EK-DMA20-UD-002

DMA20 MEMORY BUS ADAPTER UNIT DESCRIPTION

digital equipment corporation • marlborough, massachusetts

1st Edition, March 1975
2nd Edition (Rev), October 1976

The material in this manual is for informational purposes and is subject to change without notice. Digital Equipment Corporation assumes no responsibility for any errors which may appear in this manual.

Printed in U.S.A.

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

CONTENTS

CONTENTS (Cont)

ILLUSTRATIONS

TABLES

PREFACE

The DMA theory chapter contains the following three levels (sections) of description:

1. Overview
2. Functional Description
3. Logic Description

The Overview section identifies and introduces the major elements of the DMA20 and provides a brief description of their individual function and how they operate collectively to execute the primary DMA functions.

The Functional Description section defines the detailed functional structure of the DMA. The basic operations and capabilities of the DMA are described in terms of this functional structure. The description is supported with appropriate introductory and support graphics. The level of detail is limited to a functional perspective; however, major logic signals are introduced and described.

The Logic Description section is the most comprehensive portion of the DMA theory chapter. The section contains a detailed logic description of the structure introduced in the Functional Description. The logic elements are described in detail in the context of how they execute the primary DMA functions. The supporting graphics are logical detail expansions of similar graphics presented in the previous section. In addition, the section provides a direct index into the logic print set, through the use of print prefixes.

SECTION 1
OVERVIEW

1.1 GENERAL DESCRIPTION

The Memory Bus Adapter (DMA20) is the interface control element for the external core memory associated with the KL10 system. The DMA20, located within the system I/O cabinet, controls the transfer of data between the MBox portion of the central processor and the destructive readout core memory. In addition, the DMA20 provides error detection and reporting capabilities.

The DMA is equipped with four storage module bus ports; from 1 to 16 storage modules may be connected to each bus port providing a maximum storage capacity of over four million, 36-bit words. Communications between the DMA bus ports and the storage modules take place over each of the KBuses. Communications between the DMA and MBox take place over the SBus. A typical system configuration is shown in Figure 1-1.

1.1.1 System and Priority Operation

The operation of the DMA is synchronous with respect to the MBox and asynchronous with respect to external core memory. Access to memory is governed by a request/response dialogue wherein the MBox initiates a memory cycle request and waits for a response from the DMA. In turn, the DMA response is determined by a like request/response dialogue from core memory.

Since each storage module connected to a DMA bus port has more than one port, the storage module is required to service additional processors, data channels, or multiplexors on a time-sharing basis. In the event of two or more simultaneous requests, the memory priority network allows access to the processor having the highest priority.

1.1.2 Basic System Operation

Memory operation is initiated when the MBox requests a memory cycle by issuing a memory location address and the cycle type (i.e., read, write) over the SBus. After decoding the request components of a read request, the DMA enables the appropriate bus port (or ports) and issues a memory cycle request, together with the memory address, to those storage modules connected to the selected ports. In the case of a write request, the data is loaded into a DMA data buffer prior to issuing the cycle request to the storage modules.

If the addressed storage module is not busy, it reads the address into its input buffer and responds with an acknowledge signal. The acknowledge signal is gated through the DMA to the MBox indicating acceptance of the memory request. The DMA then controls the requested cycle type transferring data between the MBox and core memory. During a multiple data word request, if the addressed storage module is inoperative or in some way off line, the DMA will time-out and set an internal nonexistent memory error flag provided the first data word has been acknowledged.

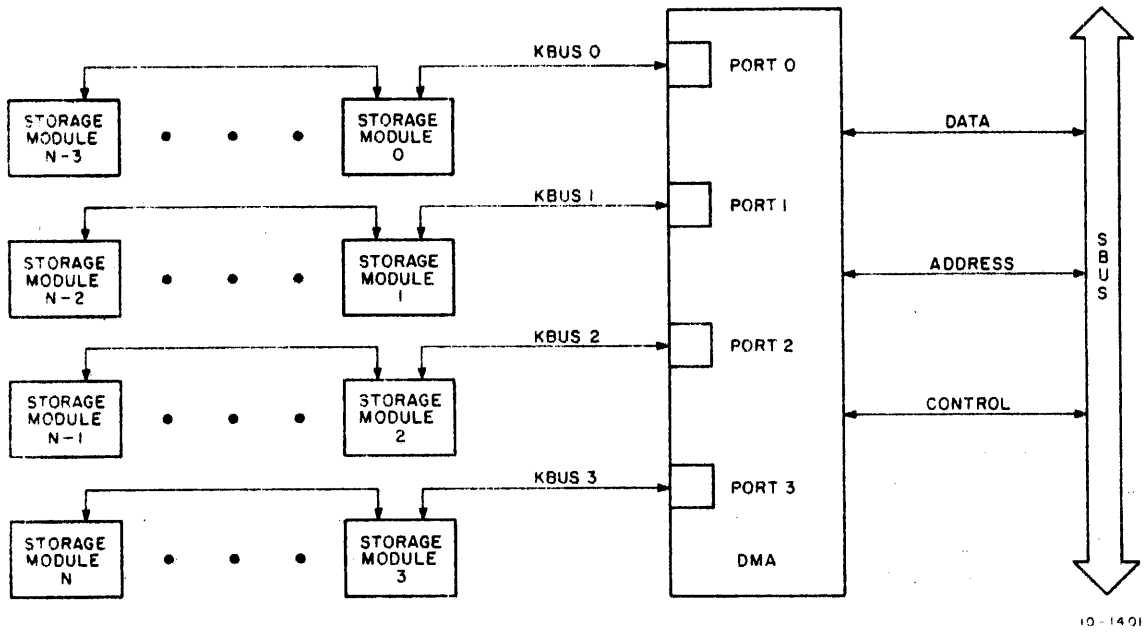


Figure 1-1 Typical System Configuration

1.1.3 Memory References

Memory references to the DMA can access from one to four 36-bit word locations. Normally the references are directed to four-word groups in memory, designated quadwords (quadword = Words 0, 1, 2, and 3). More than one location can be addressed at once, since four word request lines are included as part of the SBus, in addition to the set of 22-bit address lines. Each request line acts in conjunction with the address lines to specify one word within the quadword. In operation, the MBox supplies the starting address and the word request lines determine the number and the words accessed. Thus the MBox can specify any or all of the words within a quadword.

Although one to four words may be addressed simultaneously on the SBus, data transfer and other communications are on a serial basis. After the first word is accessed, words are cycled in ascending order, modulo 4. For example, a memory reference requesting a quadword with a starting address of 01 will cause the DMA to transfer data in the word order 1, 2, 3, 0. Likewise, a quadword starting address of 00 will result in words being cycled in the order 0, 1, 2, and 3.

1.1.4 Operational Modes

The DMA has three operational modes: One Bus Mode, Two Bus Mode, and Four Bus Mode. The operational modes determine the number of KBuses used during memory cycle requests and the associated storage module interleave configurations. Interleave configurations are analogous to the noninterleaved and 2-way and 4-way interleave modes used with KA/KI systems. That is, storage modules connected to bus ports in multiples of two or four with equal amounts of storage may be 2-way or 4-way interleaved, respectively. Storage module interleaving rotates successive memory cycles between interleaved modules if the addressing is sequential.

Operational and storage module interleave modes are enabled by software conditioning of the DMA and physically setting the appropriate interleave switches provided on storage module switch panels.

Mode and interleave associations are specified below:

a. One Bus Mode (1 bus mode): storage modules noninterleaved, or 2-way or 4-way interleaved; generally the DMA employs one KBus for memory requests provided all words requested are located in storage modules connected to the same bus. However, the DMA can employ all buses if the requested words are located in storage modules connected to different buses.

b. Two Bus Mode (2 bus mode): storage modules 2-way or 4-way interleaved; DMA employs two buses for memory requests.

c. Four Bus Mode (4 bus mode): storage modules 4-way interleaved; DMA employs four buses for memory cycle requests.

Operation in bus modes 2 and 4 effectively decreases cycle time, thus reducing processor idle time. In 2 bus mode and 2-way interleaving, the DMA can read/write up to two words per KBus cycle time with the two words distributed between two storage modules via two different KBuses. In 4 bus mode and 4-way interleaving, the DMA can read/write up to four words per KBus cycle time with the four words distributed between four storage modules via four different KBuses.

Normally the DMA operates in 4 bus mode. However, degraded operation is allowed in 2 bus mode, and further memory system degradation is allowed in 1 bus mode. Generally, memory system degradation is a user decision based on the amount of memory he is willing to lose as against the reduced access speed. For example, a time-sharing user operating 4-bus mode loses a storage module. In this case he may choose to reconfigure the memory system to 1 bus mode, deciding on a reduced memory system speed while maintaining as much memory on line as possible.

1.2 MEMORY CYCLE TYPES

The memory system has three cycle types: read/restore, clear/write, and read/modify/write. The read/restore cycle reads and transfers data from an addressed memory location through the DMA to the MBox. The clear/write cycle transfers data from the MBox through the DMA and writes it into an addressed memory location. The read/modify/write cycle reads and transfers data from an addressed memory location through the DMA to the MBox. The cycle then pauses while the data is modified. After modification the data is restored through the DMA to the original addressed memory location.

1.2.1 Read Restore Cycle

After cycle initiation as described in Subsection 1.1.2, the read/restore cycle transfers data from a selected memory location through a bus-associated data buffer in the DMA to the MBox. When the addressed storage module is not busy the module responds to the DMA request with an address acknowledge signal, indicating acceptance of the request.

During the read portion of the cycle, the memory reads data from the addressed location and loads it into the memory data latches. The data, together with parity, is then transferred to the associated DMA data buffer. After reception of the address acknowledge signal, the associated DMA data buffer is enabled allowing the data through the buffer to the SBus data lines. A memory-generated restart signal is then gated through the DMA to the MBox indicating that the requested data has been transferred. In the meantime, during the restore portion of the cycle, the storage module disconnects from the KBus to allow writing of the data contained in the memory latches back into the addressed memory location (destructive readout).

With the DMA operating in 2- or 4-bus mode and the storage modules interleaved and executing a multiple word request, the words will be retrieved from more than one storage module. In this case, although the words are transferred to the MBox in a specified order, they are not necessarily retrieved from the storage modules and stored in the DMA in that order. It is the responsibility of the internal DMA control logic to interpret the required word transfer order and comply with it.

1.2.2 Clear/Write Cycle

After cycle initiation as described in Subsection 1.1.2, the clear/write cycle transfers data from the MBox through a bus-associated data buffer in the DMA to a selected memory location. When the addressed storage module is not busy, the module responds to the DMA request with an address acknowledge signal, indicating acceptance of the request.

During the clear portion of the cycle the storage module reads data from the addressed memory location. At this point, the write data is available in the DMA data buffer. The internally generated DMA restart signal enables the associated DMA data buffer, gating the data and parity from the buffer to the KBus and storage module. The storage module then completes the cycle by writing the data and parity (contained in the memory data latches) into the addressed location.

1.2.3 Read/Modify/Write Cycle

After cycle initiation as described in Subsection 1.1.2, the read/modify/write cycle transfers data from a selected memory location through the DMA data buffer to the MBox for modification. After modification, the data is written into the same memory location. This operation is executed only using a one-word request.

If the addressed storage module is not busy, the module executes a normal read cycle. At this point the memory does not restore the data in its latches to the original location, instead the storage module clears the latches and waits. While the storage module pauses the MBox modifies the data (modify portion) and initiates the write portion of the cycle. The MBox places the modified data on the SBus and issues a write restart signal to the DMA. The restart signal enables the associated DMA data buffer, gating the modified data through the buffer to the memory latches. The DMA initiates the internal restart signal to the addressed storage module, which writes the modified data and parity (contained in the latches) into the original location.

1.3 ERROR DETECTION AND REPORTING

The DMA provides the core memory system with an error detection and reporting capability. Data parity is checked in the DMA during the three cycle operations. Should a parity error be detected during any portion (address/data) of a data transfer, it is flagged in the DMA error register and an appropriate error signal is sent to the MBox. In addition, checking is provided for nonexistent memory (NXM) references. As with a parity error, an NXM reference (on other than the first cycle) will set an associated flag in the error register. When the software desires error status it performs a diagnostic cycle request. The DMA will then enable the error status to the SBus data lines where it is retrieved and analyzed by the software.

The diagnostic cycle request also provides several functions for diagnosing the DMA portion of the memory system. In addition, the diagnostic cycle provides the capability of setting the DMA to the desired operational mode.

1.4 PHYSICAL CONFIGURATION

The DMA has a total complement of 10 modules; its physical configuration is illustrated in Figure 1-2. Each module is briefly described in the following paragraphs.

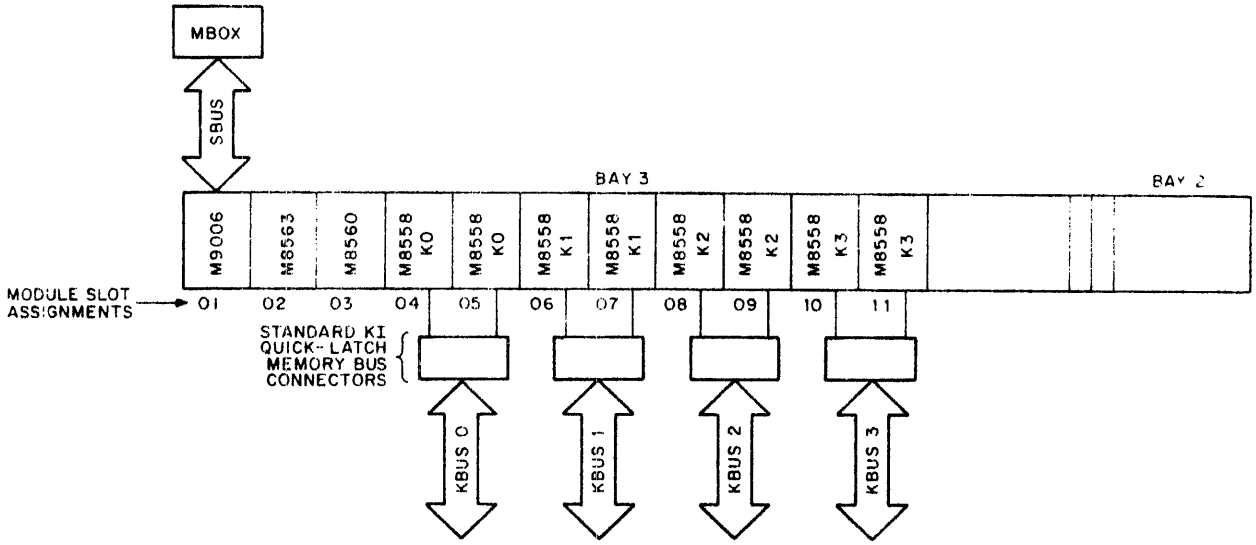


Figure 1-2 Physical Configuration

Module M8558: Eight boards, illustrated on field maintenance prints MA1-6. Contains the KBus data buffers and corresponding address lines. In addition, the logic module contains the storage module response logic. Each module contains 18 data bits plus a parity bit and 11 address lines. Thus, each of the four KBuses has two M8558 modules associated with it.

Module M8560: One board, illustrated on field maintenance prints DTR1-5. Contains the SBus receivers/drivers, error and clock logic.

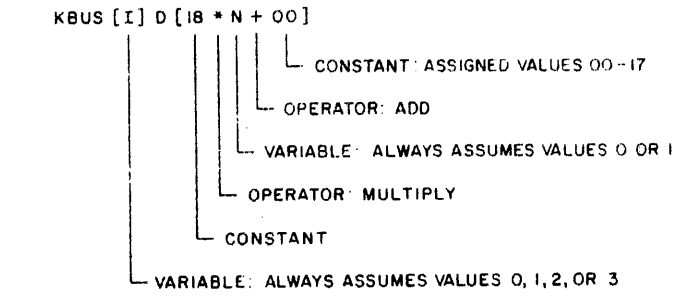
Module M8563: One board, illustrated on field maintenance prints DMC1-7. Contains the request response, KBus selection and timing logic.

1.5 LOGIC DIAGRAM NOTATION

A special function notation system is incorporated on logic prints MA1-6. The notation system reduces (by a factor of eight) the number of prints required to illustrate use of the actual logic contained on the eight M8558 modules.

A function name is represented by one of three types of arithmetic expressions. Each expression contains variables and operators which are evaluated, using integer arithmetic, to produce a particular function name or indicate a specific bit position. The three types of expressions are described in Figures 1-3 through 1-5, together with examples of expression evaluation. Note that the third type (Figure 1-5) is actually a combination of the first two types.

In addition, module backplane slot position numbers are specified by combinations of the assumed values of variables N and I. Table 1-1 lists the N and I value combinations and their associated slot numbers. Note that these N and I combinations are applicable for all expression types.



EXAMPLE EVALUATION

GIVEN:

KBUS [I] D [18 * N + 00] AND SUBSTITUTING I = 0, N = 1

EVALUATING THE EXPRESSION:

KBUS [0] = KBUS 0

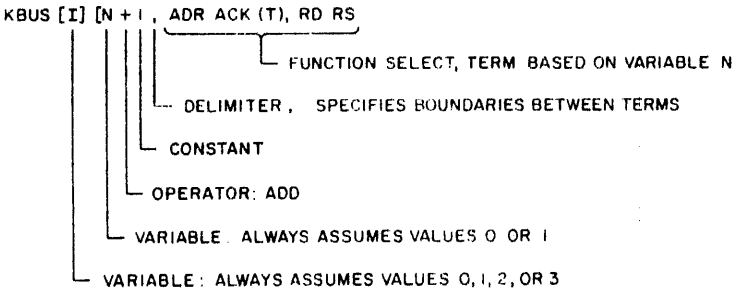
D [18 X 1 + 00] = D18

THEREFORE:

KBUS0 D18

10-1447

Figure 1-3 Type 1 Expression Description/Evaluation



EXAMPLE EVALUATION

GIVEN:

KBUS [I] [N + 1, ADR ACK (T), RD RS] AND SUBSTITUTING I = 1, N = 1

EVALUATING THE EXPRESSION:

KBUS [1] = KBUS1

N + 1 = 1 + 1 = 2, USE SECOND TERM (RD RS)

THEREFORE:

KBUS1 RD RS

10-1448

Figure 1-4 Type 2 Expression Description/Evaluation

```
[(3*I+6)/4, SBUS I ADR [10* N+18] L, [N+1, SBUS I START A H, SPARE SBUS TERM 8], SBUS I D [18*N+(I-1) /2*10+04] L]

EXAMPLE EVALUATION

THE FIRST TERM [(3*I+6)/4] SPECIFIES THE EXPRESSION TERM TO BE USED.
REMAINING TERMS ARE EVALUATED AS IN TYPES 1 AND 2.

SUBSTITUTING I = 1, N = 0

EVALUATING THE EXPRESSION:
(3*1+6)/4 = 3+6/4 = 9/4 = 2 (ROUNDED OFF) = USE SECOND TERM, ALSO IDENTIFIES BOARD PAIR (KBUS 1)

[N+1, SBUS I START A H, SPARE SBUS TERM 8]
   0+1 = 1 = USE FIRST TERM IN BRACKETS
         = SBUS I START A H

THEREFORE:
   FROM EVALUATION: FUNCTION = SBUS I START A H

FROM TABLE 1-2: MODULE SLOT NUMBER = 6
```

10-1783

Figure 1-5 Type 3 Expression Description/Evaluation

Table 1-1 N/I Value Combinations

Variable I	Variable N	Slot Number
00	0	4
00	1	5
01	0	6
01	1	7
02	0	8
02	1	9
03	0	10
03	1	11

The slot number in the Figure 1-3 example (I = 0, N = 1), the module slot number is 5. Likewise, in the Figure 1-4 example (I = 1, N = 1) the module slot number is 7.

1.6 CORE MEMORY SPECIFICATIONS

Core memory is constructed of coincident current, ferrite core, 3D, 3-wire storage modules. Depending on the type connected, storage module capacity may vary from 16K to 128K 37-bit words (36 data bits and 1 parity bit). Tables 1-2 through 1-4 list the major specifications for each type of storage module that may be connected to a DMA. If additional storage module information is required, refer to the document reference in the appropriate table.

Table 1-2 ME10 Storage Module Specifications

Characteristics	Specification
Cycle Time	1.0 μs maximum
Read Access Time	610 ns maximum
Address Acknowledge Time	200 ns maximum
Word Length	36-bits plus parity
Module Size	16,384 words
Access Ports	Four
Interleaving Modes	2-way and 4-way
Document Reference	A-MN-ME10-0-MAN-1

Table 1-3 MF10 Storage Module Specifications

Characteristics	Specification
Cycle Time	950 ns maximum
Read Access Time	610 ns maximum
Address Acknowledge Time	200 ns maximum
Word Length	36-bits plus parity
Module Size	65,536 words maximum
Access Ports	Four
Interleaving Modes	2-way and 4-way
Document Reference	A-MN-MF10-0-MAN-1

Table 1-4 MG10 Storage Module Specifications

Characteristics	Specification
Cycle Time	1.0 μs maximum
Read Access Time	660 ns maximum
Address Acknowledge Time	230 ns maximum
Word Length	36-bits plus parity
Module Size	131,072 words maximum
Access Ports	Eight
Interleaving Modes	2-way and 4-way
Document Reference	EK-MG10-MM-001

1.7 TEXT DEFINITIONS

The following terms are defined to avoid reader confusion with the several cycle types and general function references encountered in the remainder of the text (Sections 2 and 3).

DMA Cycle: Defined as the total internal DMA processing required to service any MBox read or write request regardless of the number of words involved in that request.

Word Cycle: (Actually a subset of the DMA cycle.) Defined as the internal DMA processing required for a single-word read or write operation.

KBus Cycle: Defined as a single-word read or write operation over a particular KBus (or port). The word and KBus cycles are regarded as separate items, and their definitions do not imply that a KBus cycle necessarily overlaps a word cycle (or vice versa).

Memory Cycle: Defined as the storage module processing required to read or write a single word.

N: Defined as a substitute for the term 0–3 (i.e., N = 0–3). N appears when a function is used in a general rather than a specific context (e.g., KBUS N = KBUS 0–3; DMC5 KN WR CLK = DMC5 K0–3 WR CLK).

SECTION 2
FUNCTIONAL DESCRIPTION

2.1 GENERAL SYSTEM DESCRIPTION

The DMA20 provides the control and data paths between the MBox portion of the central processor and external core memory as shown in Figure 1-1. Depending on the memory cycle requested, the DMA data path logic bidirectionally transfers data between the addressed memory location and the SBus. The DMA control logic selects the required port and associated KBus. It also performs sequential address incrementing and other memory control functions during memory cycle execution.

2.2 FUNCTIONAL LOGIC AREAS

For descriptive purposes the DMA is divided into the functional areas shown in Figure 2-1 and described in the following paragraphs.

Data Path/Control: Contains the address line translators and four data buffers. A set of address line logic level translators, data buffer, and data buffer logic level translators are associated with each memory bus port.

Read/Write Control: Generates the read and write operation signals and their synchronization, as well as controls the corresponding SBus signals. In addition, it generates the enabling functions during read operations and timing functions during write operations.

Request/Acknowledge Control: Generates the memory cycle request and acknowledge control signals during memory operations. In addition, the logic synchronizes those signals at the earliest optimum time.

Address/Bus Selection: Provides bus port (and associated KBus) selection as determined by the SBus address and word request set. The logic also supplies the synchronizing bus selection functions to other functional areas.

Error Detection and Reporting: Provides address and data parity checking and error storage logic. When used with the MBox diagnostic functions, the error logic provides control for the reporting capability as well as controlling the internal DMA functions (e.g., operational bus mode selection).

Internal Clock: Contains the DMA clock generators, which are driven from the external MBox clock. The generators provide a phased clock at the MBox clock rate and several additional phased clocks at twice the MBox clock rate.

KBus/Port Functions: A general area used to graphically represent the grouping of all KBus, functions with a selected port and its associated KBus, which includes the synchronizers and logic level translators.

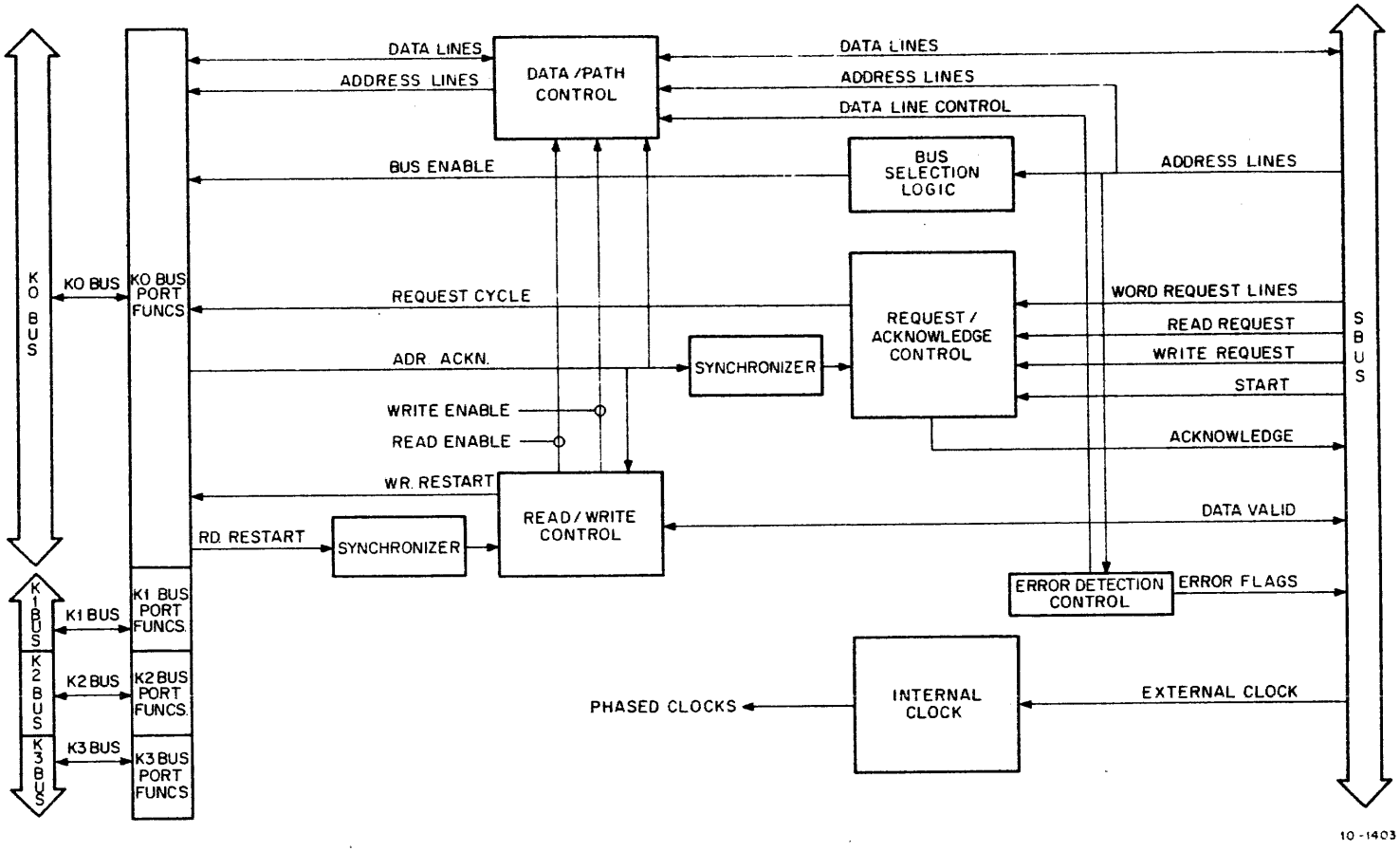


Figure 2-1 DMA Functional Areas

2.3 POWER AND INITIALIZATION

During power up and power down, all DMA memory cycle request lines are disabled to prevent the possibility of any cycle initiation. During periods when the DMA is not executing memory cycle requests, it continually direct clears its data buffers and holds the KBus synchronizer request functions disabled.

2.4 SYSTEM INTERFACE DESCRIPTIONS

The memory system requires two memory buses. Communications between the MBox and the DMA take place over the SBus; communications between each DMA bus port and its storage modules take place over a corresponding K Bus. The system interfaces are shown in Figure 2-2; the interface lines are described in the following subsections. A general system timing diagram is provided in Figure 2-3.

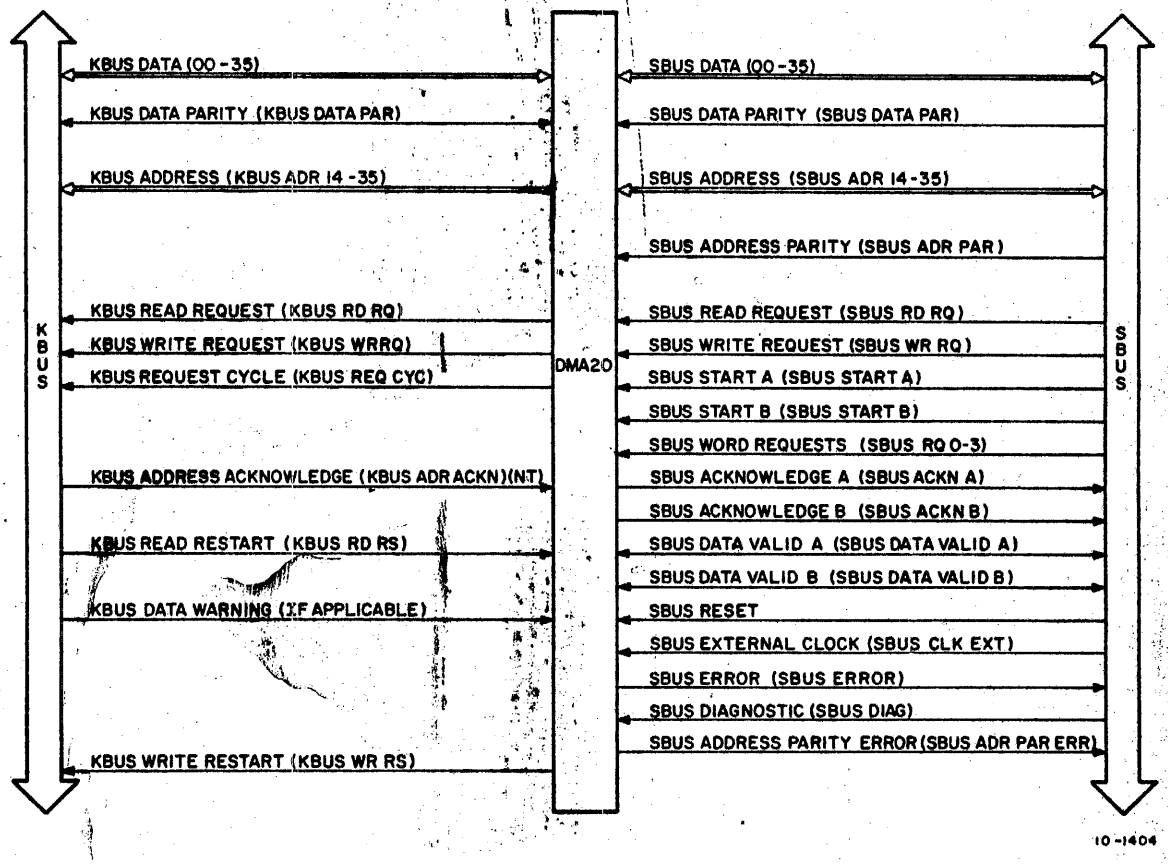


Figure 2-2 External Memory System Interface

2.4.1 SBus Signal Line Definitions

SBus Data (SBUS D00-35): Data passes between the MBox and DMA as levels (low = 1, high = 0). The 36 bidirectional data lines transfer data words to the MBox during a read/restore cycle and the read portion of a read/modify write cycle. Likewise, the lines transfer data words to the DMA during a clear/write cycle and the write portion of a read/modify/write cycle.

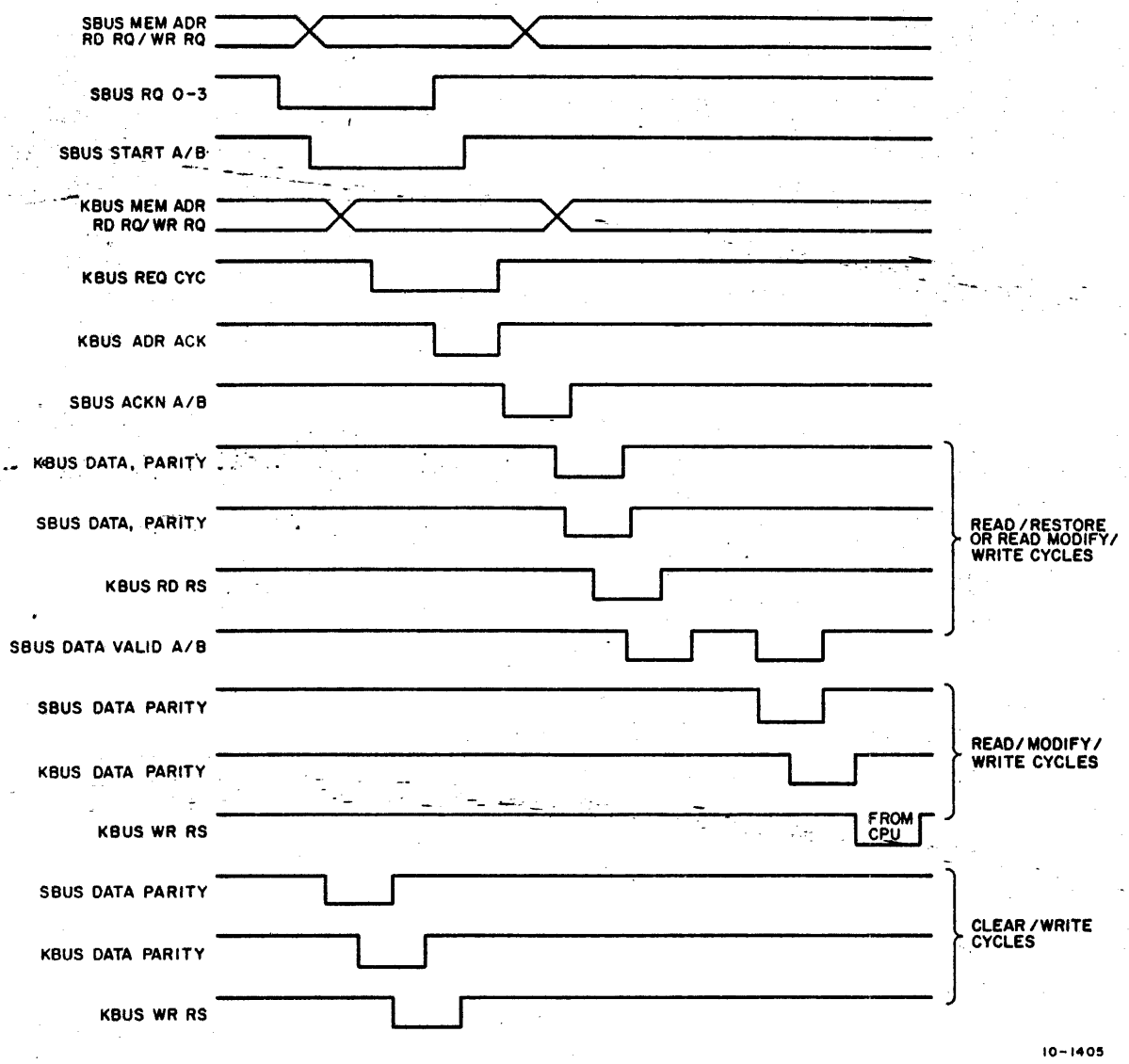


Figure 2-3 Interface Timing Diagram

SBus Data Parity (SBUS DATA PAR): The bidirectional parity line transfers odd parity as computed on the associated data words during read/write cycles. Write data parity is checked in the DMA and transferred to the storage module. The storage module does not compute parity, only stores it with the associated word. Read parity is read out of memory, checked in the DMA, and transferred to the MBox. Should a read parity error be detected the DMA will set appropriate error flags and transfer the bad parity to the MBox.

SBus Address (SBUS ADR 14–35): The address lines designate the specific address (physical location) to read/write in the selected storage module. Bits 14–33 address a quadword; bits 34–35 address the first word to be accessed within the quadword.

SBus Address Parity (SBUS ADR PAR): The address parity line transfers odd parity with the read/write cycle address. As with data parity the storage module does not compute parity, only stores and checks it. Should an address parity error be detected, the KBus cycle request is inhibited.

SBus Clock (SBUS CLK EXT): This clock (external) drives the DMA internal clock generator which, in turn, provides phased (deskewed) clock signals at the SBus clock rate and at twice the SBus clock rate. The Phase A clock is generated on the fall time of the external clock.

SBus Read Request (SBUS RD RQ): The MBox asserts this line to request a memory read cycle. The signal is applied to the Request/Acknowledge control, which initiates a DMA read request to the addressed storage module. With both SBUS RD RQ and SBUS WR RQ asserted, the MBox is requesting a read/modify/write cycle from the addressed storage module.

SBus Write Request (SBUS WR RQ): The MBox asserts this line to request a memory write cycle. The signal is applied to the Request/Acknowledge control, which initiates a DMA write request to the addressed storage module. With both SBUS WR RQ and SBUS RD RQ asserted, the MBox is requesting a read/modify/write cycle from the addressed storage module.

SBus Start A (SBUS START A): The MBox asserts this line on Phase A of its clock. START A is applied to the Request/Acknowledge control, which initiates a DMA request for a memory cycle from the addressed storage module. Memory cycle type is determined by the SBUS RD RQ/WR RQ line asserted.

SBus Start B (SBUS START B): The MBox asserts this line on Phase B of its clock and is functionally identical to SBUS START A.

SBus Word Requests (SBUS RQ 0–3): The four request lines (0–3), in conjunction with the operational mode and starting address (SBUS ADR 34–35), determine DMA bus selection and data word accessing.

SBus Acknowledge A (SBUS ACKN A): Represents the DMA generated address acknowledge as returned from the storage module indicating that the storage module has accepted the memory cycle requested. ACKN A is returned to the MBox on the Phase A clock.

SBus Acknowledge B (SBUS ACKN B): This line is functionally identical to SBUS ACKN B except it is returned to the MBox coincident with the Phase B clock.

SBus Data Valid A (SBUS DATA VALID A): Transfers an equivalent memory-generated read restart signal from the DMA in sync with the Phase A clock. When true, indicates that data and parity are valid and have been transferred to the MBox during a read cycle or read portion of a read/modify/write cycle. Only during the write portion of a read/modify/write cycle is DATA VALID sent to the DMA. In this case it represents an equivalent write restart signal, which initiates the write portion of the cycle in the DMA.

SBus Data Valid B (SBUS DATA VALID B): SBUS DATA VALID B is functionally identical to SBUS DATA VALID A, except it is in sync with the Phase B clock.

SBus Error (SBUS ERROR): SBUS ERROR is a cumulative error line; when true represents the detection of a read or write data parity error or NXM error. The MBox may retrieve the particular DMA error flag by executing the appropriate diagnostic cycle.

SBus Address Error (SBUS ADR PAR ERR): This line true represents the detection of an address parity error. On detection of this type error, the DMA will inhibit any cycle requests over the KBuses.

SBus Diagnostic (SBUS DIAG): SBUS DIAG is used only in the diagnostic mode. When true, SBUS DIAG initiates a diagnostic operation in the DMA. The particular diagnostic function executed is determined by the diagnostic information placed on the SBus.

SBus Reset (SBUS RESET): Causes a reset of the DMA circuitry to a known logic state.

2.4.2 KBus Signal Line Definitions

KBus Data (KBUS D00-35): Data passes between the DMA and the attached storage modules as pulses (words), pulse = 1, no pulse = 0. The 36 bidirectional data lines transfer data words to the DMA during a read/restore cycle and the read portion of a read/modify/write cycle. Likewise, the lines transfer data words to the storage modules during a clear/write cycle and the write portion of a read/modify/write cycle.

KBus Data Parity (KBUS DATA PAR): The bidirectional parity line transfers odd parity as computed on the associated data words during read/write cycles. Write data parity from the MBox is checked in the DMA and transferred to the addressed storage module. The storage module does not compute parity, only stores it with the associated word. Read parity is read out of memory, checked in the DMA, and transferred to the MBox. Should a read parity error be detected, the DMA will set appropriate error flags and transfer the bad parity to the MBox.

KBus Address (KBUS ADR 14-35): The address lines designate the specific address (physical location) to read/write in the selected storage module. Bits 14-33 address a quadword; bits 34-35 address specific words within the quadword. While the SBus address lines supply only the starting address (for a quadword), the DMA Address Selection logic supplies the correct bits (34-35) to the enabled KBuses.

KBus Request Cycle Fast (KBUS RQ CYC FAST): With RQ CYC asserted, a memory cycle is requested from the addressed storage module. Cycle type is dependent on the RD RQ/WR RQ. Although fast, slow, and immediate type cycle requests (cycle overlapping) are not used with the DMA, they are required for bus compatibility with KA/KI storage modules. On any cycle request all three lines (on the selected KBuses) are asserted simultaneously. Note that the address and RD RQ and WR RQ lines must be valid prior to setting RQ CYC true. This requires that the RQ CYC line be false whenever the ADR, RD RQ, and WR RQ lines are changing.

KBus Request Cycle Slow (KBUS RQ CYC SLOW): This line is functionally identical to KBUS RQ CYC FAST/IMMEDIATE, i.e., required for bus compatibility.

KBus Request Cycle Immediate (KBUS RQ CYC IMM): This line is functionally identical to KBUS RQ CYC FAST/SLOW, i.e., required for bus compatibility.

KBus Read Request (KBUS RD RQ): The DMA asserts this line to request initiation of a read/restore cycle from the selected storage module. RD RQ must be true prior to KBus RQ CYC going true. With both KBUS RD RQ/WR RQ true, a read/modify/write cycle is requested.

KBus Write Request (KBUS WR RQ): The DMA asserts this line to request initiation of a clear/write cycle from the selected storage module. WR RQ must be true prior to KBUS RQ CYC going true. With both KBUS WR RQ/RD RQ true, a read/modify/write cycle is requested.

KBus Read Restart (KBUS RD RS): Sent to the DMA from the selected storage module to indicate that data and parity are valid and have been transferred to the DMA. RD RS is generated only during read/restore and the read portions of read/modify/write cycles.

KBus Write Restart (KBUS WR RS): Sent to the selected storage module from the DMA indicating that data and parity are valid and have been transferred to the storage module. WR RS is generated only during clear/write and the write portions of read/modify/write cycles. If WR RS is not generated during these cycles, the storage module may hang and require clearing by the operator.

KBus Address Acknowledge [KBUS ADR ACK (NT)]: Returned from selected MA and MB10 storage modules indicating acceptance of the memory request.

KBus Data Warning (KBUS DATA WARNING): Returned from selected MD, ME, MF, and MG10 storage modules. It is sent prior to read restart and indicates the subsequent reception of data. Refer to the *KL10 Service Manual* for the adjustment procedure. Note that this procedure is different from that of the KI CPU.

2.5 CLOCK OPERATION

The external SBus clock (SBUS CLK EXT) provides a timing train to drive the internal DMA clock generator. The clock generator outputs sequence logic and synchronize operations with the MBox. In addition, the clock generator has delay networks to provide deskewing for the propagation time delay encountered on the SBus. The external clock input is used to generate a phased clock train at the external clock frequency with a nominal pulse width of 20 ns. This clock, called the Phase A 125 ns clock, is generated on the fall time of the external clock (Figure 2-4). Included with the clock is a phasing network used to detect and indicate the period prior to Phase A being true, and when Phase A is true.

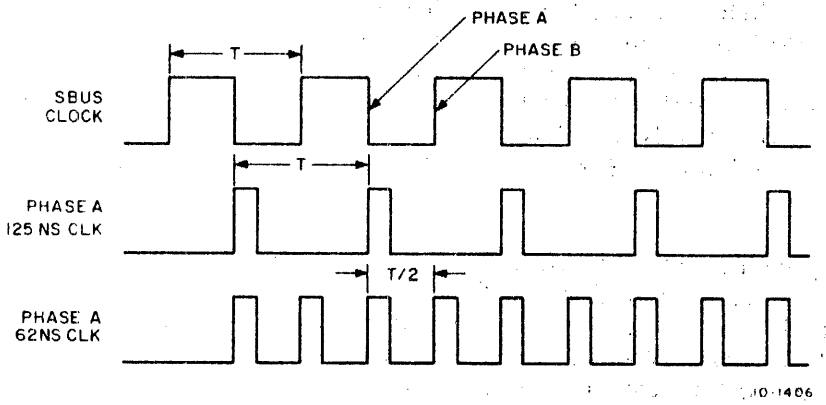


Figure 2-4 Clock Generator Timing

With the 125 ns clock synchronized in both the MBox and DMA, and with control bus propagation delays less than the period between clocks, SBus control signals generated on one end of the bus by a particular clock phase can be received at the other end of the bus on the next clock of the same phase without the need for synchronizing logic. For example, SBUS ERROR is transmitted in the controller on Phase A and strobed in the MBox after one external clock period, also on Phase A. Other SBus signals, such as START, are also linked to a particular phase. With no time lost in synchronizing the bus signals to internal logic, memory access times are held to a minimum.

In addition to generating the Phase A 125 ns clock, the external clock is effectively frequency divided to produce six additional clock trains having a repetition rate of twice the external clock frequency and a nominal width of 20 ns. These clocks, called the 62 ns clocks, are distributed through inverters to their associated logic modules. However, because of its heavy fanout, the M8563 logic module requires a discrete component clock driver.

NOTE

The internal clocks (125 ns and 62 ns) are labeled assuming an external clock rate of 8 MHz. The external clock rate is system dependent, however, and is a direct function of the basic machine (CPU) clock.

By using the 62 ns clock, those functions that are not dependent on the 125 ns clock can be executed with a subsequent saving in delay time. For example, when KBUS ADR ACK is returned from the storage module it is gated with the selected port function to supply the set requirement to the internal ACKN flip-flop, which is synced by a 62 ns clock. Thus, SBus ACKN is generated after a synchronizing delay of half an external clock period, rather than a full clock period.

2.6 MEMORY REQUEST OVERVIEW

The following subsections provide an overview of the memory request cycle operations in terms of a basic one-word request. Included in the description are the major functions implemented.

2.6.1 Memory Cycle Initiation

A read restore memory cycle is initiated when the MBox asserts a starting address (SBUS ADR 14–35), the required word request lines (SBUS RQ 0–3), cycle type (SBUS RD RQ), and start signal (SBUS START A or SBUS START B) on the SBus (Figure 2-5). The START signal set depends on which portion of the MBox clock (Phase A or B) synchronized cycle initiation. The address lines are separated at the DMA SBus interface. The 20 high-order bits (SBUS ADR 14–33) are fed into the address buffer register, where they are stored for the remainder of the cycle request. From the buffer register, the address is coupled to the logic level line translators of the Data Path/Control to the corresponding lines on all KBuses. The two low-order bits (DTR1 ADR 34–35) are applied to the address counter in the Address/Bus selection. An additional address line path is directed to the error logic.

START A B is applied to the start sequence flip-flops in the Request/Acknowledge control to generate the cycle start function (DMC4 CYC START). SBUS RD RQ/WR RQ and SBUS RQ 0–3 are also applied to the Request/Acknowledge control and locked up in the request latches (DMC4 REQ 0–3 and DMC4 RD/DMC4 WR, respectively). REQ 0–3, as well as ADR 34–35, are also applied to the address counter for bus selection and bus address determination.

The request lines, as with address lines, are also applied to the Error Detection control. The Error Detection control checks the address, request, and data lines for odd parity. Should a data parity error be detected, the cumulative error line (SBUS ERROR) is set and transferred to the MBox. In the case of an address parity error, an associated flag is set and also transferred to the MBox. In addition, the cycle request to the storage module is inhibited and the current address is strobed into the request error register. Should the MBox require exact error status, it must execute a diagnostic function as described in Subsection 2.9.2.

The operational bus mode, word request lines, and reference address determine bus port selection as follows:

a. One Bus Mode: With an even or odd address, all buses (K0–K3) are enabled.

b. Two Bus Mode: With an even address, the two even buses (K0 and K2) are enabled; with an odd address, the two odd ports (K1 and K3) are enabled.

c. Four Bus Mode: The two low-order bits of the starting address enable the corresponding numbered bus (e.g., ADR 34–35 = 01, bus K1 is enabled).

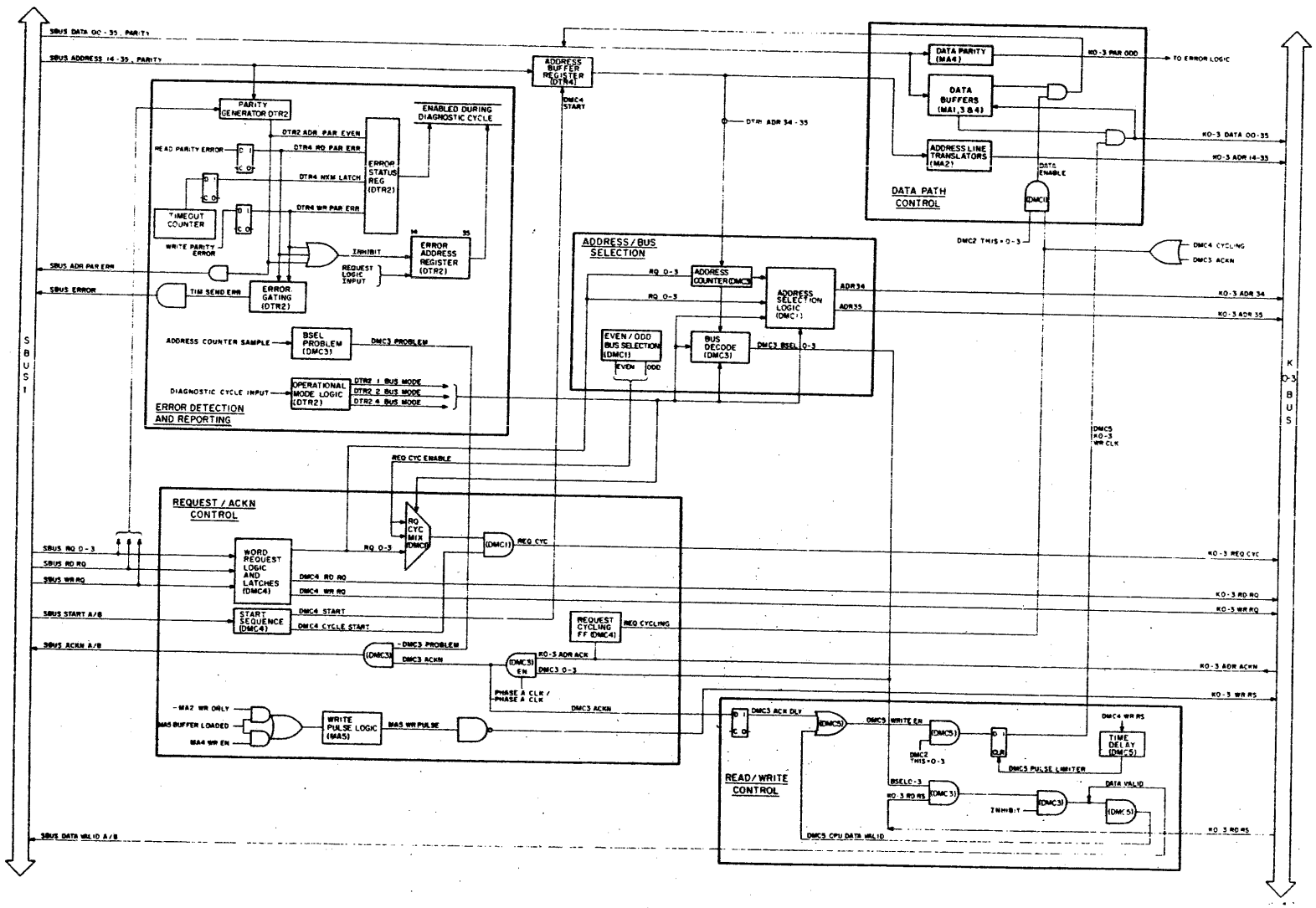


Figure 2-5 Functional Block Diagram

When the start sequence is complete (DMC4 CYC START set), the selected port REQ CYC line is enabled from the Request/Acknowledge control to the port-associated KBus. If the storage module has no higher priority requests and is ready to start another cycle, it responds to the DMA with Address Acknowledge (KBUS 0-3 ADR ACK). ADR ACK is returned over the selected KBus to the Request/Acknowledge control. Here it is gated with the selected port signal (BSEL = 0-3) from the Address Bus selection and if not to be inhibited sets the internal acknowledge function (BSEL ACKN). BSEL ACKN is then gated onto the SBus as SBUS ACKN A/B, depending on whether it was synchronized with a Phase A or B clock, and transferred to the MBox indicating storage module acceptance of the request.

A clear write memory cycle is initiated when the MBox asserts SBUS ADR 14-35, SBUS RQ 0-3, SBUS WR RQ, and SBUS START A/B. In addition, the MBox places the write data on the data lines (SBUS D00-35). As in a read cycle, the request components generate the address and bus selection functions including the cycle request function, RQ CYC. However, during a write request, REQ CYC is not coupled to the selected KBus until the appropriate DMA data buffer is loaded from the SBus, thus inhibiting a cycle request to the storage module. A data buffer write clock is then generated, strobing the SBus data into the buffer and setting the buffer loaded function. With the buffer loaded, RQ CYC is enabled to the KBus and connected storage module. As in the read operation, if the addressed storage module has no higher priority it generates ADR ACK, which is returned to the MBox as SBUS ACKN A/B.

2.6.2 Read/Restore Cycle

If a read/restore cycle is requested, the DMA issues KBUS N ADR 14-35, KBUS N RD RQ, and KBUS N REQ CYC over the KBus (Figure 2-5). The storage module responds with KBUS N ADR ACK. The selected storage module reads the data out of the addressed location placing it in its data buffer latches. As described in Subsection 2.6.1, ADR ACK is transferred to the MBox as SBUS ACKN A/B indicating request acceptance.

The KBus data is transferred and direct sets the corresponding KBus data buffer flip-flops in the Data Path/Control. KBUS N ADR ACK generates DMC3 ACKN, which is applied to the Data Path/Control where it is gated with DMC4 RD and the latched bus selection function (DMC2 THIS = 0-3) to generate the appropriate data buffer enabling function (K N DATA EN). DATA EN enables the SBus drivers gating the data buffer outputs to the SBus. A delayed acknowledge function, DMC3 ACK DLY, sets the cycling flip-flop (DMC4 CYCLING) in the Request/ Acknowledge control, indicating the DMA is processing a request.

In the meantime, the selected storage module has sent the read restart signal (KBUS N RD RS). To increase timing efficiency for those storage modules which return DATA WARNING prior to RD RS, DATA WARNING is used to initiate the internal DMA restart sequence. When the subsequent RD RS is coupled into the DMA it is logically ignored. At this point (reception of DATA WARNING/RD RS) the storage module disconnects from the KBus to restore the data in its latches to the addressed location. KBUS N RD RS is gated with DMC3 BSEL 0-3 in the Read/Write control and if not to be inhibited sets the internal restart function (DMC3 RD RS). If DMC3 RD RS is not to be inhibited, it is gated onto the SBus as SBUS DATA VALID A/B after synchronization with a Phase A or B clock. DMC3 RD RS INH sets the completed operation flip-flop (DMC4 COMPLETE), which essentially terminates the read operation.

2.6.3 Clear/Write Cycle

If a clear/write cycle is requested, the DMA issues KBUS 0-3 ADR 14-35 over the KBus (Figure 2-5). However, note that as described in Subsection 2.6.1 KBUS 0-3 RQ CYC (and KBUS 0-3 WR RQ as well) is not issued over the KBus until the appropriate DMA data buffer is loaded.

During the start sequence a delayed function (DMC4 START DLY1) generates the write enable function (DMC5 WRITE EN). WRITE EN is gated with the latched bus selection function (DMC2 THIS = 0–3) from the Request/Acknowledge control to set the appropriate data buffer write clock flip-flop (DMC5 KN WR CLK). WR CLK strobes the write data from the SBus receivers into the selected data buffer as well as generates the load function (MA5 BUFFER LOADED). BUFFER LOADED then enables RQ CYC (and WR RQ) to the KBus.

If the storage module can accept the request, it reads the data out of the addressed memory location, discards it, and responds with KBUS N ADR ACK. In addition to performing identical functions as in the read cycle, KBUS N ADR ACK enables the write function (MA5 WREN). WREN is ANDed with BUFFER LOADED which, after a delay, generates the write pulse (MA5 WR PULSE). WR PULSE then enables the DMA data buffer output to the KBus and subsequently into the storage module input buffer. WR PULSE is also gated onto the KBus as KBUS N WR RS. On receiving WR RS the storage module disconnects from the KBus to write the data in its input buffer into the addressed location.

In the meantime, WRITE EN initiates a cascaded restart flip-flop sequence in the Request/Acknowledge control to generate the internal write restart function (DMC4 WR RS). DMC4 WR RS drives a 75 ns delay line (tapped at 25 ns) whose output (DMC5 PULSE LIMITER) limits (except at full processor clock speed) the WR CLK pulse width by direct clearing the write clock flip-flop. The write operation is essentially terminated when the bus is not busy (-MA5 KN BUSY).

2.6.4 Read/Modify/Write Cycle

The read/modify/write cycle is only executed for one-word requests. If this type cycle is requested the DMA issues KBUS 0–3 ADR 14–35, KBUS 0–3 RD RQ and WR RQ, and REQ CYC over the KBus (Figure 2-5). If the storage module can accept the request, it reads the data out of the addressed location, places it into its data buffer latches, and responds with KBUS 0–3 ADR ACK. As described in Subsection 2.6.1, ADR ACK is transferred to the MBox as SBUS ACK A/B, indicating a request acceptance. The DMA then executes a normal read cycle as described in Subsection 2.6.2. After sending DATA WARNING/RD RS to the DMA (and in turn DATA VALID to the MBox), the storage module pauses while the data is modified by the central processor. After modification, the MBox places the data on the SBus and asserts its DATA VALID line.

DATA VALID initiates the write portion of the cycle in the DMA by generating DMC5 WRITE EN in the Read/Write control. The subsequent data buffer enabling and write functions, as described in Subsection 2.6.3, transfer the modified data from the SBus receivers through the DMA data buffer to the storage module input buffer. After receiving WR RS from the DMA, the storage module disconnects from the KBus to write the data in its input buffer back into the original location. The write portion is essentially terminated when the bus is not busy (-MA5 KN BUSY).

2.7 WORD SELECTION

Each memory reference over the SBus is made to a four-word block in available memory. The four-word block is referred to as a quadword, where each quadword contains Words 0, 1, 2, and 3. Using a combination of SBus address and word request lines (SBUS RQ 0–3), a quadword may be accessed with one MBox memory reference. There is a direct correlation between a particular word and its word request line (i.e., Word 0 is associated with SBUS RQ 0, Word 1 is associated with SBUS RQ 1, etc.). The address lines provide the starting address and the word request lines specify the words within the quadword to be accessed.

SBUS ADR 14–33 represents the most significant part of the quadword address, i.e., it is equal to the overall quadword address including module selection. SBUS ADR 34–35 represents the starting address and points to the first word in the quadword to be accessed. The words are accessed in ascending order modulo 4, beginning with the word specified by the starting address (word access order = 0, 1, 2, 3 or 1, 2, 3, 0 or 2, 3, 0, 1 or 3, 0, 1, 2).

Since the address of each word within the quadword differs only in the value of the two least significant bits, each RQ line corresponds to one of the word addresses within the quadword. Thus, for a quadword request the MBox places the required address on SBUS ADR 14–35 and asserts SBUS RQ 0–3. The combination of SBUS ADR 34–35 and its corresponding request line determines the first word accessed. The other three request lines enable the DMA address logic to control the KBus address lines and access the remaining three words of the quadword.

Although one to four words can be simultaneously requested over the SBus, its data lines are only one-word wide. This requires that words (as well as control information) be transferred serially for multi-word requests.

Since memory references must be sequential, the word request lines provide the additional flexibility to skip addresses not required. For example, a memory reference may be issued requiring three words at addresses 00, 01, and 03. In this case SBUS RQ 0, 1, and 3 are enabled. The DMA addressing logic will access addresses 00, 01, and 03 skipping 02. Thus, time is not wasted requesting an access to an unwanted memory address.

Requests of two or three words usually occur during cache writeback cycles, where only some of the words in a quadword have been modified since they were last read. If the SBus request line associated with the starting address is not true, an error has occurred.

2.8 OPERATIONAL MODES

Addressing and word accessing is implemented by the three DMA operational modes (4 bus, 2 bus, and 1 bus modes) and their associated storage module interleave modes:

a. 4 bus mode is the normal operating mode and provides the capability to access up to four consecutive words (quadword) over sequential bus ports with one MBox memory reference. All storage modules must be set for 4-way interleaving. Storage modules on KBus N must respond to address bits 34–35 = N.

b. 2 bus mode provides a degraded capability to access up to four consecutive words (with one MBox memory reference), alternating between the two odd ports (KBus 1 and 3) and the two even ports (KBus 0 and 2). In this mode, accessing requires two KBus cycles per port. All storage modules must be set for 2-way or 4-way interleaving. Storage modules on KBuses 0 and 2 must respond to even addresses; storage modules on KBuses 1 and 3 must respond to odd addresses.

c. 1 bus mode (noninterleaved) provides a further degraded capability to access from one to four words over any bus port with one MBox memory reference. This mode requires a complete KBus cycle per port for each word requested with the storage module address and interleave switches set in any configuration.

The DMA operational mode is set when the MBox issues a Function 0 Diagnostic Cycle (Subsection 2.9.2). The appropriate bit configuration is applied to the Error Detection control, where it is decoded and subsequently sets the required control flip-flops. However, to operate in 4 or 2 bus modes requires an even number of storage modules with equal storage capacities connected to each DMA bus port.

2.8.1 4 Bus Mode Operation

With 4 bus mode (and 4-way interleaving) the contents of the storage modules are arranged sequentially in quadword sets (one word/storage module) such that each word is accessed over its corresponding KBus. As shown in Figure 2-6, a quadword is distributed across four storage modules with all Word 0s accessed over KBus 0, all Word 1s accessed over KBus 1, etc. To facilitate correct module and word selection, the two low-order module selection bits (20–21) are interchanged with the two low-order word selection bits (34–35) through the interleave switches on the storage modules.

ADDRESS BIT POSITIONS	INITIAL CONFIGURATION 17 18 19 " 34 35	PORT SELECTED	INTERCHANGED CONFIGURATION 17 18 19 " 34 35	RESULTANT SELECTION MODULE	RESULTANT SELECTION LOCATION	WORD RETRIEVAL ORDER
QUAD-WORD REQUEST START ADDRESS 00						
START ADDRESS	0 0 0 0 0	EVEN-KBUS 0	0 0 0 0 0	0 0 0	0 0	WORD 0
RQ1 INCREMENT	0 0 0 0 1	ODD-KBUS 1	0 0 1 0 0	0 0 1	0 0	WORD 1
RQ2 INCREMENT	0 0 0 1 0	EVEN-KBUS 2	0 1 0 0 0	0 1 0	0 0	WORD 2
RQ3 INCREMENT	0 0 0 1 1	ODD-KBUS 3	0 1 1 0 0	0 1 1	0 0	WORD 3
QUAD-WORD REQUEST START ADDRESS 01						
START ADDRESS	0 0 1 0 1	ODD-KBUS 1	0 0 1 0 1	0 0 1	0 1	WORD 1
RQ2 INCREMENT	0 0 1 1 0	EVEN-KBUS 2	0 1 0 0 1	0 1 0	0 1	WORD 2
RQ3 INCREMENT	0 0 0 1 1	ODD-KBUS 3	0 1 1 0 1	0 1 1	0 1	WORD 3
RQ0 INCREMENT	0 0 1 0 0	EVEN-KBUS 0	0 0 0 0 1	0 0 0	0 1	WORD 0
3 WORD REQUEST START ADDRESS 00 OMIT WORD 02						
START ADDRESS	0 1 0 0 0	EVEN-KBUS 0	0 0 0 1 0	0 0 0	1 0	WORD 0
RQ1 INCREMENT	0 1 0 0 1	ODD-KBUS 1	0 0 1 1 0	0 0 1	1 0	WORD 1
RQ3 INCREMENT	0 1 0 1 1	ODD-KBUS 3	0 1 1 1 0	0 1 1	1 0	WORD 3

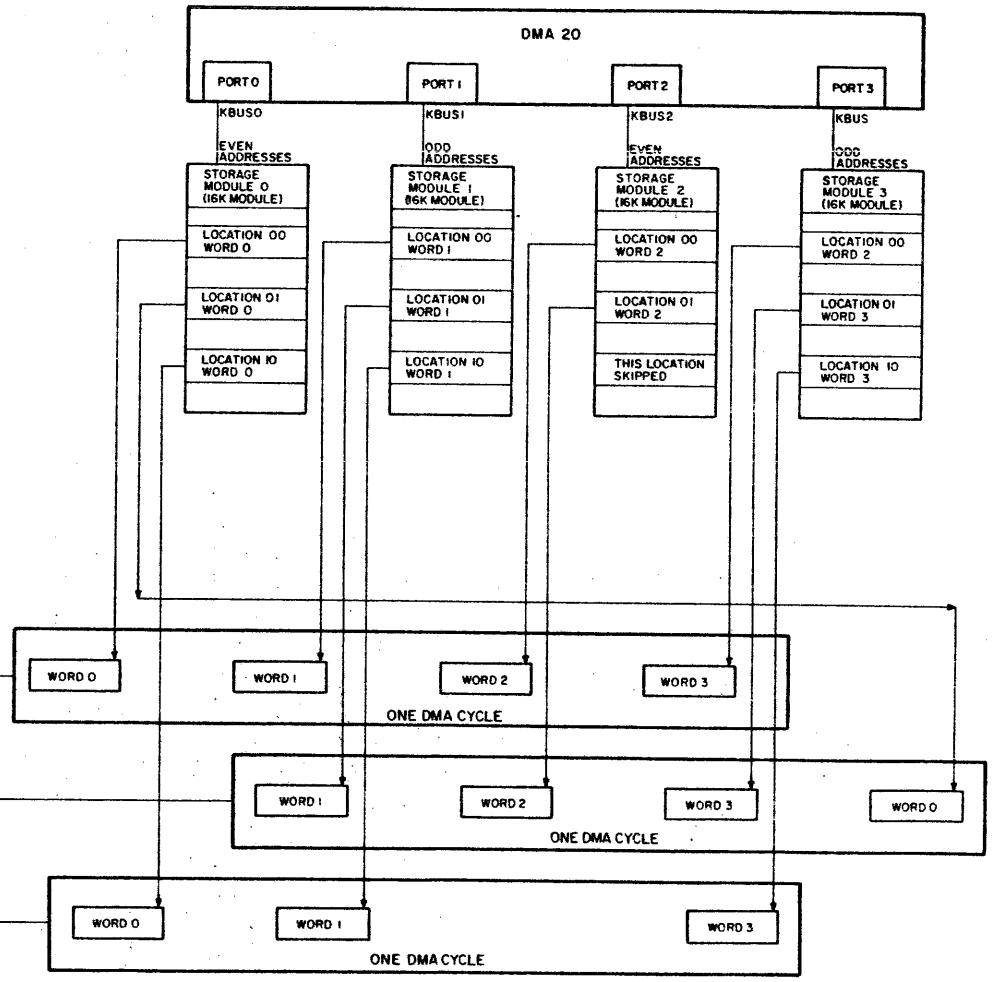


Figure 2-6 4 Bus Mode Operation

As shown in the table of Figure 2-6, this interchange of module and word selection bits has the effect of addressing the identical location in each of the four selected storage modules. The module accessed first is determined by the starting address; the number of words accessed is determined by the number of request lines enabled. If a quadword request were required all request lines (RQ 0–3) would be enabled. However, as in the previous example, if address 02 were to be skipped RQ 0, 1, and 3 would be enabled and RQ 2 disabled.

The request lines, through the word request logic, also provide an incrementing value to the address counter in the bus selection logic. The address counter is initialized to a value equal to the two low-order address bits and incremented by a count of 1, 2, or 3 after each ADR ACK is returned to the MBox. The counter output is gated with the operational mode to provide decoding for bus (port) selection. The bus selection logic is such that storage modules on KBus N respond when address bits 34–35 = N. For example, if bits 34–35 = 00, KBus 0 port is enabled; likewise, if bits 34–35 = 01, KBus 1 port is enabled. In conjunction with the bus selection logic, the address selection logic directs the low-order address bits (34–35) to the appropriate bus as determined by the current operational mode. The bus enabling sequence and subsequent word access order are sequential from the starting address in ascending order, modulo 4 (i.e., 0, 1, 2, 3; 2, 3, 0, 1, etc.)

The appropriate latches in the word request logic are initially set to reflect the SBus request lines enabled. Each request line latch is cleared when its associated request is performed. When the word request logic output is cleared (-DMC4 ANY RQ), it indicates that all requested words have been transferred and all KBus RQ CYC lines have been disabled. In addition, the cleared word request logic output, together with the request completed function (DMC4 COMPLETE) or all buses not busy (-MA5 KN BUSY), effectively terminates a DMA cycle.

Module selection and access order, as a function of address and request line control, are tabulated in Figure 2-6. Two additional examples are presented to demonstrate module selection and access/retrieval order using a different starting address and module select code, as well as selection and retrieval during a three-word request.

For a quadword read operation starting at address 00 (SBUS ADR 34–35 = 00), the MBox raises the appropriate address and data lines, and enables RQ 0–3 as well as the other required SBus lines (Subsection 2.6.1). Word request latches are set; the address counter is initialized to 00, which after decoding enables the KBus 0 response functions. Since this is a quadword request the Request/Acknowledge control issues RQ CYC (FAST, SLOW, IMM) over all KBuses. In the meantime, the address selection logic, in conjunction with the latched word request functions, has arranged the low-order KBus address bits to correspond to the requested words (i.e., KBUS ADR 34–35 = 00 for Word 0, KBUS ADR 34–35 = 01 for Word 1, etc.).

As described in Subsection 2.6.1, the selected modules respond with ADR ACK, which will be transferred to the MBox indicating request acceptance. It is not necessary that the storage modules respond with the ACKN and data in the order requested. The ACKN responses from the storage modules are latched into their respective KBus synchronizers; the incoming data is direct set into its corresponding data buffer. The acknowledge and data are thus stored until the bus selection logic selects a particular KBus ACKN and data for transfer to the MBox. Since the starting address was 00, the ADR ACK from KBus 0 is initially selected and transferred to the SBus as SBUS ACKN A/B depending on the synchronizing clock (Phase A/B).

The initial ADR ACK generates DMC3 ACK DLY, which is gated with the appropriate latched bus selection function (DMC2 THIS = 0–3) to generate enabling functions DMC1 KN DATA EN (in this case K0 DATA EN). Each DATA EN sequentially enables the SBus drivers gating the data buffer outputs to the SBus in a word-serial fashion. The subsequent KBUS DATA WARNING/RD RS signals from the storage modules are sent to the DMA and coupled into their respective synchronizers. When the bus selection logic selects a particular KBus (in this case KBus 0), the associated RD RS is

synced with a Phase A or B clock, generating appropriate DATA VALID A or B functions to the MBox.

At RD RS DLY time the address counter is effectively incremented (changed to the next address) on a 62 ns clock by the enabled RQ 1 line. The word request latch associated with RQ 0 is cleared since the word requested by SBUS RQ 0 has been transferred. Since the address counter is now 01, KBus 1 is selected and the corresponding ADR ACK and data buffer output (Word 1) are gated and followed by RD RS through the DMA to the MBox on Phase A or B clocks. The address counter is again incremented on each of two 62 ns clocks by RQ 2 and 3, enabling and transferring the KBus 2 and 3 ADR ACKs, data words, and RD RS. On the transfer of Word 3, the DMC4 COMPLETE flip-flop is again set and together with the cleared word request logic output essentially terminates the DMA read operation.

For a quadword write operation starting at address 00, the MBox raises the appropriate address and data lines, and enables SBUS RQ 0–3 as well as the other required SBus lines (Subsection 2.6). All word request latches are set; the address counter is initialized to 00, which after decoding enables the KBus 0 response functions.

As described in Subsection 2.6.3, RQ CYC is not coupled to the KBuses until the first word (Word 0) is loaded into its associated data buffer. At this point, the K0 buffer write clock (DMC5 K0 WR CLK) is generated, strobing the SBus data into the KBus 0 data buffer and setting the buffer loaded function (MA5 BUFFER LOADED). At BUFFER LOADED time, RQ CYC is enabled to all KBuses. As in the read operation, the address selection logic, in conjunction with the request lines, arranges the low-order KBus address bits with the requested words and their corresponding KBuses.

The ACKN responses from the storage modules are latched into their respective KBus synchronizers. On receiving KBus 0 ADR ACK the DMA generates write enable (WR EN) which, after a delay, generates the write pulse (WR PULSE). K0 WR PULSE enables Word 0 from the output of the data buffer to the KBus data lines. In addition, WR PULSE is sent over the KBus as a write restart (WR RS) function. The storage module completes its cycle by writing the data in its input buffer into the addressed location.

Meanwhile, KBUS 0 ADR ACK is transferred to the MBox indicating request acceptance. At this time the MBox places the new data word (Word 1) on the SBus data lines. At DMC3 ACK DLY time the address counter is effectively incremented on a 62 ns clock by the enabled RQ1 line. ACK DLY also sets the cycling flip-flop (DMC4 CYCLING) indicating the DMA is processing requests. With DMC4 CYCLING set, an equivalent K1 ADR ACK is generated in the ACKN logic and gated through the DMA to the MBox. The delayed ACKN (DMC3 ACK DLY) generates the write enable and write clock functions in the Read/Write control to strobe the incoming SBus data (Word 1) into the K1 data buffer. On generating WR PULSE and KBUS1 WR RS the data is transferred from the K1 data buffer to KBus 1 where the storage module writes the data into the addressed location.

At each ACK DLY time the address counter is incremented on 62 ns clocks by the enabled RQ2 and 3 lines. The acknowledge sequence and write operations over KBus 2 and KBus 3 are performed in an identical manner as described for Word 1. On the transfer of Word 3 to the KBus, the DMC4 COMPLETE flip-flop is again set. The write operation is essentially terminated when all buses are not busy (-MA5 K0–3 BUSY), resetting DMC4 CYCLING.

2.8.2 2 Bus Mode Operation

As shown in Figure 2-7 the contents of the storage modules are arranged such that for the first 16K of core even words are stored in even locations within even numbered storage modules, and odd words are stored in even locations within the odd numbered storage modules. For the second 16K (when address bit 19 is incremented to one) even words are stored in odd locations within the even storage modules, and odd words are stored in odd locations within the odd storage modules.

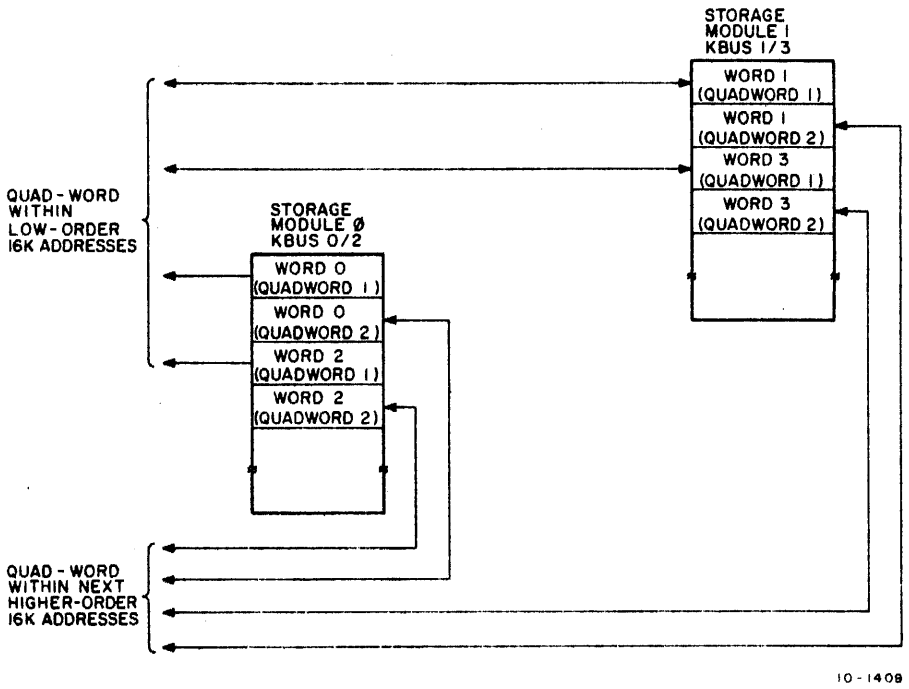


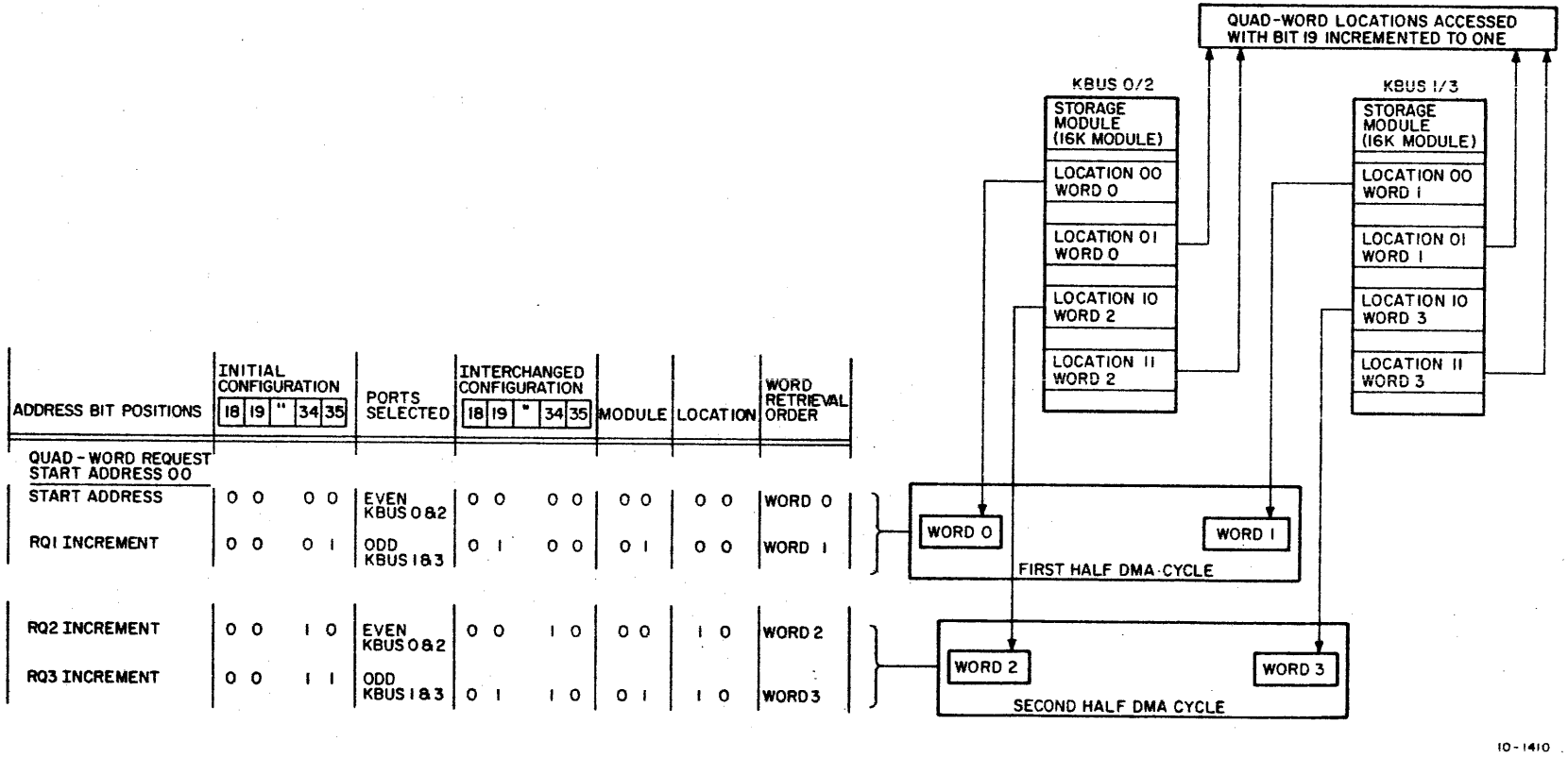
Figure 2-7 Storage Module Content Interleave 2 Mode

Depending on the setting of the storage module interleave switches either one or two low-order module select bits (bit positions 21 or 20 and 21) are interchanged with one or two low-order word select bits (bit positions 35 or 34 and 35). This interchange of module and selection bits has the effect of addressing the identical location within each module of the interleaved pair. The module accessed first is determined by the starting address; the number of words requested and accessed is determined by the word request lines enabled. In a quadword request, all word request lines (RQ 0–3) would be enabled. However, as in the example in Subsection 2.7, if address 02 were to be skipped, RQ0, 1, and 3 would be enabled and RQ2 would be disabled.

As in 4 bus mode, the word request lines set their associated latch in the word request logic and an initializing value to the address counter. The address counter is initialized to a value equal to the two low-order address bits.

The output of the address counter is gated with the current operational mode to provide decoding for bus selection. The bus selection logic is such that storage modules on KBus 0 and 2 respond when address bits 34–35 are even. Likewise, storage modules on KBus 1 and 3 respond when bits 34–35 are odd. Module selection and access order as a function of address and request line control are tabulated in Figure 2-8.

For a quadword read operation starting at address 00, the MBox raises the appropriate address and data lines and enables SBUS RQ 0–3 as well as the other required SBus lines (Subsection 2.6.1). SBUS RQ 0–3 sets all word request latches. The address counter is initialized to 00, which after decoding enables the KBus 0 and 2 response functions. The Request/Acknowledge control issues RQ CYC and RD RQ over both selected KBuses. (For discussion purposes, a response is expected on KBus 0.) After receiving KBUS 0 ADR ACK from the responding storage module, DMC3 ACKN generates SBUS ACKN A/B to the MBox. In addition, ADR ACK disables the even portion of the request cycle mixers and subsequently the KBUS 0 and 2 RQ CYC. DMC3 ACKN is also gated with the latched bus selection function (DMC2 THIS = 0) to generate DMC1 K0 DATA EN. DATA EN enables the SBus drivers gating the data buffer outputs to the SBus.



ADDRESS BIT POSITIONS	INITIAL CONFIGURATION 18 19 " 34 35	PORTS SELECTED	INTERCHANGED CONFIGURATION 18 19 " 34 35	MODULE	LOCATION	WORD RETRIEVAL ORDER
QUAD-WORD REQUEST START ADDRESS 00						
START ADDRESS	0 0 0 0	EVEN KBUS 0&2	0 0 0 0	0 0	0 0	WORD 0
RQ1 INCREMENT	0 0 0 1	ODD KBUS 1&3	0 1 0 0	0 1	0 0	WORD 1
RQ2 INCREMENT	0 0 1 0	EVEN KBUS 0&2	0 0 1 0	0 0	1 0	WORD 2
RQ3 INCREMENT	0 0 1 1	ODD KBUS 1&3	0 1 1 0	0 1	1 0	WORD 3

Figure 2-8 2 Bus Mode Operation

The subsequent DATA WARNING/RD RS generates the appropriate SBUS DATA VALID A or B functions to the MBox. In addition, at DMC3 RD RS DLY time the address counter is effectively incremented by the enabled RQ1 line. Since the address counter is now 01, KBus 1 and 3 are selected and read requests are issued over these buses. KBUS 1 ADR ACK generates K1 DATA EN enabling the data buffer outputs to the SBus. RD RS is synced with Phase A or B clock enabling DATA VALID A or B to the MBox, and completing the first half of the DMA cycle.

The address counter is incremented to 10 on the Word 1 DMC3 RD RS DLY, again selecting KBUS 0 and 2. The second half of the DMA cycle is initiated for the third and fourth words of the quadword request. When the DMA receives the fourth ADR ACK, the word request latches have been cleared and all KBus RQ CYC lines disabled. The subsequent Word 3 RD RS generates the fourth REQ COMPLETE function and together with the request logic output resets REQ CYCLING, terminating the second half of the DMA cycle and completing the quadword request. Note that at each REQ COMPLETE a clear function is generated, which clears the associated data buffer.

For a quadword write operation starting at address 00, the MBox raises the appropriate data and address lines and enables SBUS RQ 0–3 as well as the other required SBus lines (Subsection 2.6.1). SBUS RQ 0–3 sets all word request latches. The address counter is initialized to 00, which after decoding selects the KBus 0 and 2 response functions. However, as in 4 bus mode, RQ CYC is not coupled to the KBuses until the first SBus data word (Word 0) is loaded into its appropriate data buffer. At this point WR CLK is generated, strobing Word 0 into the K0 data buffer, setting BUFFER LOADED, and initiating RQ CYC over the even buses.

As described in Subsection 2.6.3, the storage module responds with ADR ACK, which is transferred to the MBox on Phase A or B indicating request acceptance. On reception of ADR ACK the MBox places the Word 1 data on the SBus. In the meantime, KBUS ADR ACK generates the write enable function (WR EN) which, after a delay, generates WR PULSE gating Word 0 onto the KBus.

The delayed ADR ACK function causes incrementing of the address counter by the enabled RQ1 line, which effectively initiates selection of the KBus 1 and 3 functions. A write request is then issued over the odd KBuses. At the same time, in the word request logic, the RQ0 latch is cleared. Since the address counter is now 01 and the odd bus is not busy (-DMC4 ODD BUSY), an equivalent K1 ADR ACK is gated through the DMA to the MBox. The WR CLK and WR PULSE are generated, and the Word 1 data is coupled through the K1 data buffer to the KBus. The subsequent WR RS function causes the storage module to disconnect from the KBus to complete its write cycle.

The address counter is incremented and the second half of the DMA cycle is initiated to write the third and fourth words of the quadword request. When the DMA receives the fourth ADR ACK all word request latches have been cleared. The subsequent WR RS generates the fourth DMC4 COMPLETE. The write operation is essentially terminated when all buses are not busy (-MA5 K0–3 BUSY), resetting DMC4 CYCLING.

2.8.3 1 Bus Mode Operation

With the DMA configured for 1 bus mode, the storage module contents are arranged sequentially throughout the modules. As in the other operational modes, the number of words requested and accessed is determined by the word request lines enabled. However, in this mode a complete word and KBus cycle are required for each location accessed (Figure 2-9).

As in the other operational modes, the request lines provide an incrementing value to the address counter and set their associated latch in the word request logic. The address counter is initialized to a value equal to the two low-order address bits. However, unlike 2 and 4 bus modes, the output of the address counter has no effect on bus port selection. The interleave mode functions are applied to the bus decode logic enabling all bus selection functions (BSEL 0–3) throughout the quadword request.

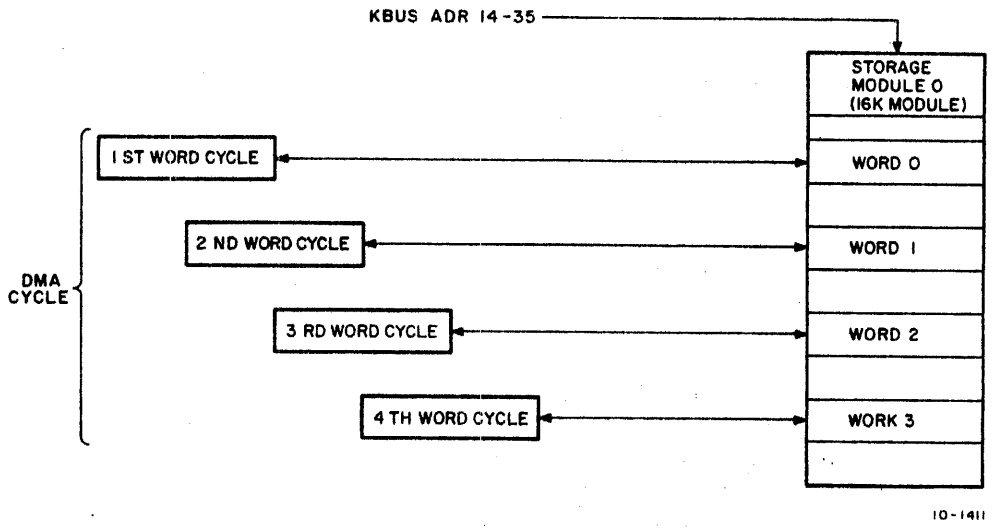


Figure 2-9 1 Bus Mode Access Order

For a quadword read operation starting at address 00, the MBox raises the appropriate address and data lines and enables SBUS RQ 0–3 as well as the other required SBus lines (Subsection 2.6.1). SBUS RQ 0–3 set all word request latches, and the address counter is initialized to 00. The Request/Acknowledge control issues REQ CYC and RD RQ over all buses. (For this discussion, a response is expected only from a storage module on KBus 0.) After receiving ADR ACK from the responding module, REQ CYCLING is set and together with the bus selection function generates K0–3 DATA EN. In the meantime, the Word 0 read data direct sets the corresponding bit flip-flops in the K0 data buffer. DATA EN then enables the SBus drivers, gating the K0 data buffer output to the SBus. The delayed ACK function clears the RQ0 latch in the word request logic.

The subsequent DATA WARNING/RD RS function generates the appropriate SBUS DATA VALID A/B to the MBox. At RD RS DLY time the address counter is effectively incremented by the enabled RQ 1 line. The request completed function is then generated indicating completion of the first word cycle. At the same time, the clear function direct clears the K0 data buffer.

At this point, with the request cycle mixers and gating enabled, a second cycle request is issued over each of the KBuses. Retrieval and transfer of Words 1–3 are executed in exactly the same manner as for Word 0. As Word 3 is retrieved its associated word request latch is cleared, disabling the request cycle mixers. The fourth DMC4 COMPLETE function is enabled and together with -DMC4 ANY RQ effectively terminates the quadword request, with the clear function direct clearing the K0 data buffer.

For a quadword write request starting at address 00, the MBox raises the appropriate address and data lines, and enables SBUS RQ 0–3 as well as the other required SBUS lines (Subsection 2.6.1). All word request latches are set; the address counter is initialized to 00 and in this mode selects all KBus response functions. As in the other bus modes, the first data word must be loaded into its respective data buffer before a cycle request is initiated. WR CLK strobes the Word 0 data into the K0 data buffer. The RQ CYC is then issued over all KBuses.

As in the previous read cycle description, a response is expected only from a storage module on KBus 0. On reception of K0 ADR ACK, the WR PULSE enables the K0 buffer data to the KBus. In the meantime, the address counter is incremented and the associated word request latch (RQ0) is cleared. The subsequent WR RS function causes the storage module to disconnect from the KBus to complete its write cycle. DMC4 COMPLETE is then generated terminating the first word cycle, with the clear function direct clearing the K0 data buffer.

At this point with the request cycle mixers and gating enabled, a second cycle request is issued over each of the KBuses. Transferring of Words 1–3 is executed in exactly the same manner as for Word 0. As Word 3 is transferred its associated word request latch is cleared disabling the request cycle mixers. As in the other modes the write operation is essentially terminated when all buses are not busy (-MA5 K0–3 BUSY), thus resetting DMC4 CYCLING.

2.9 DIAGNOSTIC CAPABILITIES

The overall diagnostic capabilities are provided by the DMA Error Detection control, and the exercising of diagnostic functions through execution of a diagnostic cycle from the MBox. In addition, the diagnostic cycle is used for communications between the MBox and DMA to control setup of the operational modes.

The error detection logic detects address parity and read and write data parity errors as well as NXM references. Each of the error types will set an appropriate flag in the status register. In addition, the logic will store the address of the last error type and the enabled word request lines in the request error register. If required, the error and status registers may be read by the software when it exercises one of its diagnostic functions as described in Subsection 2.9.2.

Generally the diagnostic cycles have the capability to retrieve the status and request error register contents, as well as determine memory system type. A diagnostic cycle can set and verify any of the three operational bus modes, and enable the execution of data transfers between the MBox and a selected DMA data buffer (without going through memory cores).

2.9.1 Error Detection Operation

As shown in Figure 2-5, the address, its parity, and the word request lines are applied to the parity generators in the error logic and checked for odd parity. Should an address parity error be detected, the even parity function (DTR2 ADR PAR EVEN) is set. In turn, DTR2 ADR PAR EVEN sets an error flip-flop (DTR2 PAR ADR) in the status register, as well as the SBus address error line (SBUS ADR PAR ERR).

The read/write data parity is checked in the Data Path/Control logic. The parity checking output (DMC6 PAR ODD) is gated with its complement. Should a parity error occur the read/write error function (DMC6 RD ERR/DMC6 WR ERR) is generated. Either error function sets its associated error flip-flop (DTR2 RD PAR ERR/DTR2 WR PAR ERR), in turn generating DMC6 SEND and SBUS ERROR. The error flip-flop also sets the appropriate error flag (DTR2 PAR RD/DTR2 PAR WR) in the status register. In addition, the error flip-flop output inhibits the error address clock flip-flop (DTR2 ADR CLK) retaining the current address and enabled request lines in the request error register.

The error logic also has the capability to detect NXM references. If an ACK is not received from an addressed storage module within a specified time interval after the first word is acknowledged, the NXM error function (DTR4 NXM) sets the NXM flag (DTR4 NXM LATCH) in the stauts register. ERR NXM also sets DMC6 SEND ERROR, which generates SUBS ERROR to the MBox. As with parity errors, an NXM inhibits DTR2 ADR CLK retaining the current address and enabled request lines in the request error register.

2.9.2 Diagnostic Cycle Operation

During the diagnostic cycle all communication, including addressing, between the MBox and DMA is performed over the SBus data lines. The overall diagnostic cycle consists of two havles: TO DMA and FROM DMA. During the TO DMA portion of the cycle, diagnostic conditions are set; during the FROM DMA portion those conditions are verified or in some cases error conditions retrieved. The general TO DMA field format is shown in Figure 2-10.

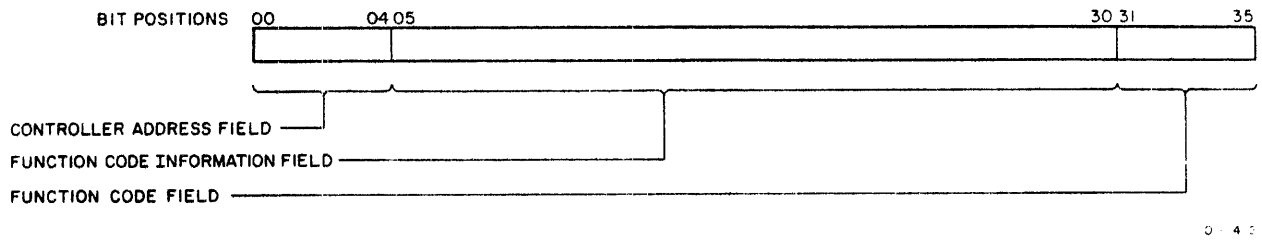


Figure 2-10 General Diagnostic Cycle Format

The diagnostic cycle is initiated and synced to the Phase A clock with a complete cycle requiring four Phase A clock periods. During the first half of the cycle the MBox sends out a DMA address, function code, and information specified by the function code. During the second half of the cycle the DMA sends information back to the MBox using the data lines as specified by the function code. As shown in Figure 2-11, the DMA strobes the SBus data lines on the third Phase A clock, and the MBox clocks the data lines on the fourth Phase A clock.

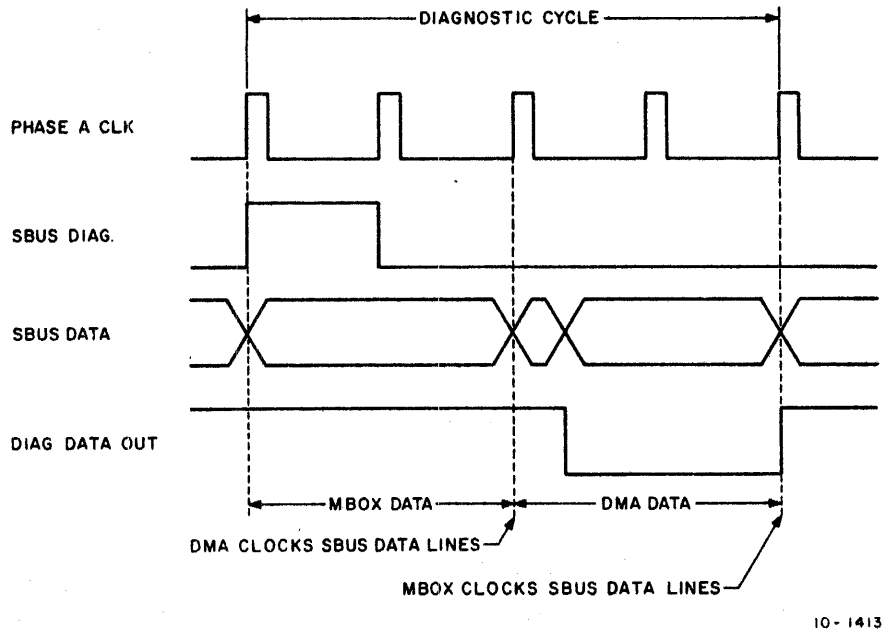


Figure 2-11 Diagnostic Cycle Timing

Two function codes are defined (Function 0 and Function 1) and their formats are described in Figures 2-12 and 2-13. SBUS DIAG effectively enables the diagnostic address decoder during the first half cycle and the transmission function for data retrieval during the second half cycle. Note that those bit positions, which set or clear conditions in the first half cycle, will contain the new data when retrieved during the second half cycle.

The following paragraphs functionally describe the impact each field has on the DMA. The paragraphs are keyed to supplement information in Figures 2-12 and 2-13.

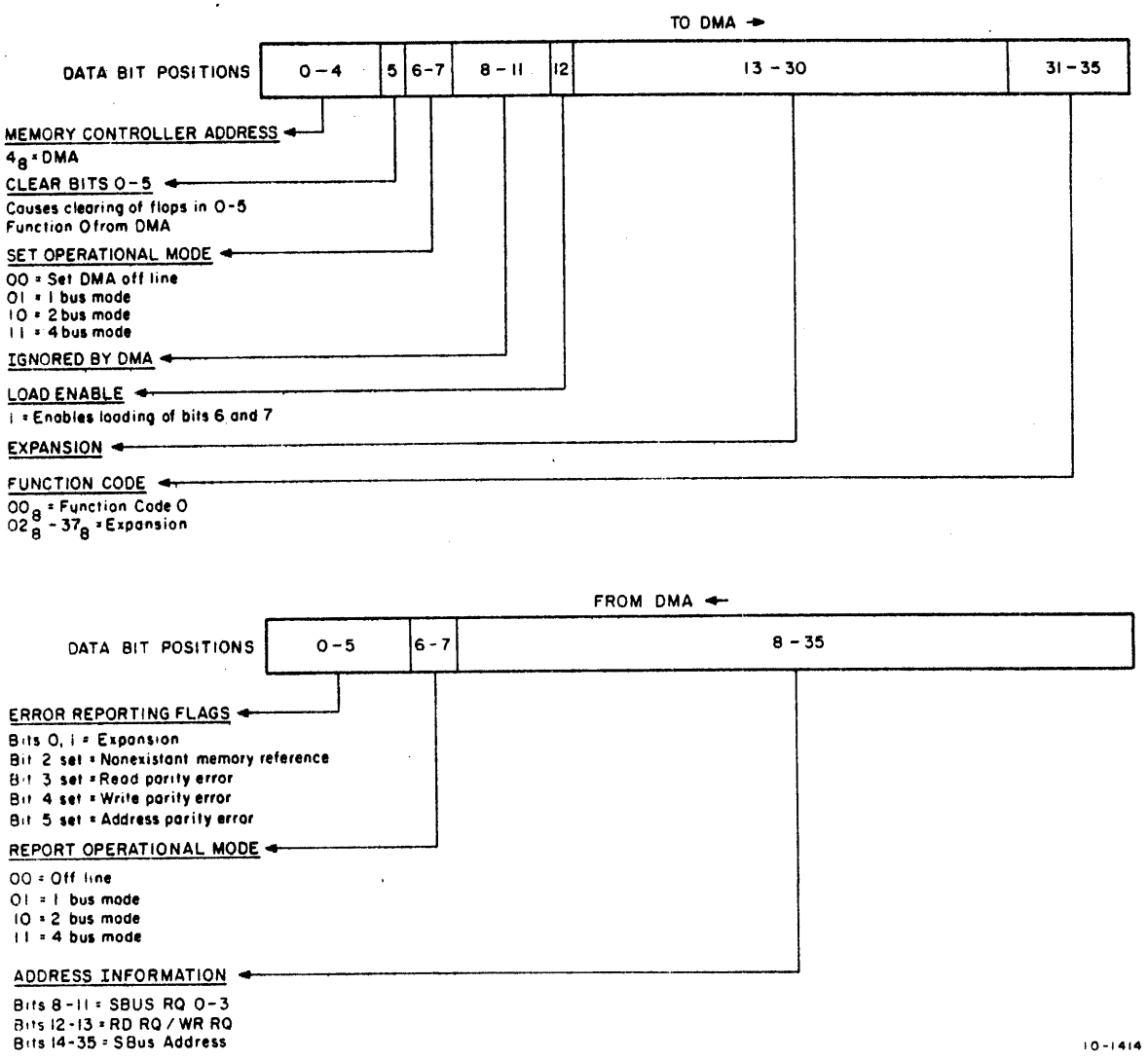


Figure 2-12 Function 0 Field Format Descriptions

2.9.2.1 Function 0 Operation to DMA (Figure 2-12) – Memory Controller Address. This field specifies the address of the controller to be exercised. The field is decoded in the address/function decoder of the error logic. In this case, the address field will decode as 4, specifying the DMA.

Clear Bit. This 1-bit field is applied to the error logic and, if true, generates the clearing function for the four flip-flops in the error status register.

Operational Mode. This 2-bit field, applied to the error logic, sets the DMA operational mode as determined by the code. With a code of 00 the DMA is set off line by resetting both bus mode flip-flops. The remaining three codes set/reset the flip-flops as required for the selected bus mode (Subsection 2.9).

Load Enable. This bit is applied to the error logic and, if true, enables gating of the field to the operational mode flip-flops. If the bit is false, bits 6 and 7 are ignored by the DMA.

Function Code. This 5-bit field specifies the function code; in this case, 00 = Function Code 0.

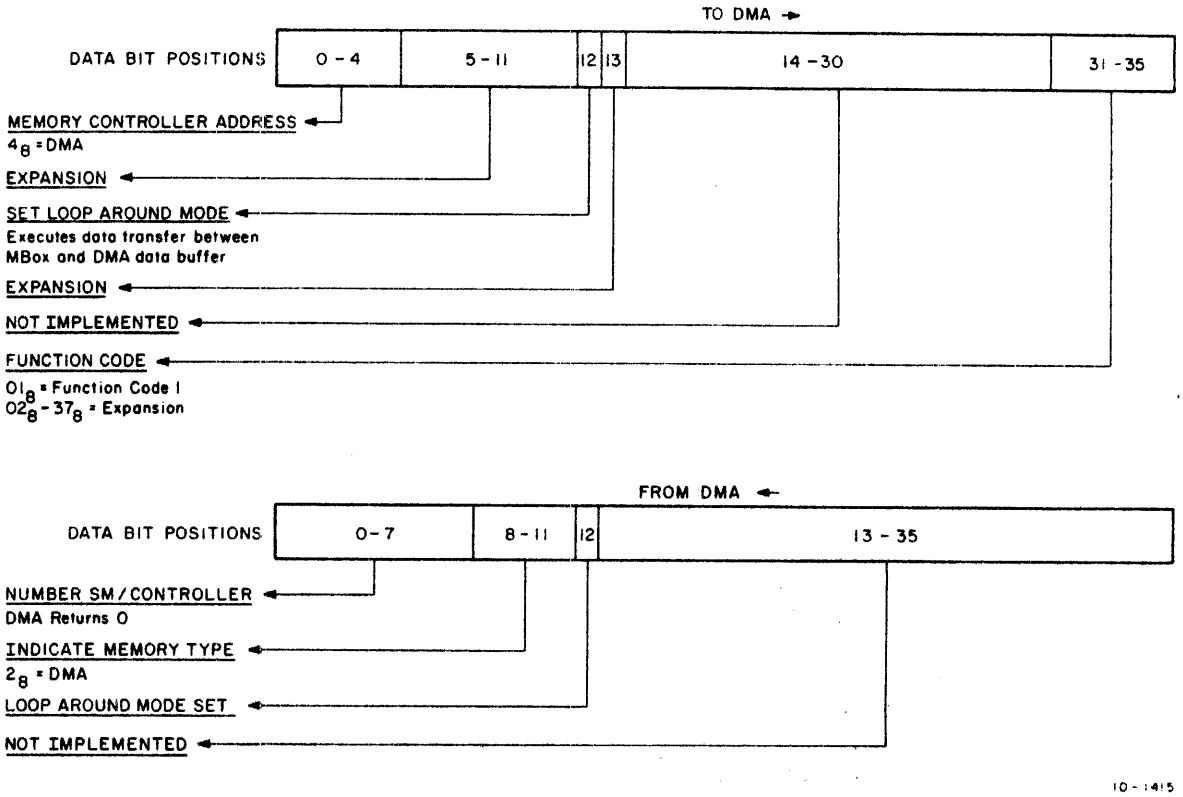


Figure 2-13 Function 1 Field Format Descriptions

2.9.2.2 Function 0 Operation from DMA (Figure 2-12) – Error Reporting Flags. This 6-bit field returns the error type from the status register in the error logic. The output of the status register is enabled to the SBus data lines by the error transmit function (ERR XMIT), which was generated on the first half cycle.

Report Operational Mode. This 2-bit field returns the current DMA operational mode. The condition of the error logic bus mode flip-flops is enabled to the data lines by ERR XMIT.

Address Information. This 22-bit field returns address information concerning the most recent memory cycle or the first parity error as specified in Figure 2-12. The outputs of the appropriate flip-flops (RD RQ, WR RQ, RQ 0–3, etc.) and the error address in the error logic are enabled to the SBus data lines by ERR XMIT.

Function Code. This 5-bit field is essentially the op code for the diagnostic function.

2.9.2.3 Function 1 Operation to DMA (Figure 2-13) – Memory Controller Address. This 5-bit field specifies the address of the controller to be exercised by the diagnostic cycle. The address is decoded in the address/function decoder of the error logic. In this case, the address field will decode as 4, specifying the DMA.

Loop Around Mode. With this bit true, the DMA is conditioned for a loop around mode operation that allows checking of the DMA-MBox data path.

Function Code. This 5-bit field is essentially the op code for the diagnostic function.

2.9.2.4 Function 1 Operation from DMA (Figure 2-13) – Number Storage Module/Controller. The DMA always returns zeros in this 8-bit field.

Indicate Memory Type. This 4-bit field indicates the type of memory connected. In this case, a bit configuration of 2^2 is returned, indicating a DMA.

Loop Around Mode. When true, this 1-bit field indicates that the DMA was conditioned for a loop around mode operation on the previous half cycle.

2.9.3 Loop Around Mode

Loop around mode is a diagnostic feature that allows the data path between the MBox and the DMA to be checked without reading/writing core. It is used mainly by the diagnostic programmer in isolating system failures. Loop around mode is set by a Function 1 diagnostic cycle with the DMA in 4 bus mode. When set and followed by a write cycle, normal SBus operation takes place with the write data being strobed into the selected DMA data buffer. However, with loop around active, the request cycle mixers are disabled, inhibiting any cycle requests over the KBus.

If an SBus read cycle is initiated next (with the DMA still in loop around mode), SBus operation is again normal with the data stored in the data buffer by the previous write transferred back to the MBox. As a result, the data path for read and write cycles may be tested independently of core memory selection and circuity. The mode is automatically cleared at termination of the read operation.

2.9.4 Single Step Operation

This operation provides the system with the capability to single-step (execute) read/write memory cycles at a reduced clock rate. For this function, the central processor is operating in its diagnostic mode at a reduced clock rate with the DMA set in 1 bus mode. However, DMA operations are essentially executed as with normal cycle requests, with the slow clock logic generating the necessary functions to transfer data from/to the KBus.

SECTION 3
LOGIC DESCRIPTION

3.1 INTRODUCTION

This section describes the logical elements that comprise the major portions of the DMA. Included is a detailed description of the DMA at the functional logic level. However, in some cases where deemed necessary, the description is detailed to the logic gate level. In these cases, in addition to the logic diagrams supplementing the text, reference is made to the appropriate print in the field maintenance print set.

3.2 CLOCK/PHASE LOGIC

The clock logic is driven synchronously with the external MBox clock and provides several clock outputs. The clock outputs (internal clocks) are distributed throughout the DMA boards to initiate and synchronize functional operations during requested memory cycles. Included with the clock logic is the phase indicator circuit, which detects and indicates the period prior to the Phase A clock and the period when the Phase A clock is true.

3.2.1 Clock Logic

The clocks are generated on alternate half cycles of the external clock. The Phase A clock is generated on the fall time of the external clock, and an equivalent Phase B clock generated in the control logic on the subsequent rise time of the external clock. The clock logic, which is detailed on field maintenance print DTR3, basically consists of three pulse generators. Each generator contains an adjustable delay line to compensate for logic delays and skewing encountered on the SBus.

As shown in Figure 3-1, the external clock (SBUS EXT CLK) is inverted and applied to an 18 ns delay line that effectively determines clock pulse width. The output of the 18 ns delay line (CLK DLY) is ANDed with DTR1 CLK EXT and -DTR1 CLK EXT generating two complementary clock trains having 20-ns-wide pulses (nominal) at the repetition rate of the external clock. One clock train is in phase with Phase A of the external clock, the other in phase with Phase B. The two outputs are gated with +3 Vdc to drive the 80 ns deskewing delay lines. The delay line associated with the 125 ns A output is adjusted such that the rising edge is coincident with the rising edge of the MBox clock, which occurs at Phase A time.

The 62 ns clock outputs are generated through an inclusive OR of the deskewed complementary clock trains. Each 62 ns delay line is adjusted such that the output is coincident with the fall (Phase A) and rise (Phase B) of the external clock. The four 62 ns B–E clocks are distributed to the M8558 board pairs associated with each KBus. The 62 ns A and 125 ns A (H) clocks synchronize the M8560 board, while the remaining clocks are distributed to the M8563 board.

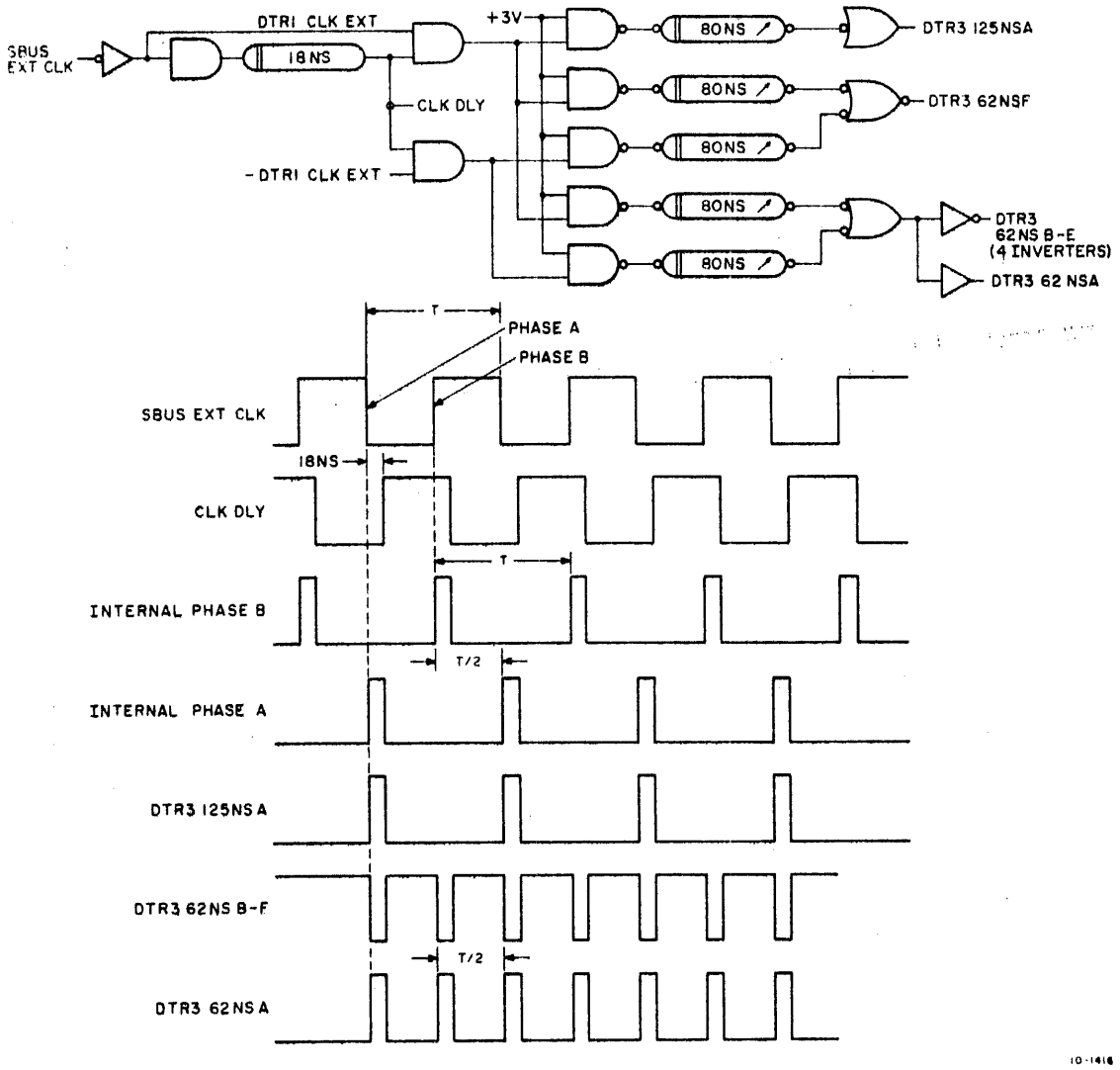


Figure 3-1 Clock Logic and Timing

3.2.2 Clock Driver

Because of fanout considerations, the 62 ns F output is buffered to a driver that supplies the clock current requirements to M8563. As shown in Figure 3-2, the clock driver is constructed of discrete components consisting of three stages. The PNP input stage couples the F clock into two NPN drivers. One driver controls an NPN Darlington pair, while the other controls a PNP Darlington pair. In this configuration, the Darlington pairs are operated in modified push-pull.

When DTR3 CLK 62 ns F goes high (2.0 V min), Q1 is driven into cutoff, in turn driving Q2 and Q3 into cutoff. Q2 then drives the upper Darlington pair (Q6 and Q7) into cutoff; Q3 drives the lower Darlington pair (Q4 and Q5) into conduction. When DTR3 CLK 62 ns F goes low (0.8 V max) Q1 conducts, turning on Q2 and Q3. This time, Q2 turns on the upper pair while Q3 drives the lower pair into cutoff.

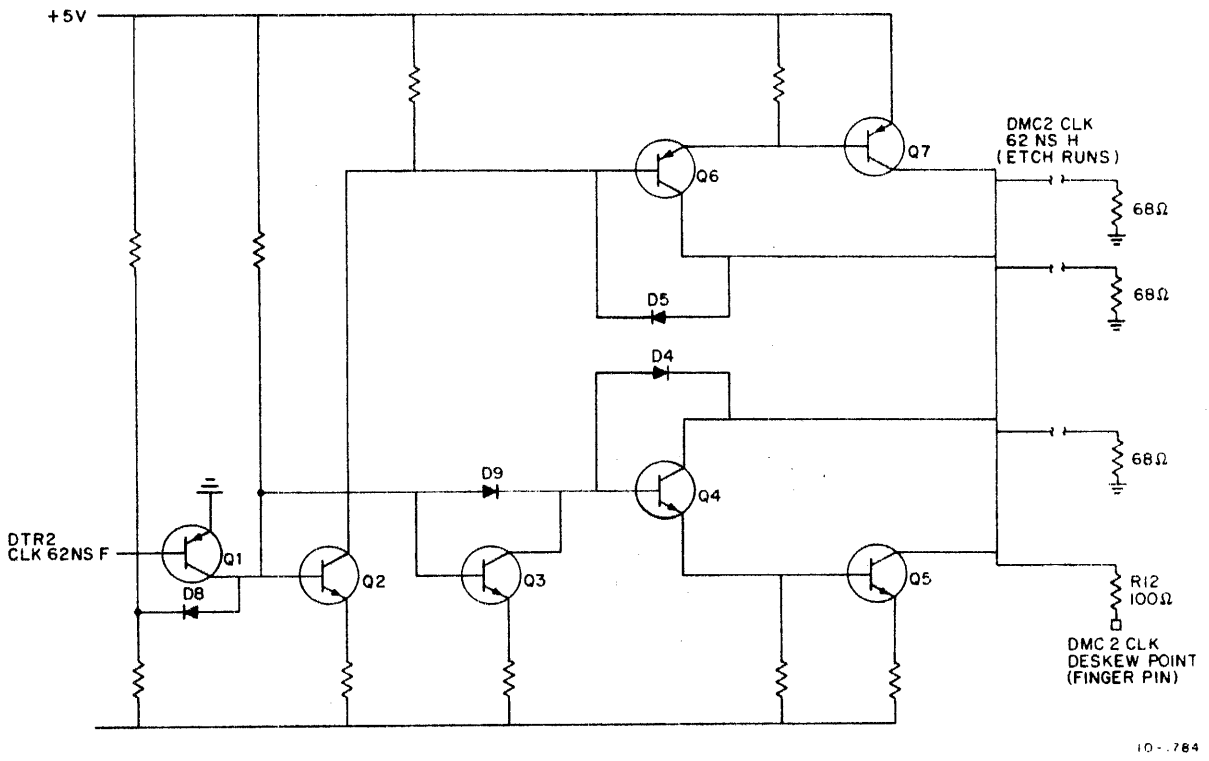


Figure 3-2 Simplified Clock Driver and Etch Runs

Note that Q4 and Q6 of the output stages are shunted with Schottkey diodes (D4 and D5) to prevent either transistor from saturating and increasing recovery time. However, in this configuration Q4 and Q6 prevent Q5 and Q7, respectively, from saturating, thus maintaining the required high turn on/off speed. Likewise, Q1 and Q3 are also prevented from saturating by diodes D8 and D9.

The driver is capable of sinking 140–193 mA to the three output etch runs, with each run terminated in 68Ω. The three etch runs plus the deskew point run (DMC2 CLK DESKEW POINT) present an effective output impedance of 21.5Ω. The deskew point is connected to a board finger pin through a short etch run. The series 100Ω resistor (R12) acts to prevent reflections on this unterminated run. Should an etch run short to ground, the excessive current will break down Q7 since it has no current limiter in its collector circuit.

3.2.3 Phase Indicator

The phase indicator consists of a pair of interconnected flip-flops used to synchronize SBus functions in the control logic to Phase A of the MBox clock. The Phase A Coming flip-flop (PHS A COMING) indicates the period prior to Phase A being true, while Phase A Clock (CLK PHS A) indicates the period when Phase A is true. Note that PHS A COMING is a pseudo-signal used only for descriptive purposes and is not called out on the logic prints.

As shown at T0 in Figure 3-3, CLK 125 ns A direct sets PHS A COMING. The zero set output of PHS A COMING provides a low data input to CLK PHS A, while ground is the data input for PHS A COMING. On the following 62 ns clock both flip-flops reset, with the PHS A COMING zero set output going high. The next 62 ns clock sets CLK PHS A, while the 125 ns A clock again direct sets PHS A COMING. This cycle repeats, continuously toggling CLK PHS A true in sync with Phase A of the external clock.

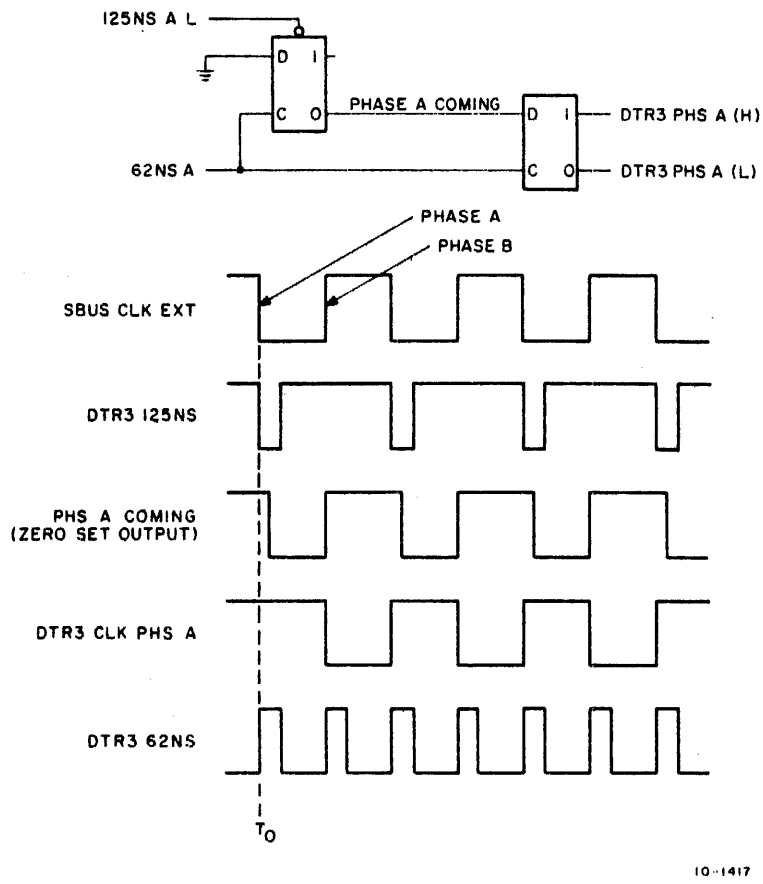


Figure 3-3 Phase Indicator Logic and Timing

3.3 OPERATIONAL MODE SETUP

Prior to initiating a memory cycle request, the DMA operational mode is set on the first half of a Function 0 diagnostic cycle issued by the MBox (Subsection 3.8.2). The interleave setup functions are generated in and share a portion of the diagnostic circuitry in the error logic.

As shown in Figure 3-4, the DMA address and function code is decoded from the SBUS D00–04 and D31–35 lines in the diagnostic function decoder. After decoding, the Function 0 Diagnostic Cycle function (DTR2 FCN 00), together with the enabling bit (SBUS D12), generates the Bus Mode Set function (DTR2 BUS SET). Since SBUS D06–07 determine the bus mode enabling code, they are ANDed with DTR2 BUS SET to set the bus mode flip-flops according to the required code.

DTR1 D07 sets the 1 Bus Mode flip-flop (DTR2 BUS 1); DTR1 D06 sets the 2 Bus Mode flip-flop (DTR2 BUS 2). As shown in Figure 3-4, both flip-flop outputs are sampled through inverters such that if DTR2 BUS 1 is set and DTR2 BUS 2 is not set, 1 bus mode is enabled (DTR2 1 BUS MODE). Likewise, if DTR2 BUS 1 is not set and DTR2 BUS 2 is set, 2 bus mode is enabled (DTR2 2 BUS MODE). The latch input for each flip-flop is provided by the opposite (disabled) bus mode function. For example, with 1 bus mode enabled (DTR2 BUS 1), -2 Bus Mode (-DTR2 2 BUS MODE) provides its latching function. The opposite is true for 2 bus mode.

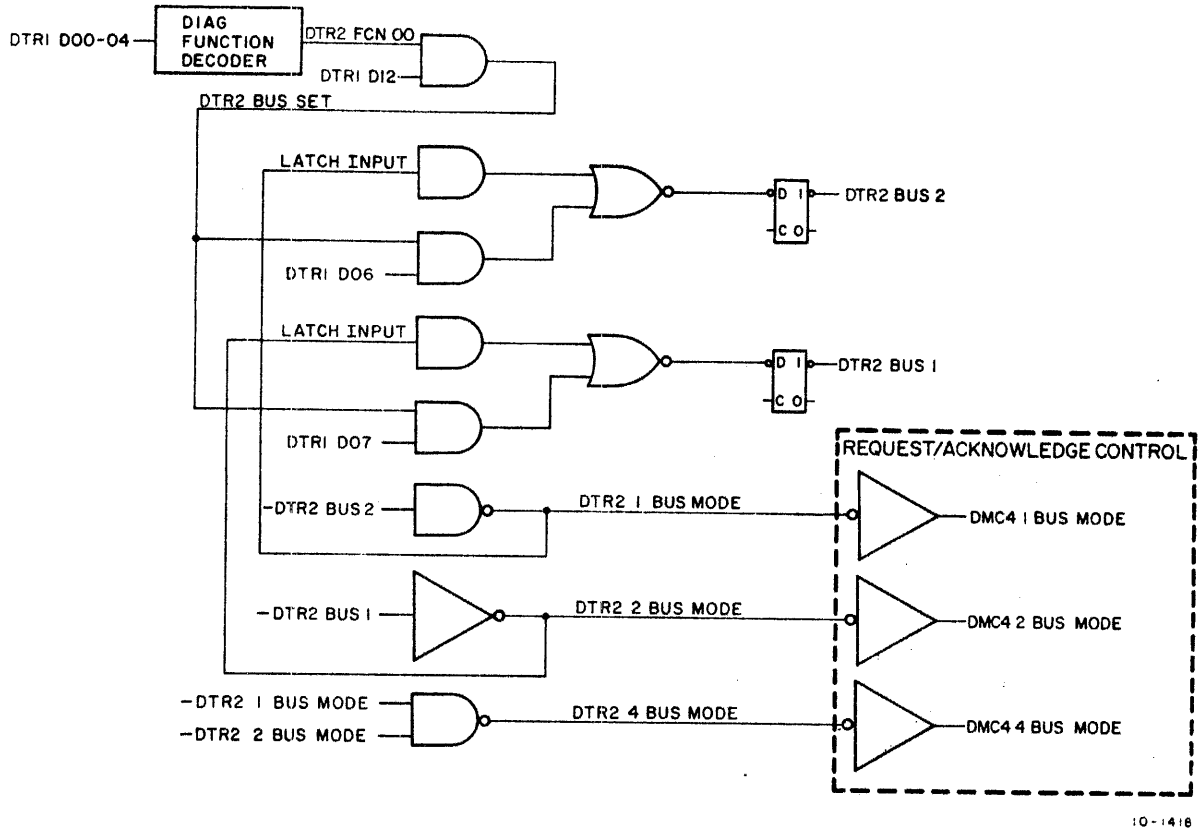


Figure 3-4 Operational Mode Logic

To enable the 4 bus mode (DTR2 4 BUS MODE), the other bus mode functions are ANDed such that with DTR2 BUS 1 and 2 set, the 1 and 2 bus modes are disabled (-DTR2 1 BUS MODE ∧ -DTR2 2 BUS MODE), thus enabling DTR2 4 BUS MODE. Note that in 4 bus mode the disabled bus mode functions provide the latching functions much the same as in 1 and 2 bus modes. The second half of the Function 0 diagnostic cycle also uses the bus mode set field (SBUS D06-07) to verify setting of the desired operational mode. The output of the bus mode sampling inverters is coupled back into the interface drivers and returned to the MBox when the transmit function (DTR2 XMIT) is true.

3.4 4 BUS MODE OPERATION

The following subsections present a detailed functional logic description of a quadword request (access order: Words 0, 1, 2, and 3) in 4 bus mode. Detailed timing diagrams and functional logic block diagrams are provided to supplement the text.

A quadword memory cycle is initiated when the MBox asserts:

a. Address lines - SBUS ADR 14-35

b. Read/write request lines - SBUS RD RQ/WR RQ

c. Word request lines - SBUS RQ 0-3

d. Cycle start line - SBUS START A/B

SBUS RD RQ/WR RQ are coupled through the interface board. They are then passed through the cycle type mixer setting the appropriate cycle type flip-flop (i.e., DMC4 RD for read cycle, DMC4 WR for write cycle, both for read/modify/write cycle). The SBUS RQ 0–3 lines are coupled through the interface board and applied to the word request logic input to determine the number of words accessed. SBUS ADR 14–33 are coupled through the interface board to the address buffer register in the data path logic. The two low-order bits (34–35) are applied to the address counter in the bus selection logic to determine the cycle starting address. SBUS START A/B is coupled through the interface board and applied to the Request/Acknowledge control to generate the cycle setup delay time.

3.4.1 Bus Selection

Initially the two low-order bits (DTR1 ADR 34–35) are applied to the address counter effectively initializing the counter output to a value equal to the value of bits 34 and 35. During memory cycle requests, the counter is locked to its current count until a delayed RD RS for a read operation or delayed ADR ACK for a write operation is generated, at which time it is incremented by the enabled word request lines. In addition, the address counter output together with the word request lines and bus mode are applied to the address selection logic. This logic functions to apply the correct low-order address bits to the appropriate KBus as determined by the current bus mode (Subsection 3.4.3).

3.4.1.1 Address Counter Setup – The functional counter is constructed to two 2 × 4 mixers and two D-type flip-flops. As shown in Figure 3-5, the selector inputs of the request line mixer are determined by the counter output (DMC3 NEXT 1 and DMC3 NEXT 2). The code for the incrementing mixer selector inputs is determined by the delayed ADR ACK for a write and delayed RD RS for a read. The output of the incrementing mixer conditions the inputs of flip-flops DMC3 NEXT 1 and DMC3 NEXT 2, which constitute the counter output. Note that each section of a cascaded pair of mixers corresponds to a bit position output of the counter.

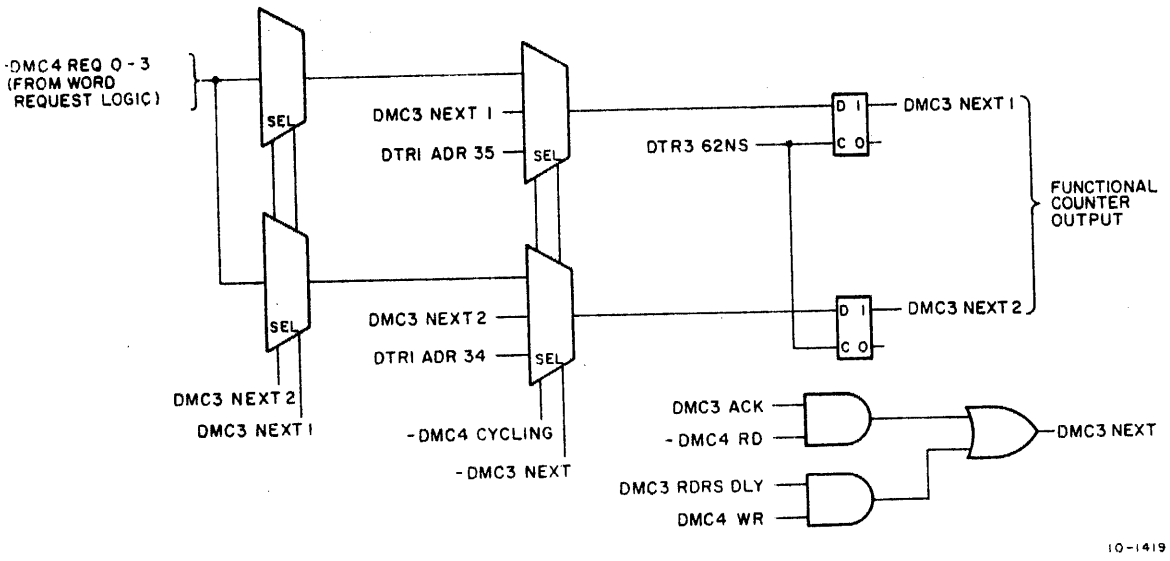


Figure 3-5 Address Counter Logic

Since the starting address was 00 (DTR1 ADR 34–35 = 00), this value is gated through the incrementing mixer resetting the counter output flip-flops to 00 (-DMC3 NEXT 1 ∧ -DMC3 NEXT 2) on CLK 62 ns A. At this point the incrementing mixer selector inputs are coded 11 (i.e., -DMC3 NEXT ∧ -DMC4 CYCLING). The counter output is coupled to the bus decode logic, which enables

the K0 bus function (DMC3 BSEL 0). Each bus function is coupled into the Read/Write and Request/Acknowledge controls. In addition, the bus functions are latched into the bus selection latches (DMC2 THIS 0–3); these latched bus functions are coupled into the data buffer enable gating of the Data Path/Control.

In the Request/Acknowledge control, the bus functions (DMC3 BSEL 0–3) are ANDed with corresponding KBus acknowledge functions (K0–3 ADR ACK from the synchronizer to ensure that the responding storage module is on the selected DMA bus port (in this case K0 bus). If the responding storage module was not on the selected bus port, the ACKN sequence would not be initiated, thus causing an NXM error at the MBox.

The bus functions are implemented in the same manner in the Read/Write control, ensuring that KBUS RD RS is received over the selected bus port. As with ACKN, should the RD RS function return over a KBus not selected SBUS DATA VALID would not be returned to the MBox, since the response sequence would not be initiated. In addition, functions are provided in both logic areas to synchronize the transmittal of SBUS ACKN and DATA VALID on alternate Phase A and B clock times.

3.4.1.2 Address Counter Increment, Write – For a write operation and with address counter output equal to 00, a code of 00 on the request line mixer select inputs allows DMC4 REQ 1 and -DMC4 REQ 1 to the input of the incrementing mixer (Figure 3-5). DMC3 ACK DLY goes true, setting DMC3 NEXT and forcing code 01 on the incrementing mixer select inputs. This action allows the enabled DMC4 REQ 1 and -DMC4 REQ 1 from the request line mixer through the increment mixer to the data inputs of DMC3 NEXT 1 and 2. On the following 62 ns clock the counter output is effectively incremented to 01 (-DMC3 NEXT 2 ∧ DMC3 NEXT 1).

At this time DMC4 CYCLING sets (and DMC3 NEXT is disabled) selecting code 10 on the increment mixer select inputs. This effectively latches the counter to the current count (due to the cross coupling of the counter output to the increment mixer inputs). The counter output, now equivalent to an address of 01, is coupled to the bus decode logic enabling the K1 bus function (DMC3 BSEL 1). DMC3 BSEL 1 is then latched into the bus selection latches setting DMC2 THIS = 1.

When the subsequent DMC3 ACK DLY is received DMC3 NEXT is again set selecting the code 00 inputs and incrementing the counter output to a state of 10 (DMC3 NEXT 2 ∧ -DMC3 NEXT 1). The mixer select and counter incrementing continue until the word request logic is cleared (-DMC4 ANY RQ) and all buses are not busy (-MA5 KN BUSY), thus effectively terminating the write cycle request.

3.4.1.3 Address Counter Increment, Read – For a read operation with the counter output equal to 00, a code of 00 on the request line mixer select inputs allows DMC4 REQ 1 and -DMC4 REQ 1 to the input of the incrementing mixer. After ADR ACK time REQ CYCLING is set, setting up the code 10 on the increment mixer select inputs. This effectively latches the counter to the current count (due to cross coupling of the counter output to the increment mixer input). On the subsequent DMC3 RD RS DLY, DMC3 NEXT sets selecting the code 00 inputs on the increment mixer allowing DMC4 REQ 1 and -DMC4 REQ 1 to the counter flip-flop inputs. At CLK 62 ns A the flip-flops are strobed producing a counter output of 01 (-DMC3 NEXT 2 ∧ DMC3 NEXT 1).

As during the write operation, the counter output (equivalent to an address of 01) is coupled to the bus decode logic enabling Bus Function 1 (DMC3 BSEL 1), which is latched into the bus selection latches setting DMC2 THIS = 1. When the subsequent DMC3 RD RS DLY (associated with KBus 1) is received, DMC2 NEXT is again set selecting the code 00 inputs and incrementing the counter output to a state of 10 (DMC3 NEXT 2 ∧ -DMC3 NEXT 1). The mixer select and counter incrementing continues until the word request logic is cleared (-DMC4 ANY RQ) and DMC4 COMPLETE is set, thus effectively terminating a complete read cycle request.

3.4.1.4 Word Request Logic – The word request logic is constructed of a decoder, 4 × 2 mixer, and a set of four D-type flip-flops. As shown in Figure 3-6, the output of each mixer is gated to an associated flip-flop. (Note that the circuit illustrated represents only one-quarter of the request logic, and is duplicated in the hardware for each of the remaining request lines.) Initially the request logic stores the state of the SBUS RQ 0–3 lines in the four request flip-flops. At each ADR ACK time the appropriate request flip-flop is reset which, in turn, clears the corresponding KBus cycle request (RQ CYC) line at the request cycle mixer input. The particular RQ CYC line disabled is determined by the access order and address counter output.

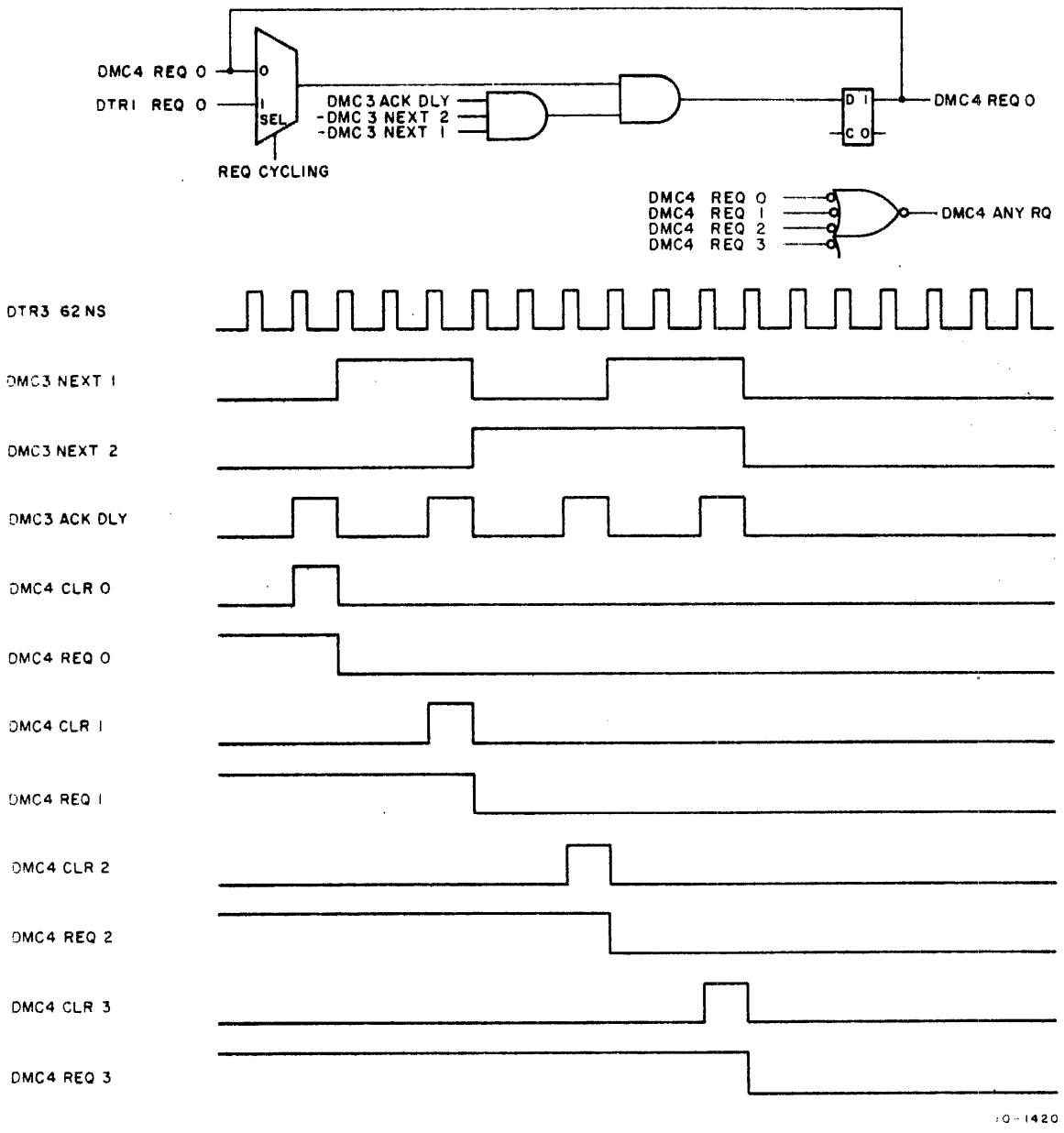


Figure 3-6 Word Request Counter and Timing

The status of request lines DTR1 RQ 0–3 are strobed into the flip-flops on a CLKS 62 ns A clock. (In this case all flip-flops are set.) Concurrent with the delayed acknowledge (DMC3 ACK DLY) and subsequent incrementing of the address counter, the flip-flop input gate is disabled resetting DMC4 RQ 0 on the next CLKS 62 ns A. (The address counter output provides the decoder inputs.) The feedback path now holds the flip-flop reset. -DMC4 REQ 0 is fed through the cycle request mixer disabling DMC1 K0 RQ CYC (mixer select code = 11).

The request flip-flops continue to reset on each DMC3 ACK DLY, disabling the corresponding request function. When all request flip-flops have reset (indicating all requests have been processed), -DMC4 ANY REQ disables a portion of the DMC4 CYCLING flip-flop latch input gating. The DMC4 COMPLETE function is again set on BSEL WR RS DLY (for write) or DMC3 RD RS INH (for read). DMC4 COMPLETE disables the remaining portion of the input gating, resetting DMC4 CYCLING and effectively terminating a read request. A write request is effectively terminated with -DMC4 ANY REQ and each of the not busy functions true, allowing DMC4 CYCLING to reset.

3.4.2 Request Cycle Start

SBUS START A/B is applied to the request cycle sequence flip-flops in the Request/Acknowledge control. The start function (SBUS START A or B) is determined by which MBox clock (Phase A or B) initiated the function. The start logic generates the last function necessary to initiate a cycle request over any KBus (DMC4 CYC START). In addition, the start sequence provides setup time for the word request and address counters, and bus selection logic assuring that the address and address lines are valid at the storage module prior to issuing the cycle request. Incorporated with this logic is the Phase A Start (DMC4 START PHS A) flip-flop, which flags the phase START was synced on and prevents unlatching of DMC4 BUS START on the opposite clock phase.

The start logic consists of a set of cascaded flip-flops. The sequence is initiated by the reception of SBUS START A/B. Both START A and B are ANDed with DTR3 CLK PHS A and -DTR3 CLK PHS A, respectively. This gating arrangement allows sequence initiation coincident with Phase A or B time (Figure 3-7). The start function is propagated through the flip-flops at successive CLK 62 ns A clocks and ultimately sets the Cycle Start function (DMC4 CYC START). A delay is generated between the reception of the request at the DMA and actual cycle initiation at the KBuses. With DMC4 CYC START true, the final requirement for the request cycle gating is satisfied.

For a read operation DMC1 KN RQ CYC is gated with -DMC4 WR ONLY to generate MA2 KN RQ CYC, which satisfies the KBus translator output AND gate. At this point, the request (KBUSN RQ CYC), together with the cycle type, is issued over the KBus. However, in the case of a write operation, MA2 KN RQ CYC is inhibited and is not issued over the KBus until the SBus write data is loaded into the appropriate data buffer.

The 20 high-order address bits (ADR 14–33) are applied to an additional buffer register so that when the MBox changes address lines for the next cycle the address for the current cycle request is not lost. Note that the two low-order bits (DTR1 ADR 34–35) are already contained in the address counter. The address buffer, located in the Data Path/Control, is functionally between the SBus address receivers and the address line translators. DMC4 START serves as the address buffer strobe function, with the address stored in the buffer until the request is complete.

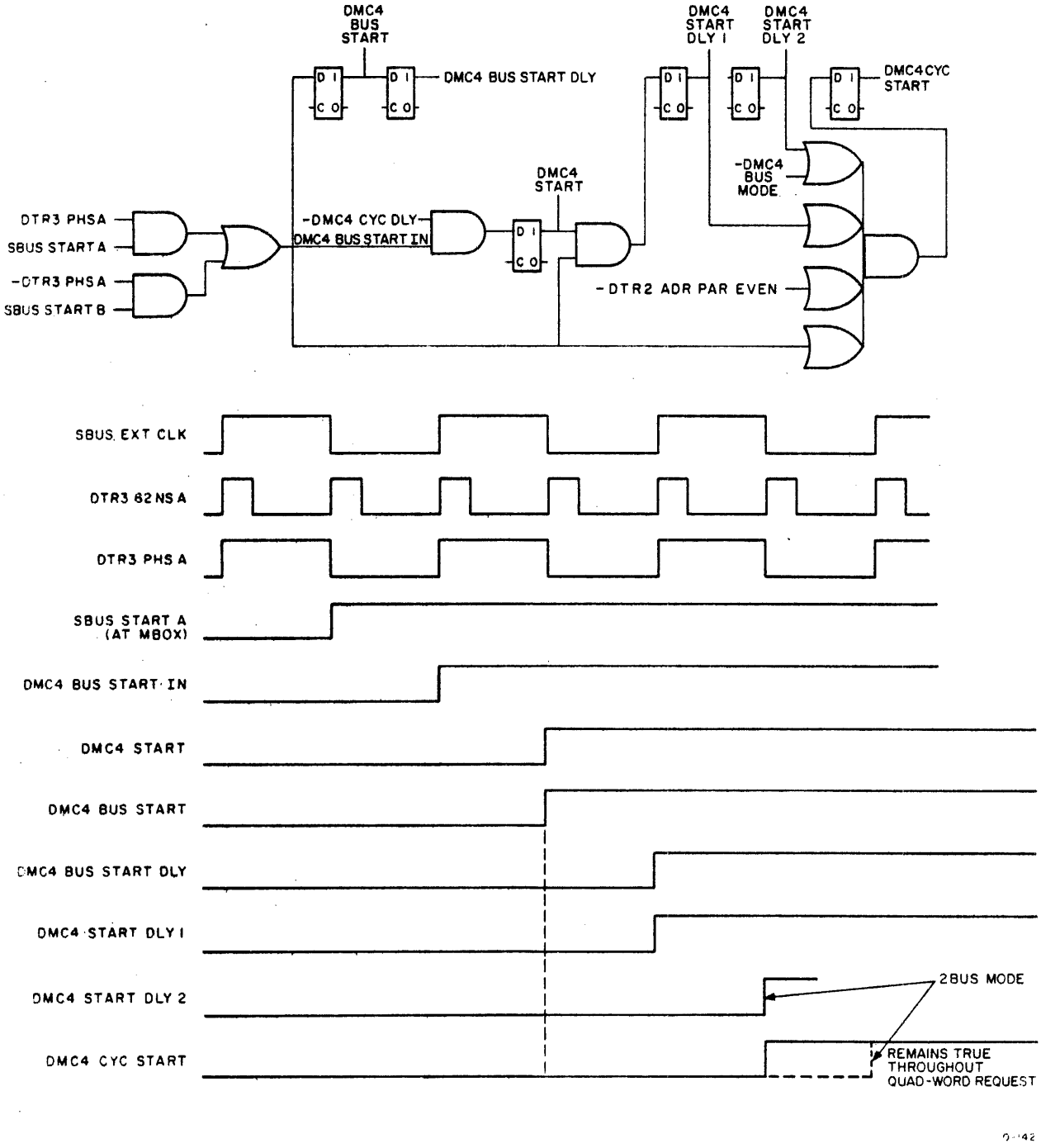


Figure 3-7 Start Logic and Timing

3.4.3 4 Bus Address Selection Logic

The address selection logic (shown on field maintenance print DMC1) provides the DMA with the capability of allowing the correct low-order address bits (34–35) to their associated KBus as determined by the current operational mode. The selection logic outputs are grouped such that there is one bit 34 output/bus (e.g., DMC1 K2 ADR 34), with odd and even buses combined for bit 35 (e.g., DMC1 K1/3 ADR 35) (Figure 3-8). The overall selection logic may be divided into two areas: output gating and 2 bus addressing. All selected addresses (regardless of operational mode) are either generated in (1 and 4 bus modes) or coupled through (2 bus mode) the output gating.

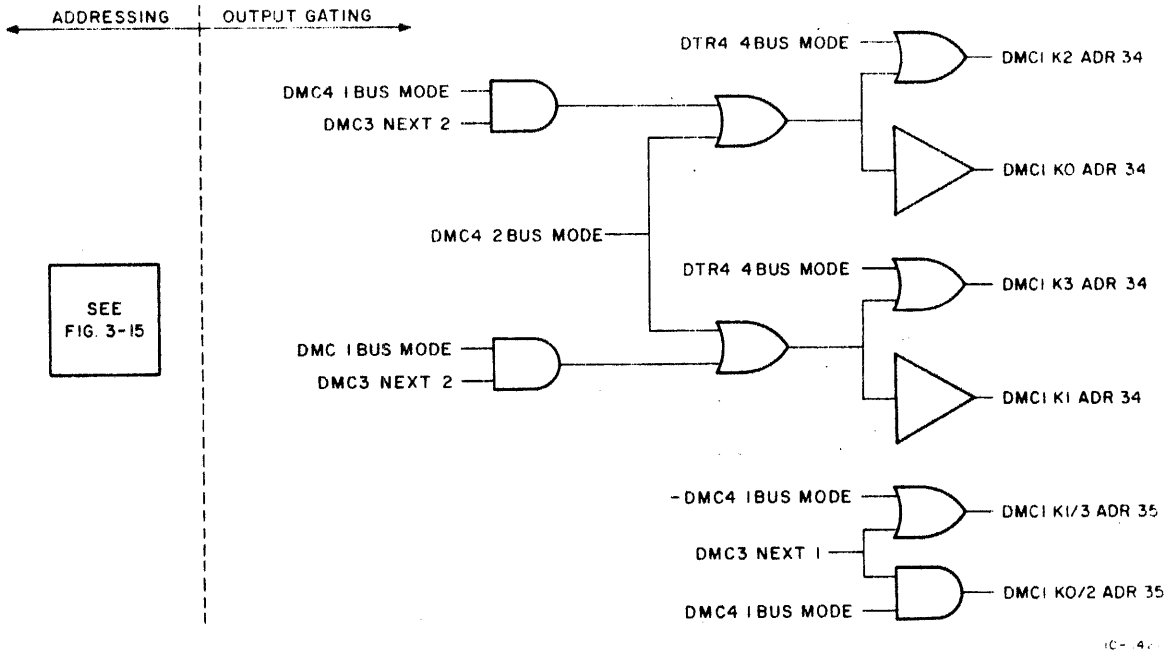


Figure 3-8 1/4 Bus Address Selection Logic

Each word accessed in 1 bus mode requires a complete word cycle with no correlation between address and KBus number, i.e., the same address is allowed over all KBuses at each request cycle initiation. The output gating logic is such that with 1 bus mode enabled (DTR2 1BUS MODE) the individual bit position outputs are directly dependent on the address counter output. For example, assume a Word 1 access; in this case the address counter output is 01 (-DMC3 NEXT 2 ∧ DMC3 NEXT 1), which causes the output gating to assert DMC1 K1/3 ADR 35 and DMC1 K0/2 ADR 35, and negate each of the DMC1 K0–3 ADR 34 outputs.

In 4 bus mode there is a direct correlation between a specific address and a KBus number (i.e., adr = 00 = KBUS 0, adr = 01 = KBUS 1, etc.). With 4 bus mode enabled (DTR2 4 BUS MODE) bits 34 on KBuses 2 and 3 are always asserted by DTR2 4 BUS MODE since they will access Words 2 and 3.

In 2 bus mode, low-order address selection (and in particular bit 34) is implemented differently to save time between DMA request cycles. That portion of the selected logic dedicated to 2 bus operations is described in Subsection 3.5.1.

3.4.4 KBus Logic Level Translators

The KBus logic level translators provide the control, data, and KBus response lines with logic level conversion between the DMA and each KBus. The bidirectional data lines are equipped with a translator having both input and output legs. All received signals on input legs (into DMA) are translated by identical logic (i.e., the receive logic for ADR ACK is the same as for a data bit). Likewise, the output leg (to storage module) for a data line translator is electrically identical to the output translator for an address/control function. As such, only the data line translator will be described.

As shown on the output leg in Figure 3-9, the write data (or address/control function) is coupled to the output gate through a 4258 transistor amplifier that acts as a level shifter. With its input low (logic 1) the 4258 is driven into conduction. The voltage drop at the junction of load resistors R1 and R2 satisfies the 8881 output AND gate (typical high = 1.1 V, typical low = -4.2 V, at point A). The output is then coupled through current limiting resistor R3 to the KBus cable and storage module.

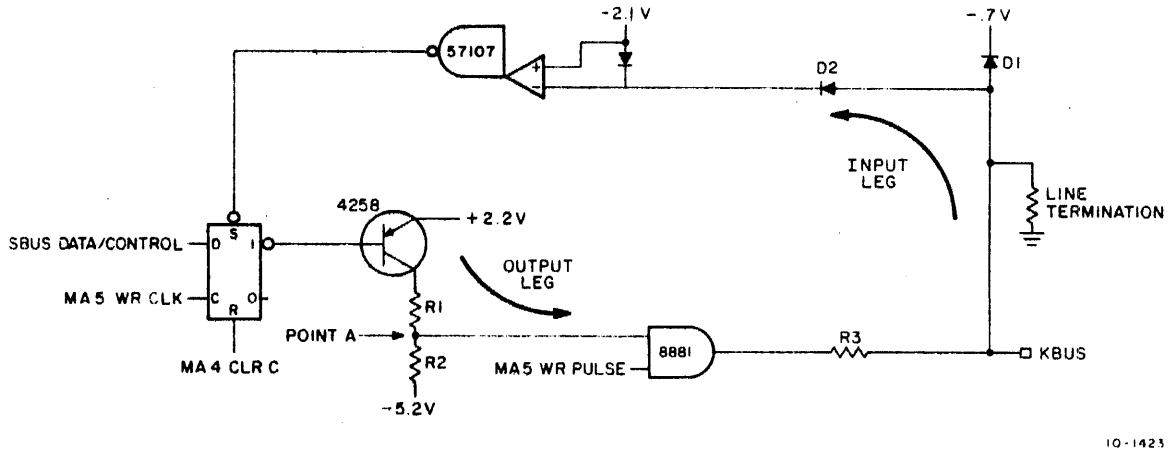


Figure 3-9 Translator Logic

The translator input leg consists of a differential comparator and the associated clamping and reference levels as shown in Figure 3-9. For descriptive purposes, the voltages used may be considered worst case.

The data (or response function) is coupled through D2 to the inverting side of the comparator. D1 acts as a clamping diode, clipping the positive overshoot on a 1 → 0 transition. With a logic 1 the voltage at the junction of D1 and D2 is -2.9 V. Allowing a 0.7 V voltage drop across D2, the voltage at the input of the comparator is -2.2 V. With the noninverting input of the comparator referenced to -2.1 Vdc, a difference of 100 mV is produced between comparator inputs. The 75107 AND leg is now satisfied (other gate legs referenced to +3 Vdc) producing a low and direct setting the associated data flip-flop.

3.4.5 Request Line Compatibility

MA2 KN RQ CYC is applied to the three request lines of each KBus: Fast, Slow, and Immediate. These request lines provide compatibility between the DMA and any storage module type connected to the KBus. Asserting all three request lines on the selected KBus on each cycle request guarantees that any type storage module connected will respond when addressed. In addition, RQ CYC is gated with the initial flip-flop level of the KBus synchronizer (MA4 ADR ACK) assuring that on the reception of ADR ACK, RQ CYC is disabled over the KBus. This ensures that the memory will only cycle once for each request.

3.4.6 Address Acknowledge Sync

Following a cycle request the storage module determines it is the addressed module. After address decoding the storage module responds to the DMA with ADR ACK, which is returned through the KBus translator and synchronizer to the Read/Write control. The synchronizer provides storage for the module responding functions (ADR ACK and DATA WARNING/RD RS). Since all storage modules will not necessarily respond with either function in the order requested, the first flip-flop level of the line translator retains the storage module response (in the case where a multiple-word request was issued).

Each level translator is provided with two inputs to accommodate response lines from all storage module types. One input accommodates ADR ACK (NT) (Not Timed) and RD RS, the other accommodates DATA WARNING. ADR ACK (or DATA WARNING/RD RS) is coupled through the input level translator and direct sets the first level flip-flop. In addition, this flip-flop output is also applied to the cycle request line gating in the Read/Write control in order to disable the appropriate RQ CYC line with the reception of ADR ACK.

The initial level flip-flop outputs are ORed and coupled into the synchronizer. The synchronizer consists of two cascaded flip-flops that synchronize the KBus asynchronous acknowledge and restart functions with the synchronous DMA logic. The K0–3 ADR ACK functions are coupled into the Address/Bus Selection logic where they are ANDed with the bus select function (DMC3 BSEL 0–3). Since the address counter has determined that the first word is to be retrieved over KBUS 0, DMC3 BSEL 0 is enabled, gating K0 ADR ACK to the input of the ADR ACK flip-flop (Figure 3-10). On the following CLK 62 ns A clock, DMC3 ACKN sets.

For efficiency DMC3 ACKN is detected as occurring on either a Phase A or B clock. DMC3 ACKN A PHS is continually toggled and indicates the next clock phase. When DMC3 ACKN sets, DMC3 ACKN A PHS is prevented from changing until ACKN is cleared. The state of ACKN A PHS, when ACKN is set, determines which SBUS ACKN (A or B) will be transmitted on. DMC3 ACKN is ANDed with DMC3 ACKN A PHS in the Request/Acknowledge control to generate SBUS ACKN A, otherwise ACKN is gated with -DMC3 ACKN A PHS generating SBUS ACKN B over the SBus.

As shown in Figure 3-10, DMC3 ACKN sets the acknowledge delay flip-flop (DMC3 ACK DLY) on the next CLK 62 ns. This function provides delay time between the SBUS ACKN functions. DMC3 ACK DLY sets DMC3 ACK INH on the next CLK 62 ns which, in turn, resets MDC3 ACKN. Thus, an ADR ACK pulse with a duration of one external clock period, is generated and transmitted to the MBox in the order requested every one and one half external clock periods (assuming all memories had responded).

3.4.7 Read/Restore Cycle Control

The storage module data is coupled through the input level translator and direct sets each applicable bit position flip-flop in the data buffer. (Note that all buffers are cleared at the start of the DMA cycle.) Since the read data direct sets the data buffer flip-flops, it is not necessary to be concerned with the order in which the read data is returned. The internal DMA timing functions will gate the data out of the buffers and onto the SBus in the order requested. With the appropriate latched bus function true (in this case DMC2 THIS = 0) the data buffer enabling function (DMC1 K0 DATA EN) is generated enabling the SBus input side of the data buffers and allowing the Word 0 read data to the SBus transmitters (Figure 3-10). The subsequent storage module restart function (KBUS0 RD RS) direct sets the flip-flop in the translator.

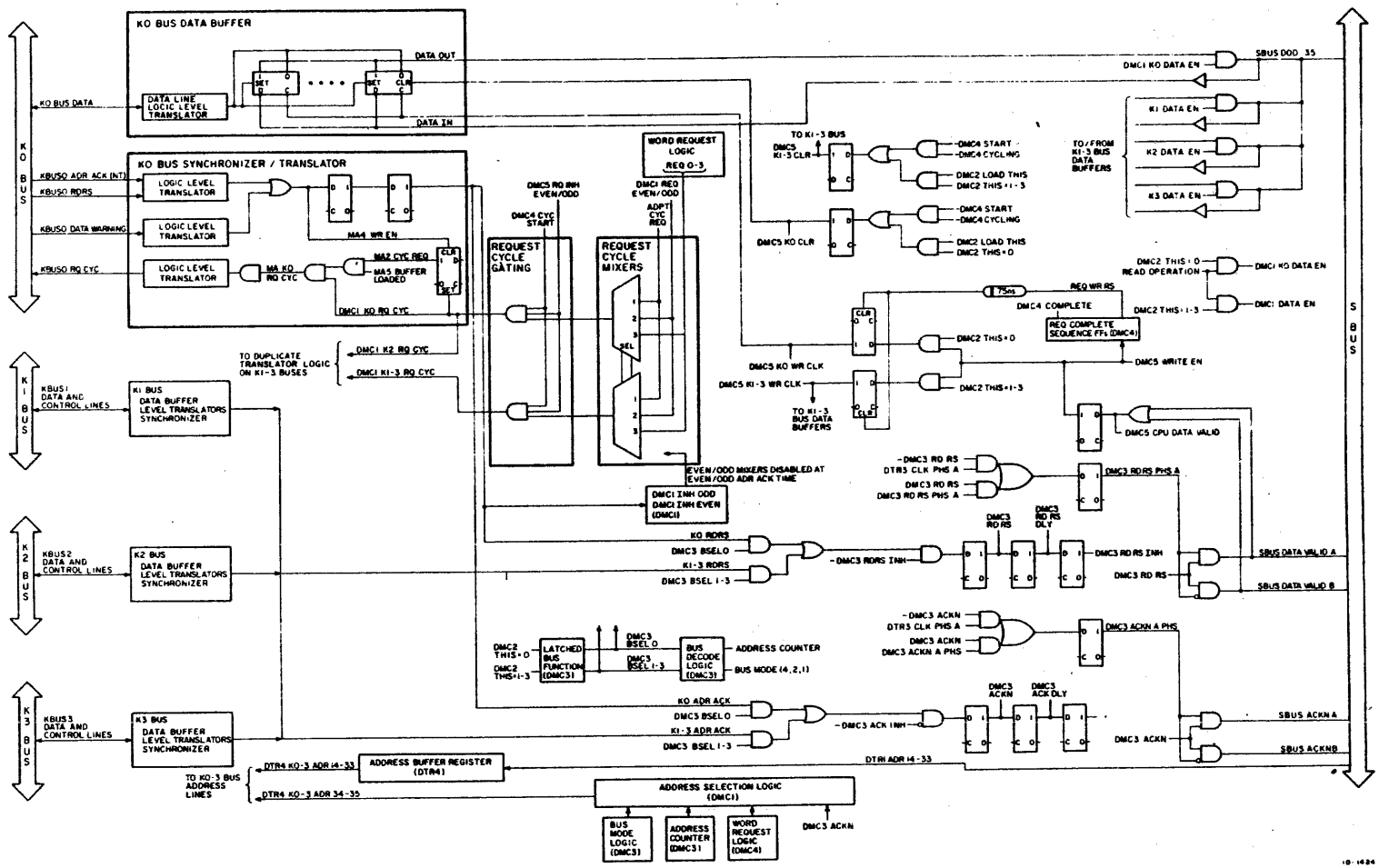


Figure 3-10 Functional Logic Block Diagram

With those storage modules which return DATA WARNING prior to RD RS, DATA WARNING is used as a logical restart function to compensate for the clocking time required for the KBus synchronizer flip-flops. The DATA WARNING function is delayed, in the storage module, to within 290 ns of RD RS. In operation, DATA WARNING is clocked through the synchronizer and Read/Write control to generate SBUS DATA VALID A/B two clock periods sooner than the subsequent RD RS. When RD RS is sent to the DMA, it is logically ignored since the synchronizer flip-flops were previously set by the DATA WARNING function.

As with ADR ACK, DMC3 RD RS is detected as occurring on either a Phase A or B clock. DMC3 RD RS A PHS is continually toggled indicating the next clock phase. When DMC3 RD RS sets, DMC3 RD RS A PHS is prevented from changing until RD RS is cleared. The state of RD RS A PHS when RD RS is set determines the phase on which DATA VALID will be transmitted. DMC3 RD RS is ANDed with DMC3 RD RS A PHS (or - DMC3 RD RS A PHS) setting SBUS DATA VALID A on the Phase A clock or SBUS DATA VALID B on the Phase B clock, indicating the Word 0 transfer- of the quadword request to the MBox.

Subsequently, DMC3 RD RS DLY is applied to a pair of cascaded flip-flops (DMC2 LOAD THIS and DMC2 LOAD THIS DLY). On successive 62 ns clocks both flip-flops set. However, DMC2 LOAD THIS NOW is generated on LOAD THIS $\wedge$ -LOAD THIS DLY. At DMC2 THIS = 0 time, DMC2 LOAD THIS NOW sets the K0 data buffer clearing flip-flop which, in turn, direct clears the K0 data buffer as well as the synchronizing flip-flops.

As the address counter is incremented, the corresponding latched bus functions (BSEL THIS = 1, 2, or 3) generate the appropriate data buffer enabling functions, strobing the data from the associated data buffer to the SBus transmitters and clearing the associated data buffers. Thus, the read data is word-serially transferred across the SBus in the order requested as determined by the word request lines.

In the meantime, DMC3 RD RS INH (generated one external clock period after DMC3 RD RS) has been applied to the request complete flip-flops in the Request/Acknowledge control. These flip-flops generate the Request Complete function (DMC4 COMPLETE) which is applied to the DMC4 CYCLING data input gating. DMC4 COMPLETE is set on each RD RS or WR RS function depending on the cycle requested. DMC4 COMPLETE, together with the output function of the word request logic (-DMC4 ANY REQ), disables the CYCLING flip-flop input. On the following 62 ns clock DMC4 CYCLING resets to allow disabling and clearing of the word request logic, address counters, and cycle type latches for the next MBox request.

3.4.8 Clear/Write Cycle Control

Unless a request is active in the DMA, the data buffers are force-cleared. Without the start (-DMC4 START) and cycling (-DMC4 CYCLING) functions, the clear flip-flops (DMC5 KN CLR) in the Read/Write control timing logic are held set. DMC5 KN CLR direct clears each bit position within the four data buffers.

In a write operation, the first word must be loaded into the appropriate data buffer prior to issuing a cycle request over the selected KBus. When DMC1 KN RQ CYC is generated it direct sets the cycle request flip-flop (MA2 CYC REQ). MA2 CYC REQ is ANDed with the buffer loaded flag (MA5 BUFFER LOADED) which at this time is reset. In the meantime, the Write Only function (DMC4 WR ONLY) from the cycle type latches generates the Write Enable function (DMC5 WRITE EN), which together with BSEL THIS = 0 generates the K0 data buffer write clock (DMC5 K0 WR CLK). The WR CLK strobes the write data from the SBus into the data buffer and direct sets the MA5 BUFFER LOADED flip-flop. BUFFER LOADED now satisfies the AND gate and generates MA2 K0-3 RQ CYC, which is coupled through the level translators to all KBuses. Note that the write data will be held in the K0 data buffer until sometime after ADR ACK is received (Figure 3-10).

ADR ACK from the responding module direct sets the translator flip-flop MA5 ADR ACK, which disables KBUS0 RQ CYC and generates the synchronizer Write Enable function (MA4 KN WREN). (ADR ACK is also returned to the MBox.) MA4 K0 WREN, together with BUFFER LOADED, drives a 100 ns delay line. After a 50 ns delay, the latch clock (MA5 LATCH CLK) is generated, and in turn generates the write pulse (MA5 WR PULSE) which enables the data buffer content to the KBus. WR PULSE is also coupled through a level translator and transmitted over the selected KBus as the Write Restart function (KBUS N WR RS). 100 ns later, MA5 150 ns DELAY L is generated cutting off the WR PULSE (100 ns pulse width) and resets the BUFFER LOADED flip-flop through its clock input.

In the meantime, DMC5 WRITE EN is also applied to the request complete flip-flops. Unlike the read operation, the DMC4 COMPLETE function in this case is implemented as a set of cascaded flip-flops set on successive CLK 62 ns clocks during the write operation (Figure 3-11). DMC5 WRITE EN initiates the sequence setting DMC4 WR RS, which is applied to the Read/Write control timing logic to drive a 250 ns delay line. The delay line output (DMC5 PULSE LIMITER) provides a 75 ns delay and direct clears the WR CLK flip-flop to ensure that the WR CLK is no longer than 100 ns. DMC4 WR RS also sets the restart delay flip-flop (DMC4 WR RS DLY) on the following CLK A clock. WR RS DLY provides a 62 ns delay between WR RS and DMC4 COMPLETE to allow propagation of WR RS to the storage module and generation of the data buffer clear function.

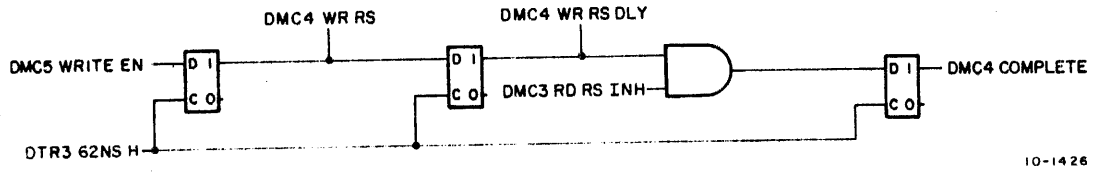


Figure 3-11 Simplified Request Complete Logic

In addition, DMC4 WR RS sets the DMC2 LOAD THIS flip-flop on the next CLK 62 ns and generates the DMC2 LOAD THIS NOW function. DMC2 LOAD THIS NOW and DMC2 THIS = 0 are ANDed setting DMC5 K0 CLR on the next 62 ns clock clearing each bit position of the K0 Bus data buffer. As the address counter is incremented the load and latched bus functions will sequentially set their corresponding clear functions clearing their respective data buffers shortly after the write clock.

When the MBox receives SBUS ACKN for Word 0 it removes Word 0 from the data lines and replaces it with Word 1. As each ACKN is received at the MBox, it sets the data lines with the next word. After receiving four ACKNs the MBox disables the data lines until the next request. On successive storage module ACKNs, DMC5 WRITE EN generates the appropriate WR CLK and WR RS functions transferring the write data through the data buffers to the KBuses. The operation is effectively terminated when all requests are cleared (-DMC4 ANY RQ) and all buses are not busy (-MA5 KN BUSY).

3.4.9 Read/Modify/Write Cycle Control

In addition to the other SBus functions, the MBox asserts both SBUS RD RQ and SBUS WR RQ indicating a read/modify/write cycle. During the read/modify/write cycle, the DMA executes a modified read/restore cycle, pauses while the central processor modifies the word, and then proceeds to execute a clear/write cycle when the MBox returns the modified data word. The read/modify/write cycle is used only for one word requests; however, as in a read or write cycle the request lines are used in conjunction with the address counter output to determine the word requested. Because the read modify/write cycle is used only for one word requests, cycle timing is identical for all three interleave modes.

After address decoding, the selected storage module responds with ADR ACK, places the requested word on the KBus data lines, and transmits DATA WARNING/RD RS. Note that in this case the address counter is not incremented and the current bus (DMC3 BSEL N) remains selected. As with the read/restore cycle described in Subsection 3.4.7, DATA WARNING/RD RS is returned to the MBox as DATA VALID A or B depending on the clock phase.

After modification, the MBox places the data on the SBus and asserts SBUS DATA VALID A/B. In this case, DATA VALID serves as an SBus write restart function. (Note that this is the only operation in which the DATA VALID function is transmitted to the DMA.) SBUS DATA VALID A/B is gated with DMC3 RD RS INH in the Read/Write Control generating an additional data valid function (DMC5 CPU DATA VALID). On the next 62 ns clock, DMC5 CPU DATA VALID generates DMC5 WRITE EN. WRITE EN, together with the latched bus function (DMC2 THIS = 0–3), generates the WR CLK function for the selected data buffer, strobing the modified data word into the buffer (Figure 3-10).

As in the write operation, WR CLK generates WR PULSE allowing the data buffer output to the KBus and the selected storage module. DMC5 WRITE EN is applied to the request complete sequence flip-flops generating DMC4 WR RS, which, after driving the pulse limiter delay line, clears the WR CLK flip-flop. DMC4 WR RS also sets DMC4 WR RS DLY on the following A clock. DMC4 WR RS DLY provides a 62 ns delay between WR RS and DMC4 COMPLETE which allows propagation of WR RS to the storage module prior to termination of the internal DMA cycle.

3.5 2 BUS MODE OPERATION

The following subsections present a detailed functional logic description of a quadword request in 2 bus mode starting at Word 0. For discussion purposes, storage modules on K0 and K1 buses are considered two-way interleaved. Those logic areas that are functionally identical to 4 bus mode are not described in detail; however, reference is made to the appropriate subsection in the 4 bus mode description.

Prior to initiating a memory cycle request, the DMA operational mode is set to 2 bus mode and verified by a Function 0 diagnostic cycle issued by the MBox (Subsection 3.8.2). A quadword memory cycle is initiated when the MBox asserts:

a. Address lines – SBUS ADR 14–35

b. Read/write request lines – SBUS RD RQ/WR RQ

c. Word request lines – SBUS RQ 0–3

d. Request cycle start – SBUS START A/B

SBUS RD RQ/WR RQ are coupled through the interface board. They are then passed through the cycle type mixer setting the appropriate cycle type flip-flop (i.e., DMC4 RD for read cycle, DMC4 WR for write cycle, both for read/modify/write cycle). The SBUS RQ 0–3 lines are coupled through the interface board and applied to the word request counter to determine word retrieval order. SBUS ADR 14–33 are coupled through the interface board to the address line translators in the Data Path/Control. The two low-order bits (34–35) are applied to the address counter to determine the cycle starting address. The counter output is applied to the address selection logic. In this mode, the logic sends the even address over K0 and 2 buses and the odd address over KBus 1 and 3 buses (Subsection 3.5.1). SBUS START A/B is coupled through the interface board and applied to the Request/Acknowledge control to generate the cycle setup delay time.

3.5.1 2 Bus Address Selection Logic

As described in Subsection 3.4.3, the address selection logic allows the correct low-order address bits (34–35) to their associated KBus as determined by the operational mode. Only that portion of the selection logic dedicated to 2 bus mode address selection is illustrated and described in this subsection. As shown in Figure 3-12, the logic is considerably more complicated for 2 bus mode than either 1 or 4 bus modes. As indicated in the simplified logic figure, the low-order address selection is different for read or write.

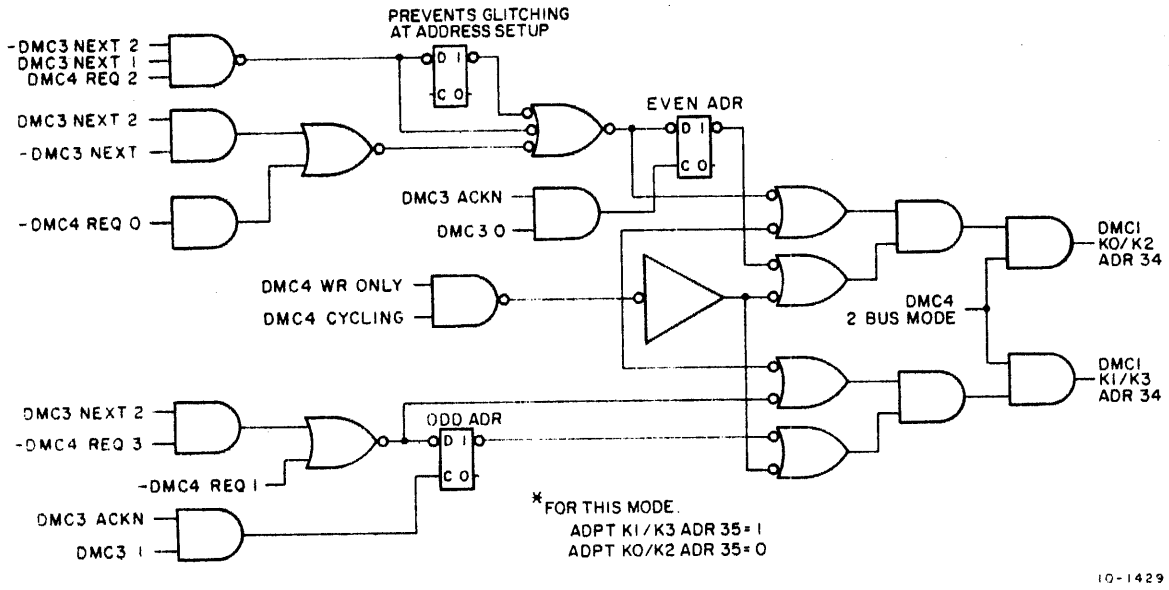


Figure 3-12 2 Bus Address Selection Logic

For a read operation, the address bits are not buffered (i.e., the even and odd address buffer flip-flops are not used), with the buffered portion of the output ORing AND gate always satisfied. In this case, address decoding is straightforward with the low-order address bits determined by the address counter output and latched request lines. However, the logic implementation is slightly different for the write operation. The problem in this case is to assure that the address on the enabled bus remains valid until the write operation has completed and still provide address setup prior to the second half of the DMA cycle.

As in the example, Word 0 is the first word to be written. Initially, the K0 and 2 buffers are loaded with the Word 0 data. The address selection logic allows 00 to K0/2 bus bits 34–35. On the subsequent ACKN from the storage module, the even address buffer flip-flop is strobed by DMC3 ACKN. Since its data input is not true, the flip-flop remains reset. DMC4 CYCLING then goes true and one-half of the output AND gates are satisfied. At the same time, the address counter is incremented to one (-DMC3 NEXT 2 ∧ DMC3 NEXT 1).

DMC3 BSEL 1 ∧ -DMC4 ODD BUSY initiates a second acknowledge sequence (simulated K1 ACKN). The subsequent DMC3 ACKN strobes the odd address buffer flip-flop, which remains reset (its data input being not true). After DMC3 ACK DLY time the address counter is incremented to 10, DMC3 BSEL 0 ∧ -DMC4 EVEN BUSY initiating a third acknowledge sequence (simulated K2 ACKN). The subsequent DMC3 ACKN strobes the even address buffer flip-flop. Since the address counter is 10 the flip-flop's data input is true and it sets, asserting DMC1 K0/K2 ADR 34. On the

subsequent ACK DLY the address counter is incremented to 4. On the fourth simulated ACKN (DMC3 BSEL 0 ∧ -DMC4 EVEN BUSY) the odd address flip-flop clock input is strobed. Since its data input is true (DMC3 NEXT2 DMC4 REQ) the flip-flop sets, asserting DMC1 K1/K3 ADR 34.

3.5.2 Read/Restore Operation

Initially, SBUS ADR 34–35 effectively initializes the address counter (described in subsection 3.4.1.1) to a value equal to the value of bits 34–35. As in 4 bus mode operation, the counter is locked to its current count until RD RS is returned, and at RD RS DLY time the counter is incremented by the enabled word request lines.

Since the starting address was 00, this value initializes the address counter output to 00 (-DMC3 NEXT 1 ∧ -DMC3 NEXT 2) on CLK 62 ns H. The counter output is coupled to the bus decode logic enabling both even bus functions (DMC3 BSEL 0 and DMC3 BSEL 2). BSEL 0, 2 are then latched into the bus selection latches (DMC2 THIS = 0, 2). Meanwhile, the word request line states are stored in the word request logic (described in Subsection 3.4.1.3) to be cleared at each ADR ACK time, disabling the request even or odd functions which, in turn, disable the even/odd request cycle mixers as required (Figure 3-9).

SBUS START A/B initiates the start sequence and after a delay sets DMC4 CYC START. Both cycle request inhibit functions from the Request/Acknowledge timing logic are false (-DMC5 RQ INH EVEN ∧ -DMC5 RQ INH ODD). The request cycle mixer select inputs are set for code -01, allowing the even and odd request functions to the request cycle gating. At this point, all cycle request requirements are satisfied and a cycle request is issued over buses K0–3. DMC1 K0–3 RQ CYC, together with the address and request types, are transferred through their corresponding logic level translators to the storage modules. In addition, the 20 high-order address bits are strobed into the address buffer by DMC4 START.

Since only storage modules on the K0 and K1 buses are interleaved, an even word response is expected only over the K0 bus and an odd word response only over the K1 bus. The other even and odd buses will be effectively ignored since no storage module should respond on those buses.

Following the cycle request the addressed storage module responds to the DMA with ADR ACK. As in 4 bus mode, the acknowledge function is coupled through the translator and sets the pre-synchronizer flip-flop MA4 WREN, disabling the KBus 0 cycle request. It is then clocked through the synchronizer, gated with DMC3 BSEL 0, and sets DMC3 ACKN. The ACKN function is then transferred on a phase A/B clock as SBUS ACKN A/B.

K0 ADR ACK also generates the even bus inhibit function (DMC1 EVEN INH) that is applied to the enable input of the request cycle mixer, effectively disabling the even portion of the request cycle mixer. DMC3 ACK DLY, set after K0 ADR ACK, together with the address counter output, sets DMC5 K0 INH EVEN disabling the even request cycle gating.

As in 4 bus mode, the K0 Word 0 read data is coupled through the translators and direct sets each corresponding bit-position flip-flop in the data buffer. K0 DATA EN then allows the read data to the SBus. Following K0 RD RS, which is transferred to the MBox as SBUS DATA VALID, DMC3 RD RS DLY effectively increments the address counter to 01 generating the address for the K1 bus storage module. The bus decoder, in turn, selects DMC3 BSEL 1 and 3 and DMC5 K0 CLR is generated, clearing the K0 data buffer and the K0 synchronizer flip-flop. With K0 ADR ACK reset, DMC1 INH EVEN goes low removing the inhibit from the even request cycle mixer inputs.

Meanwhile, the ADR ACK from the K1 bus has been stored in the synchronizer and the K3 data buffer and RQ CYC line are cleared. K1 ADR ACK performs functions identical to the ACKN function from the K0 bus, i.e., generates SBUS ACKN A/B and the odd bus inhibit function. In

addition, the odd bus DMC3 ACK DLY generates DMC5 RQ INH ODD, disabling the odd request cycle gating. The K1 Word 1 read data direct sets the K1 data buffer flip-flops; K1 DATA EN allows the read data to the SBus. K1 RD RS is transferred to the MBox as SBUS DATA VALID B. Again, DMC3 RD RS DLY effectively increments the address counter producing an output of 10.

At this point both DMC5 RQ INH EVEN and ODD reset, enabling the request cycle gating; the second half of the DMA cycle request is issued over the K0 and K2 buses. The Word 2 read data is gated through the data buffer to the SBus as on the first request. DMC3 RD RS DLY increments the address counter to 11. DMC5 K1 CLR is then generated, clearing the K1 data buffer and synchronizer flip-flop. In the meantime, -DMC5 RQ INH ODD again enables a cycle request on the odd buses (K1 and K3). As before, the bus decoder selects DMC3 BSEL 1 and 3. K1 ADR ACK is gated with BSEL 1 initiating the acknowledge sequence and transmitting SBUS ACKN. The K1 Word 3 read data is then gated through the data buffer to the SBus. The subsequent RD RS function again sets DMC4 COMPLETE, which is gated with -DMC4 ANY RQ resetting DMC4 CYCLING thus terminating the quadword read request.

3.5.3 Clear Write Cycle

As in 4 bus mode, the first word to be written must be loaded into the appropriate data buffer prior to issuing a cycle request to a storage module. When DMC1 K0-3 RQ CYC is generated, it direct sets MA2 CYC REQ. When DMC4 WR ONLY generates DMC5 WRITE EN and the subsequent DMC5 K0 WR CLK, the write data is strobed from the SBus into the data buffer and direct sets the MA5 BUFFER LOADED flag. BUFFER LOADED ∧ MA5 CYC REQ now satisfies the output KBus AND gates and generates MA2 K0-3 RQ CYC to the KBuses. DMC5 WRITE EN also initiates the internal write restart (DMC4 WR RS) and request complete (DMC4 COMPLETE) sequence on successive 62 ns clocks.

As in 4 bus mode, the responding storage module returns ADR ACK and sets the presynchronizer flip-flop MA4 K0 WREN. K0 WREN drives the WR PULSE logic generating the write pulse clearing function [N+1, MA5 K(I) WR PULSE]. This WR PULSE function, together with the K2 bus clear enable function (DMC5 2BUS CLR EN), generates the clear function, MA4 CLR C, to direct clear the K2 bus data buffer and disable KBUS2 RQ CYC. ADR ACK is coupled through the translator and synchronizer, gated with DMC3 BSEL 0, and initiates the ACKN sequence. The ACKN function is transferred on the Phase A/B clock as SBUS ACKN A/B. The MBox then places the Word 1 write data on the SBus. When MA5 WR PULSE is true (after KBUS0 ADR ACK) the Word 0 write data is gated from the data buffer to the KBus translator. WR PULSE is also coupled through its translator and transmitted as KBUS 0 WR RS. The K0 clearing logic generates MA4 CLR at MA5 DELAY ∧ -MA5 CLEAR DELAY, clearing its associated buffer and synchronizer.

K0 ADR ACK generates DMC1 INH EVEN which is applied to the enable input of the even request cycle mixer, disabling the even portions of the mixer. DMC3 ACK DLY together with the address counter output sets DMC5 RQ INH EVEN disabling the even request cycle gating. In addition, DMC3 ACK DLY effectively increments the address counter to 01 selecting bus functions BSEL 1 and 3.

A write clock is not generated during the K0 ACKN sequence (on ACK DLY) since DMC4 CYCLING has not yet set. DMC3 BSEL 1 ∧ -DMC4 ODD BUSY initiates a second ACKN sequence sending a simulated K1 ACKN to the MBox. In the meantime, the K1 DMC3 ACK DLY initiates a write clock/restart sequence strobing the Word 1 data from the SBus into the odd data buffers. The subsequent WR PULSE gates this data to the storage module. The K1 clearing logic generates MA4 CLR (MA5 DELAY ∧ -MA5 CLEAR DELAY) clearing its associated buffer and synchronizer.

Meanwhile, the MBox has received the K1 ACKN and placed the Word 2 data on the SBus. The DMC3 ACK DLY (of K1 ACKN) effectively increments the address counter to 10, selecting BSEL 0 and 2. DMC2 LOAD THIS DLY resets DMC5 RQ INH EVEN allowing DMC1 K0/2 RQ CYC to

the KBus0/2 translator AND gates. DMC3 BSEL 0 ∧ -DMC4 EVEN BUSY initiates a simulated K0 (Word 2) ACKN sequence, followed by a write clock, and write pulse/restart sequence as in Word 1. On the WR CLK, BUFFER LOADED sets, satisfying the translator AND gate and asserting KBUS0 RQ CYC.

On ACK DLY the address counter is incremented and a write clock sequence initiated. Meanwhile, the MBox has received the K2 ACKN and placed the Word 3 data on the SBus. At LOAD THIS DLY time (on K2 ACKN) DMC5 RQ INH ODD resets, allowing DMC1 K1/3 RQ CYC to direct set MA2 CYC REQ. As before, on WR CLK, BUFFER LOADED sets asserting KBUS1 RQ CYC. When the last busy function is cleared, it allows DMC4 CYCLING to reset on the following CLK 62 ns. With CYCLING reset, the address counter, request line, and type latches are deselected and conditioned for the next MBox request.

3.5.4 Non-Quadword Request

During a read or write cycle request for less than a quadword in 2 bus mode the DMA must generate unique timing functions during the time the unrequested word would normally be accessed. The timing problem is essentially the same for read or write cycle requests, since these particular functions are enabled only during the response portion of a request. The example used to illustrate the timing problem is a 3-word cycle request, starting at Word 0 and skipping Word 1. Detailed timing of the most significant functions is provided to illustrate read and write cycle timing for the 3-word request.

The timing problem logic compares the address counter output with its incrementing input. Should the counter output be even and its incrementing value also be even, the problem timer function (DMC3 PROBLEM TIMER) is set on DMC3 NEXT or -DMC2 LOAD THIS. (Note that the identical situation also exists for an odd counter output and odd incrementing value.) DMC3 PROBLEM TIMER, in turn, generates DMC3 PROBLEM which controls the data input gating of DMC3 ACK INH and DMC3 RD RS INH. DMC3 PROBLEM effectively holds DMC3 RD RS INH true for an additional two clock periods, thus allowing sufficient time for address line settling and address counter incrementing through Word 1, and K1 request cycle inhibiting and initiation of the second DMA half cycle.

3.6 1 BUS MODE OPERATION

The following subsections present a detailed logic description of a quadword request (starting at Word 0) in 1 bus mode. Those logic areas that are functionally identical to the 2 and 4 bus modes are not described in detail; however, reference is made to the appropriate subsections in the 2 and 4 bus mode descriptions.

The timing involved in 1 bus mode operations is the major difference from the other modes, i.e., cycle requests executed in this mode require extended cycle times and DMC3 PROBLEM TIMER and DMC3 PROBLEM are enabled during the DMA cycle. Furthermore, each word request requires one complete word cycle, i.e., one cycle is completed before the next request is issued.

Prior to initiation of a memory cycle request, the DMA operational mode is set to 1 bus mode and verified by a Function 0 diagnostic cycle issued by the MBox (Subsection 3.8.2). A quadword memory cycle is initiated when the MBox asserts:

a. Address lines - SBUS ADR 14–35

b. Read/Write request lines - SBUS RD RQ/WR RQ

c. Word request lines - SBUS RQ 0–3

d. Request cycle start - SBUS START

As in 2 bus mode SBUS RD RQ/WR RQ are coupled through the cycle type mixer setting the appropriate cycle type flip-flop. SBUS RQ 0–3 line states are applied to the word request logic. SBUS ADR 14–33 are coupled into the KBus translators, and bits 34–35 are applied to and initialize the address counter to determine the cycle starting address. SBUS START A/B generates the cycle setup delay time.

Note that in this mode the address counter output has no effect on bus port selection. The operational mode functions are applied to the bus decode logic and enable all bus selection functions (DMC3 BSEL 0–3) throughout the quadword request.

3.6.1 Read/Restore Operation

As in 2 and 4 bus modes, SBUS ADR 34–35 effectively initialize the address counter (described in Subsection 3.4.1.1) to a value equal to the value of bits 34–35. Since Word 0 is the first word requested, the address counter is initialized to 00 (-DMC3 NEXT1 ∧ -DMC3 NEXT 2) on CLK 62 ns. As in the other modes, the counter is locked to its current count until RD RS DLY is received, at which time the counter is incremented by the enabled word request lines. Meanwhile, the word request line states are stored in the word request flip-flops (described in Subsection 3.4.1.4) and cleared at each DMC3 ACK DLY time. The word request logic output provides one of the cycle enabling inputs (DMC1 CYC RQ) to the request cycle mixer and remains true until the last word is retrieved.

SBUS START A/B initiates the start sequence and after a delay sets DMC4 CYC START; the cycle request inhibit functions are false (-DMC5 RQ INH EVEN ∧ -DMC5 RQ INH ODD). The request cycle mixer selector inputs are set for code 10 allowing DMC4 ANY RQ to the request cycle gating. At this point, all cycle request requirements are satisfied, and a cycle request is issued over all buses (K0–3). In addition, DMC3 BSEL 0–3 are enabled for the storage module response and remain enabled throughout the quadword request.

The addressed storage module responds with ADR ACK disabling RQ CYC over K0 bus. Since the addressed location is assumed to be on the K0 bus, K0 ADR ACK and BSEL 0 initiate the acknowledge sequence and return SBUS ACKN A/B to the MBox. (Note that with BSEL 0–3 enabled a response from a storage module on any other bus would also initiate the ACKN sequence.) K0 ADR ACK generates the even and odd request inhibit functions (DMC1 INH EVEN, DMC1 INH ODD) disabling the request cycle mixer and terminating the cycle requests over all the buses. At DMC3 ACK DLY time, DMC5 RQ INH EVEN and ODD set, disabling the request cycle input gating.

At DMC3 NEXT time DMC3 PROBLEM TIMER and DMC3 PROBLEM are generated and perform the same functions as in 2 bus mode. As in 2 and 4 bus modes, the K0 Word 0 read data is coupled through the translator and direct sets each corresponding bit-position flip-flop in the data buffer. K0 DATA EN then allows the read data to the SBus. K0 RD RS generates SBUS DATA VALID A/B which is returned to the MBox on a Phase A/B clock.

At DMC2 LOAD THIS DLY time both request inhibit functions are reset (-DMC5 RQ INH EVEN ∧ -DMC5 RQ INH ODD). DMC5 K0 CLR clears the data buffer and the ACK and RD RS flip-flops in the K0 synchronizer. With K0 ADR ACK reset, DMC1 INH EVEN and DMC1 INH ODD go low enabling the request cycle mixers. A second word request cycle is then issued over all KBuses. The subsequent Word 1 ADR ACK and RD RS are received over K0 bus. The Word 1 read data is enabled through the K0 data buffer to the SBus as before.

The remaining words are accessed and retrieved over the K0 Bus in exactly the same manner and timing as the Word 0 and 1 data. As Word 3 is retrieved the last word request flip-flop is cleared (-DMC4 ANY RQ) and coupled with DMC4 COMPLETE resets DMC4 CYCLING, effectively terminating the quadword request.

3.6.2 Clear/Write Operation

As in the other operational modes, the first word to be written must be loaded into the appropriate data buffer prior to issuing a cycle request to a storage module. The buffer loading sequence is basically the same as described in Subsection 3.5.3 for 2 bus mode. With BUFFER LOADED true, MA2 K0–3 RQ CYC is issued over all KBuses.

Since the location accessed is assumed to be on KBUS0, KBUS0 ADR ACK is coupled through the translator and synchronizer, disabling the cycle request at its synchronizer. ADR ACK is gated with BSEL 0 initiating the ACKN sequence and transferred to the MBox as SBUS ACKN A/B. The MBox then places the Word 1 write data on the SBus data lines. When WR PULSE is true (after ADR ACK) the Word 0 write data is gated from the data buffer to the KBus. WR PULSE is also transferred over the KBus as KBUS0 WR RS.

K0 ADR ACK generates DMC1 INH EVEN and DMC1 INH ODD, which disables the request cycle mixers and, in turn, the cycle requests over the K1–3 buses. DMC3 ACK DLY sets both DMC5 RQ INH EVEN/ODD disabling the request cycle output gating. DMC3 NEXT increments the address counter and sets DMC3 PROBLEM TIMER enabling DMC3 PROBLEM to perform the same function as in 2 bus mode. With DMC2 LOAD THIS DLY set, both request inhibit functions are reset (-DMC5 RQ INH EVEN ∧ -DMC5 RQ INH ODD). At DMC2 LOAD THIS NOW time the clear functions (K0–3 CLR) are enabled, direct clearing the data buffer and bus functions. With K0 ADR ACK reset, DMC1 INH EVEN and DMC1 INH ODD go low, enabling request cycle mixers and, in turn, the request cycle gating.

However, as with the first word cycle, the Word 1–3 write data must be loaded into the data buffer prior to issuing the cycle request over the KBuses. The exectuion and timing of the remaining word requests are identical to the first. As in 2 and 4 bus modes, the operation is effectively terminated when all requests are cleared (-DMC4 ANY RQ) and all buses are not busy (-MA5 KN BUSY).

3.7 IDLE AND POWER CLEAR

During periods when the DMA is not executing memory cycles requests the data buffers are held clear and the synchronizer request functions disabled. Each pair of data inputs to the data buffer clear flip-flops (DMC5 K0–3 CLR) are ORed. One side of the OR enables the flip-flop data input during DMA processing (DMC2 THIS = 0–3 ∧ DMC2 LOAD THIS NOW), and the other enables the data input when the DMA is not processing cycle requests (-DMC4 CYCLING ∧ -DMC4 START). Thus, with no cycle requests from the MBox the four data buffers are direct cleared by DMC5 KN CLR and held in the reset state until a cycle request is received. In addition, the same function clears the synchronizer cycle request flip-flop (MA2 CYC REQ) disabling the input gating of the request line level translation on each KBus.

To eliminate the possibility of power line transients disturbing the DMA during power up and power down, a CROBAR function is applied to the KBus request logic. With CROBAR true, the level shift gating for all KBus RQ CYC lines is disabled. As shown in Figure 3-13, Q1 is normally cut off and Q2 is conducting, enabling one side of the 8881 level shift output gate. When CROBAR goes true, Q1 conducts, driving Q2 into cutoff forcing the junction of R1 and R2 to -5.2 V and disabling one leg of the level shift output gate.

3.8 ERROR DETECTION AND REPORTING

The parity error logic detects and stores parity errors encountered during all DMA cycle operations. The error logic also has the capability to detect, store, and report NXM references after the first word is acknowledged. The diagnostic logic, which is exercised during diagnostic cycle execution, provides the error reporting and diagnostic capabilities. The diagnostic logic is also used to set the DMA operational bus modes. Two additional diagnostic features are provided: Single Step operation and

Loop Back mode. With the DMA operating in Single Step, cycle operations are executed at a reduced clock rate and may be stopped at any point for diagnostic purposes. Loop Back mode allows exercising of the DMA control and data paths without accessing the addressed storage module.

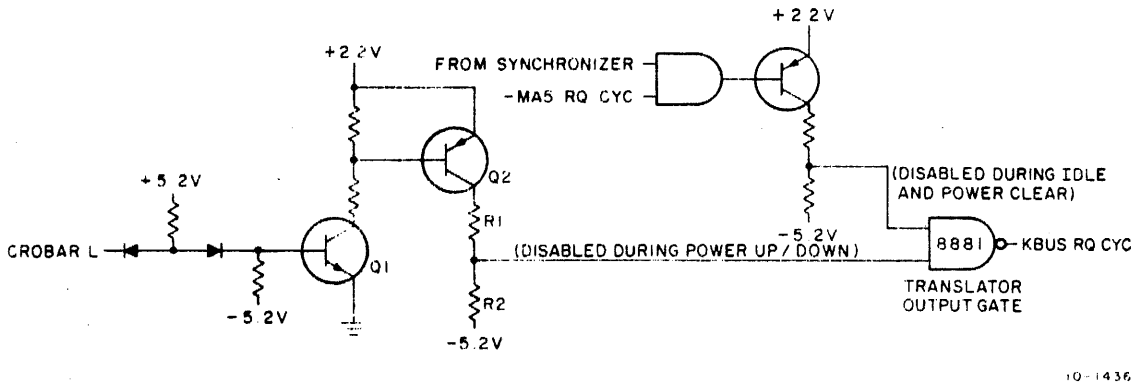


Figure 3-13 Simplified Crobar Logic

3.8.1 Error Detection

Address and data parity is checked on all DMA operations. Should a parity error be detected an associated flag is set in the error status register. On address and data parity errors the current address together with the word request lines and cycle type are retained in the request error register. Note that the correct error address is maintained by retaining the equivalent current address (DMC2 THIS 1 and DMC 2 THIS 2) together with the 20 high-order address bits (SBUS 34–35). The same components are also retained in the request error register if an NXM reference is detected. In addition, the NXM flag is set in the error status register.

During no-error operations the request error register strobe (DTR2 ADR CLK) is generated by the negated error flags at each ACK DLY time. Thus, the ADR CLK continually strobes the current address request lines and request type into the register. Should a parity error or NXM be detected the ADR CLK is inhibited, retaining the current (error) address and request components in the register.

The contents of both registers can be read when the MBox issues a diagnostic cycle. However, error flags set during an SBus operation are not automatically cleared on the next memory reference. Once set, the error flags can be cleared only by the execution of a Function 0 diagnostic cycle with appropriate bits set.

3.8.1.1 Address Parity – All address bits including parity, word request lines, and cycle request lines are combined and tested for odd parity in a parity tree containing four, 9-bit parity generators. Should a parity error be detected, the even output of the last parity generator (DTR2 ADR PAR EVEN) is true. As shown in Figure 3-14, DTR2 ADR PAR EVEN together with DMC4 BUS START generates DMC4 ADR PAR ERR. ADR PAR ERR together with ADR PAR EVEN sets the address parity error flag (DTR2 PAR ADR) in the error status register, as well as prevents the setting of DMC4 CYC START, thus inhibiting a cycle request over the selected KBus.

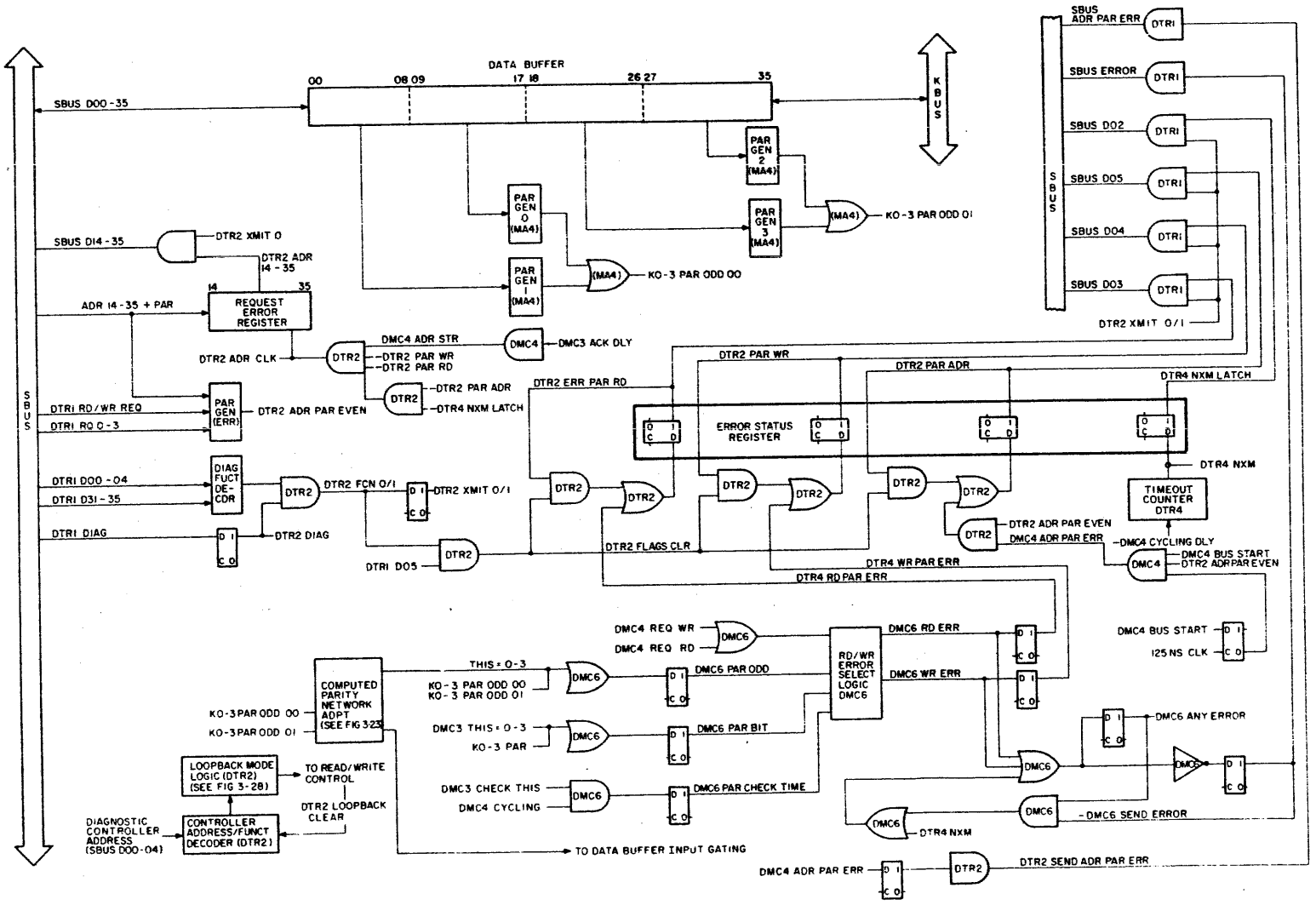


Figure 3-14 Simplified Error Logic and Timing

At DMC4 BUS START time, with DTR2 ADR PAR EVEN true, DMC4 ADR PAR ERR is true. In turn, ADR PAR ERR sets the send address error flip-flop (DTR2 SEND ADR PAR ERR), which sets on a 125 ns clock. Once set, DTR2 SEND ADR PAR ERR is coupled to and enables SBus driver SBUS ADR PAR ERR, transferring an address parity error indication to the MBox. Meanwhile, DMC4 BUS START has set a delay flip-flop, which disables DMC4 ADR PAR ERR removing the data input from DTR2 SEND ADR PAR ERR. On the next 125 ns clock DTR2 SEND ADR PAR ERR resets, in turn disabling SBUS ADR PAR ERR. Note that ADR PAR ERR was true for one clock period. In the meantime, DTR2 PAR ADR inhibits the request error register clock (DTR2 ADR CLK). The current address, word request lines, and request type are thus retained in the register.

3.8.1.2 Read Data Parity - Data parity is checked on all read/restore and read portions of read/modify/write cycle requests (Figure 3-14). In addition, odd parity is computed (but never modified) on the read data in the DMA; should a discrepancy exist between the memory and computed parity, an error flag is set in the error status register.

The incoming parity (K0–3 PAR) is coupled to the parity bit flip-flop (DMC6 PAR BIT). DMC6 PAR BIT will set if the incoming parity bit is ON. As shown in Figure 3-14, the 36-bit data word is separated and fed to two pairs of 9-bit parity generators. The outputs of each pair of generators is then compared in a bus-associated parity computing network. The entire network accepts the parity generators from the four KBuses (KN PAR ODD N). Each bus pair parity functions are compared in the network and coupled with the appropriate latched bus function (DMC2 THIS = 0–3) to the odd parity flip-flop (DMC6 PAR ODD).

If the incoming parity is correct and no errors are encountered in the computing network, the network outputs are complementary. An example of network operation is provided in Figure 3-15. The example illustrates generation of computed parity from a data word and its parity.

Parity check time (DMC6 PAR CHECK TIME) is generated by DMC3 CHECK THIS after the outputs of DMC6 PAR ODD and DMC6 PAR BIT are compared. Should both flip-flops be set/reset (indicating a noncomplementary parity network output condition), the read error indicator (DMC6 RD ERR) is generated. On the next 62 ns clock DMC6 RD ERR sets the read parity error flip-flop (DTR4 RD PAR ERR) and the error flip-flop (DMC6 ANY ERROR). On the following 125 ns clock, DTR4 RD PAR ERR sets error flag DTR2 PAR RD, which in turn inhibits the error address clock (DTR2 ADR CLK) retaining the current address, word requests, and cycle type in the request error register. On the same 125 ns clock, DMC6 RD ERR sets DMC6 SEND ERROR enabling SBUS ERROR and transferring an error indication to the MBox.

The latching function for DMC6 ANY ERROR is provided by -DMC6 SEND ERROR. Likewise, the latching function for DMC6 SEND ERROR is provided by DMC6 ANY ERROR. When DMC6 SEND ERROR sets, the latch function is disabled, resetting TIM ANY ERROR on the next 62 ns clock. The latch on DMC6 SEND ERROR is then disabled causing it to reset on the following 125 ns clock, providing a send function duration of one clock period.

3.8.1.3 Write Data Parity - The hardware implementation for write data parity errors is similar to the read parity logic; the timing is different, however. Data parity is checked on all clear/write and write portions of read/modify/write cycle requests.

As in the read cycle, the data is checked in the four parity generators and compared in the parity computing network after the data has been loaded into the appropriate buffer. At DMC6 PAR CHECK TIME, generated by DMC3 CHECK THIS (set by the 62 ns clock after DMC4 WR RS sets), DMC6 PAR ODD and DMC6 PAR BIT are compared. Should both flip-flops be set/reset, DMC6 WR ERR is set. On the next 62 ns clock TIM WR ERR sets the write parity error flip-flop (DMC4 WR PAR ERR) and the error flip-flop (DMC6 ANY ERROR). On the following 125 ns clock, DTR4

WR PAR ERR sets error flag DTR2 PAR WR, which in turn inhibits DTR2 ADR CLK retaining the current address, word requests, and cycle type into the request error register. On the same 125 ns clock, DMC6 WR ERR sets the DMC6 SEND ERROR flip-flop enabling SBUS ERROR and transferring an error indication to the MBox.

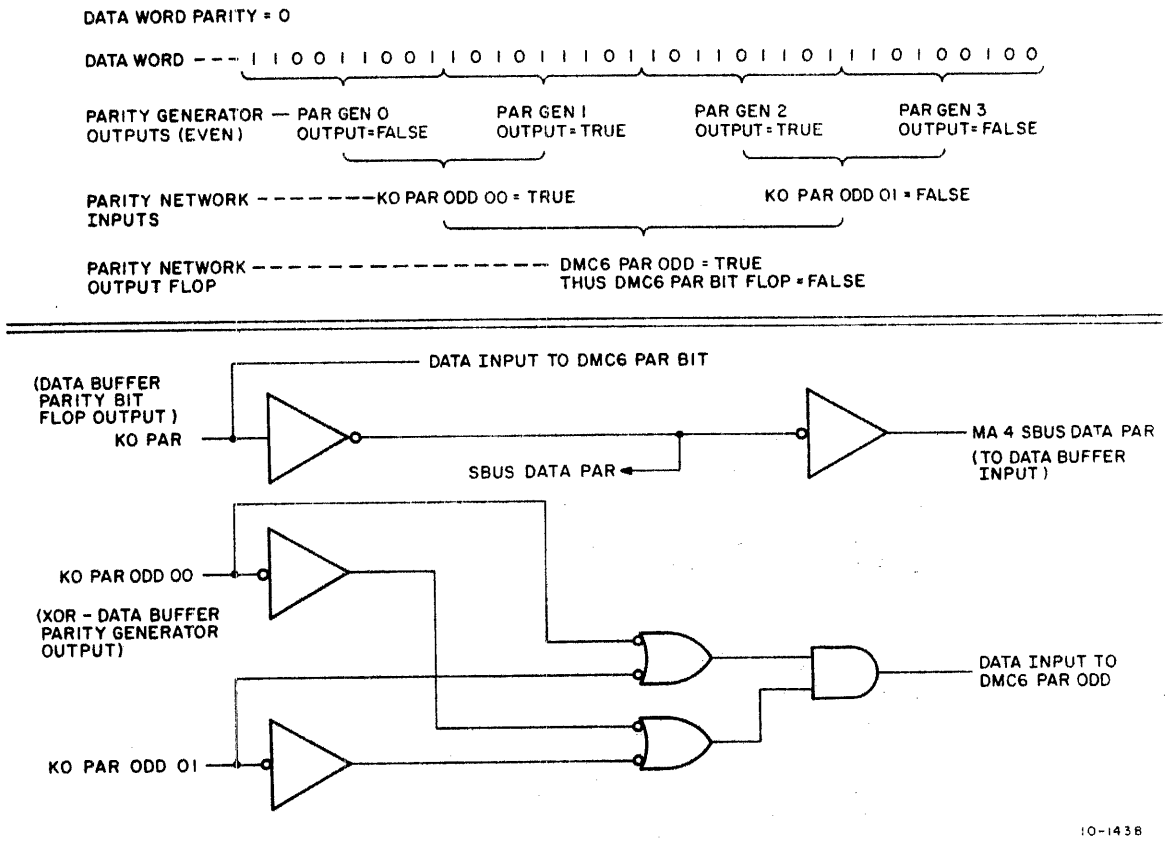


Figure 3-15 Computed Parity Network Example

3.8.1.4 Nonexistent Memory Reference – The DMA provides checking for an NXM reference after the first ACKN. If an ADR ACK is not received from a subsequently selected storage module (indicating presence of memory), a time-out sequence (equal to 225 external clock periods) terminates the operation, thus preventing a hung condition on the KBus. The MBox initiates its time-out sequence at the assertion of SBUS START A/B. The DMA does not initiate its time-out sequence until after it has received an ACKN on the first word accessed. Should an NXM time-out occur, the DMA sets the NXM flag and loads the request error register with the current address, word requests, and cycle type.

During DMA operations in which an NXM is encountered, certain data words are lost to the system depending on the current bus mode. During 1-, 2- and 4-bus read and 1- and 4-bus write operations the word not acknowledged and its subsequent words are lost. Obviously, if no ACKN is received for the fourth (or last) word in any bus mode, only that word is lost. However, since alternate buses are used, the 2-bus write operation is a special case. If no ACKN is received on the second word, only words 2 and 4 are lost. If no ACKN is received on the third word only word 3 is lost.

The NXM logic in the DMA consists of a pair of cascaded synchronous up/down counters, a pair of D type flip-flops, and associated gates. As shown in Figure 3-16, the counters are connected such that the carry output of the first counter forces an up-count in the second counter.

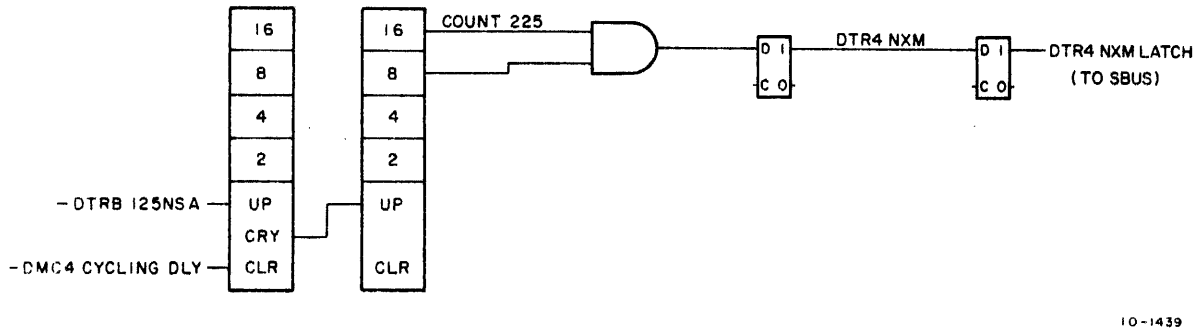


Figure 3-16 NXM Counter Logic

After reception of the first accessed word ACKN (DMC4 ACK DLY) DMC4 CYCLING sets, which in turn sets DMC4 CYCLING DLY on the following 62 ns clock. With DMC4 CYCLING DLY true, the clear function is removed from the counter. The counter then starts up-counting on successive 125 ns clocks. After each 16 counts of the first counter, its carry output forces one up-count in the second counter. If no further ACKNs are received, the count (time-out sequence) continues for 225 counts. At count 225, the two high-order counter outputs set the NXM error flip-flop (DTR4 NXM).

DTR4 NXM sets the NXM flag (DTR4 NXM LATCH) in the error status register. Note that the DTR4 NXM LATCH flag is not cleared until the MBox executes a Function 0 diagnostic cycle resetting the designated flip-flops. DTR4 NXM sets DMC6 SEND ERROR, which generates SBUS ERROR to the MBox. As with parity errors, NXM inhibits DTR2 ADR CLK retaining the current address and enabled request lines in the request error register. In addition, DTR4 NXM direct clears DMC4 CYCLING, which eventually resets DMC4 CYC START removing the current KBus request, preventing a hung bus condition, and inhibiting any further requests.

3.8.2 Diagnostic Cycle Logic

The diagnostic logic implements those functions specified by a diagnostic cycle issued by the MBox. The logic provides the reporting capability allowing the MBox to retrieve the contents of the parity error and request error registers, as well as execute a loop-around operation.

During diagnostic cycle execution, all communications between the MBox and the DMA are performed over the SBus data lines. The two diagnostic function cycles are initiated and synced to the Phase A clock with a complete cycle requiring four Phase A clock periods. During the first two clock periods the MBox sends out a DMA address, function code, and information specified by the function code. During the second two clock periods, the DMA sends information back to the MBox using the data lines as specified by the function code. The field significance of each diagnostic function cycle is summarized in Figure 3-17 and cycle timing is illustrated in Figure 3-18.

3.8.2.1 Diagnostic Cycle Operation – During the first half (to DMA) of either diagnostic cycle, the four high-order bits provide the DMA address. The decoded address function enables the operational mode decoding logic as well as the error clear and the error transmit functions.

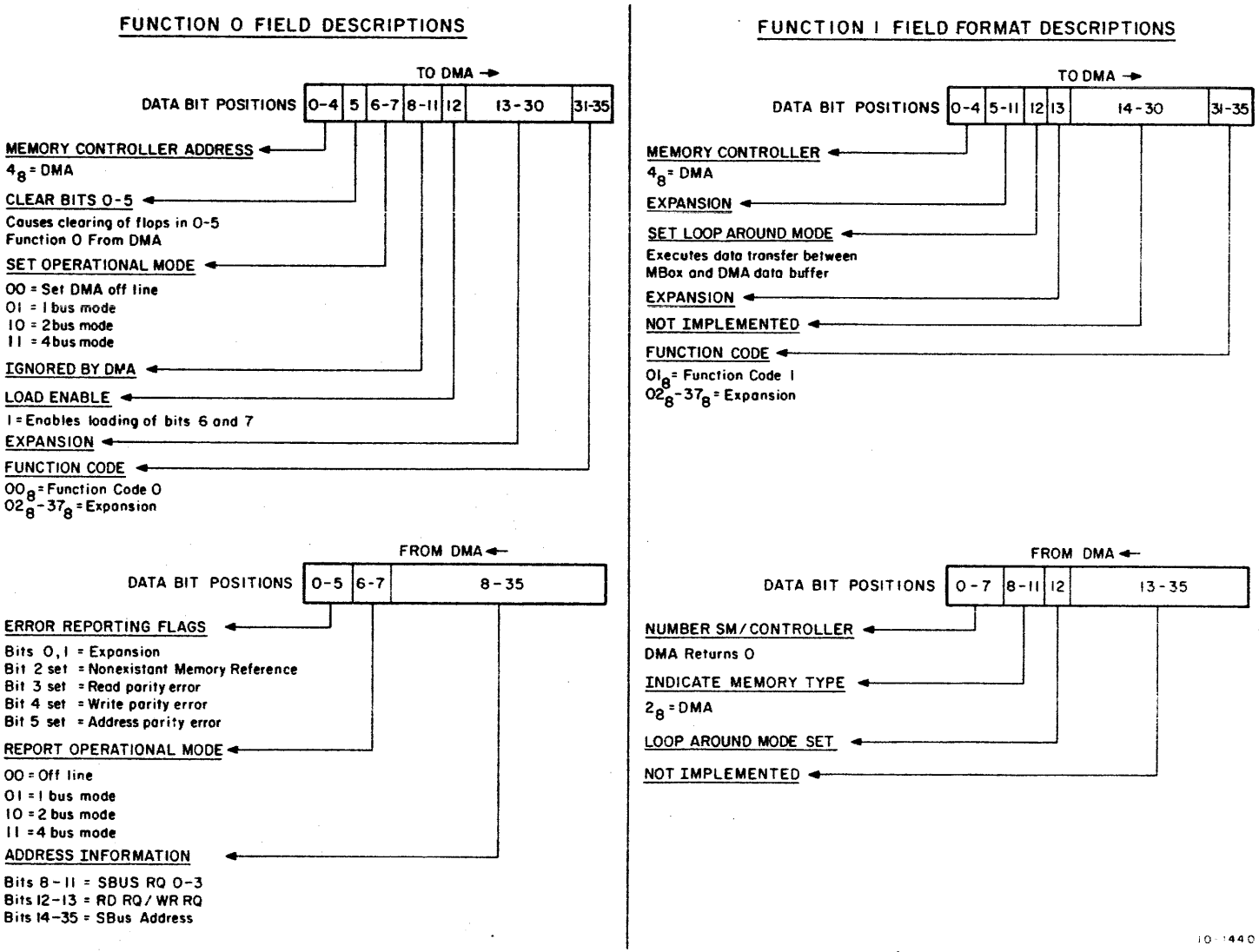


Figure 3-17 Diagnostic Cycle Summary

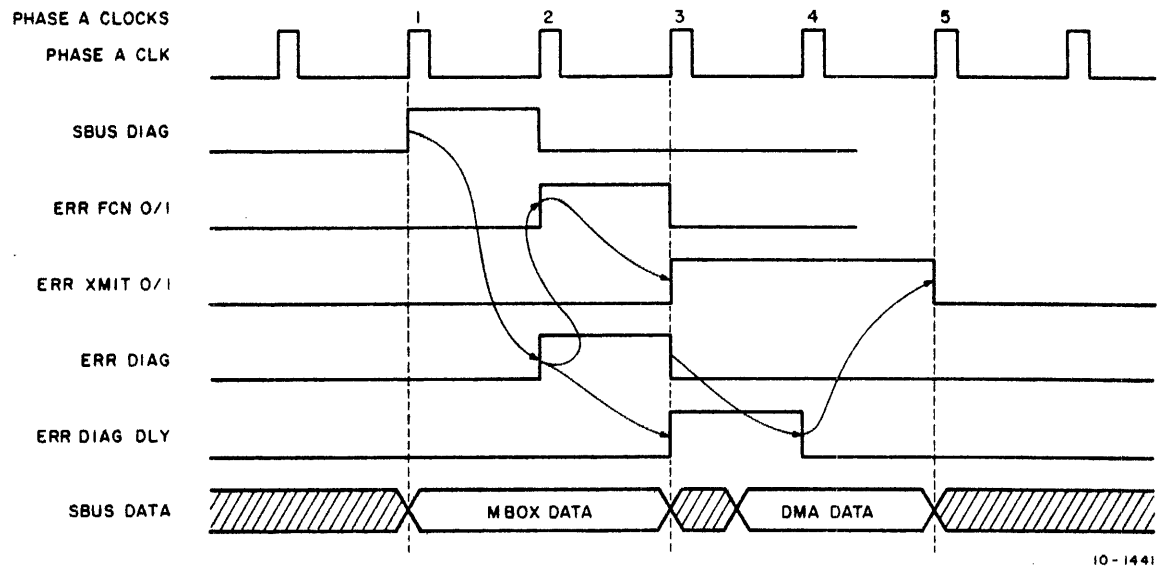


Figure 3-18 Diagnostic Cycle Timing

At cycle initiation time, the MBox asserts the appropriate data lines together with the diagnostic function (SBUS DIAG). SBUS D00–04 and D31–35 are coupled to the diagnostic function decoder (DMA address = 48); SBUS DIAG is coupled to the data input of the diagnostic flip-flop (DTR2 DIAG) (Figure 3-14). On the following CLK 125 ns A clock DTR2 DIAG sets, enabling the function decoder outputs (DTR2 FCN 00 or DTR2 FCN 01). These output functions in turn set their associated error transmit functions (DTR2 XMIT 0 or DTR2 XMIT 1). With either XMIT true, the SBus is enabled with the requested error flags, addressing information, or operational bus mode.

Since SBUS DIAG is negated one Phase A clock period after cycle initiation (Figure 3-18), DTR2 DIAG is reset on the following CLK 125 ns A clock. DTR2 DIAG then disables the address decoder causing the enabled DTR2 XMIT to reset, disabling the DMA SBus data lines. Meanwhile (four Phase A clock periods after cycle initiation) the MBox strobes the SBus data lines retrieving the enabled DMA error, address, or operational mode functions. As indicated in Figure 3-17, those functions retrieved by the MBox are determined by the diagnostic cycle type initiated.

Both error registers and operational mode flip-flops are direct cleared when the MBox issues a master clear function. In addition, the parity error register is cleared when the MBox issues a Function 0 cycle with its CLEAR bit set. The DMA decoded diagnostic function, DTR2 FCN 00, together with DTR1 D05 disable the data input gating of the parity error register (DTR2 FLAGS CLR). The register is then reset on the following CLK 125 ns A clock.

Either of the three DMA operational modes is set when the MBox issues a Function 0 cycle. The bus mode set is determined by the bit configuration of the operational mode field as described in Subsection 3.3. After address decoding, DTR2 FCN00 is ANDed with the load enable field (SBUS D12) generating DTR2 BUS SET. BUS SET, together with the operational mode field (SBUS D06–07), set the desired mode flip-flops.

3.8.2.2 Single Step Operation – This operation provides the system with the capability to single-step through read/write memory cycles for diagnostic purposes. For this function the central processor is operating in its diagnostic mode at a reduced clock rate with the DMA set in 1 bus mode. Although during single-step operations the external clock is running at a reduced rate, DMA operations are essentially executed as with normal cycle requests.

Because of the reduced clock rate the clear/write operation is a special case, an example of which is in the following discussion. As in the normal write operation, the data buffer is loaded prior to issuing a cycle request over the KBuses. When the write request is issued, the addressed storage module immediately clears the addressed location. At this point the storage module waits, expecting to receive the write data with WR RS. Should the storage module time-out before the DMA data is placed on the KBus, that data is lost. To prevent this situation, the DMA slow clock logic generates the necessary functions to gate the data into the data buffer, issue the cycle request, place the data on the KBus, and generate WR RS (after receiving ADR ACK from the address storage module) even with the clock stopped. Note that data loss is not a problem with a read cycle operation since the storage module normally restores the data in its latches after it disconnects from the memory bus.

The reduced clock rate is detected by the slow clock generating logic. As shown in Figure 3-19 this logic consists of two cascaded flip-flops that monitor the Phase A clock. The first flip-flop in the cascade is an intergrating one-shot clocked by CLK 125 ns A. Since the clock is retriggered at a rate faster than its time-out rate (as determined by the external RC timing components) the flip-flop remains set. Should the Phase A clock rate fall below 1 μs (or stop) the one-shot is reset which, in turn, sets the Slow Clock flip-flop (DTR4 SLOW CLK). DTR4 SLOW CLK is gated with -DMC4 RD (for a clear/write) or CPU DATA VALID (for the write portion of a read/modify/write) to generate the slow write restart function (DMC2 SLOW WR RS). DMC2 SLOW WR RS now replaces the normal restart function, DMC4 WR RS, to complete the clear/write request.

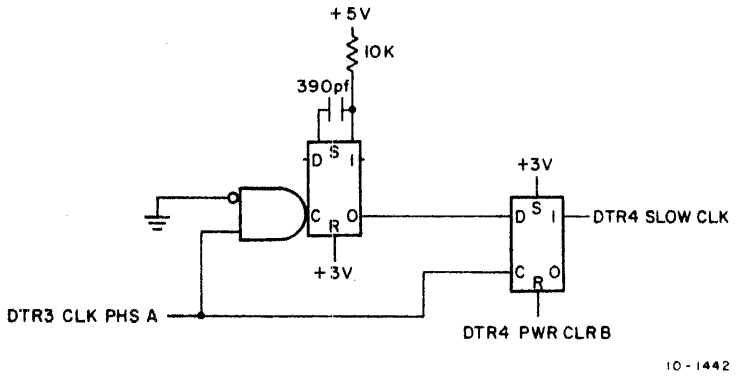


Figure 3-19 Slow Clock Detection Logic

At this point DMC2 SLOW WR RS follows two logic paths:

a. Direct sets DMC5 K0–3 WR CLK strobing the SBus data into the selected data buffer and setting BUFFER LOADED, which subsequently generates the KBUS RQ CYC.

b. Drives a delay line network setting DMC5 PULSE LIMITER, which direct clears WR CLK.

MA4 WR EN is generated on the reception of ADR ACK, driving a delay line network producing MA5 WR PULSE. As in a normal write operation WR PULSE gates the data buffer output to the KBus and enables WR RS on the KBus. Note that the slow clock portion of the logic is not dependent on the speed of the clock for its operation. Thus, a write operation can be completed (after ADR ACK) without loss of data with the DMA clock stopped.

3.8.2.3 Loopback Mode Operation – The loopback mode diagnostic feature allows checking of the DMA read and write control and data paths. The feature is executed in 4 bus mode with a one-word request. Initially, the loopback mode is set by a diagnostic cycle. The MBox then issues a normal write cycle, loading the write data into the appropriate buffer; however, the cycle request is inhibited from the KBus. At completion of the write operation the selected data buffer is not cleared. (Note that the DMA generates the loopback K and SBus response functions, e.g., ACKN, RD RS, etc.) A subsequent read cycle retrieves the data in the DMA buffer for analysis in the MBox. For a no-fault condition the retrieved data should be identical to that deposited in the buffer during the previous write cycle.

To set loopback mode, the MBox issues a Function 1 diagnostic cycle with SBUS D12 set. The controller address and function type are decoded in the diagnostic function decoder. With SBUS D12 enabled, the loopback flip-flop is set causing the DMA to enter the loopback mode (Figure 3-20). DTR2 LOOPBACK sets both inhibit flip-flops (DMC5 RQ INH EVEN/ODD) disabling the request cycle mixer output gating (-DMC1 KN RQ CYC) to the KBuses. In addition, LOOPBACK disables the select inputs of the RQ CYC mixers.

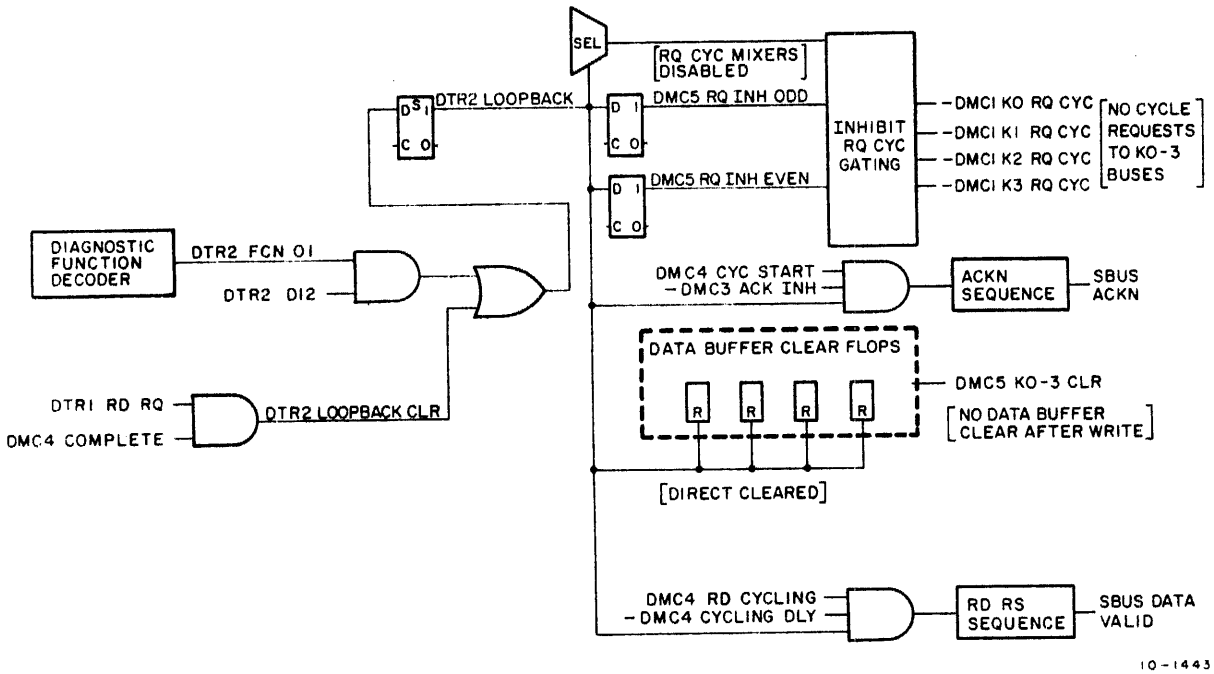


Figure 3-20 Loopback Mode Logic

The MBox then issues a normal write cycle. As in a normal write cycle, the DMA data buffer is loaded prior to issuing the request over the KBus. DMC4 CYC START is generated during the start sequence; however, with the request cycle mixers and gating already disabled, no request is issued. At this point, DMC4 CYC START, -DMC3 ACK INH, and DTR2 LOOPBACK direct set DMC3 ACKN initiating the internal acknowledge sequence returning SBUS ACKN to the MBox.

In a normal operation, the selected data buffer is cleared when the storage module disconnects from the KBus to complete the write cycle. However, in loopback mode DTR2 LOOPBACK is gated into the Read/Write Control as DMC5 LOOPBACK to hold the data buffer clearing flip-flops direct cleared, thus preserving the write data.

The MBox then issues a read request cycle. DMC4 CYC START together with DTR2 LOOPBACK generates the simulated ACKN. At the same time DMC1 KN DATA EN is enabled, allowing the previously written data buffer content to the SBus. DMC4 CYCLING B together with DTR2 LOOPBACK and -DMC4 CYCLING DLY direct sets DMC3 RD RS producing a simulated RD RS in the same manner as the simulated ACKN. The subsequent restart sequence generates DMC4 COMPLETE. DMC4 COMPLETE and the read request (DTR1 RD RQ) generates the loopback mode clearing function (DTR2 LOOPBACK CLR). LOOPBACK CLR disables the latch circuit allowing DTR2 LOOPBACK to reset disabling the loopback mode. In the meantime, DMC4 COMPLETE resets DMC4 CYCLING effectively terminating the read request cycle and clearing the data buffer with DMC5 KN CLR.

APPENDIX A
ABBREVIATIONS AND MNEMONICS

ACK - Acknowledge
ACKN - Acknowledge
ADR - Address
B - Buffered
BSEL - Bus Select
CLK - Clock
CLR - Clear
CYC - Cycle
D - Data
DIAG - Diagnostic
DLY - Delay
EN- Enable
ERR - Error
EXT - External
FCN - Function
GEN - Generator
INH - Inhibit
K - KBus
MADR - Memory Address
MEM - Memory
NS - Nanoseconds
NXM - Nonexistent Memory
PAR - Parity
PHS - Phase
PWR - Power
RD - Read
REQ - Request
RQ - Request
RS - Restart
SM - Storage Module
STR - Strobe
WR - Write
WREN - Write Enable
XMIT - Transmit

INDEX

DMA20 MEMORY BUS ADAPTER
UNIT DESCRIPTION
EK-DMA20-UD-002

Reader's Comment

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 do you find with the manual? _____

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

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

Would you please indicate any factual errors you have found. _____

Please describe your position. _____

Name _____ Organization _____

Street _____ Department _____

City _____ State _____ Zip or Country _____

CUT OUT ON DOTTED LINE

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
Technical Documentation Department
Digital Park, PK3-2
Maynard, Massachusetts 01754