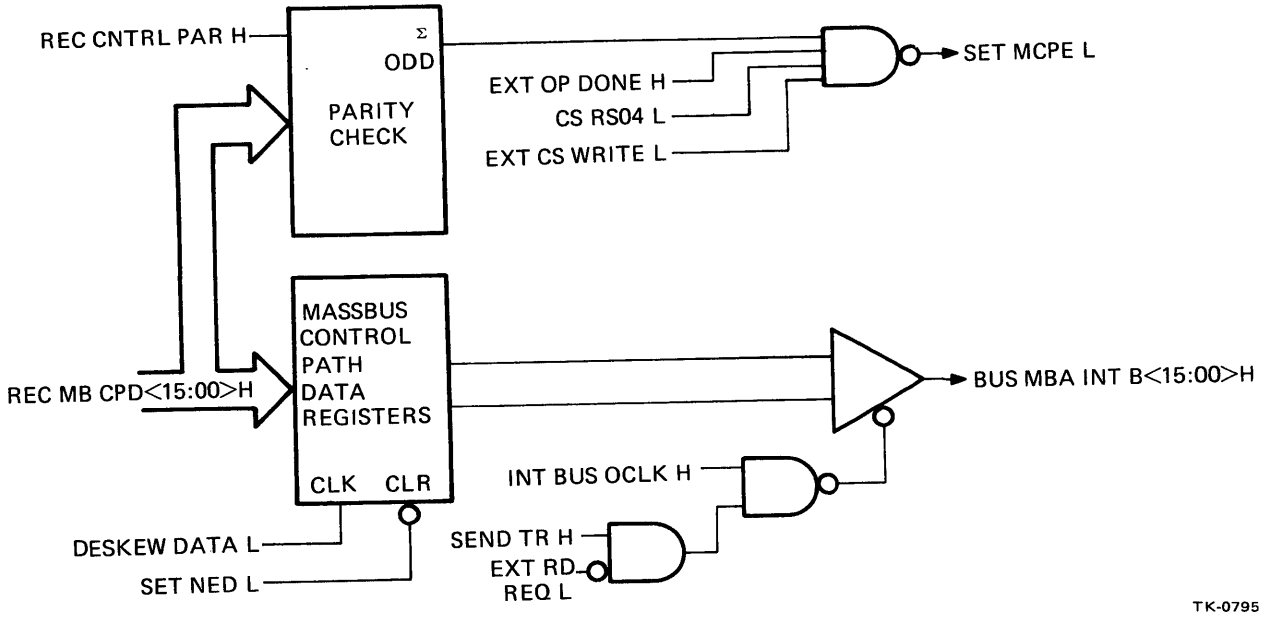


Figure 4-36 Massbus Control Path Data Registers and Parity Check (Received Massbus Data)

When the MBA is in the maintenance mode, MB MAINT MODE L (MCPE) prevents DEM from being asserted on the Massbus. The MBA performs a write asserting data on the Massbus followed by a read of the data just put on the Massbus. The MBA uses a timeout to complete the read. The register select and drive select lines are also read to ensure that the correct data has been put on these lines. The diagnostic register is read to check the validity of the operation.

4.5.3 Massbus Receivers/Drivers

Information is transferred to and from the Massbus via the Massbus control path receivers and drivers. Massbus information is summarized in Chapter 2 of this manual. The control path receivers are always enabled. Transmit enable signals are summarized in Table 4-4.

Table 4-4 Massbus Transmit Enable Signals

Control Path Signal	Massbus Signal	Enabling Signal
EXT CS BIT<7:5>H	MASS DS<2:Ø>	(always enabled)
EXT CS BIT<4:Ø>H	MASS RD<4:Ø>	(always enabled)
XMIT MB DEM H	MASS DEM	MB MAINT MODE L OR XMIT MB DEM H
XMIT MB INIT H	MASS INIT	(always enabled)
XMIT MB CPD<15:ØØ>H	MASS C(15:ØØ>	EXT CS WRITE L NAND MB MAINT MODE L
XMIT CNTRL RAR H	MASS CPA	(always enabled)
EXT CS WRITE H	MASS CTOD	(always enabled)
SIMULATE OCC H	MASS OCC	SIMULATE OCC H AND MB MAINT MODE H
XMIT MB EXC H	MASS EXC	(always enabled)
MB FAIL	MASS FAIL	(always enabled)

4.5.4 End Data Transfer Logic (MCPA)

End data transfer logic is responsible for generating the appropriate signals at the end of a data transfer. The logic monitors various signals in the control path and MBA interface to produce the following:

DT END L	Data transfer complete
CLR DT BUSY L	Clear data transfer busy
BLOCK SEND CMD H	Block sending the command/address
SET DT ABORTED L	Set data transfer aborted
SET DT COMPLETE L	Set data transfer complete

Figure 4-37 illustrates the logic used to accomplish these functions.

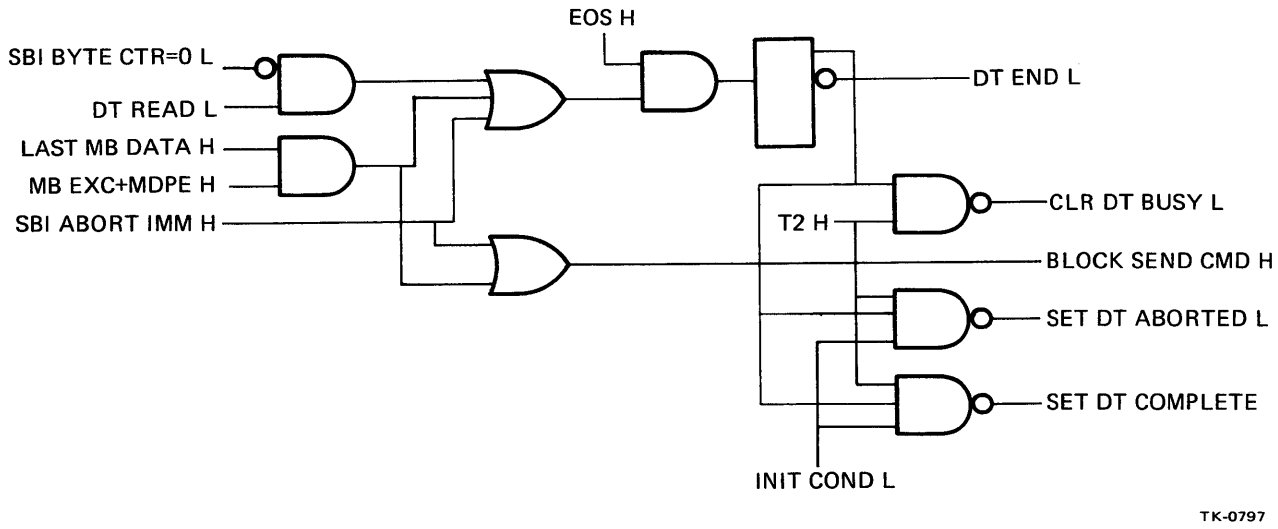


Figure 4-37 End Data Transfer Logic

DT END L will be asserted only after the MBA has dropped RUN and the drive has asserted EBL if one of the following conditions occur.

1. The data transfer function was a read (DT READ H), the SBI byte counter has gone to zero (SBI BYTE CNTR=Ø H), and memory has acknowledged the second data word (WD2 ACK H).

2. The data transfer was a write or write check function (DT READ L not asserted) and the SBI byte counter has gone to zero.

3. The data transfer is to be aborted immediately due to an error (SET SBI ABORT IMM H).

4. There has been a Massbus exception or parity error (MB EXC + MDPE H).

DT END L is synchronized with TØ. If DT END L has been asserted, CLR DT BUSY L will be produced at the following T2. Block SEND CMD L will be asserted as a result of an abort condition, Massbus exception, or a Massbus data parity error. SET DT ABORTED L will be asserted at T2 of the SBI cycle if DT END H is asserted, a Massbus exception or parity error occurs, and there is no initialize condition (INIT COND L not asserted). SET DT ABORT L will also be asserted if a missed transfer error has occurred (SET MXF L). SET DT COMPLETE L will be asserted at T2 if there is an initialize condition and DT END H has been asserted.

RH780 MASSBUS ADAPTER
TECHNICAL DESCRIPTION
EK-RH780-TD-001

Reader's Comments

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

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

What features are most useful? _____

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

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

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

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

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

Additional copies of this document are available from:

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

Order No. EK-RH780-TD-001

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

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

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

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

Postage will be paid by:

Digital Equipment Corporation
Communications Development and Publishing
1925 Andover Street
Tewksbury, Massachusetts 01876