Signetics

VMXbus Specification Manual Rev A

Oct. 1983

VMXbus

SPECIFICATION MANUAL

PUBLISHED BY

THE

VMEbus MANUFACTURERS' GROUP

Revision A

October, 1983

NOTE: While considerable effort has been expended to make this document comprehensive, reliable and unambiguous, it is still being published in preliminary form for public study and comment. Please direct any comments, or suggestions in writing to the VMEbus Manufacturers' Group secretary:

Rick Main
Signetics Corporation
811 E. Arques Avenue
P.O. Box 409
Sunnyvale, California 94086

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1 VMXbus SPECIFICATION OBJECTIVES

1.2 VMXbus STRUCTURE
1.2.1 Physical Structure
1.2.2 Functional Structure

1.3 SIGNAL LINE STATES

1.4 USE OF THE ASTERISK (*)
1.4.1 Strobe lines
1.4.2 Bused Signal Lines
1.4.3 Other Signal Lines

CHAPTER 2 VMXbus DATA TRANSFER BUS

2.1 SIGNAL LINES

2.2 MASTER AND SLAVE MODULES

2.2.1 D16 MASTERS and D16 SLAVES
2.2.2 D32 MASTERS and D32 SLAVES
2.2.3 Mixing D16 and D32 MASTERS and SLAVES

2.2.4 SEQ MASTERS and SEQ SLAVES
2.2.5 Mixing SEQ and non-SEQ MASTERS and SLAVES

2.2.6 IMA MASTERS

CHAPTER 3 EXAMPLES OF DATA TRANSFER BUS OPERATION

3.1 DATA TRANSFER CYCLE TYPES

3.1.1 Single Write Cycle
3.1.2 Single Read Cycle
3.1.3 Sequential Write Cycle
3.1.4 Sequential Read Cycle
3.1.5 Indivisible Single Address Cycle
3.1.6 Indivisible Multiple Address Cycle

CHAPTER 4 MASTER AND SLAVE TIMING SPECIFICATIONS

4.1 General Information

4.2 Single Write Cycle

4.3 Single Read Cycle

4.4 Sequential Write Cycle

4.5 Sequential Read Cycle

4.6 Indivisible Single Address Cycle

4.7 Indivisible Multiple Address Cycle

4.8 Withdrawn Cycle

CHAPTER 5 VMXbus ARBITRATION PROTOCOL

5.1 SIGNAL LINES

5.2 EXAMPLES OF ARBITRATION OPERATION
5.2.1 PRIMARY MASTER Grants VMX-DTB To SECONDARY MASTER
5.2.2 PRIMARY MASTER Receives VMX-DTB From SECONDARY MASTER

5.3 ARBITRATION TIMING DIAGRAMS
5.3.1 PRIMARY MASTER Grants VMX-DTB
5.3.2 PRIMARY MASTER Receives VMX-DTB

CHAPTER 6 ELECTRICAL SPECIFICATIONS

6.1 INTRODUCTION

6.2 POWER DISTRIBUTION

6.3 ELECTRICAL SIGNAL CHARACTERISTICS

6.4 DRIVER SPECIFICATIONS
6.4.1 Totem-pole Drivers
6.4.2 Tri-state Drivers
6.4.3 Transceivers
6.4.4 Open-collector Drivers

6.5 RECEIVER SPECIFICATIONS

6.6 SIGNAL LINE TERMINATIONS

6.7 ISOLATION LINES

6.8 BOARD SIGNAL LOADING

6.9 BACKPLANE RIBBON CABLE SPECIFICATION

CHAPTER 7 MECHANICAL SPECIFICATIONS

7.1 DEFINITION OF TERMS

7.2 VMXbus BACKPLANE DESCRIPTION

7.3 VMXbus EUROCARD DESCRIPTION

7.4 VMXbus PIN ASSIGNMENT

CHAPTER 1
INTRODUCTION

1.1 VMXbus SPECIFICATION OBJECTIVES

The VMXbus is a "subsystem bus" which has been designed for use with VMEbus. It provides a high speed secondary path which is optimized for connecting up to six boards in a subsytem configuration. This subsystem can transfer data from board to board over its VMXbus interface without waiting for and without burdening the primary bus (VMEbus). This specification has been written with the following objectives:

a) To specify the electrical characteristics required to design boards that will reliably transfer data between each other over the VMXbus.

b) To specify the mechanical characteristics required to be compatible with VMEbus systems.

c) To specify protocols that precisely define the interaction between the boards that are interfaced to each other over the VMXbus.

d) To specify the terminology used to describe VMXbus based subsystems.

1.2 VMXbus STRUCTURE

The structure of the VMXbus can be described from two points of view: its physical structure and its functional structure. The following sections describe each in detail.

1.2.1 Physical Structure

The physical structure of the VMXbus is composed of an "expanded" VMEbus card rack, one to six plug-in PC boards, and a backplane ribbon cable. The expanded card rack provides both a P1 backplane and a P2 backplane. The P2 backplane buses only the "b" row of pins on the three-row connectors. (The outer two rows of connector pins are left unbused by the P2 backplane: they typically protrude through the rear of the P2 backplane as wire wrap pins and are bused for VMXbus use with a ribbon cable which is pushed onto the wirewrap pins. This cable has from two to six connectors on it. The cable buses the two outer rows of pins of adjacent P2 connectors allowing the boards in these slots to transfer data over the cable.

A board that is inserted from the front of the card rack may have a VMEbus interface, a VMXbus interface, or both. For example, a global I/O board might have only a VMEbus interface, a CPU or a memory board might have both, and a

math processor might have only a VMXbus interface. Any board that has a VMXbus interface uses the two outer rows of the J2 connector.

The use of a ribbon cable to bus the P2 connectors allows any group of up to six adjacent slots to function as a subsystem. The user can install two or more cables to create several VMXbuses in a single card rack. Each of these VMXbus subsystems can operate independently of the primary system bus (VMEbus) and independently of each other. It also permits some slots to be used for other purposes such as I/O signals.

1.2.2 Functional Structure

The VMXbus consists of two groups of signal lines, called the VMXbus Data Transfer Bus (VMX-DTB), an arbitration bus, and a collection of "functional modules" which are configured as required to interface devices to these buses. The functional modules communicate with one another by means of bus signal lines provided by the ribbon cable.

NOTE

> The "functional modules" defined in the specification are used as vehicles for discussion of the bus protocol, and need not be considered a constraint to logic design. For example, the board designer may choose to design a board which has both a MASTER and a SLAVE module on it.

The VMXbus Data Transfer Bus (VMX-DTB) consists of the data and address pathways and associated control signals. Functional modules called MASTERS and SLAVES use the VMX-DTB to transfer data between each other.

Since there can be two MASTERS in a VMXbus group a means must be provided to transfer control of the VMX-DTB between these two MASTERS in an orderly manner, and to guarantee that only one MASTER controls the VMX-DTB at a given time. The Arbitration Bus is defined to do this.

NOTE

> The terms PRIMARY MASTER and SECONDARY MASTER are used in this specification to distinguish between the MASTER that arbitrates the VMX-DTB and the MASTER that must request, and be granted, the VMX-DTB before using it.

1.3 SIGNAL LINE STATES

The VMXbus specification describes all activity on its signal lines in terms of levels and transitions. A signal line is always assumed to be in one of two levels or in transition between these levels. Whenever the term "high" is used, it refers to a high TTL voltage level (>= +2.0 V). The term "low" refers to a low TTL voltage level (<= 0.8 V). A signal line is "in transition" when its voltage is moving between + 0.8 V and + 2.0 V.

There are two possible transitions which can appear on a signal line, and these will be refered to as "edges". A RISING EDGE is defined as the time period during which a signal makes its transition from a low level to a high level. A FALLING EDGE is defined as the time period during which a signal line makes its transition from a high level to a low level.

1.4 USE OF THE ASTERISK (*)

To help define usage of their signal lines, some signal line mnemonics have an asterisk suffix. Depending on the type of signal this suffix has one of three meanings:

1.4.1 Strobe lines

Where an asterisk is used on a strobe line, it indicates that the data being strobed is valid and is sampled on the FALLING edge of the stobe. Strobe lines which fit this category are UAS*, DS1*, and DS0*.

Where there is no asterisk on a strobe line, it indicates that the data being strobed is valid and is sampled on the RISING edge of the strobe. There is one strobe line which fits into this category: LAS.

1.4.2 Bused Signal Lines

The address and data buses carry binary information (i.e. one/zero). The fact that these lines have no asterisk suffix indicates that a "one" is represented by a high level and a "zero" by a low level.

1.4.3 Other Signal Lines

Where an asterisk is used on a signal line other than a strobe line it indicates that a low level on that line represents a "true" state and a high level a "false" state. Signal lines which fit this category are LWORD*, DERR*, ACK*, SMREQ*, and SMGNT*.

Where there is no asterisk it indicates that the high level represents a "true" state. Thus the READ signal line

indicates a read operation when high and a write operation when low.

CHAPTER 2
VMXbus DATA TRANSFER BUS

2.1 SIGNAL LINES

The VMXbus contains a high speed asynchronous parallel Data Transfer Bus (VMX-DTB). This VMX-DTB is used by a PRIMARY MASTER and a SECONDARY MASTER to select SLAVE locations and to transfer data to or from those locations. The PRIMARY MASTER is typically a processor, the SECONDARY MASTER is typically an intelligent I/O controller, and the SLAVE is typically a memory array.

The VMX-DTB can be logically divided into address, data, and control line groups:

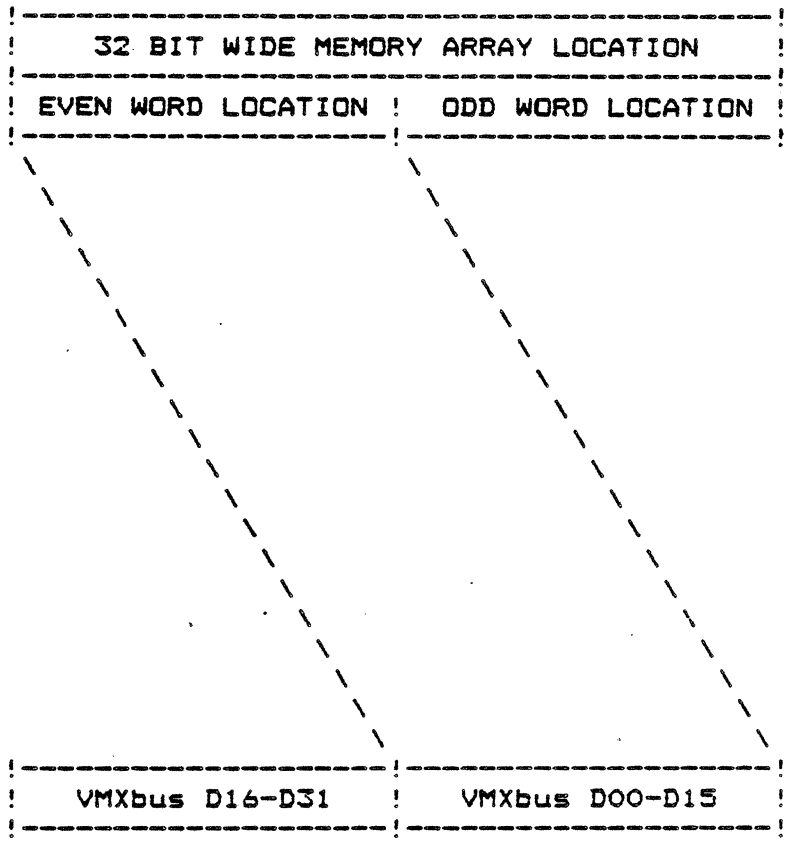
LWORD/A12 - A11/A23	(23) Multiplexed Address lines + longword (LDS* + UDS* provide a 24th bit)
D00-D31	(32) Data lines
LAS	Lower Address Strobe
UAS*	Upper Address Strobe
LDS*	Lower (Odd Byte) Data Strobe
UDS*	Upper (Even Byte) Data Strobe
ACK*	Acknowledge
DERR*	Data Error
READ	Read/Write Control

Data is transferred 8 bits, 16 bits, or 32 bits at a time. Eight bit transfers are called "byte transfers". Byte locations which are addressed with an even address are called "even byte locations" and byte locations addressed with an odd address are called "odd byte locations." Byte transfers to even byte locations are conveyed by D08-D15, and byte transfers to odd byte locations are conveyed by D00-D07. Sixteen bit transfers are called "word transfers". These words are conveyed by D00-D15. Thirty-two bit transfers are called "longword transfers". longwords are conveyed by D00-D31.

Data is stored into memory 8, 16, or 32 bits a time and then read 8,16, or 32 bits at a time. Figure 2-1 shows that when the most significant 16 bits of a longword location is read from (or written to) the data is transferred over D00-D15 instead of D16-D31 as one might expect. This means that 32 bit wide memory boards must have on-board logic to translate the most significant 16 bits of the memory array to the lower 16 data lines of the VMXbus (D00-D15)

FIGURE 2-1

WHEN A WORD ACCESS IS DONE THE DATA IS ALWAYS TRANSFERRED ON VMXbus lines D00-D15.



When 32 bits of data are written they are stored in a single longword location. When reading this data 16 bits at a time the even word address (A01 = 0) corresponds to the longword location bits D16-D31.

Likewise if two 16 bit words of data are stored in adjacent word locations (an even word location and the next higher odd word location) they may be read as a longword with the even word (A01 = 0) forming the most significant 16 bits (D16-D31)

Two bytes of data may be stored in each word location. The even byte location is the most significant byte of the word location. The odd byte is the least significant. The even byte is selected when UDS* is driven low by the MASTER, the odd byte is selected when LDS* is driven low, and both are selected for a word access when both LDS* and UDS* are low. Figure 2-2 summarizes how the various signal lines on VMXbus are used to do 8, 16, and 32 bit transfers.

FIGURE 2-2

SIGNAL LEVELS USED TO ACCESS
BYTES, WORDS, AND LONGWORDS

	LONGWORD LOCATION			
	EVEN WORD LOCATION		ODD WORD LOCATION	
	EVEN BYTE LOCATION	ODD BYTE LOCATION	EVEN BYTE LOCATION	ODD BYTE LOCATION
	D31-D24	D23-D16	D15-D08	D07-D00
BYTE ACCESS				
LWORD	HIGH	HIGH	HIGH	HIGH
A01	LOW	LOW	HIGH	HIGH
UDS*	LOW	HIGH	LOW	HIGH
LDS*	HIGH	LOW	HIGH	LOW
WORD ACCESS				
LWORD	HIGH		HIGH	
A01	LOW	Note 1	HIGH	Note 1
UDS*	LOW		LOW	
LDS*	LOW		LOW	
LONGWORD ACCESS				
LWORD	LOW			
A01	LOW	Note 2	Note 3	Notes 2 & 3
UDS*	LOW			
LDS*	LOW			

NOTES:

1. Not legal to access 16 bits of data on an odd byte address.

2. Not legal to access 32 bits of data on an odd byte address.

3. Not legal to access 32 bits of data on an odd word address.

2.2 MASTER AND SLAVE MODULES

All VMXbus MASTERS and SLAVES can do 8 bit and 16 bit data transfers. Some MASTERS and SLAVES can also do 32 bit transfers. MASTERS that can only do 8 and 16 bit transfers are called D16 MASTERS, while those that can also do 32 bit transfers are called D32 MASTERS. SLAVES that can handle only 8 and 16 bit transfers are called D16 SLAVES while those that can also handle 32 bit transfers are called D32 SLAVES.

There are four categories of cycles used to transfer data on the VMXbus: single cycles, sequential cycles, indivisible single address cycles, and indivisible multiple address cycles. In order to allow VMXbus board vendors to specify which categories of cycles their boards can do, the following abbrieviated notation has been developed:

SGL	Single cycles supported
SEQ	Sequential cycles supported
ISA	Indivisible single address cycles supported
IMA	Indivisible multiple address cycles supported

These abbrieviations are added, as suffixes, to the data width to give a complete description of a board. For example a D32 MASTER capable of all cycles would be called a D32, SGL, SEQ, ISA, IMA MASTER.

NOTE:

ALL MASTERS MUST SUPPORT, AS A MINIMUM, SINGLE CYCLES. ALL SLAVES MUST SUPPORT, AS A MINIMUM, SINGLE CYCLES, INDIVISIBLE SINGLE ADDRESS CYCLES, AND INDIVISIBLE MULTIPLE ADDRESS CYCLES.

2.2.1 D16 MASTERS And D16 SLAVES

The D16 MASTER definition allows 16 bit processors and 16 bit DMA devices to be interfaced to the VMXbus. The D16 SLAVE provides 16 bit memory. Since 16 bit processors sometimes access memory one byte at a time, D16 SLAVES must allow their even and odd byte locations to be accessed one byte at a time. Whenever a byte transfer is done to or from an odd byte location, it is done over D00-D07. Whenever a byte transfer is done to or from an even byte location it is done over D08-D15. (Transferring the even and odd bytes over the same data lines that are used in 16 bit transfers minimizes the multiplexing logic required on the D16 memory slave.)

2.2.2 D32 MASTERS And D32 SLAVES

D32 MASTERS can do LONGWORD (32 bit) transfers over D00-D31, but they can also act like D16 MASTERS (i.e. they can do byte transfers and word transfers: they do odd byte transfers over D00-D07, even byte transfers over D08-D15, and word transfers over D00-D15). This means that they can access D16 slaves as well as D32 SLAVES.

D32 SLAVES allow each of their locations to be accessed one byte at a time, one word at a time, or one longword at a time. When doing byte transfers they transfer even bytes over D08-D15, and their odd bytes over D00-D07. When doing word transfers they always use D00-D15. This means that D32 SLAVES must have logic that allows data in the upper 16 bits of its 32 bit memory array to be accessed through D00-D15. Because D32 SLAVES allow access in all of these ways they can be accessed by D16 MASTERS as well as D32 MASTERS.

2.2.3 Mixing D16 and D32 MASTERS and SLAVES.

VMXbus systems can be built with mixtures of D16 and D32 MASTERS and SLAVES. This can be done in any combination, and these boards can still communicate data successfully, with one qualification: if a D32 MASTER attempts a 32 read or write on D16 SLAVE that SLAVE must respond by driving DERR* low, indicating an error. (The D32 MASTER can still transfer data to and from this SLAVE by doing it 16 bits at a time.)

2.2.4 SEQ MASTERS And SEQ SLAVES.

SEQ MASTERS are able to do sequential access cycles as described in section 3.1.3 and 3.1.4. Most SEQ MASTERS are DMA controllers, but there is no reason why a processor board couldn't be designed to use sequential cycles. (e.g. a processor board with a cache might read blocks of data using the sequential mode.)

SEQ SLAVES are able to respond to sequential access cycles as described in sections 3.1.3 and 3.1.4. During a sequential access cycle a SEQ SLAVE latches the initial address from the bus and then increments that address each time data is read or written. Since there are actually three types of sequential access cycles (byte, word, and longword) the SEQ SLAVE must increment its internal counter in such a way that the next access results in a transfer on the next location in memory. (Since memory SLAVES may have 16, 32 or even 64 bit wide internal memory arrays, the VMXbus spec doesn't dictate how the internal address counter is incremented.)

2.2.5 Mixing SEQ and non-SEQ MASTERS and SLAVES

VMXbus systems can be built with mixtures of SEQ and non-SEQ MASTERS and SLAVES. If an SEQ MASTER attempts a sequential cycle on a non-SEQ SLAVE that SLAVE will respond by driving DERR* low, indicating an error. (The SEQ MASTER can still transfer data to and from this SLAVE by doing single cycles.

2.2.6 IMA MASTERS

IMA MASTERS are able to do indivisible multiple address cycles as described in sections 3.1.6. These cycles allow a VMXbus MASTER to access VMXbus SLAVE locations while excluding all other MASTERS from access to the same SLAVE locations.

All SLAVES must be able to handle IMA cycles. Many VMXbus SLAVES are "dual ported", allowing VMXbus MASTERS to access their locations through one port and VMEbus MASTERS to access the same locations through the second post. These dual ported boards must be designed so that whenever they respond to an indivisible multiple address cycle on the VMXbus they "lock out" all accesses from VMEbus until all of the "reads" and "writes" of the cycle are completed.

CHAPTER 3
EXAMPLES OF DATA TRANSFER BUS OPERATION

3.1 DATA TRANSFER CYCLE TYPES

Putting aside, for the moment, the fact that data may be transferred over VMXbus one byte, two bytes, or four bytes at a time, there are six basic cycle types:

1) Single write cycle
2) Single read cycle
3) Sequential write cycle
4) Sequential read cycle
5) Indivisible single address cycle
6) Indivisible multiple address cycle

There are some cases in which a MASTER may direct a SLAVE to do something that it cannot do. For example, a D16 SLAVE may be told to do a 32 bit read, or a non-SEQ SLAVE may be told to do a sequential cycle. When this happens the SLAVE "aborts" the cycle. This results in six more cycle types:

7) Aborted single write cycle
8) Aborted single read cycle
9) Aborted sequential write cycle
10) Aborted sequential read cycle
11) Aborted indivisible single address cycle
12) Aborted indivisible multiple address cycle

A SLAVE may acknowledge a cycle from the MASTER and then find itself unable to complete the cycle. For example, it may detect a parity error when retrieving data from its memory array. To indicate this to the MASTER it aborts the cycle. This sequence results in six more cycle types:

13) Late aborted single write cycle
14) Late aborted single read cycle
15) Late aborted sequential write cycle
16) Late aborted sequential read cycle
17) Late aborted indivisible single address cycle
18) Late aborted indivisible multiple address cycle

If no SLAVE responds (e.g. the SLAVE has malfunctioned, or there is no SLAVE at the address provided) a timer on the MASTER board terminates the cycle, resulting in six more cycle types:

19) Timed-out single write cycle
20) Timed-out single read cycle
21) Timed-out sequential write cycle

22) Timed-out sequential read cycle
23) Timed-out indivisible single address cycle
24) Timed-out indivisible multiple address cycle

The VMXbus spec allows a MASTER to begin a cycle BEFORE it has determined whether the location it needs to access is on the VMXbus or elsewhere. This feature can improve performance on processor boards where some addresses result in accesses over the VMXbus while others result in on-board accesses or accesses to other buses, such as VMEbus.

The processor begins such a cycle in the usual manner, by sending the "lower address" to the VMXbus SLAVES, before it knows whether or not a VMXbus access is required. If, after decoding the address, it determines that a VMXbus access is not required, then it doesn't proceed with the rest of the cycle. Such a cycle is said to be "withdrawn" by the MASTER. Since all six of the basic cycles types begin in the same way (with the transmission of the lower address) they all result in similar timing when they are withdrawn. This results in one more additional cycle type:

25) Withdrawn cycle

Example flow diagrams are given for the six basic cycle types 1) through 6) in sections 3.1.1 through 3.1.6.

3.1.1 Single Write Cycle

There are three kinds of single write cycles: byte write, word write, and longword write. Figure 3-1 shows the flow diagram for one of these: a single word write cycle, and should be referred to while reading the following cycle flow description.

The cycle begins when the MASTER drives LAS to low. LAS is then held low until the MASTER drives the twelve address bus lines with address bits A01 through A11, and the "longword" level appropriate for the word transfer (low). All these lines must be valid before LAS is driven to high.

The SLAVE latches the "lower address" and the longword bit on the rising edge of LAS. After a specified hold time, the MASTER removes the lower address and longword bit from the address bus and then places the "upper address" on it. (A12-A23). When these new address bits have stabilized, the MASTER drives UAS* to low. Each SLAVE, upon receiving the upper address, determines whether it is the one being addressed.

While the SLAVE is decoding the address the MASTER drives READ low to indicate that a write cycle is being done. Then the MASTER places its 16 bits of data on data bus lines D00-D15 and drives the two data strobes low, commanding the SLAVE to write the data into memory.

When the selected SLAVE detects the data strobes driven to low it must respond in three different ways:

1) If it is a D16 SLAVE, and has been directed to do a 32 bit write, it responds by driving DERR* low.

2) If it is able to handle the required write it drives ACK* low and stores the data into the selected memory location.

3) If it drives ACK* low, but then encounters some difficulty in completing the write (for example, writing to a ROM) it drives DERR* low within a specified time limit to indicate that the write was not successful.

NOTE

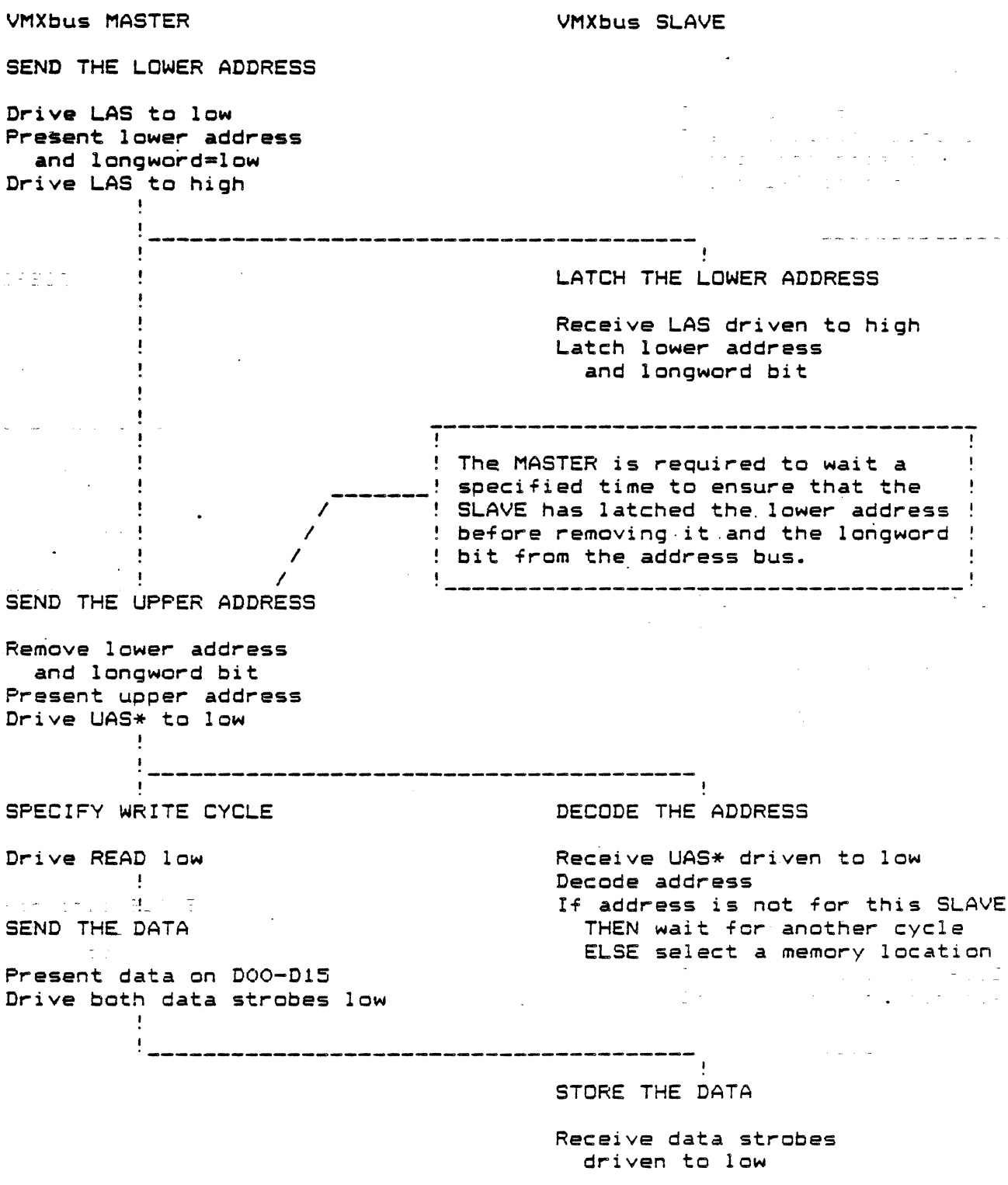
Some VMXbus MASTERS can handle late-aborted cycles while others cannot. To guarantee compatibility between VMXbus boards all SLAVES should be configurable to operate in the normal aborted cycle mode. Those SLAVE boards which can also be configured to provide late-aborted cycles typically provide a range of ACK* to DERR* delays from 0 to 125 nS. The amount of delay may be selected either

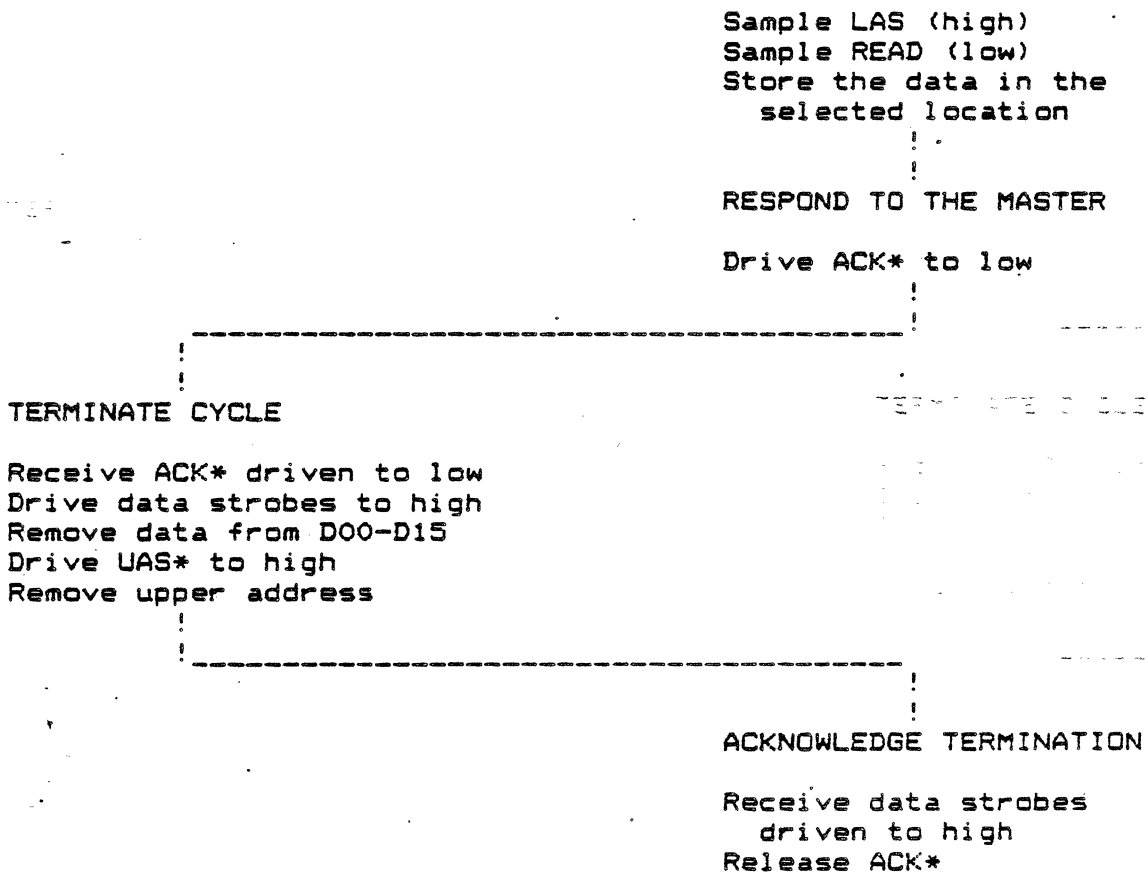
statically (e.g. by jumpers) or dynamically (e.g. by control registers.)

Figure 3-1 shows the most likely of these possiblities: the successful write. When the SLAVE responds by driving ACK* to low the MASTER removes the data from D00-D15, removes the upper address from the address bus, and drives UDS*,LDS* and UAS* high in preparation for the next cycle.

When the SLAVE detects the data strobes driven to high, it releases ACK*, allowing the data bus to be used for the next cycle.

Figure 3-1
A Single Word Write Cycle




Sample LAS (high)
Sample READ (low)
Store the data in the
selected location
RESPOND TO THE MASTER
Drive ACK* to low
TERMINATE CYCLE
Receive ACK* driven to low
Drive data strobes to high
Remove data from D00-D15
Drive UAS* to high
Remove upper address
ACKNOWLEDGE TERMINATION
Receive data strobes
driven to high
Release ACK*

3.1.2 Single Read Cycle

There are three kinds of single read cycles: byte read, word read, and longword read. Figure 3-2 shows the flow diagram for one of these: a single word read cycle, and should be referred to while reading the following cycle flow description.

The cycle begins when the MASTER drives LAS low. LAS is then held low until the MASTER drives the twelve address bus lines with address bits A01 through A11 and the "longword" level appropriate for the word transfer (low). All these lines must be valid before LAS is driven to high.

The SLAVE latches the "lower address" and the longword bit on the rising edge of LAS. After a specified hold time, the MASTER removes the lower address and the longword but from the address bus and then places the "upper address"on it (A12-A23). When these new address bits have stabilized, the MASTER drives UAS* to low. Each SLAVE, upon receiving the upper address, determines whether it is the one being addressed.

While the SLAVES are decoding the address the MASTER drives READ high to indicate that a read cycle is being done. Then the MASTER drives the two data strobes to low, commanding the SLAVE to retrieve data from the selected memory location and place it on the bus.

When the selected SLAVE detects the data strobes driven to low it responds in one of three different ways:

1) If it is a D16 SLAVE, and has been directed to do a 32 bit read, it responds by driving DERR* low.

2) If it is able to handle the required read it drives ACK* to low, retrieves the data from the selected location, and places it on the data bus.

3) If it drives ACK* low, but then encounters some difficulty in retrieving data from memory (possibly because of a parity error) it drives DERR* within a specified time limit to indicate that the data on the data bus is not valid.

NOTE

Some MASTERS require that ACK* be driven low prior to the time that data has been placed on the bus to give maximum performance, while others must see valid data on the bus when ACK* falls. To guarantee compatibility between all VMXbus boards, all SLAVES must be configurable so that they present valid data at the same

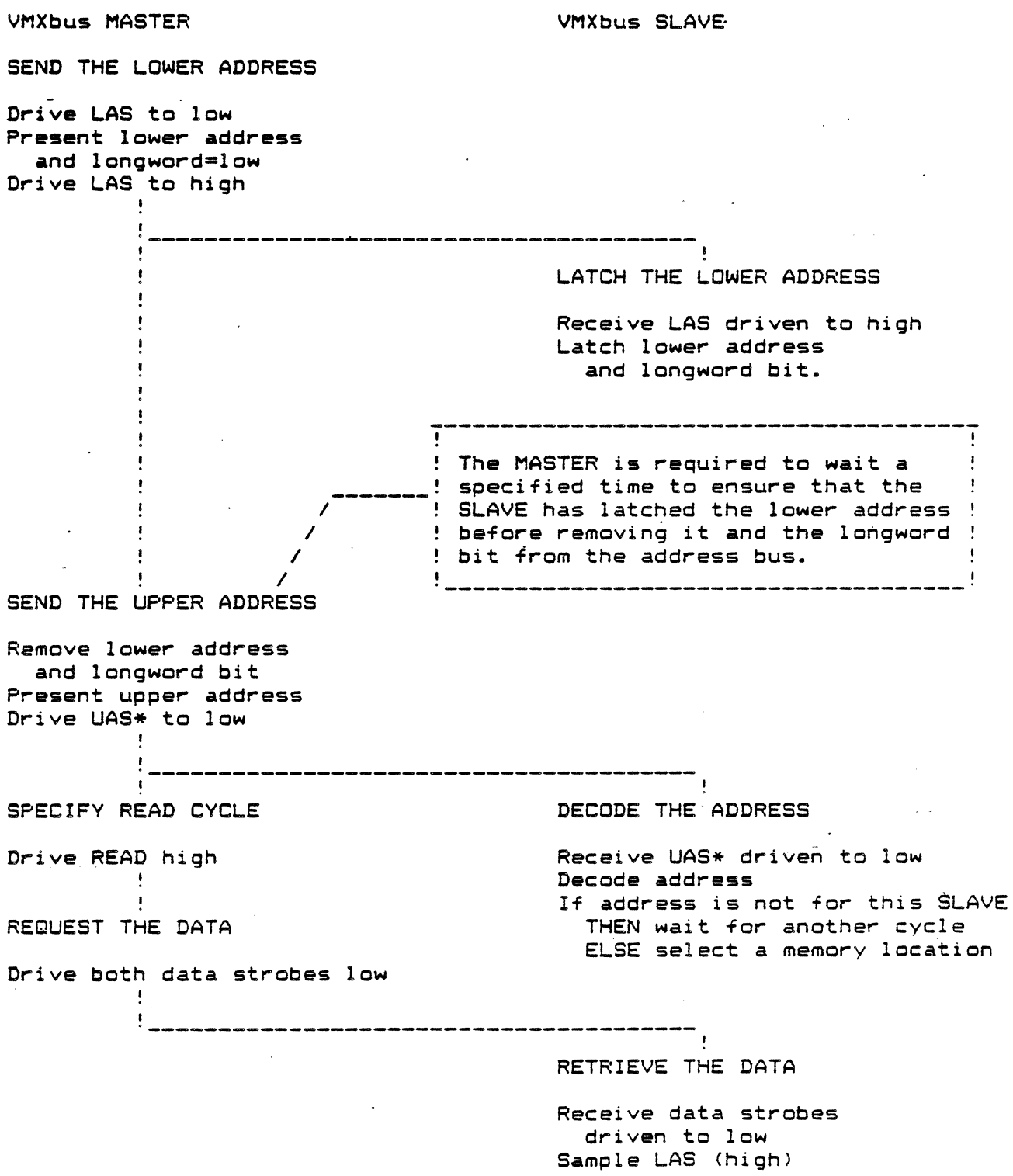
time they drive ACK* low. Those SLAVES which can ALSO be configured to drive ACK* low earlier provide a range of "data valid" to ACK* delays of 0 to 125 nS. (The amount of delay may be selected either statically (e.g. by jumpers) or dynamically (e.g by control registers) to give optimum performance.

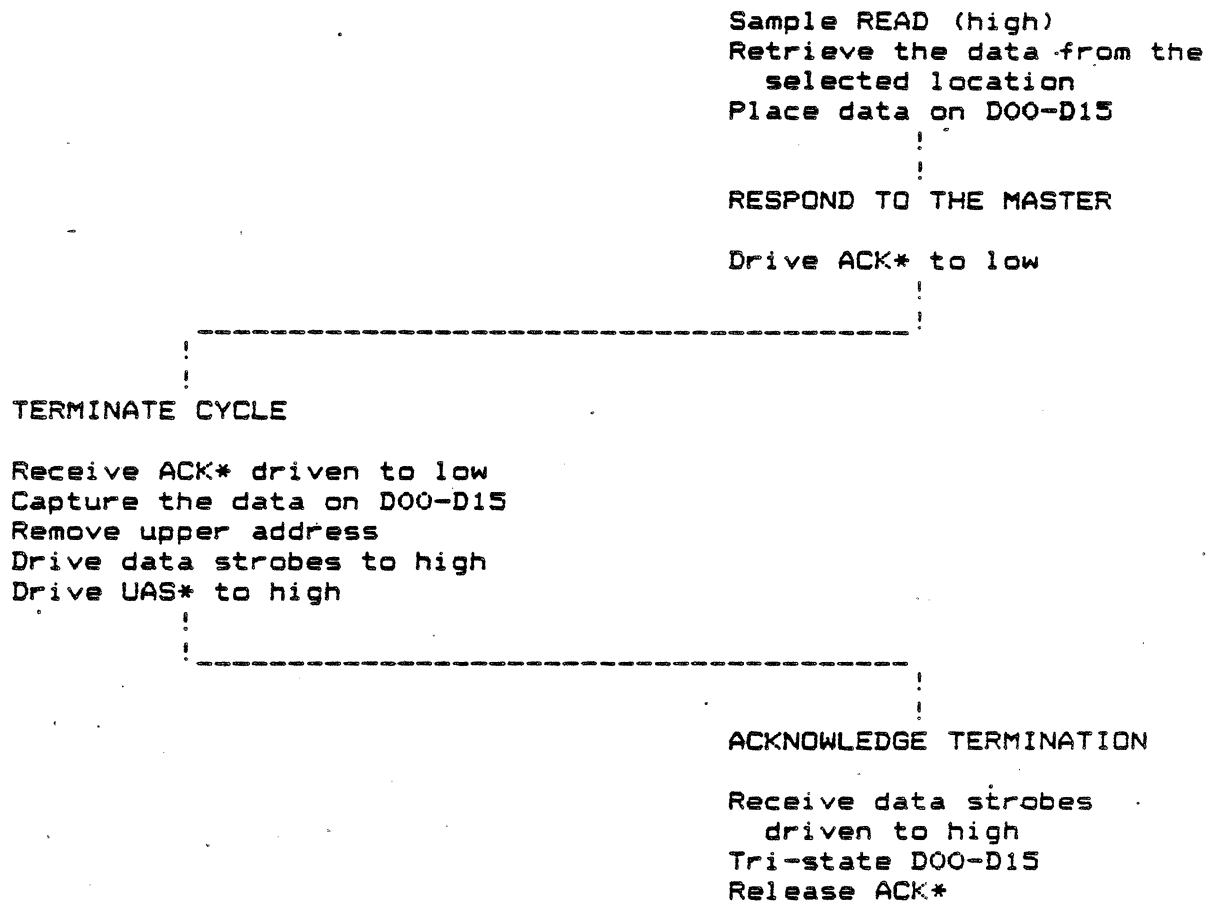
Some VMXbus MASTERS can handle late-aborted cycles while others cannot. To guarantee compatibility between VMXbus boards all SLAVES must be configurable to operate in the normal aborted cycle mode. Those SLAVE boards which can also be configured to provide late-aborted cycles typically provide a range of ACK* to DERR* delays from 0 to 125 nS. The amount of delay may be selected either statically (e.g. by jumpers) or dynamically (e.g. by control registers).

Figure 3-2 shows the most likely of these possibilities: the successful read. When the SLAVE responds by driving ACK* to low, the MASTER captures the data, removes the upper address from the address bus, and drives UDS*, LDS*, and UAS* high in preparation for the next cycle.

When the SLAVE detects the data strobes driven to high, it tri-states its data bus drivers, and releases ACK* to high, allowing the data bus to be used for the next cycle.

Figure 3-2
A Single Word Read Cycle




Sample READ (high)
Retrieve the data from the
selected location
Place data on D00-D15
RESPOND TO THE MASTER
Drive ACK* to low
TERMINATE CYCLE
Receive ACK* driven to low
Capture the data on D00-D15
Remove upper address
Drive data strobes to high
Drive UAS* to high
ACKNOWLEDGE TERMINATION
Receive data strobes
driven to high
Tri-state D00-D15
Release ACK*

3.1.3 Sequential Write Cycle

There are three kinds of sequential write cycles: byte write, word write, and longword write. Figure 3-3 shows the flow diagram for one of these: a sequential word write cycle, and should be referred to while reading the following cycle flow description.

The cycle begins when the MASTER drives LAS low. LAS is then held low until the MASTER drives the twelve address bus lines with address bits A01 through A11 and the "longword" level appropriate for the word transfer (low). All these lines must be valid before LAS is driven to high.

The SLAVE latches the "lower address" and the longword bit on the rising edge of LAS. After a specified hold time, the MASTER removes the lower address and longword bit from the address bus and then places the "upper address" on it (A12-A23). When these new address bits have stabilized, the MASTER drives UAS* to low. Each SLAVE, upon receiving the upper address, determines whether it is the one being addressed.

while the SLAVES are decoding the address the MASTER drives READ low to indicate that a write cycle is being done. In order to indicate that this is a sequential write cycle it drives LAS low again. (LAS is then held low through all of the data transfers to the end of the sequential write cycle.) Then the MASTER places its 16 bits of data on the data bus lines D00-D15 and drives the two data strobes low, commanding the SLAVE to write the data into memory.

When the selected SLAVE detects the data strobes driven to low it responds in one of three different ways:

1) If it is a D16 SLAVE, and has been directed to do a 32 bit write, or if it is a non-SEQ SLAVE and has been directed to do a sequential access, it responds by driving DERR* low.

2) If it is able to handle the required write, it drives ACK* low and stores the data into the selected memory location.

3) If it drives ACK* low, but then encounters some difficulty in completing the write (for example, writing to a ROM) it drives DERR* low within a specified time limit to indicate that the write was not successful.

NOTE

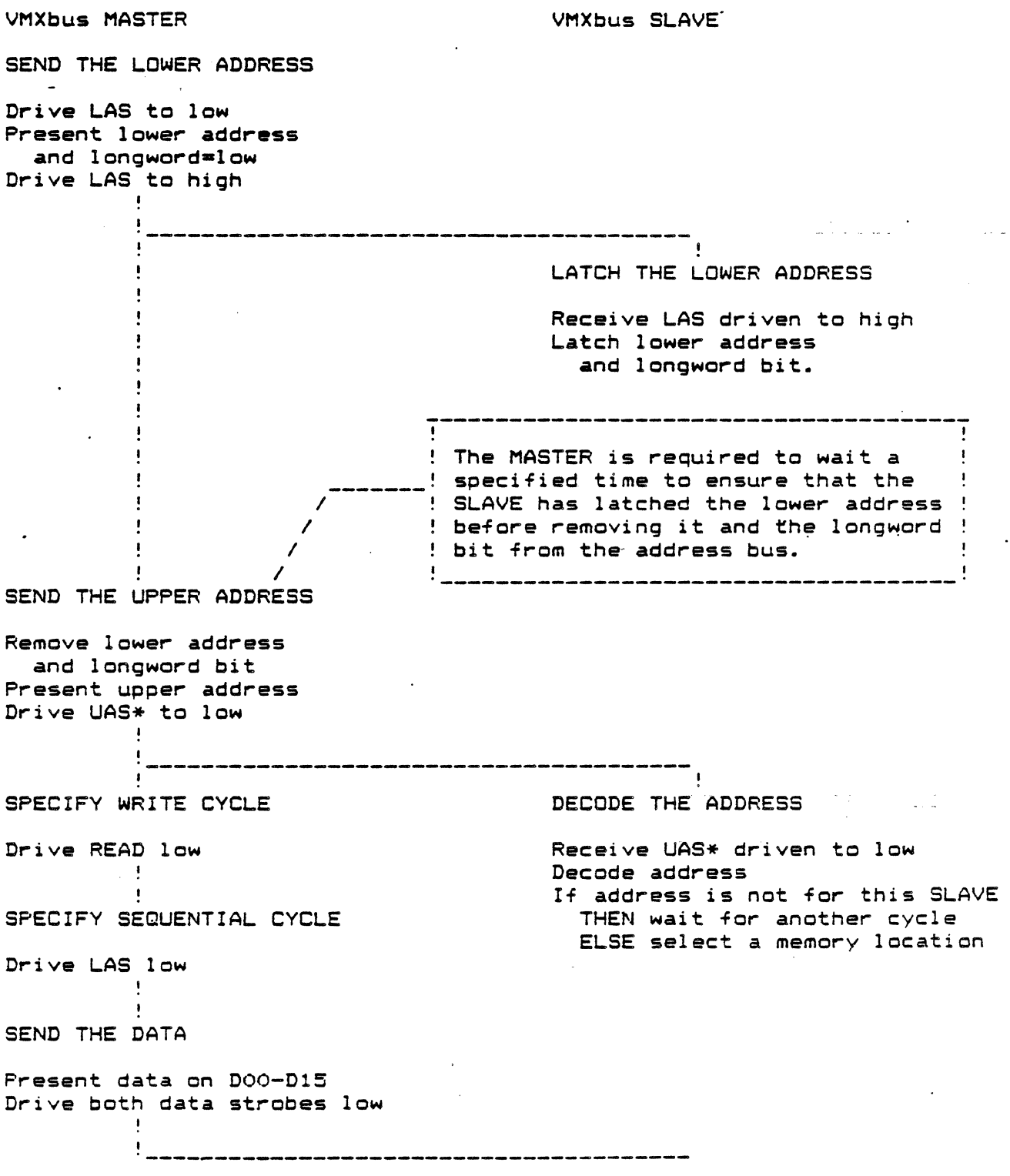
Some VMXbus MASTERS can handle late-aborted cycles while others cannot. To guarantee compatibility between VMXbus

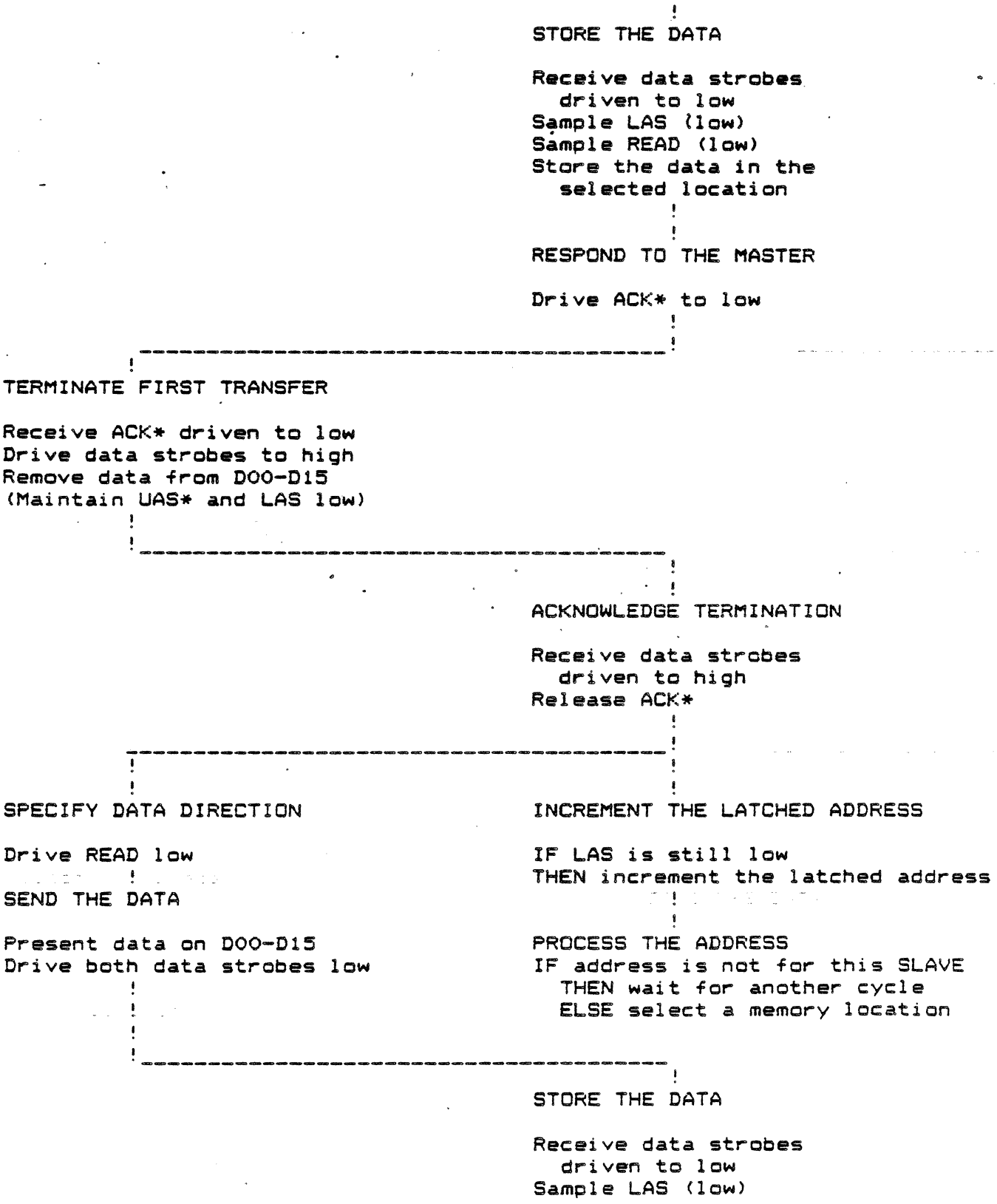
boards all SLAVES should be configurable to operate in the normal aborted cycle mode. Those SLAVE boards which can also be configured to provide late-aborted cycles typically provide a range of ACK* to DERR* delays from 0 to 125 nS. The amount of delay may be selected either statically (e.g. by jumpers) or dynamically (e.g. by control registers.)

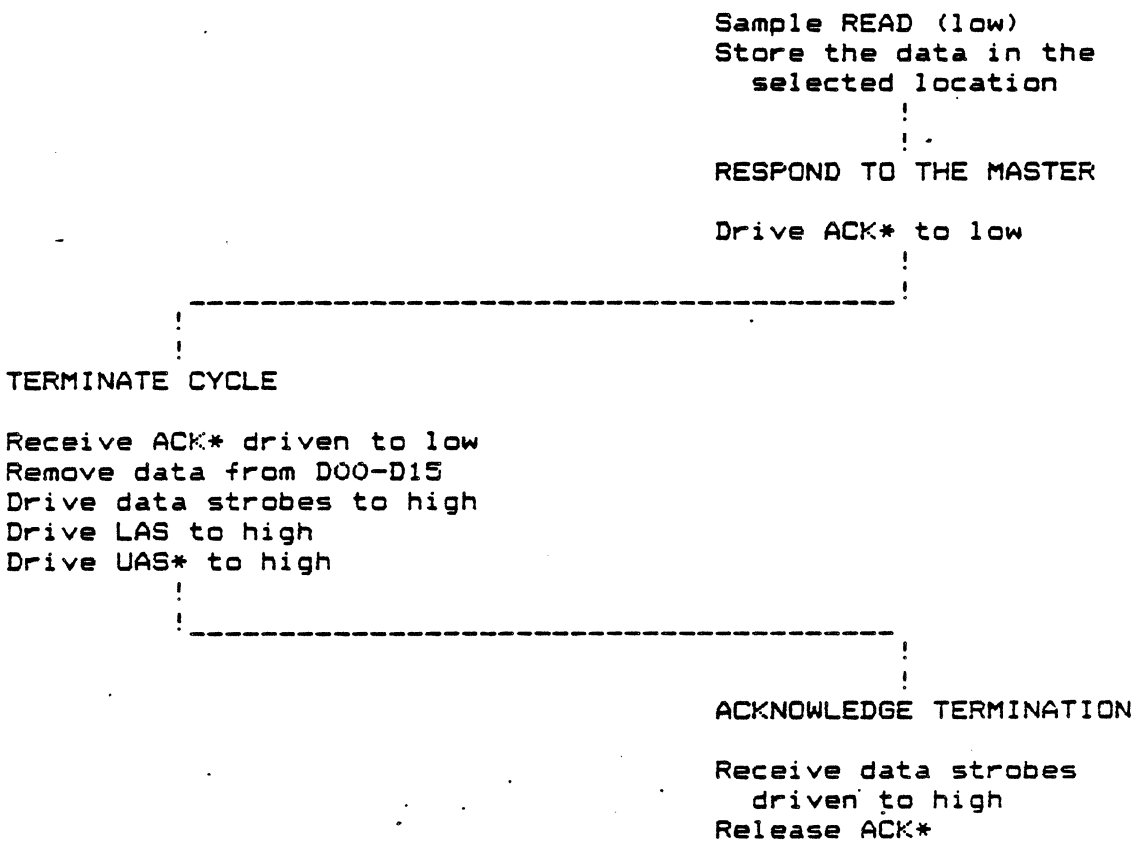
Figure 3-3 shows the most likely of these possibilities: the successful write. When the SLAVE responds by driving ACK* to low, the MASTER removes the upper address from the address bus, and drives the data strobes high, but maintains LAS and UAS* low. When the SLAVE detects the data strobes driven to high it releases ACK*, allowing the data bus to be used for the next cycle. When the MASTER sees ACK* high as well as DERR* already high it initiates another write cycle. Since LAS has been held low the SLAVE has incremented the address from the last cycle and selected a new location.

This process repeats itself for each write until all of the data has been written. On the last transfer the MASTER responds to the low ACK* by driving LAS, UDS*, LDS* and UAS* to high in preparation for the next cycle.

Figure 3-3
A Sequential Word Write Cycle




STORE THE DATA
Receive data strobes driven to low
Sample LAS (low)
Sample READ (low)
Store the data in the selected location
RESPOND TO THE MASTER
Drive ACK* to low
TERMINATE FIRST TRANSFER
Receive ACK* driven to low
Drive data strobes to high
Remove data from D00-D15
(Maintain UAS* and LAS low)
ACKNOWLEDGE TERMINATION
Receive data strobes driven to high
Release ACK*
SPECIFY DATA DIRECTION
Drive READ low
SEND THE DATA
Present data on D00-D15
Drive both data strobes low
INCREMENT THE LATCHED ADDRESS
IF LAS is still low
THEN increment the latched address
PROCESS THE ADDRESS
IF address is not for this SLAVE
THEN wait for another cycle
ELSE select a memory location
STORE THE DATA
Receive data strobes driven to low
Sample LAS (low)


Sample READ (low)
Store the data in the
selected location
RESPOND TO THE MASTER
Drive ACK* to low
TERMINATE CYCLE
Receive ACK* driven to low
Remove data from D00-D15
Drive data strobes to high
Drive LAS to high
Drive UAS* to high
ACKNOWLEDGE TERMINATION
Receive data strobes
driven to high
Release ACK*

3.1.4 Sequential Read Cycle

There are three types of sequential read cycles: byte read, word read, and longword read. Figure 3-4 shows the flow diagram for one of these: a sequential word read cycle, and should be referred to while reading the following cycle flow description.

The cycle begins when the MASTER drives LAS low. LAS is then held low until the MASTER drives the twelve address bus lines with address bits A01 through A11 and the "longword" level appropriate for the word transfer (low). All these lines must be valid before LAS is driven to high.

The SLAVE latches the "lower address" and longword bit on the rising edge of LAS. After a specified hold time, the MASTER removes the lower address and the longword bit from the address bus and then places the "upper address" on it. (A12-A23). When these new address bits have stabilized, the MASTER drives UAS* to low. Each SLAVE, upon receiving the upper address, determines whether it is the one being addressed.

while the SLAVES are decoding the address, the MASTER drives READ high to indicate that a read cycle is being done. In order to indicate that this is a sequential read cycle it drives LAS low again. (LAS is held low through all of the data transfers to the end of the sequential read cycle.) Then the MASTER drives the two data strobes to low, commanding the SLAVE to retrieve data from the selected memory location and place it on the bus.

When the selected SLAVE detects the data strobes driven to low it responds in one of three different ways:

1) If it is a D16 SLAVE, and has been directed to do a 32 bit read, it responds by driving DERR* low.

2) If it is able to handle the required read, it drives ACK* to low, retrieves the data from the selected location, and places it on the data bus.

3) If it drives ACK* low, but then is unable to retrieve data from the memory, (possibly because of a parity error) it drives DERR* within a specified time limit to indicate that the data is not valid.

NOTE

Some MASTERS require that ACK* be driven low prior to the time that data has been placed on the bus to give maximum performance, while others must see valid data on the bus when ACK* falls. To guarantee compatibility

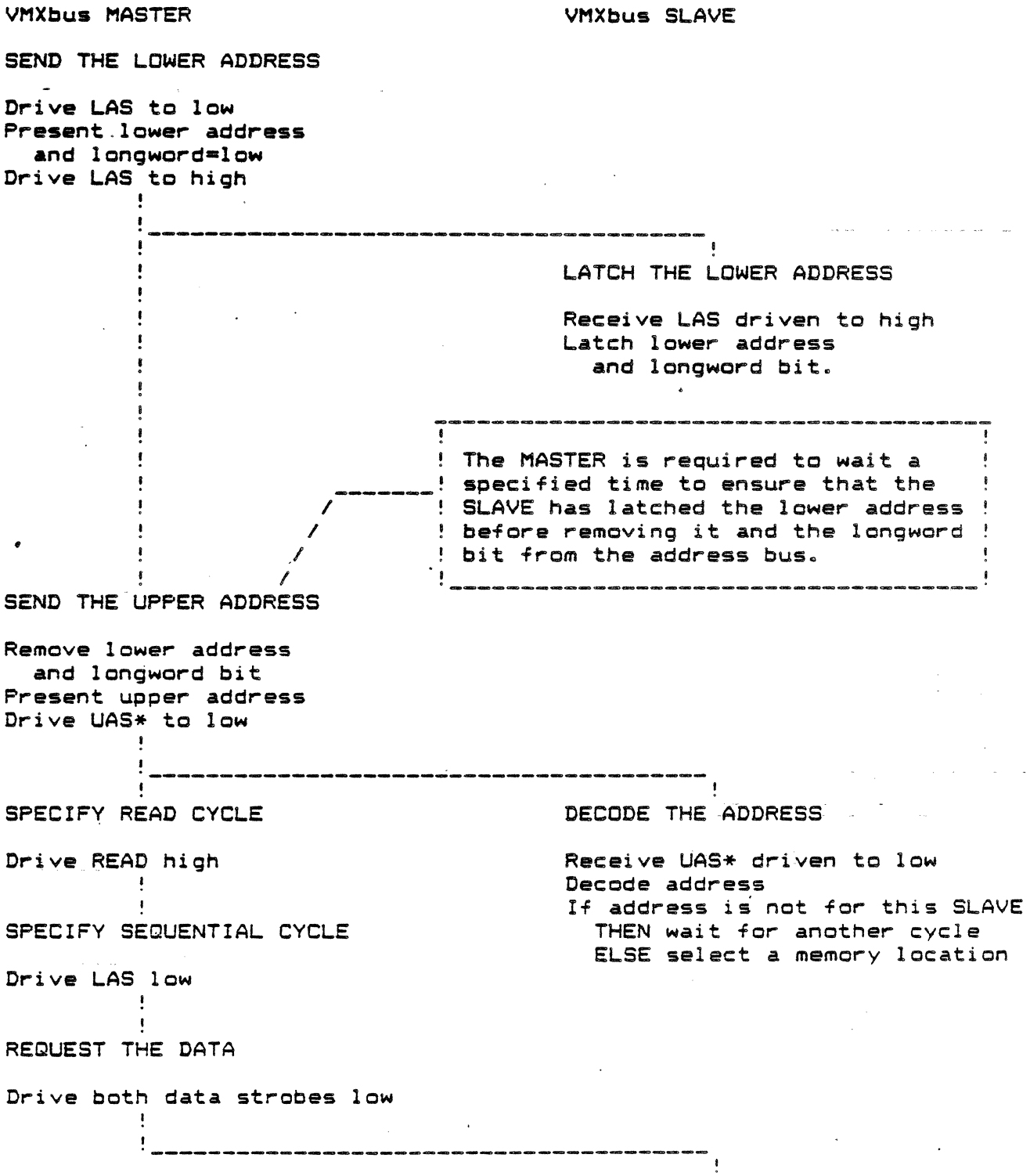
between all VMXbus boards all SLAVES must be configurable so that they present valid data at the same time they drive ACK* low. Those SLAVES which can also be configured to drive ACK* low provide a range of "data valid" to ACK* delays of 0 to 125 nS. (The amount of delay may be selected either statically (e.g. by jumpers) or dynamically (e.g by control registers) to give optimum performance.

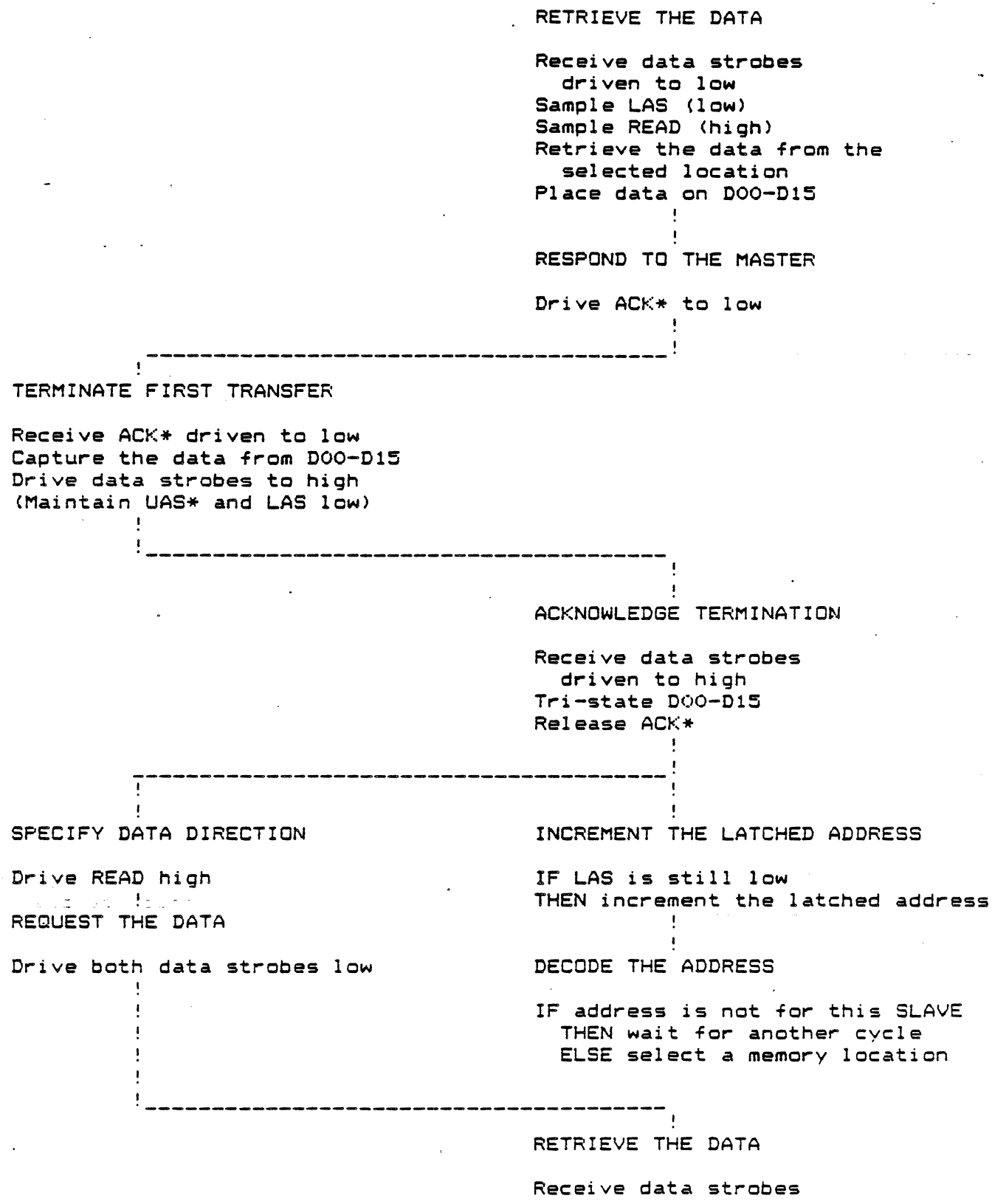
Some VMXbus MASTERS can handle late-aborted cycles while others cannot. To guarantee compatibility between VMXbus boards all SLAVES should be configurable to operate in the normal aborted cycle mode. Those SLAVE boards which can also be configured to provide late-aborted cycles typically provide a range of ACK* to DERR* delays from 0 to 125 nS. The amount of delay may be selected either statically (e.g. by jumpers) or dynamically (e.g. by control registers.)

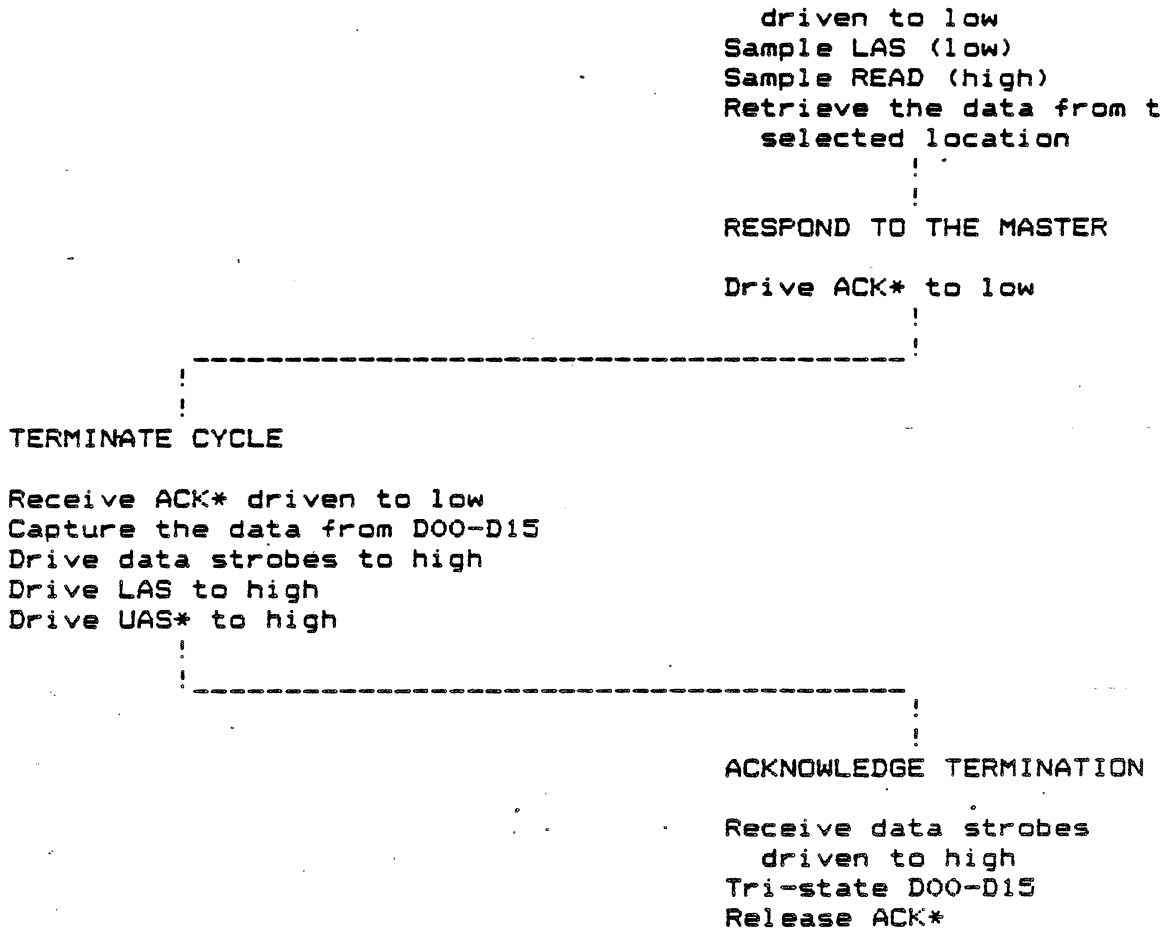
Figure 3-4 shows the most likely of these possiblities: the successful read. When the SLAVE responds by driving ACK* low, the MASTER captures the data. Then it maintains LAS and UAS* low, but drives the data strobes to high When the SLAVE detects the data strobes driven to high it tri-states its data bus drivers and releases ACK* to high. When the MASTER sees ACK* high as well as DERR* already high it drives the data strobes low for another read.

This process repeats itself for each read until all of the data has been read. On the last transfer the MASTER responds to the low ACK* by driving LAS, UDS*, LDS*, and UAS* to high in preparation for the next cycle.

Figure 3-4
A Sequential Word Read Cycle




RETRIEVE THE DATA
Receive data strobes
driven to low
Sample LAS (low)
Sample READ (high)
Retrieve the data from the
selected location
Place data on D00-D15
RESPOND TO THE MASTER
Drive ACK* to low
TERMINATE FIRST TRANSFER
Receive ACK* driven to low
Capture the data from D00-D15
Drive data strobes to high
(Maintain UAS* and LAS low)
ACKNOWLEDGE TERMINATION
Receive data strobes
driven to high
Tri-state D00-D15
Release ACK*
SPECIFY DATA DIRECTION
Drive READ high
REQUEST THE DATA
Drive both data strobes low
INCREMENT THE LATCHED ADDRESS
IF LAS is still low
THEN increment the latched address
DECODE THE ADDRESS
IF address is not for this SLAVE
THEN wait for another cycle
ELSE select a memory location
RETRIEVE THE DATA
Receive data strobes


driven to low
Sample LAS (low)
Sample READ (high)
Retrieve the data from the selected location
RESPOND TO THE MASTER
Drive ACK* to low
TERMINATE CYCLE
Receive ACK* driven to low
Capture the data from D00-D15
Drive data strobes to high
Drive LAS to high
Drive UAS* to high
ACKNOWLEDGE TERMINATION
Receive data strobes driven to high
Tri-state D00-D15
Release ACK*

3.1.5 Indivisible Single Address Cycle

Indivisible single address cycles are used to read and/or write to a single memory location without allowing any other MASTER to access the location. This is accomplished by beginning the cycle in the normal manner (i.e. sending the lower and upper address) but then holding LAS high and UAS* low while the data strobes are used to read and write repetitively to the selected location. Any sequence and any number of read and write cycles are permitted, as long as they are all of the same data width.

The indivisible single address cycle might be used to do a read-modify-write sequence, commonly used for semaphores.

There are three kinds of indivisible single address cycles: byte, word and longword. Figure 3-5 shows the flow diagram for one of these: an indivisible single address word cycle. Since such a cycle may consist of any combination of reads and writes, figure 3-5 shows the simple case of one read followed by one write. This figure should be referred to while reading the following cycle flow description.

The cycle begins when the MASTER drives LAS low. LAS is then held low until the MASTER drives the twelve address bus lines with address bits A01 through A11, and the "longword" level appropriate for the word transfer (low). All these lines must be valid before LAS is driven to high.

The SLAVE latches the "lower address" and longword bit on the rising edge of LAS. After a specified hold time, the MASTER removes the lower address and longword bit from the address bus and then places the "upper address" on it (A12-A23). When these new address bits have stabilized, the MASTER drives UAS* to low. Each SLAVE, upon receiving the upper address, determines whether it is the one being addressed.

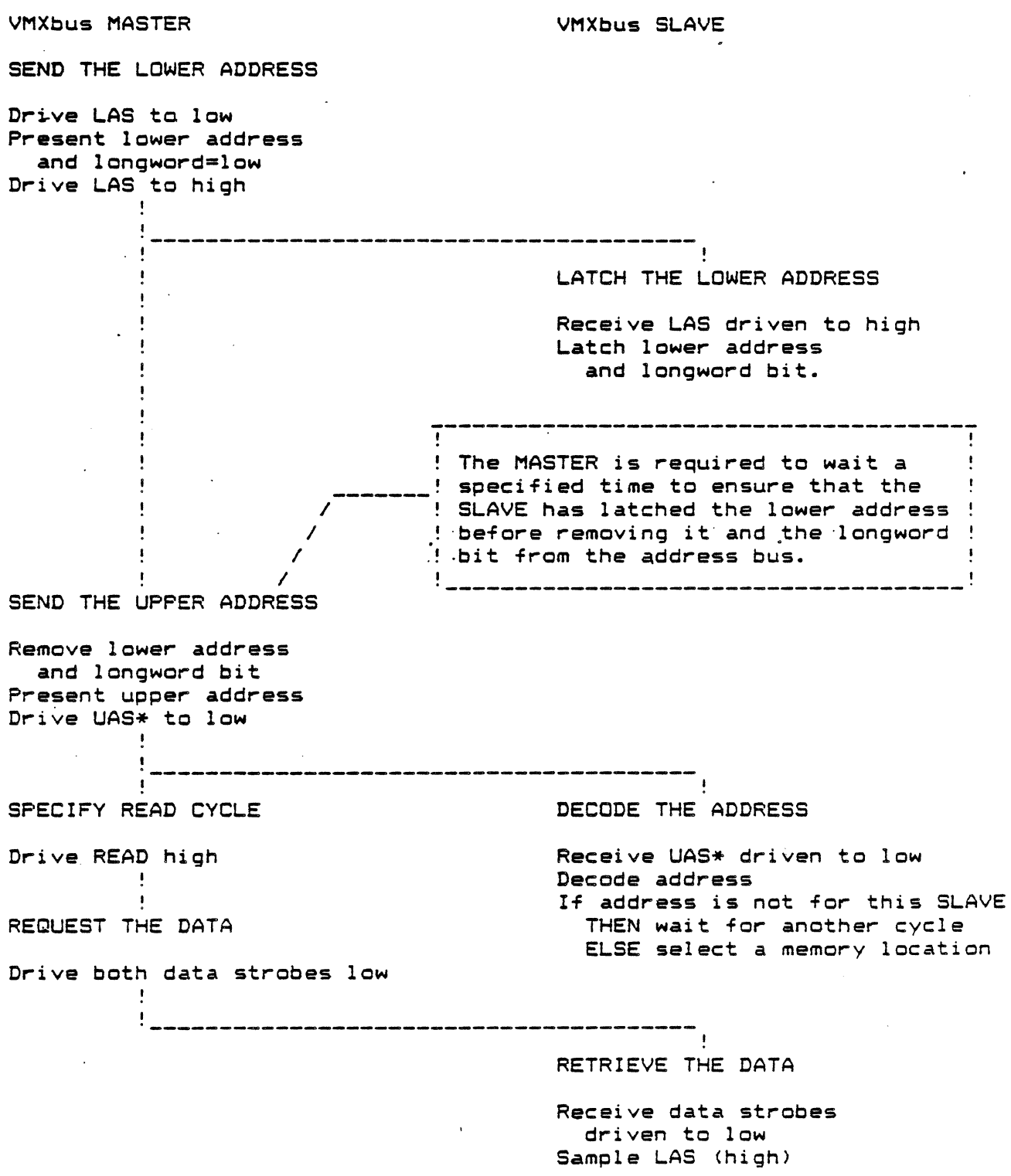
while the SLAVES are decoding the address the MASTER drives READ either high or low depending on whether a read or a write cycle is being done. In figure 2-9 a read is done first, so READ is driven high. Then the MASTER either places its data on the data bus and drives the data strobes low (write cycle), or it just drives the data strobes low (read cycle). (In figure 2-9 it does the latter.)

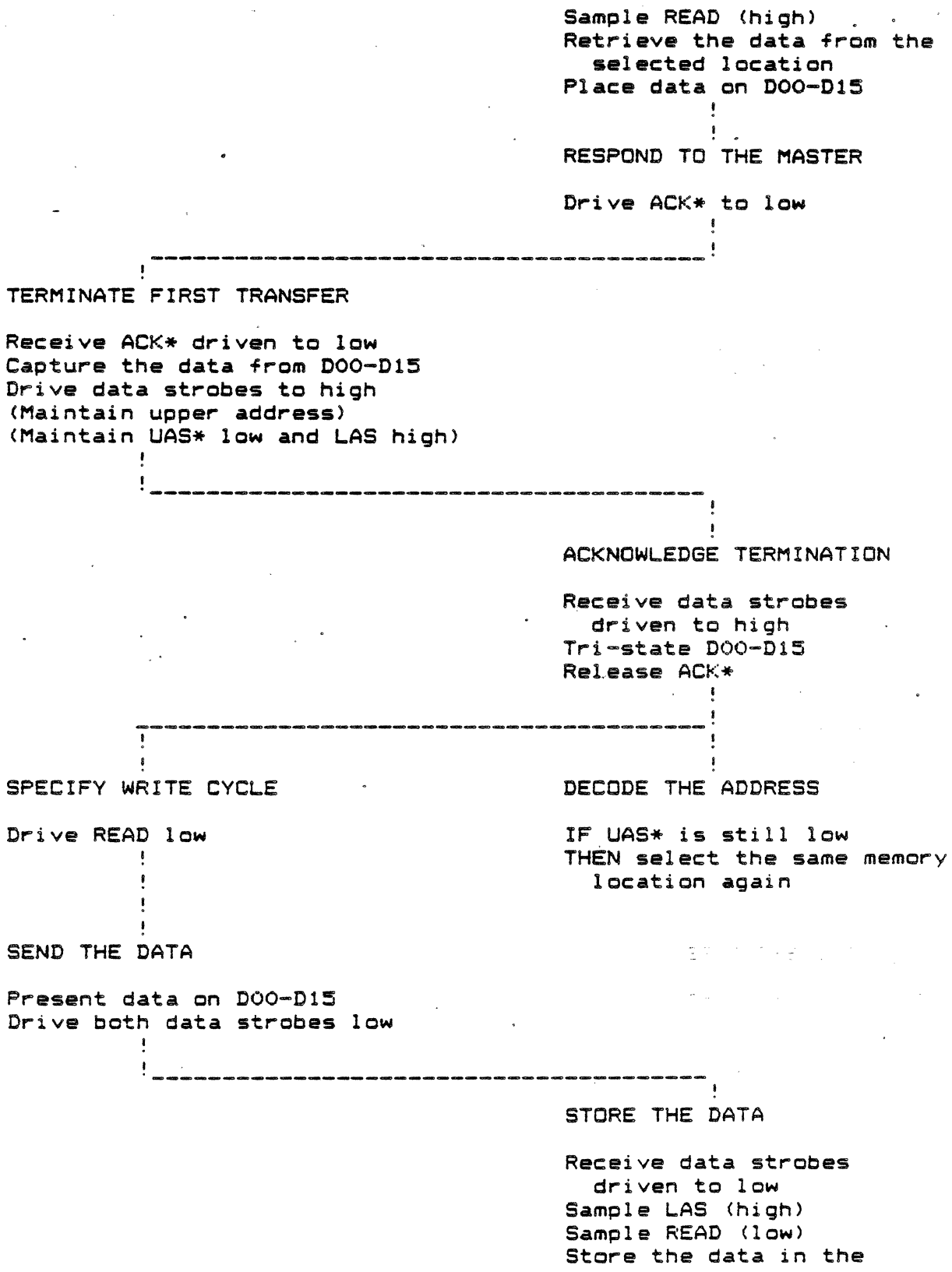
When the selected SLAVE detects the data strobes driven to low it responds in a manner similar to the earlier examples: driving DERR* low, or driving ACK* low, or driving ACK* low followed by DERR* low.

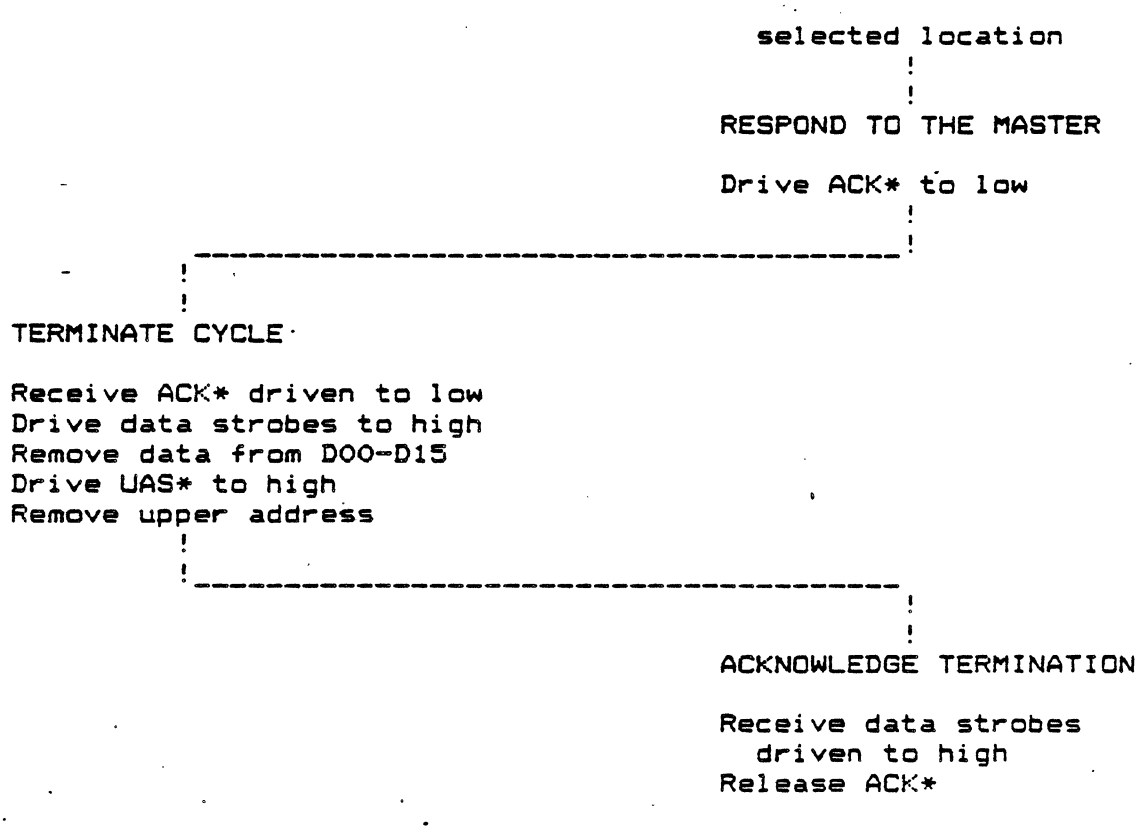
Figure 3-5 shows the most likely of these possibilities: the successful read. When the SLAVE drives ACK* to low the MASTER maintains UAS* low and LAS high and initiates another cycle by changing the level of READ as required and then driving

the data strobes to low again. Figure 3-5 shows the MASTER doing a write after the read is complete.

Figure 3-5
An Indivisible Single Address Word Cycle




Sample READ (high)
Retrieve the data from the selected location
Place data on D00-D15
RESPOND TO THE MASTER
Drive ACK* to low
TERMINATE FIRST TRANSFER
Receive ACK* driven to low
Capture the data from D00-D15
Drive data strobes to high
(Maintain upper address)
(Maintain UAS* low and LAS high)
ACKNOWLEDGE TERMINATION
Receive data strobes driven to high
Tri-state D00-D15
Release ACK*
SPECIFY WRITE CYCLE
Drive READ low
DECODE THE ADDRESS
IF UAS* is still low
THEN select the same memory location again
SEND THE DATA
Present data on D00-D15
Drive both data strobes low
STORE THE DATA
Receive data strobes driven to low
Sample LAS (high)
Sample READ (low)
Store the data in the

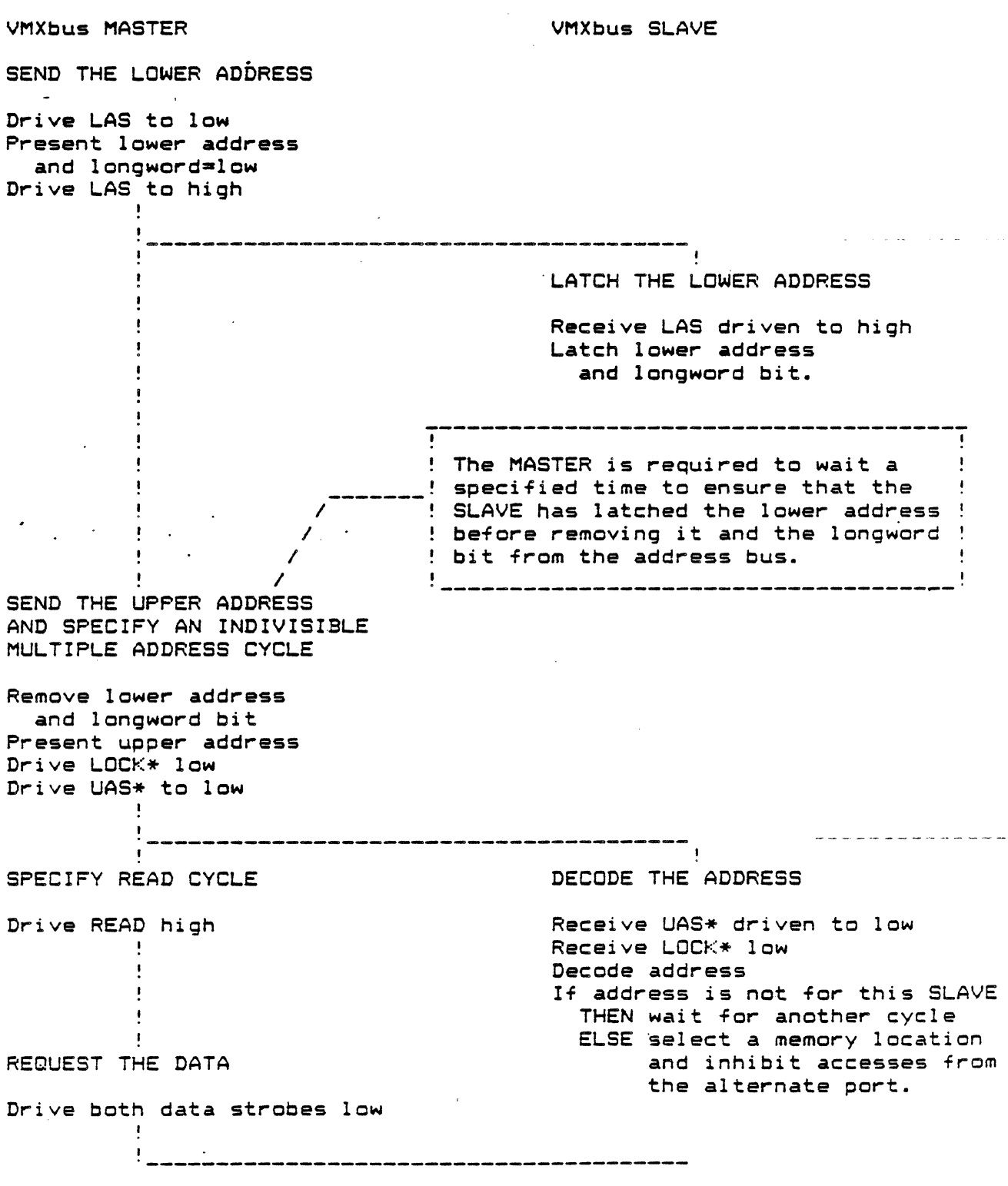

selected location
RESPOND TO THE MASTER
Drive ACK* to low
TERMINATE CYCLE
Receive ACK* driven to low
Drive data strobes to high
Remove data from D00-D15
Drive UAS* to high
Remove upper address
ACKNOWLEDGE TERMINATION
Receive data strobes
driven to high
Release ACK*

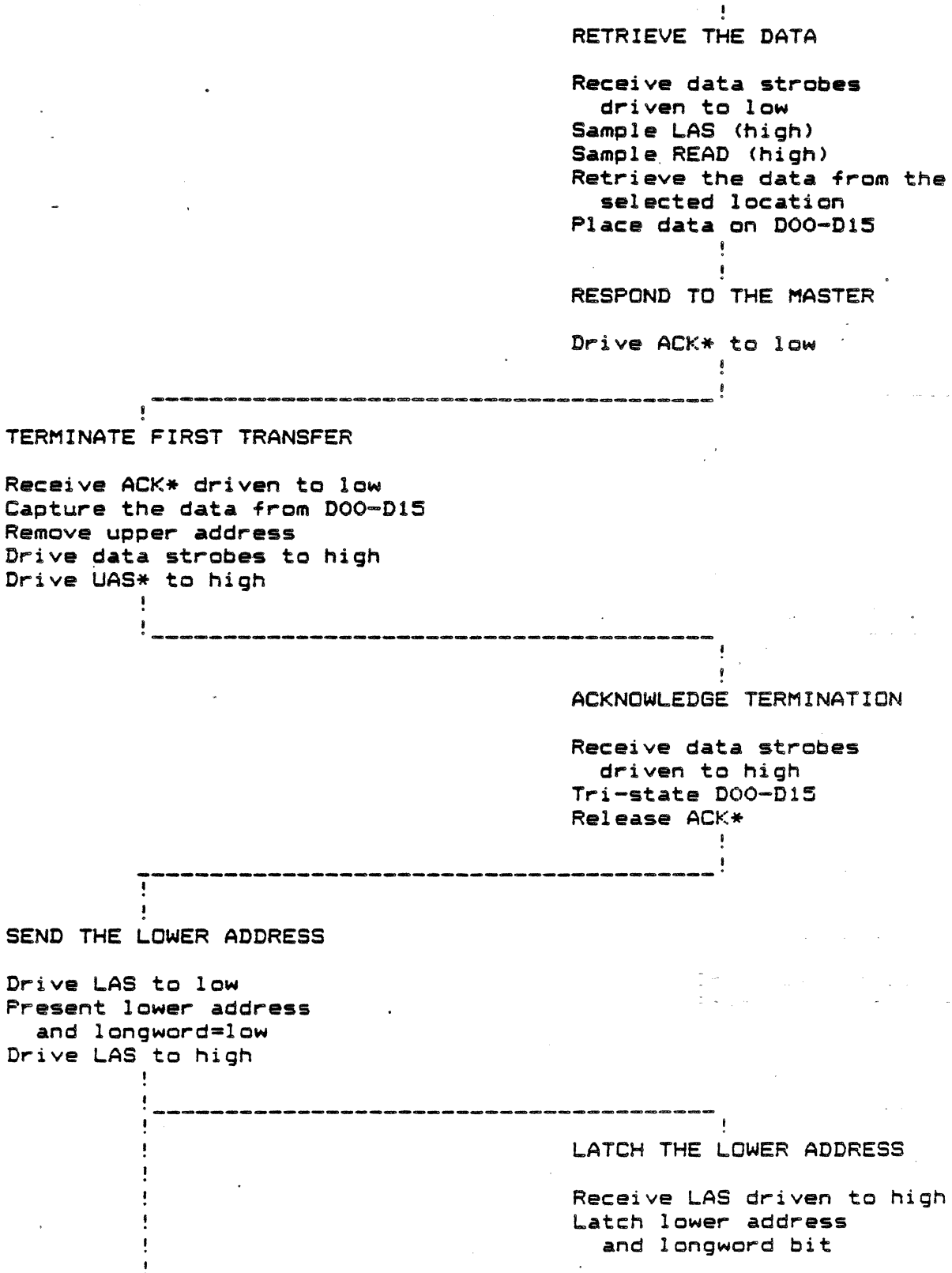
3.1.6 Indivisible Multiple Address Cycle

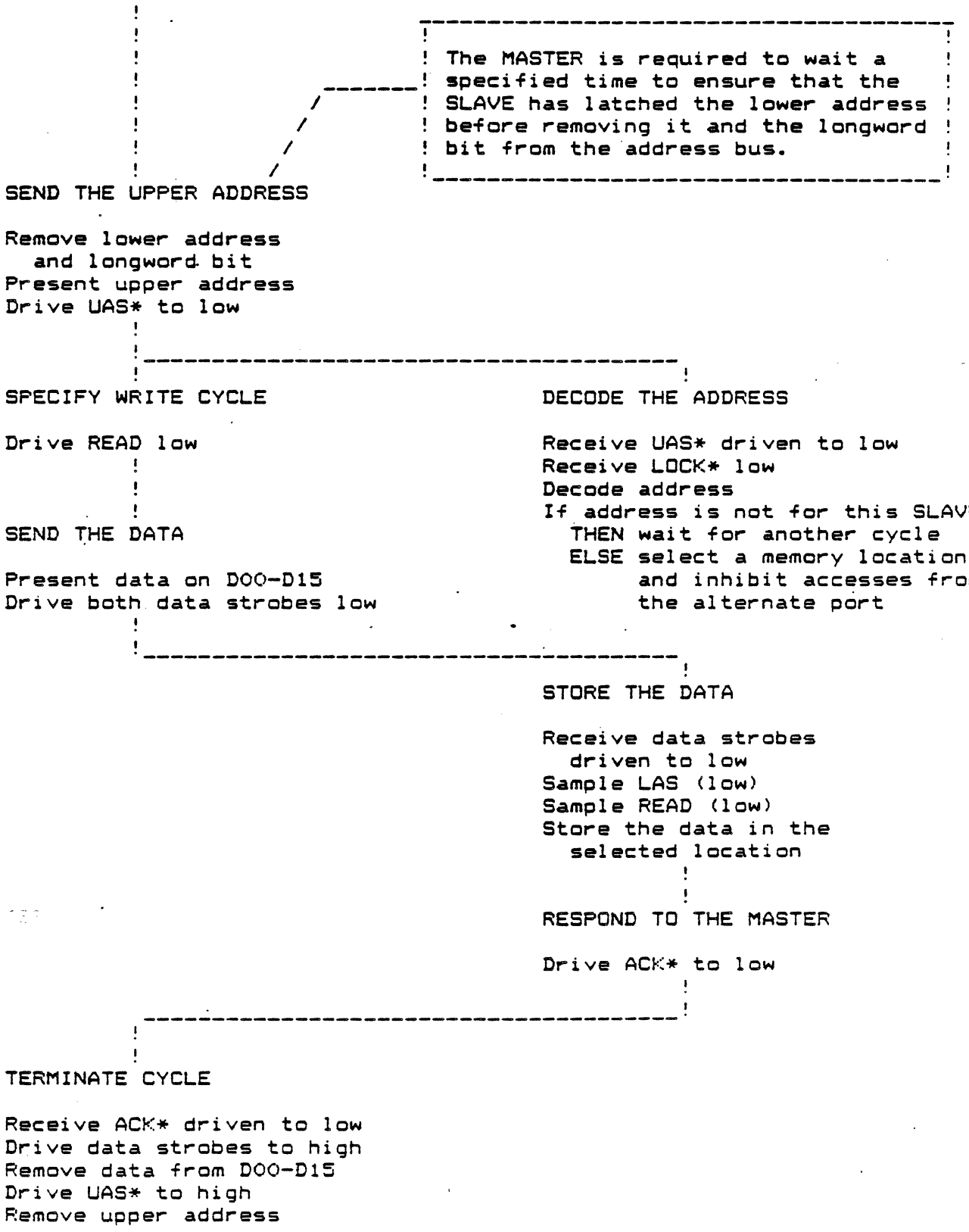
Indivisible multiple address cycles are used to read from and write to several different memory locations without allowing any other MASTER to access these locations. Access by other MASTERS is prevented through the use of a "lock" line (LOCK*). The MASTER doing the indivisible multiple address cycle holds this LOCK* line low while it does any number or combination of read and write cycles. (The individual reads and writes look exactly like single reads cycles and single write cycles except that LOCK* is low.)

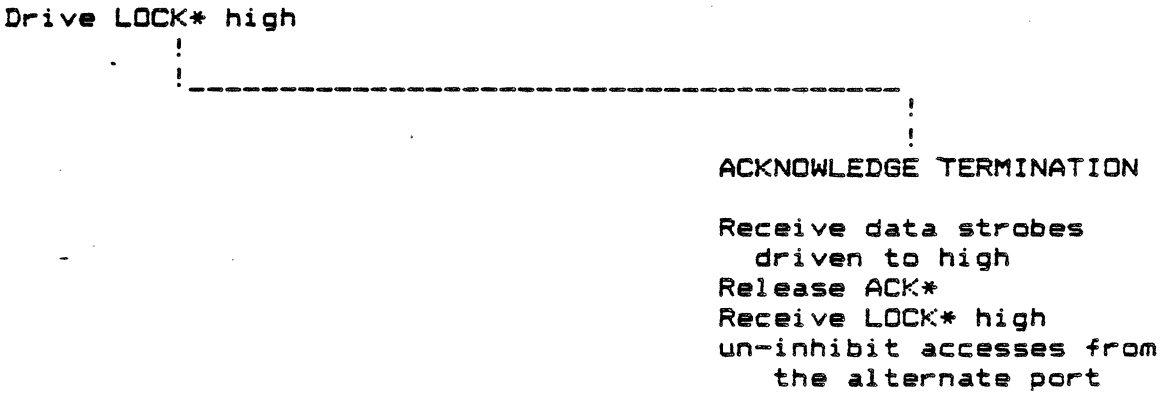
There are three kinds of indivisible multiple address cycles: byte, word, and longword. Figure 2-10 shows the flow diagram for one of these: an indivisible multiple address word cycle. Since such a cycle may consist of any combination of reads and writes, figure 3-6 shows the simple case of one read followed by one write. (Since figure 3-6 is so similar to figures 3-1 and 3-2 a detailed description won't be provided here.)

Figure 3-6
An Indivisible Multiple Address Word Cycle




RETRIEVE THE DATA
Receive data strobes driven to low
Sample LAS (high)
Sample READ (high)
Retrieve the data from the selected location
Place data on D00-D15
RESPOND TO THE MASTER
Drive ACK* to low
TERMINATE FIRST TRANSFER
Receive ACK* driven to low
Capture the data from D00-D15
Remove upper address
Drive data strobes to high
Drive UAS* to high
ACKNOWLEDGE TERMINATION
Receive data strobes driven to high
Tri-state D00-D15
Release ACK*
SEND THE LOWER ADDRESS
Drive LAS to low
Present lower address and longword=low
Drive LAS to high
LATCH THE LOWER ADDRESS
Receive LAS driven to high
Latch lower address and longword bit


The MASTER is required to wait a specified time to ensure that the SLAVE has latched the lower address before removing it and the longword bit from the address bus.
SEND THE UPPER ADDRESS
Remove lower address and longword bit
Present upper address
Drive UAS* to low
SPECIFY WRITE CYCLE
Drive READ low
SEND THE DATA
Present data on D00-D15
Drive both data strobes low
DECODE THE ADDRESS
Receive UAS* driven to low
Receive LOCK* low
Decode address
If address is not for this SLAVE
THEN wait for another cycle
ELSE select a memory location and inhibit accesses from the alternate port
STORE THE DATA
Receive data strobes driven to low
Sample LAS (low)
Sample READ (low)
Store the data in the selected location
RESPOND TO THE MASTER
Drive ACK* to low
TERMINATE CYCLE
Receive ACK* driven to low
Drive data strobes to high
Remove data from D00-D15
Drive UAS* to high
Remove upper address


Drive LOCK* high
ACKNOWLEDGE TERMINATION
Receive data strobes
driven to high
Release ACK*
Receive LOCK* high
un-inhibit accesses from
the alternate port

CHAPTER 4
DATA TRANSFER BUS TIMING DIAGRAMS

4.1 General Information

This chapter provides six pairs of timing diagrams which describe the 25 cycle types listed in section 3.1 as follows:

Fig. Nos.	Figure Title	Cycle types
4-1, 4-2	Single Write Cycle	1, 7, 13, 19
4-3, 4-4	Single Read Cycle	2, 8, 14, 20
4-5, 4-6	Sequential Write Cycle	3, 9, 15, 21
4-7, 4-8	Sequential Read Cycle	4, 10, 16, 22
4-9, 4-10	Indivisible Single Address Cycle	5, 11, 17, 23
4-11, 4-12	Indivisible Multiple Address Cycle	6, 12, 18, 24
4-13, 4-14	Withdrawn Cycle	25

A special notation has been used to describe the data strobe timing. The two data strobes (UDS* and LDS*) will not always make their transitions simultaneously. For purposes of these timing diagrams, DS"A"* represents the FIRST data strobe to make its transition, whether that is UDS* or LDS*, while DS"B"* represents the second. The reader should realize that the first data strobe to fall might NOT be the first to rise. Thus DS"A"* may represent one data strobe on its falling edge and the other on its rising edge.

4.2 Single Write Cycle

TABLE 4-1
Single Write Cycle: MASTER timing values

PARAMETER NUMBER	(NOTE A) MIN	MAX	NOTES
1	20		B
2	20		B
3	10		B
4	10		B
5	30		B
6	5		B
7	50		B
8	40		B
9	5		B
10	10		B
11	0	5	B
12	35		B
13	10		B
14	0	200	D, F
15		0	E, G
17	0		C
18	0		C
19	10		B
20	20		B
21	0	500	C
22	0	5	B
23	10		B
24	10		B
25	0	40	D

(SEE NOTES ON FOLLOWING PAGE)

TABLE 4-1 NOTES

A. All times are given in nanoseconds unless otherwise noted.

B. The MASTER must guarantee this timing between two of its outgoing signal transitions.

C. The MASTER must wait for the incoming signal edge from the SLAVE before changing the level on its outgoing signal.

D. This is a guarantee that the SLAVE will not change the incoming signal until the MASTER changes its outgoing signal.

E. The MASTER is guaranteed this timing between two of its incoming signal transitions.

F. This maximum is specified in Microseconds.

G. In order to ensure compatibility between boards, all MASTERS must be able to abort the cycle if DERR* is driven low at the same time as ACK*. In addition, some MASTERS may also have on-board jumpers or control registers that allow it to accept and respond to falling edges on DERR* as late as 125 nanoseconds after the falling edge of ACK*.

FIGURE 4-i
Single Write Cycle: MASTER timing diagram

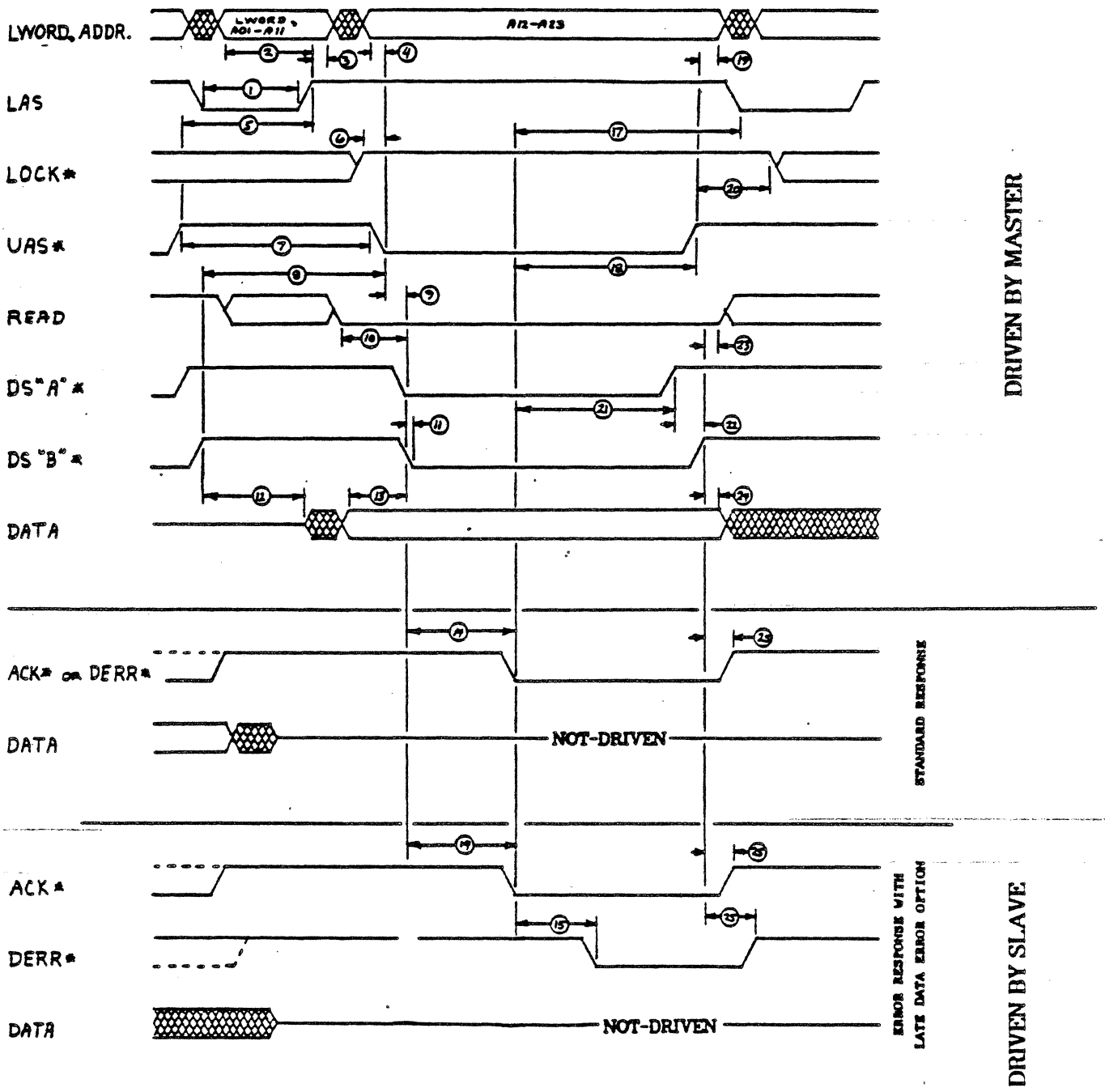


TABLE 4-2
Single Write Cycle: SLAVE timing values

PARAMETER NUMBER	(NOTE A) MIN	MAX	NOTES
1	15		B
2	15		B
3	5		B
4	5		B
5	20		B
6	0		B
7	45		B
8	35		B
9	5		B
10	10		B
11	0	10	B
12	30		B
13	5		B
14	0	200	D, F
15		0	E, G
16	0		C
18	0		C
19	5		B
20	15		B
21	0	505	C
22	0	10	B
23	5		B
24	5		B
25	0	30	D

(SEE NOTES ON FOLLOWING PAGE)

TABLE 4-2 NOTES

A. All times are given in nanoseconds unless otherwise noted.

B. The SLAVE is guaranteed this timing between two of its incoming signal transitions.

C. This is a guarantee that the MASTER will not change the incoming signal until the SLAVE changes its outgoing signal.

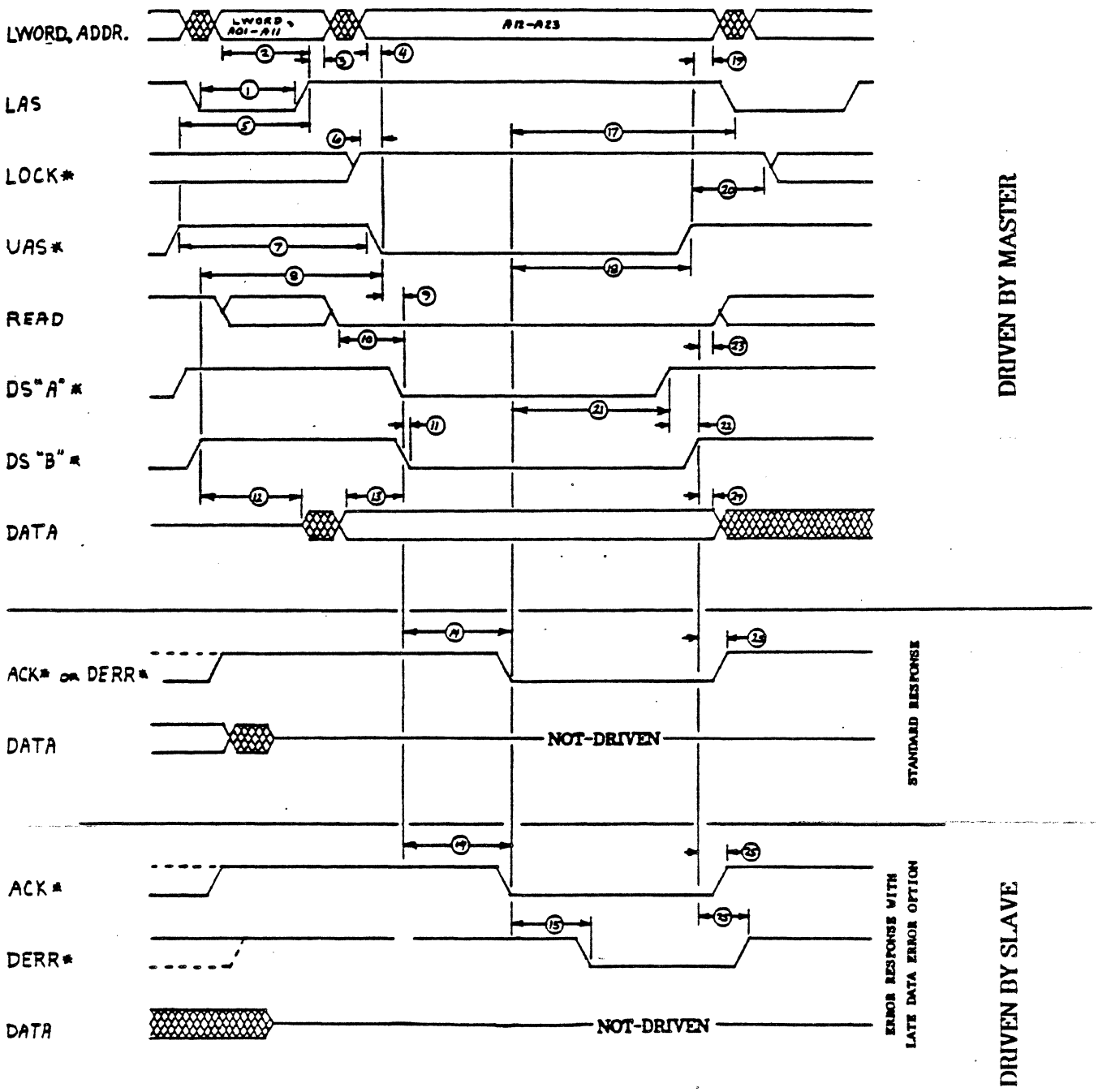
D. The SLAVE must wait for the incoming signal edge from the MASTER before changing the level on its outgoing signal.

E. The SLAVE must guarantee this timing between two of its outgoing signal transitions.

F. This maximum is specified in Microseconds.

G. In order to ensure compatibility between boards, all SLAVES must be able to provide a falling edge on DERR* no later than the falling edge on ACK*. In addition, some SLAVES may also have on-board jumpers or control registers that allow it to generate a falling edge on ACK* as much as 125 nanoseconds prior to the falling edge on DERR*.

FIGURE 4-2
Single Write Cycle: SLAVE timing diagram



4.3 Single Read Cycle

TABLE 4-3
Single Read Cycle: MASTER timing values

PARAMETER NUMBER	(NOTE A) MIN	MAX	NOTES
1	20		B
2	20		B
3	10		B
4	10		B
5	30		B
6	5		B
7	50		B
8	40		B
9	5		B
10	10		B
11	0	5	B
12	0		B
13	0	200	D, F
14	0		D
15		5	E, H
16		0	E, G
18	0		C
19	0		C
20	10		B
21	20		B
22	0	500	C
23	0	5	B
24	10		B
25	35		B
26	0	40	D
27	0		D
28	0	35	D

(SEE NOTES ON FOLLOWING PAGE)

TABLE 4-3 NOTES

A. All times are given in nanoseconds unless otherwise noted.

B. The MASTER must guarantee this timing between two of its outgoing signal transitions.

C. The MASTER must wait for the incoming signal edge from the SLAVE before changing the level on its outgoing signal.

D. This is a guarantee that the SLAVE will not change the incoming signal until the MASTER changes its outgoing signal.

E. The MASTER is guaranteed this timing between two of its incoming signal transitions.

F. This maximum is specified in Microseconds.

G. In order to ensure compatibility between boards, all MASTERS must be able to abort the cycle if DERR* is driven low at the same time as ACK. In addition, some MASTERS may also have on-board jumpers or control registers that allow it to accept and respond to falling edges on DERR* as late as 125 nanoseconds after the falling edge of ACK*.

H. All SLAVE boards can be configured to guarantee that the data on the data bus will be valid when ACK* falls, but some may be configurable to generate an "early ACK*" as much as 125 nanoseconds before the data is valid. In order to take advantage of the performance benefits that this offers, some MASTERS may be equipped with on-board jumpers or control registers that allow them to accept an early ACK* falling edge as much as 125 nanoseconds before the data is valid.

FIGURE 4-3
Single Read Cycle: MASTER timing diagram

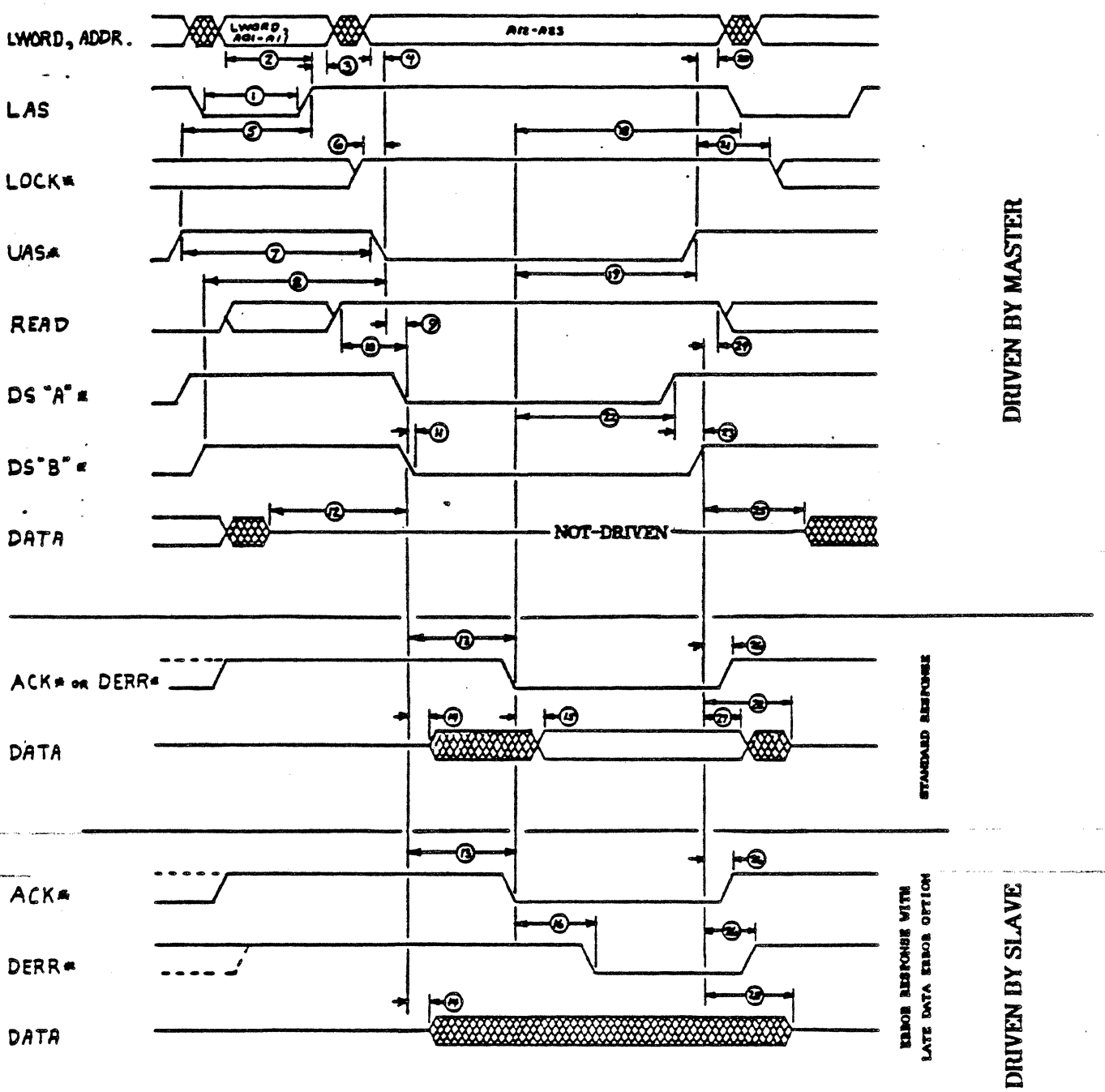


TABLE 4-4
Single Read Cycle: SLAVE timing values

PARAMETER NUMBER	(NOTE A) MIN	MAX	NOTES
1	15		B
2	15		B
3	5		B
4	5		B
5	20		B
6	0		B
7	45		B
8	30		B
9	0		B
10	5		B
11	0	10	B
12	0		B
13	0	200	D, F
14	0		E
15		0	E, H
16		0	E, G
18	0		C
19	0		C
20	5		B
21	15		B
22	0	505	C
23	0	10	B
24	5		B
25	30		B
26	0	30	D
27	0		D
28	0	30	D

(SEE NOTES ON FOLLOWING PAGE)

TABLE 4-4 NOTES

A. All times are given in nanoseconds unless otherwise noted.

B. The SLAVE is guaranteed this timing between two of its incoming signal transitions.

C. This is a guarantee that the MASTER will not change the incoming signal until the SLAVE changes its outgoing signal.

D. The SLAVE must wait for the incoming signal edge from the MASTER before changing the level on its outgoing signal.

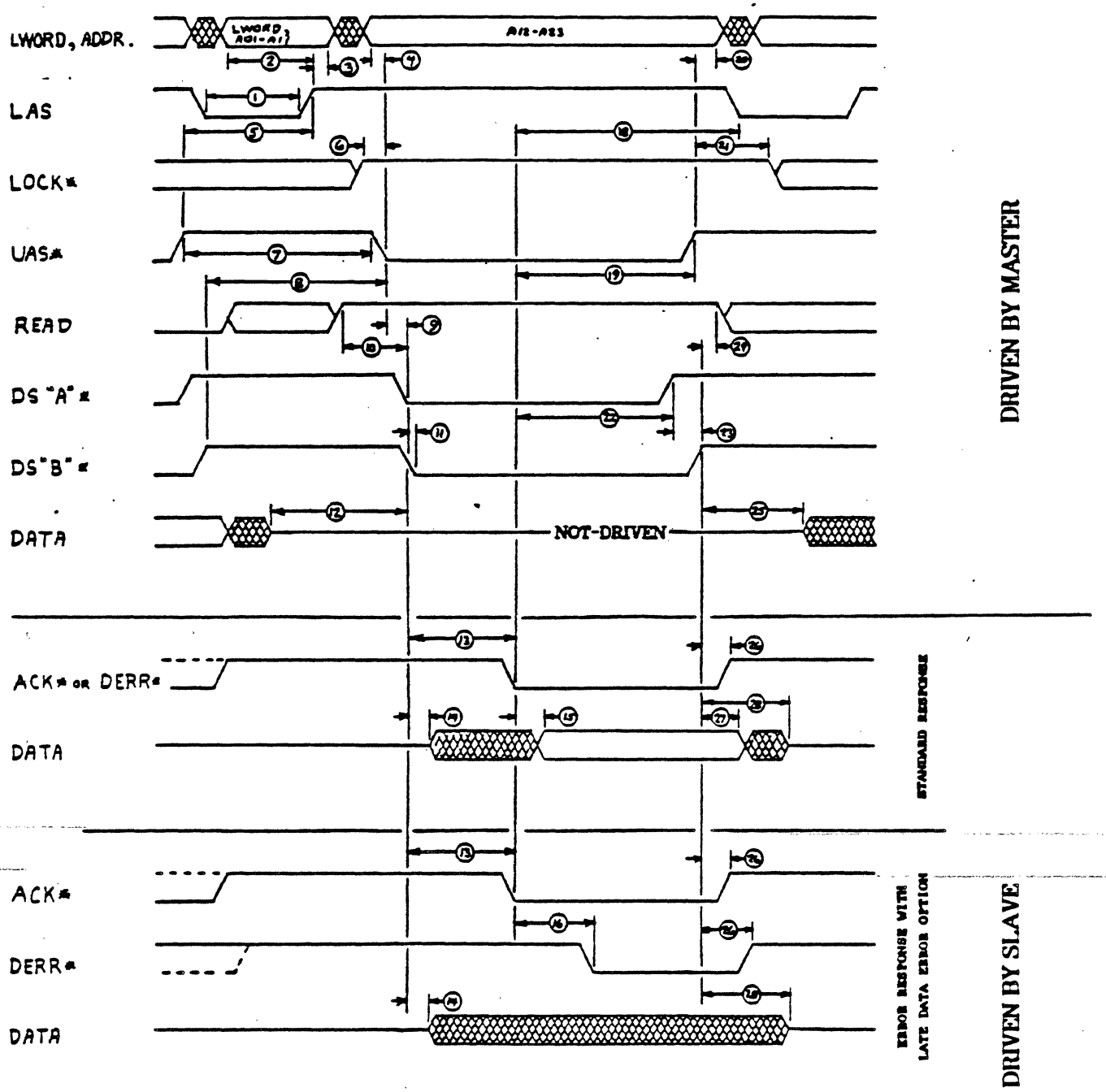
E. The SLAVE must guarantee this timing between two of its outgoing signal transitions.

F. This maximum is specified in Microseconds.

G. In order to ensure compatibility between boards, all SLAVES must be able to provide a falling edge on DERR* no later than the falling edge on ACK*. In addition, some SLAVES may also have on-board jumpers or control registers that allow them to generate a falling edge on ACK* as much as 125 nanoseconds prior to the falling edge on DERR*.

H. In order to ensure compatibility between boards, all SLAVES must be configurable to provide valid data no later than the falling edge of ACK*. In addition, some SLAVES may also have on-board jumpers or control registers that allow them to generate a falling edge on ACK* as much as 125 nanoseconds prior to placing valid data on the data bus.

FIGURE 4-4
Single Read Cycle: SLAVE timing diagram



4.4 Sequential Write Cycle

TABLE 4-5
Sequential Write Cycle: MASTER timing values

PARAMETER NUMBER	(NOTE A) MIN	MAX	NOTES
1	20		B
2	20		B
3	10		B
4	10		B
5	30		B
6	20		B
7	5		B
8	5		B
9	50		B
10	40		B
11	10		B
12	10		B
13	0	5	B
14	35		B
15	10		B
16	0	200	D, F
18		5	E, G
19	0	500	C
20	0	5	B
21	10		B
22	10		B
23	0	40	D
24	0		C
25	0		C
26	20		B
27	20		B
28	10		B
29	30		B
30	30		B

(SEE NOTES ON FOLLOWING PAGE)

TABLE 4-5 NOTES

A. All times are given in nanoseconds unless otherwise noted.

B. The MASTER must guarantee this timing between two of its outgoing signal transitions.

C. The MASTER must wait for the incoming signal edge from the SLAVE before changing the level on its outgoing signal.

D. This is a guarantee that the SLAVE will not change the incoming signal until the MASTER changes its outgoing signal.

E. The MASTER is guaranteed this timing between two of its incoming signal transitions.

F. This maximum is specified in Microseconds.

G. In order to ensure compatibility between boards, all MASTERS must be able to abort the cycle if DERR* is driven low at the same time as ACK. In addition, some MASTERS may also have on-board jumpers or control registers that allow it to accept and respond to falling edges on DERR* as late as 125 nanoseconds after the falling edge of ACK*.

FIGURE 4-5
Sequential Write Cycle: MASTER timing diagram

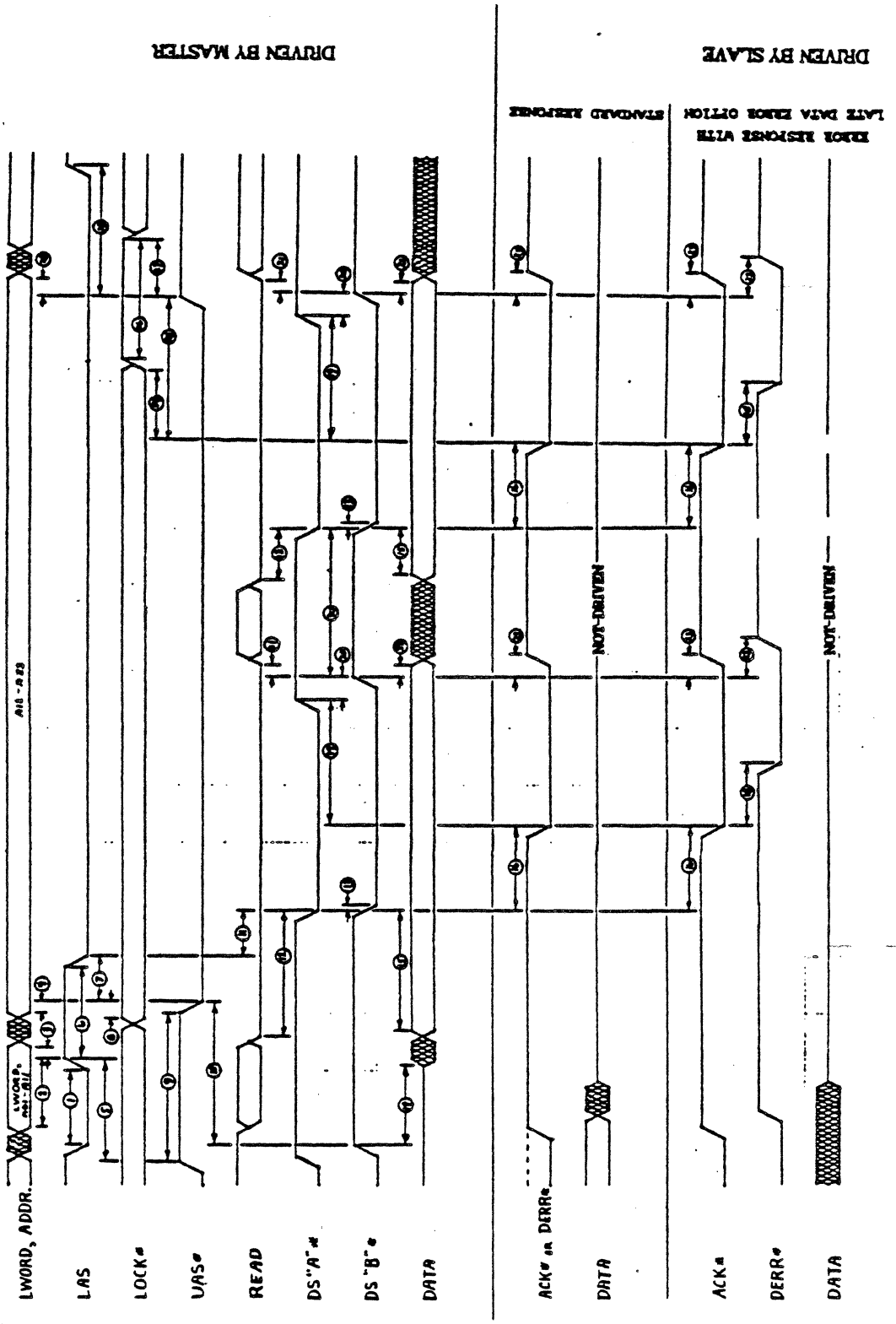


TABLE 4-6
Sequential Write Cycle: SLAVE timing values

PARAMETER NUMBER	(NOTE A) MIN	MAX	NOTES
1	15		B
2	15		B
3	15		B
4	5		B
5	20		B
6	15		B
7	0		B
8	0		B
9	45		B
10	30		B
11	5		B
12	5		B
13		10	B
14	30		B
15	5		B
16	0	200	D, F
18		0	E, G
19	0	505	C
20	0	10	B
21	5		B
22	5		B
23	0	30	D
24	0		C
25	0		C
26	15		B
27	15		B
28	5		B
29	25		B
30	30		B

(SEE NOTES ON FOLLOWING PAGE)

TABLE 4-6 NOTES

A. All times are given in nanoseconds unless otherwise noted.

B. The SLAVE is guaranteed this timing between two of its incoming signal transitions.

C. This is a guarantee that the MASTER will not change the incoming signal until the SLAVE changes its outgoing signal.

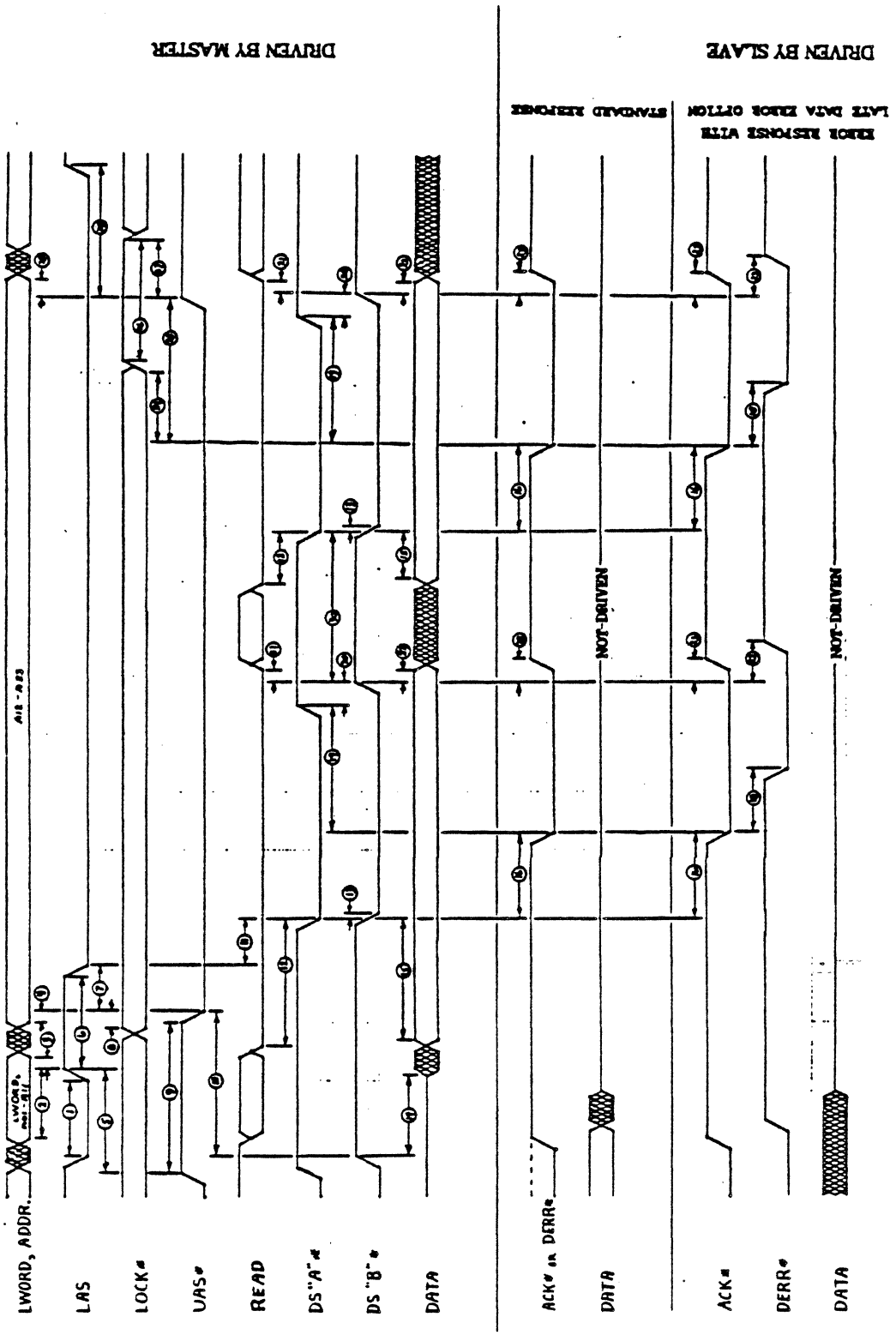
D. The SLAVE must wait for the incoming signal edge from the MASTER before changing the level on its outgoing signal.

E. The SLAVE must guarantee this timing between two of its outgoing signal transitions.

F. This maximum is specified in Microseconds.

G. In order to ensure compatibility between boards, all SLAVES must be able to provide a falling edge on DERR* no later than the falling edge on ACK*. In addition, some SLAVES may also have on-board jumpers or control registers that allow it to generate a falling edge on ACK* as much as 125 nanoseconds prior to the falling edge on DERR*.

FIGURE 4-6
Sequential Write Cycle: SLAVE timing diagram



4.3 Sequential Read Cycle

TABLE 4-7
Sequential Read Cycle: MASTER timing values

PARAMETER NUMBER	(NOTE A). MIN	MAX	NOTES
1	20		B
2	20		B
3	10		B
4	10		B
5	30		B
6	20		B
7	5		B
8	5		B
9	50		B
10	40		B
11	10		B
12	10		B
13	0	5	B
14	0		B
15	0	200	D, F
16	0		D
17		5	E, H
19		5	E, G
20	0	500	C
21	0	5	B
22	10		C
23	0	40	D
24	0		D
25	0	35	D
26	40		B
27	0		C
28	0		C
29	20		B
30	10		B
31	30		B
32	20		B

(SEE NOTES ON FOLLOWING PAGE)

TABLE 4-7 NOTES

A. All times are given in nanoseconds unless otherwise noted.

B. The MASTER must guarantee this timing between two of its outgoing signal transitions.

C. The MASTER must wait for the incoming signal edge from the SLAVE before changing the level on its outgoing signal.

D. This is a guarantee that the SLAVE will not change the incoming signal until the MASTER changes its outgoing signal.

E. The MASTER is guaranteed this timing between two of its incoming signal transitions.

F. This maximum is specified in Microseconds.

G. In order to ensure compatibility between boards, all MASTERS must be able to abort the cycle if DERR* is driven low at the same time as ACK. In addition, some MASTERS may also have on-board jumpers or control registers that allow it to accept and respond to falling edges on DERR* as late as 125 nanoseconds after the falling edge of ACK*.

H. All SLAVE boards can be configured to guarantee that the data on the data bus will be valid when ACK* falls, but some may be configurable to generate an "early ACK*" as much as 125 nanoseconds before the data is valid. In order to take advantage of the performance benefits that this offers, some MASTERS may be equipped with on-board jumpers or control registers that allow them to accept an early ACK* falling edge as much as 125 nanoseconds before the data is valid.

FIGURE 4-7
Sequential Read Cycle: MASTER timing diagram

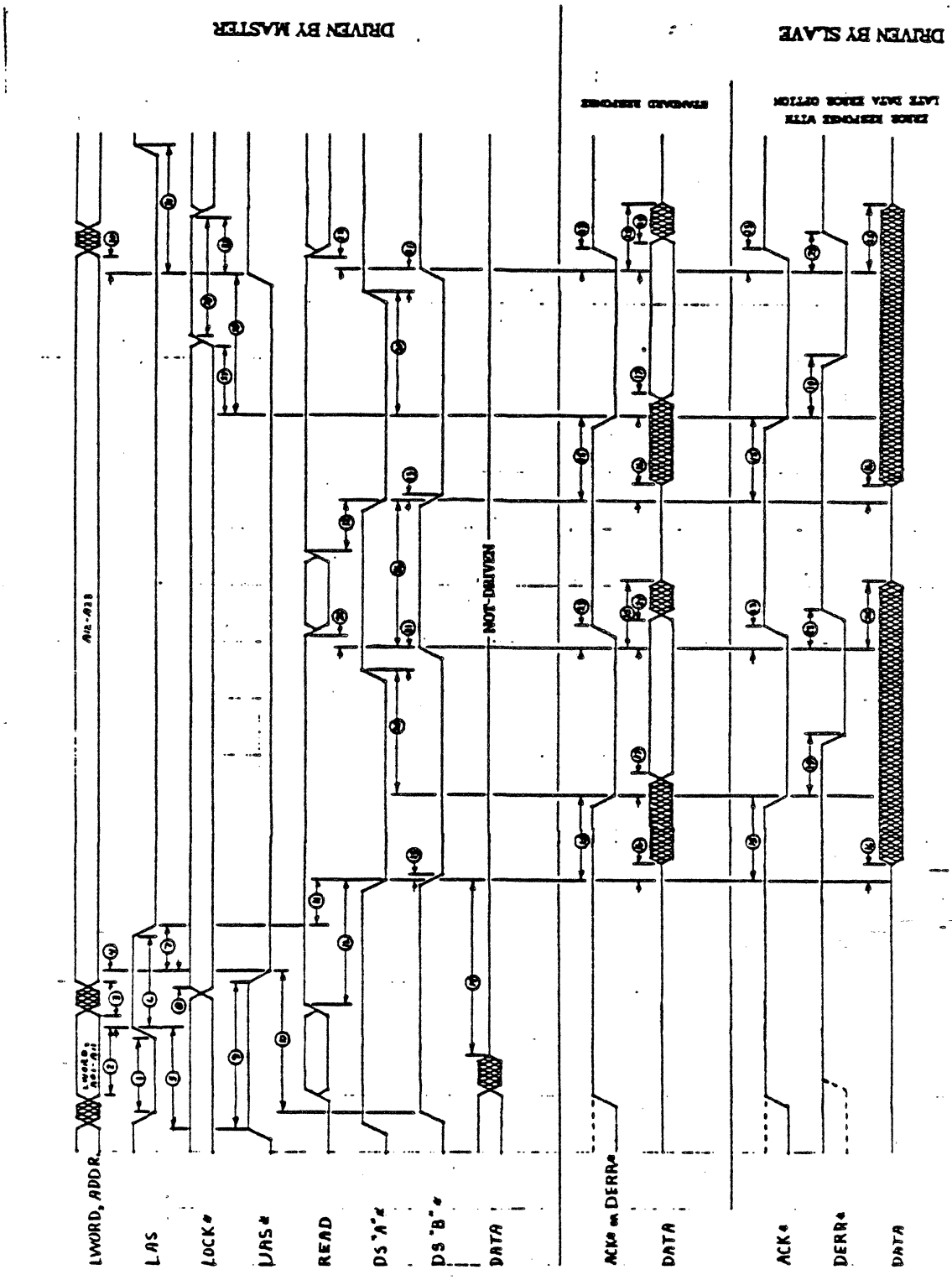


TABLE 4-8
Sequential Read Cycle: SLAVE timing values

PARAMETER NUMBER	(NOTE A) MIN	MAX	NOTES
1	15		B
2	15		B
3	5		B
4	5		B
5	20		B
6	15		B
7	0		B
8	0		B
9	45		B
10	30		B
11	5		B
12	5		B
13	0	10	B
14	0		B
15	0	200	D, F
16	0		D
17		0	E, H
19		0	E, G
20	0	505	C
21	0	10	B
22	5		C
23	0	30	D
24	0		D
25	0	30	D
26	30		B
27	0		C
28	0		C
29	15		B
30	5		B
31	25		B
32	15		B

(SEE NOTES ON FOLLOWING PAGE)

TABLE 4-8 NOTES

A. All times are given in nanoseconds unless otherwise noted.

B. The SLAVE is guaranteed this timing between two of its incoming signal transitions.

C. This is a guarantee that the MASTER will not change the incoming signal until the SLAVE changes its outgoing signal.

D. The SLAVE must wait for the incoming signal edge from the MASTER before changing the level on its outgoing signal.

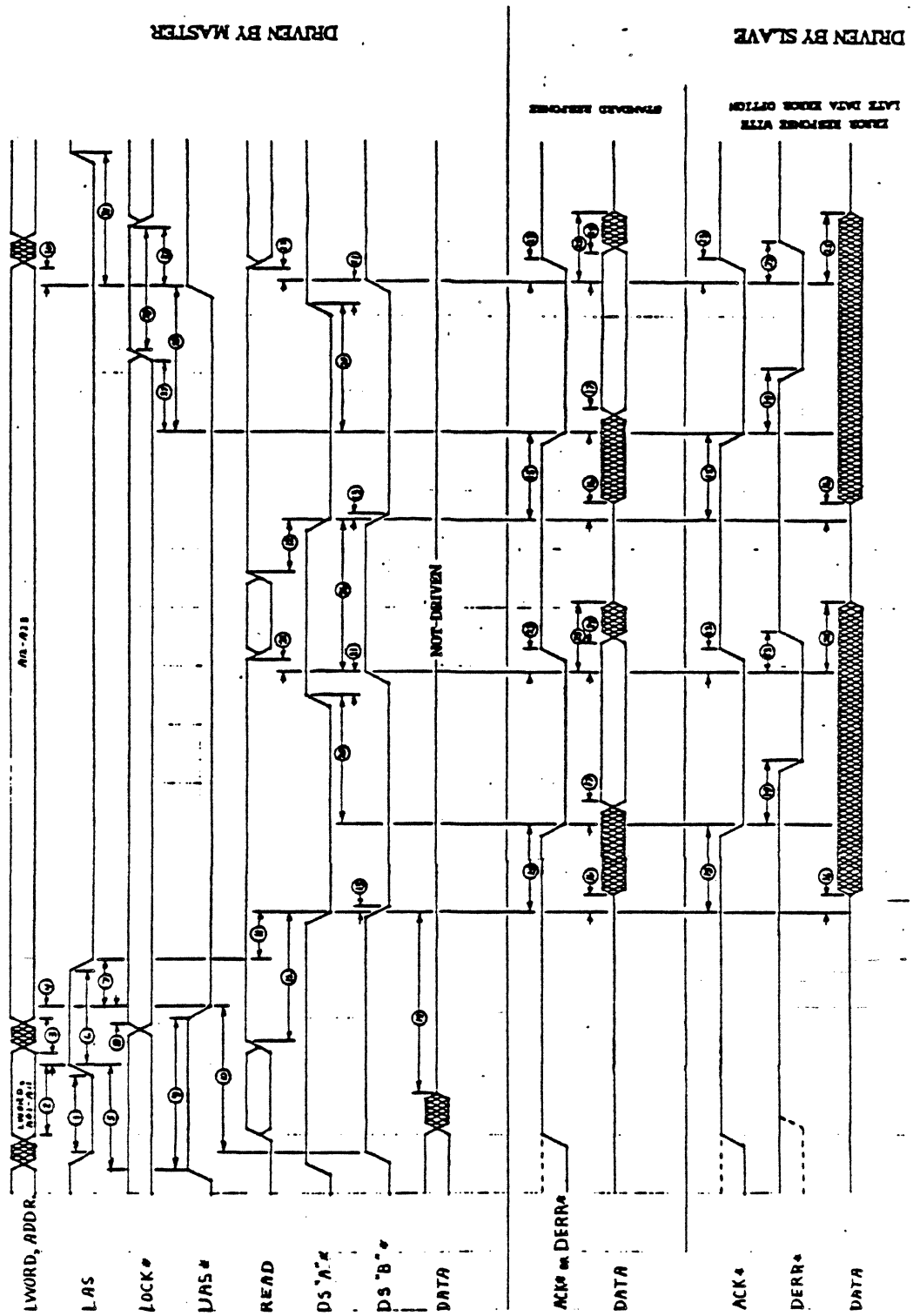
E. The SLAVE must guarantee this timing between two of its outgoing signal transitions.

F. This maximum is specified in Microseconds.

G. In order to ensure compatibility between boards, all SLAVES must be able to provide a falling edge on DERR* no later than the falling edge on ACK*. In addition, some SLAVES may also have on-board jumpers or control registers that allow it to generate a falling edge on ACK* as much as 125 nanoseconds prior to the falling edge on DERR*.

H. In order to ensure compatibility between boards, all SLAVES must be configurable to provide valid data no later than the falling edge of ACK*. In addition, some SLAVES may also have on-board jumpers or control registers that allow them to generate a falling edge on ACK* as much as 125 nanoseconds prior to placing valid data on the data bus.

FIGURE 4-8
Sequential Read Cycle: SLAVE timing diagram



4.6 Indivisible Single Address Cycle

TABLE 4-9
Indivisible Single Address Cycle: MASTER timing values

PARAMETER NUMBER	(NOTE A) MIN	MAX	NOTES
1	20		B
2	20		B
3	10		B
4	10		B
5	30		B
6	5		B
7	50		B
8	40		B
9	5		B
10	10		B
11	0	5	B
12	0		B
13	0	200	D, F
14	0		D
15		5	E, H
16		5	E, G
18	0	500	C
19	10		B
20	0	5	B
21	45		B
22	35		B
23	0	40	D
24	0	35	D
25	0		D
26	10		B
27	0		C
28	0		C
29	0		C
30	20		B
31	10		B
32	20		B
33	10		B

(SEE NOTES ON FOLLOWING PAGE)

TABLE 4-9 NOTES

A. All times are given in nanoseconds unless otherwise noted.

B. The MASTER must guarantee this timing between two of its outgoing signal transitions.

C. The MASTER must wait for the incoming signal edge from the SLAVE before changing the level on its outgoing signal.

D. This is a guarantee that the SLAVE will not change the incoming signal until the MASTER changes its outgoing signal.

E. The MASTER is guaranteed this timing between two of its incoming signal transitions.

F. This maximum is specified in Microseconds.

G. In order to ensure compatibility between boards, all MASTERS must be able to abort the cycle if DERR* is driven low at the same time as ACK. In addition, some MASTERS may also have on-board jumpers or control registers that allow it to accept and respond to falling edges on DERR* as late as 125 nanoseconds after the falling edge of ACK*.

H. All SLAVE boards can be configured to guarantee that the data on the data bus will be valid when ACK* falls, but some may be configurable to generate an "early ACK*" as much as 125 nanoseconds before the data is valid. In order to take advantage of the performance benefits that this offers, some MASTERS may be equipped with on-board jumpers or control registers that allow them to accept an early ACK* falling edge as much as 125 nanoseconds before the data is valid.

FIGURE 4-9
Indivisible Single Address Cycle: MASTER timing diagram.

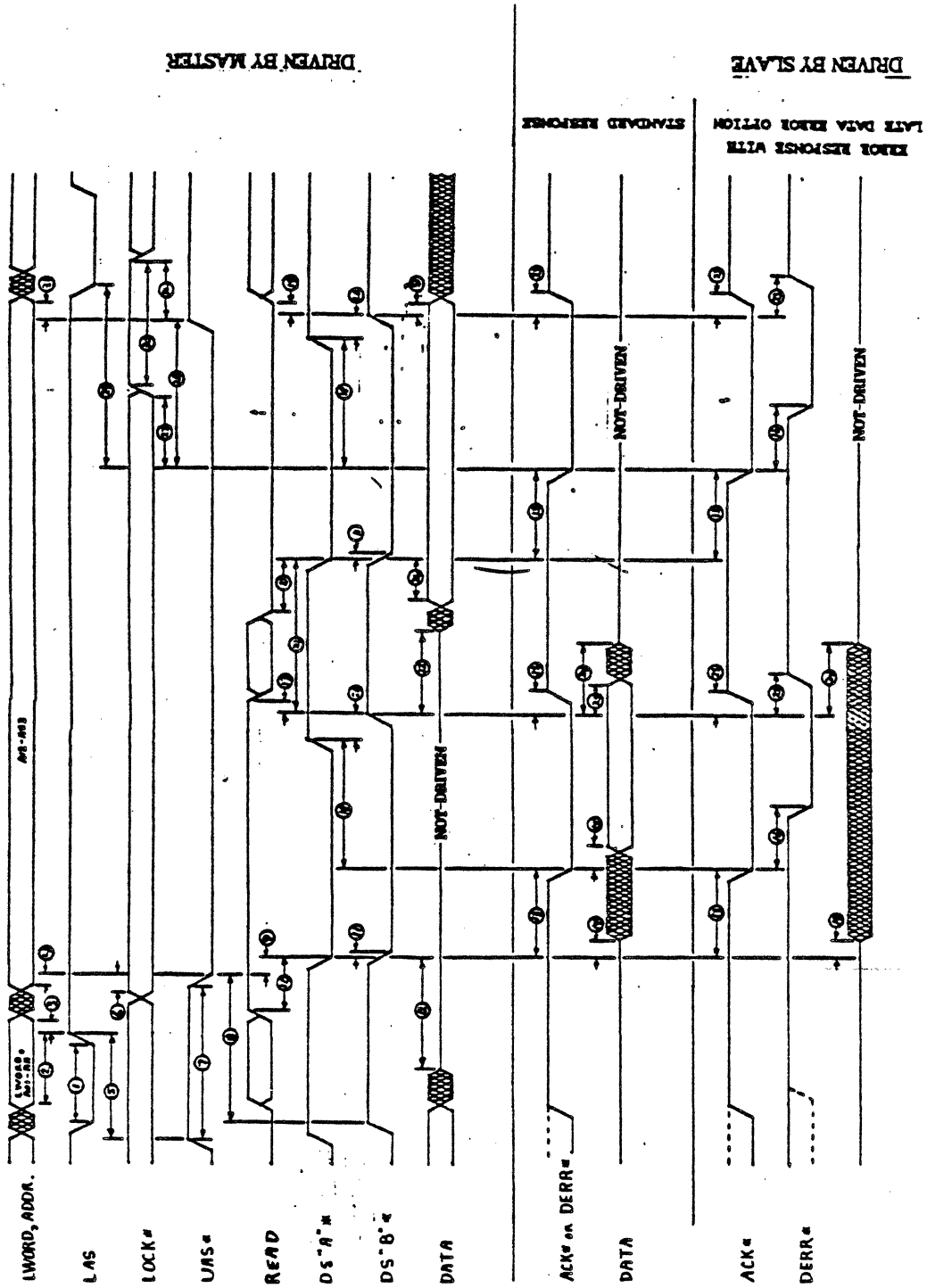


TABLE 4-10
Indivisible Single Address Cycle: SLAVE timing values

PARAMETER NUMBER	(NOTE A) MIN	MAX	NOTES
1	15		B
2	15		B
3	5		B
4	5		B
5	20		B
6	0		B
7	45		B
8	30		B
9	0		B
10	5		B
11	0	10	B
12	0		B
13	0	200	D, F
14	0		D
15		0	E, H
16		0	E, G
18	0	505	C
19	5		B
20	0	10	B
21	40		B
22	35		B
23	0	30	D
24	0	30	D
25	0		D
26	5		B
27	0		C
28	0		C
29	0		C
30	15		B
31	5		B
32	15		B
33	5		B

(SEE NOTES ON FOLLOWING PAGE)

TABLE 4-10 NOTES

A. All times are given in nanoseconds unless otherwise noted.

B. The SLAVE is guaranteed this timing between two of its incoming signal transitions.

C. This is a guarantee that the MASTER will not change the incoming signal until the SLAVE changes its outgoing signal.

D. The SLAVE must wait for the incoming signal edge from the MASTER before changing the level on its outgoing signal.

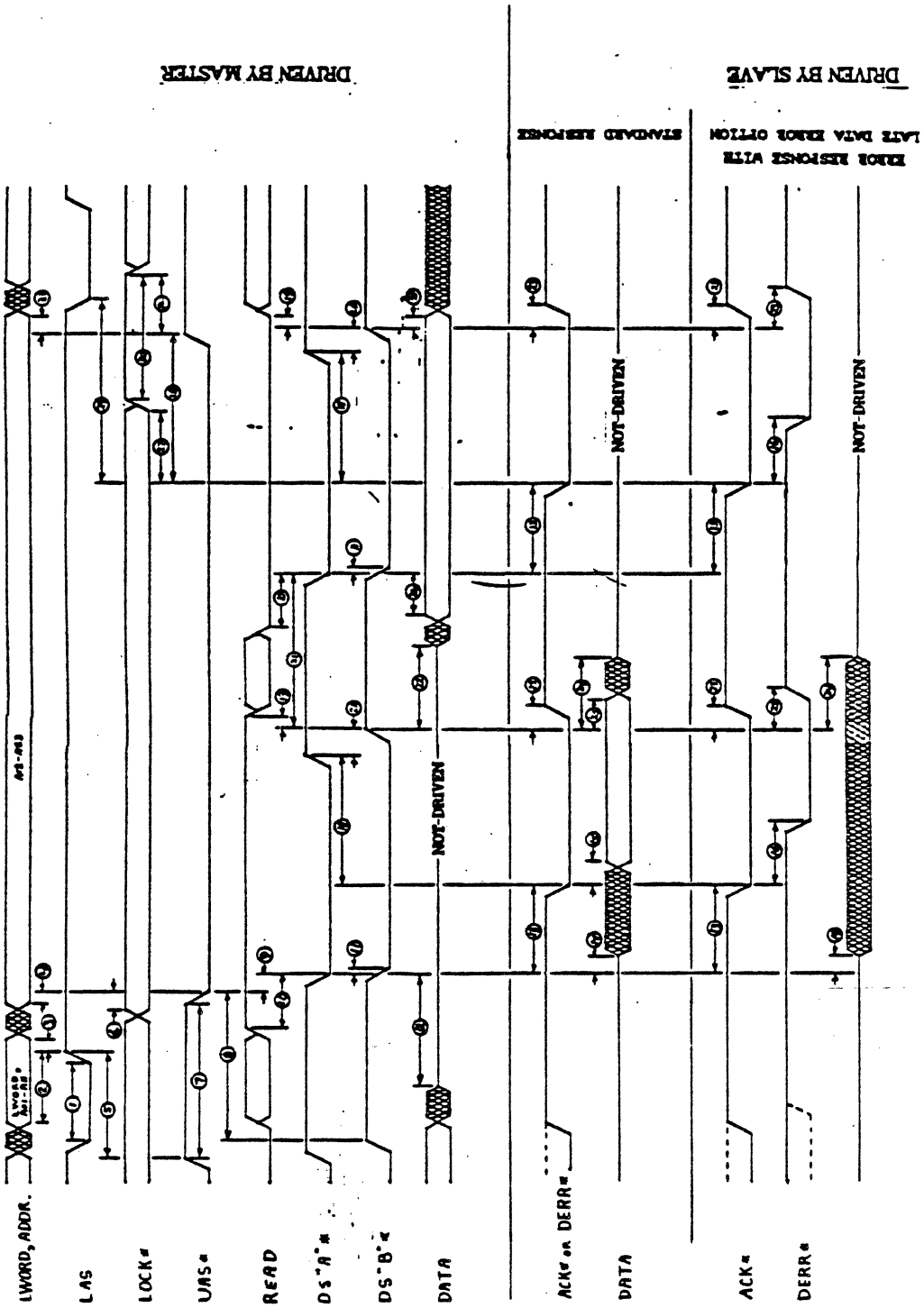
E. The SLAVE must guarantee this timing between two of its outgoing signal transitions.

F. This maximum is specified in Microseconds.

G. In order to ensure compatibility between boards, all SLAVES must be able to provide a falling edge on DERR* no later than the falling edge on ACK*. In addition, some SLAVES may also have on-board jumpers or control registers that allow it to generate a falling edge on ACK* as much as 125 nanoseconds prior to the falling edge on DERR*.

H. In order to ensure compatibility between boards, all SLAVES must be configurable to provide valid data no later than the falling edge of ACK*. In addition, some SLAVES may also have on-board jumpers or control registers that allow them to generate a falling edge on ACK* as much as 125 nanoseconds prior to placing valid data on the data bus.

FIGURE 4-10
Indivisible Single Address Cycle: SLAVE timing diagram



4.7 Indivisible Multiple Address Cycle

TABLE 4-11
Indivisible Multiple Address Cycle: MASTER timing values

PARAMETER NUMBER	(NOTE A) MIN	MAX	NOTES
1	20		B
2	20		B
3	10		B
4	10		B
5	30		B
6	5		B
7	50		B
8	40		B
9	5		B
10	10		B
11	0	5	B
12	0		B
13	0	200	D, F
14	0		D
15		5	E, H
16		5	E, G
18	0		C
19	0		C
20	10		B
21	0	500	C
22	10		B
23	0	5	B
24	0	40	D
25	0		D
26	0	35	D
27	35		B
28	10		B
29	0		C
30	20		B
31	20		B
32	10		B

(SEE NOTES ON FOLLOWING PAGE)

TABLE 4-11 NOTES

A. All times are given in nanoseconds unless otherwise noted.

B. The MASTER must guarantee this timing between two of its outgoing signal transitions.

C. The MASTER must wait for the incoming signal edge from the SLAVE before changing the level on its outgoing signal.

D. This is a guarantee that the SLAVE will not change the incoming signal until the MASTER changes its outgoing signal.

E. The MASTER is guaranteed this timing between two of its incoming signal transitions.

F. This maximum is specified in Microseconds.

G. In order to ensure compatibility between boards, all MASTERS must be able to abort the cycle if DERR* is driven low at the same time as ACK. In addition, some MASTERS may also have on-board jumpers or control registers that allow it to accept and respond to falling edges on DERR* as late as 125 nanoseconds after the falling edge of ACK*.

H. All SLAVE boards can be configured to guarantee that the data on the data bus will be valid when ACK* falls, but some may be configurable to generate an "early ACK*" as much as 125 nanoseconds before the data is valid. In order to take advantage of the performance benefits that this offers, some MASTERS may be equipped with on-board jumpers or control registers that allow them to accept an early ACK* falling edge as much as 125 nanoseconds before the data is valid.

FIGURE 4-11
Indivisible Multiple Address Cycle: MASTER timing diagram

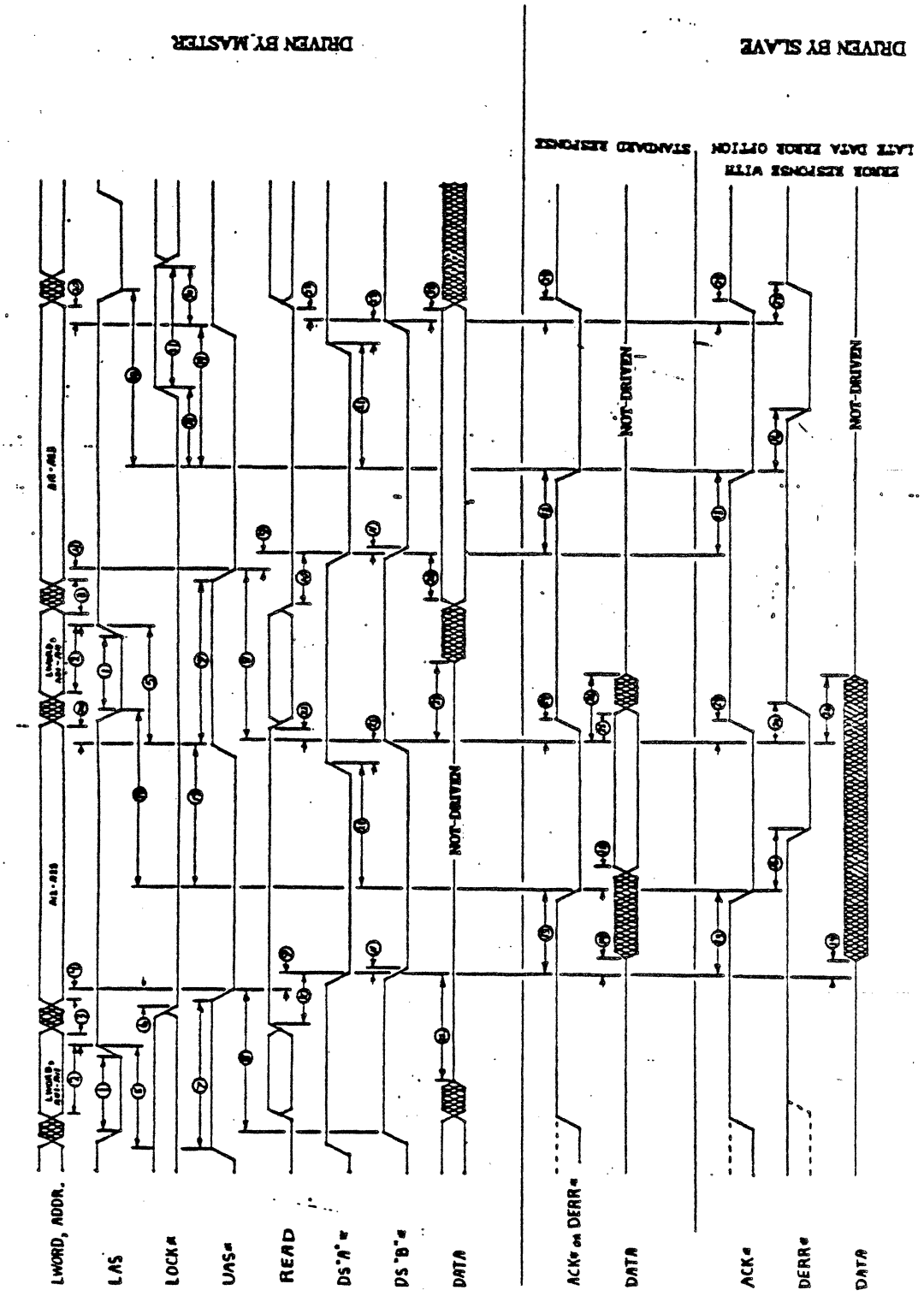


TABLE 4-12
Indivisible Multiple Address Cycle: SLAVE timing values

PARAMETER NUMBER	(NOTE A) MIN	MAX	NOTES
1	15		B
2	15		B
3	5		B
4	5		B
5	20		B
6	0		B
7	45		B
8	30		B
9	0		B
10	5		B
11	0	10	B
12	0		B
13	0	200	D, F
14	0		D
15		0	E, H
16		0	E, G
18	0		C
19	0		C
20	5		B
21	0	505	C
22	5		B
23	0	10	B
24	0	30	D
25	0		D
26	0	30	D
27	35		B
28	5		B
29	0		C
30	15		B
31	15		B
32	5		B

(SEE NOTES ON FOLLOWING PAGE)

TABLE 4-12 NOTES

A. All times are given in nanoseconds unless otherwise noted.

B. The SLAVE is guaranteed this timing between two of its incoming signal transitions.

C. This is a guarantee that the MASTER will not change the incoming signal until the SLAVE changes its outgoing signal.

D. The SLAVE must wait for the incoming signal edge from the MASTER before changing the level on its outgoing signal.

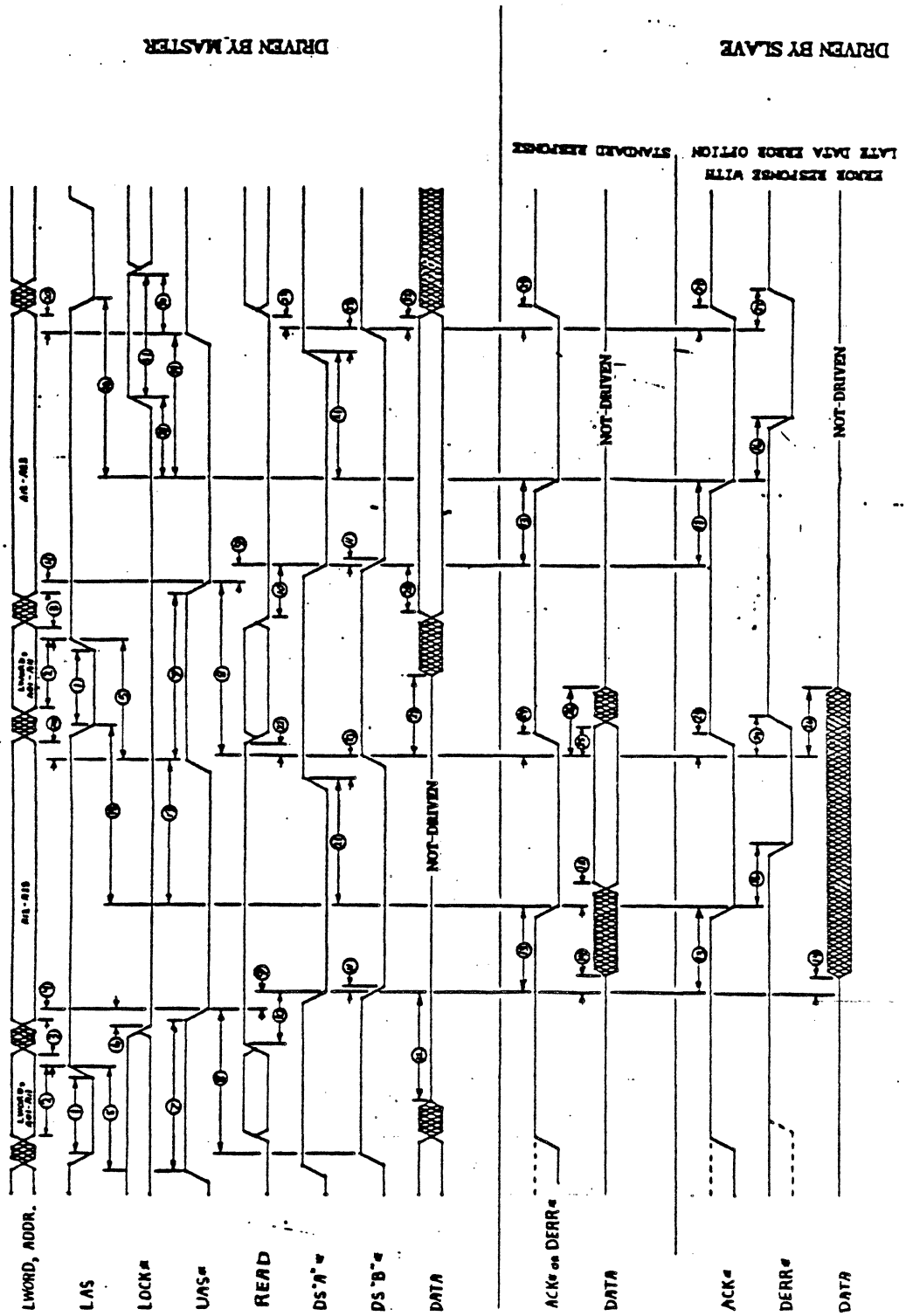
E. The SLAVE must guarantee this timing between two of its outgoing signal transitions.

F. This maximum is specified in Microseconds.

G. In order to ensure compatibility between boards, all SLAVES must be able to provide a falling edge on DERR* no later than the falling edge on ACK*. In addition, some SLAVES may also have on-board jumpers or control registers that allow it to generate a falling edge on ACK* as much as 125 nanoseconds prior to the falling edge on DERR*.

H. In order to ensure compatibility between boards, all SLAVES must be configurable to provide valid data no later than the falling edge of ACK*. In addition, some SLAVES may also have on-board jumpers or control registers that allow them to generate a falling edge on ACK* as much as 125 nanoseconds prior to placing valid data on the data bus.

FIGURE 4-12
Indivisible Multiple Address Cycle: SLAVE timing diagram



4.8 Withdrawn Cycle

TABLE 4-13
Withdrawn Cycle: MASTER timing values

PARAMETER NUMBER	(NOTE A) MIN	MAX	NOTES
1	20		B
2	30		B
3	20		B
4	35		B

NOTES:

A. All times are given in nanoseconds unless otherwise noted.

B. The MASTER must guarantee this timing between two of its outgoing signal transitions.

FIGURE 4-13
Withdrawn Cycle: MASTER timing diagram

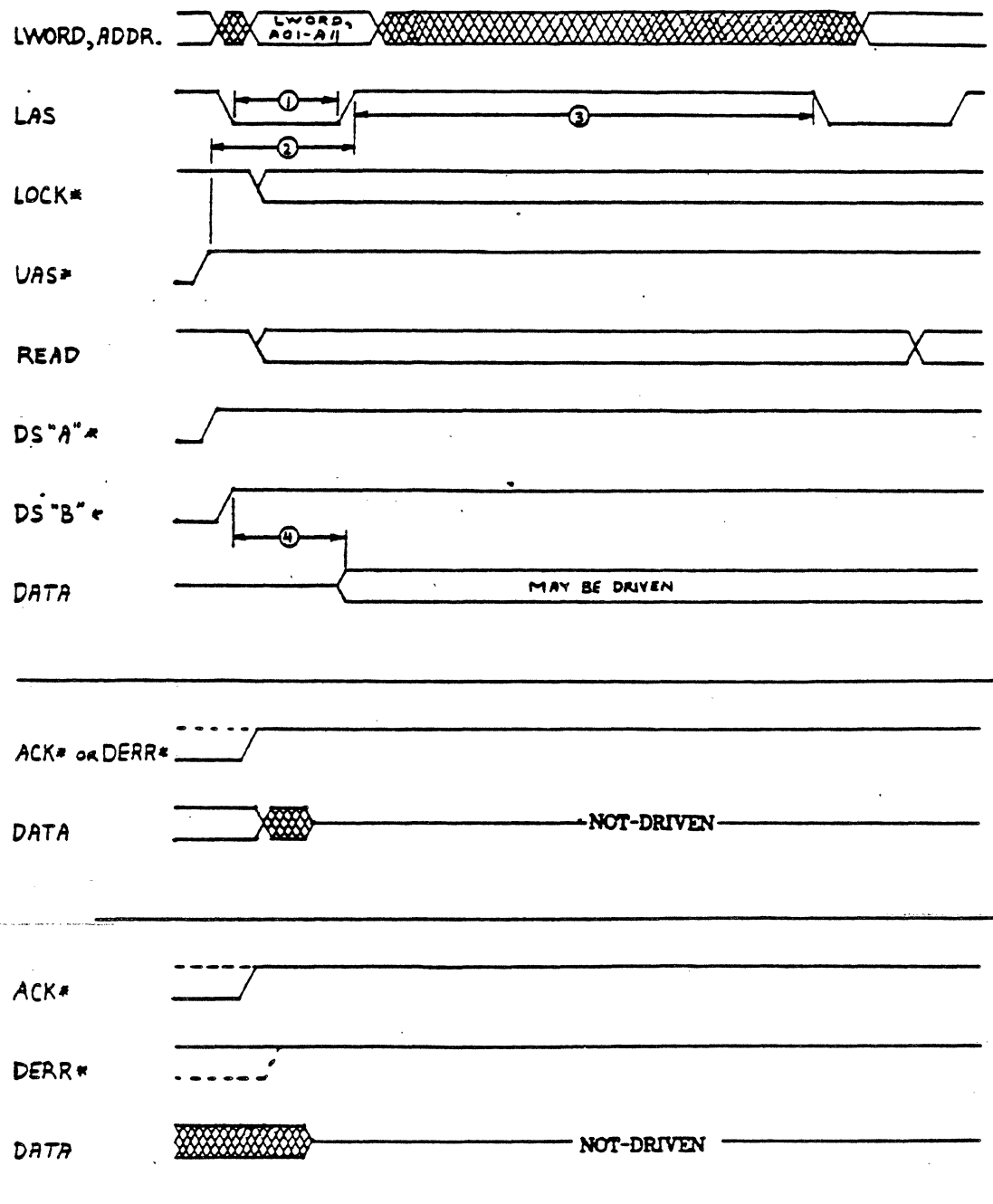


TABLE 4-14
Withdrawn Cycle: SLAVE timing values

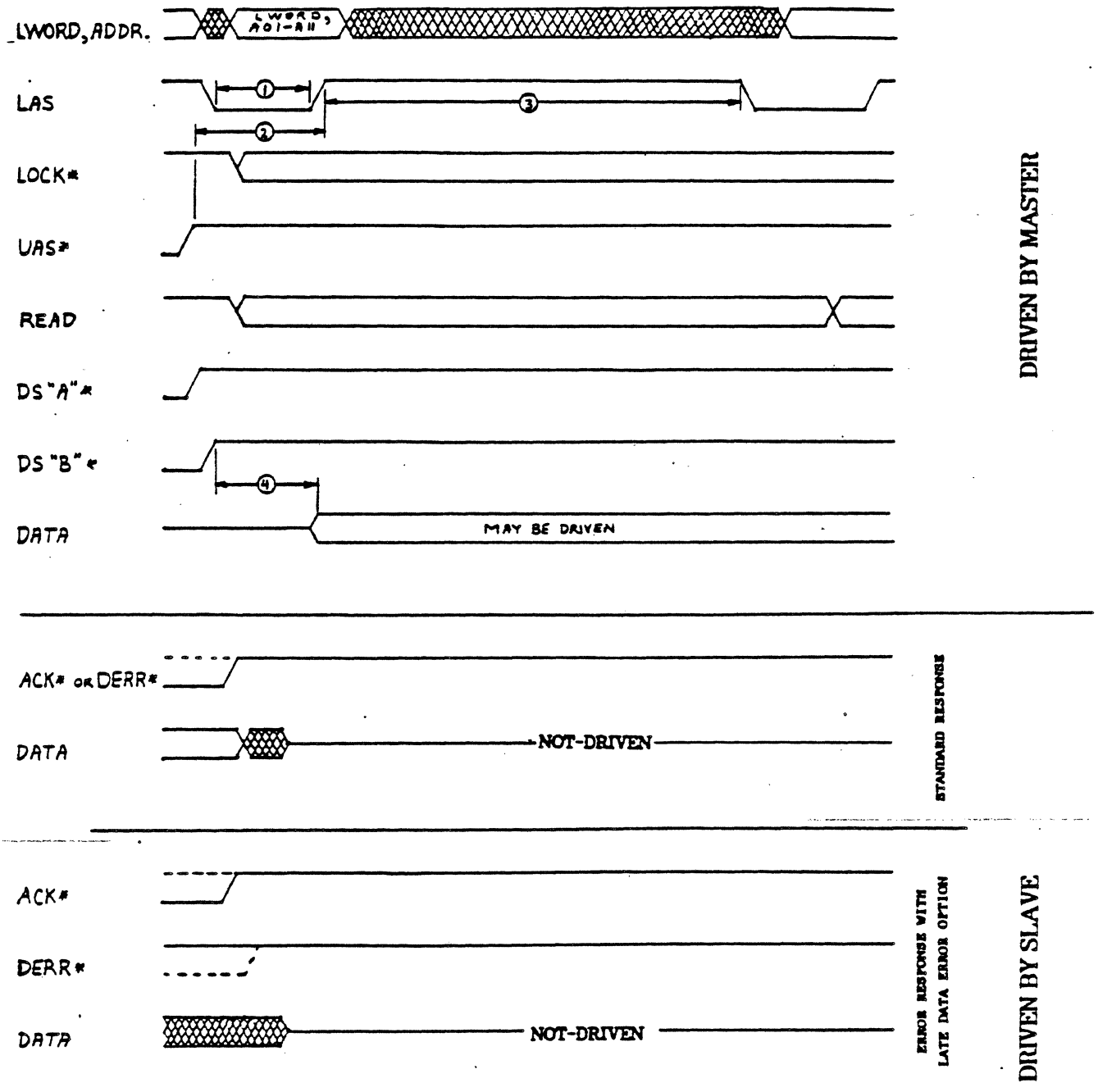
PARAMETER NUMBER	(NOTE A) MIN	MAX	NOTES
1	15		B
2	20		B
3	15		B
4	30		B

NOTES:

A. All times are given in nanoseconds unless otherwise noted.

B. The SLAVE is guaranteed this timing between two of its incoming signal transitions.

FIGURE 4-14
Withdrawn Cycle: SLAVE timing diagram



CHAPTER 5
VMXbus ARBITRATION BUS

5.1 SIGNAL LINES

The VMXbus contains two arbitration lines which are used to transfer control of the VMX-DTB between the PRIMARY MASTER and the SECONDARY MASTER:

SMRQ*	SECONDARY MASTER bus request
SMACK*	SECONDARY MASTER acknowledge

These lines are collectively called the "arbitration bus". The SECONDARY MASTER drives SMRQ* low to when it needs to read or write data to VMXbus memory. The PRIMARY MASTER may drive SMACK* low in response to the low level on SMRQ* to permit the SECONDARY MASTER to use the VMX-DTB.

The PRIMARY MASTER drives SMACK* high whenever the system is reset by the VMEbus SYSRESET* line and maintains it high at least until SMRQ* is driven low by the SECONDARY MASTER. The SECONDARY MASTER drives SMRQ* high when the system is reset and maintains it high at least until SYSRESET* goes high again. NOTE:

> When the PRIMARY MASTER is in control of the VMX-DTB there are no timing constaints on how long the PRIMARY MASTER may take to grant the VMX-DTB. Accordingly, it is permissible to design a PRIMARY MASTER which NEVER grants the VMX-DTB. Clearly if such a PRIMARY MASTER were plugged into the same bus with a SECONDARY MASTER, the SECONDARY MASTER would never be able to move any data over the bus. For this reason any board vendor selling such a PRIMARY MASTER must clearly indicate its limitations on the product's data sheets.

5.2 EXAMPLES OF ARBITRATION OPERATION

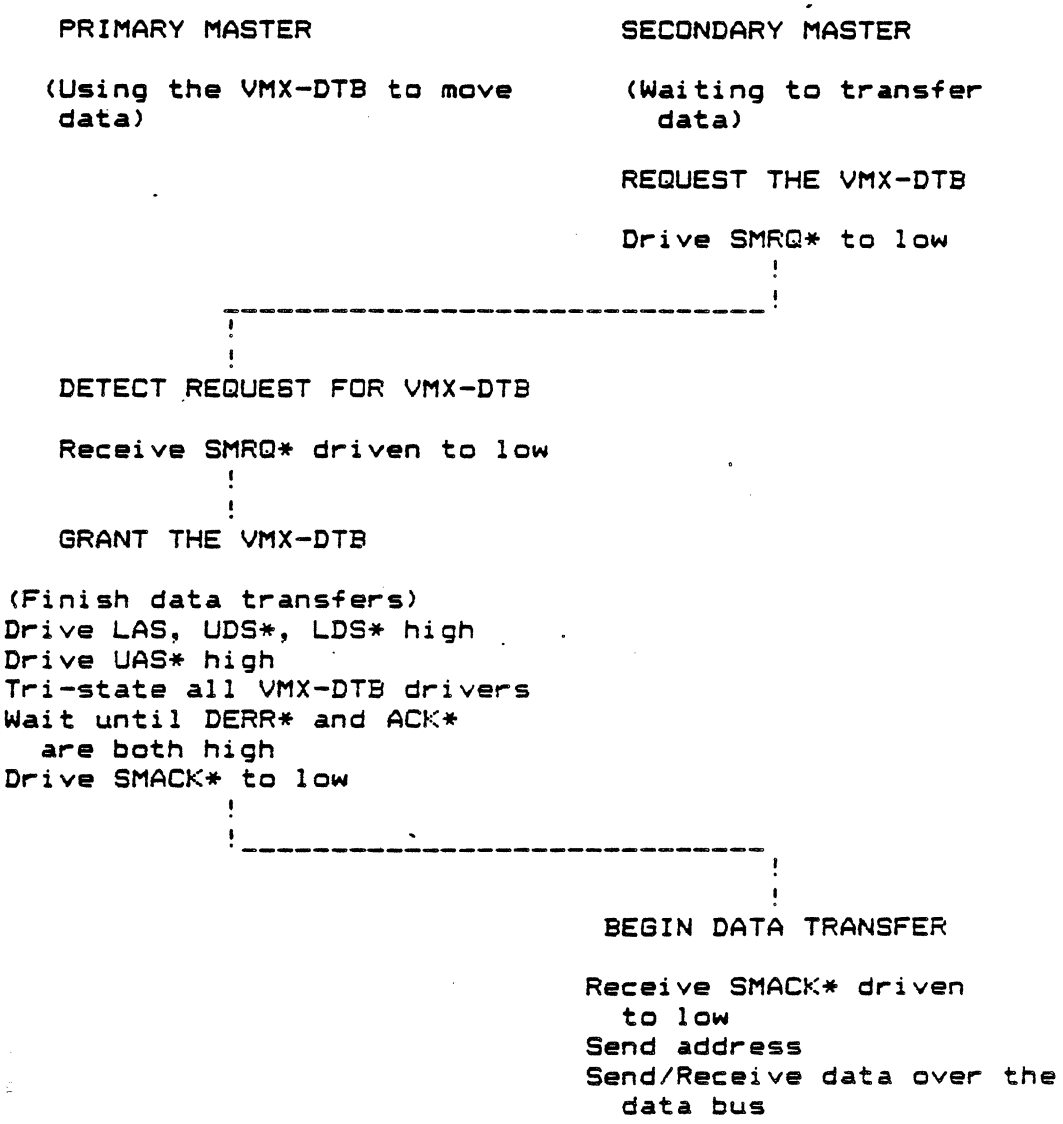
Since only two MASTERS are allowed on each VMXbus, (a PRIMARY MASTER and a SECONDARY MASTER) there are only two basic arbitration sequences. The first is when the PRIMARY MASTER grants control of the VMX-DTB to the SECONDARY MASTER. The second is when the SECONDARY MASTER gives control of the VMX-DTB back to the PRIMARY MASTER. The following two sections illustrate these sequences.

5.2.1 PRIMARY MASTER grants VMX-DTB To SECONDARY MASTER.

Figure 5-1 shows the flow diagram for the case where the SECONDARY MASTER is granted the bus. When the sequence begins the PRIMARY MASTER is using the VMX-DTB. When the SECONDARY MASTER drives SMRQ* to low, and when the PRIMARY MASTER has finished using the VMX-DTB, it drives SMACK* low, granting the VMX-DTB to the SECONDARY MASTER.

As long as the SECONDARY MASTER holds SMRQ* low it may continue to use the VMX-DTB without interuption. This allows it to transfer data to or from the memory at the maximum data rate, since it need not check before each transfer to see whether it still has control of the VMX-DTB.

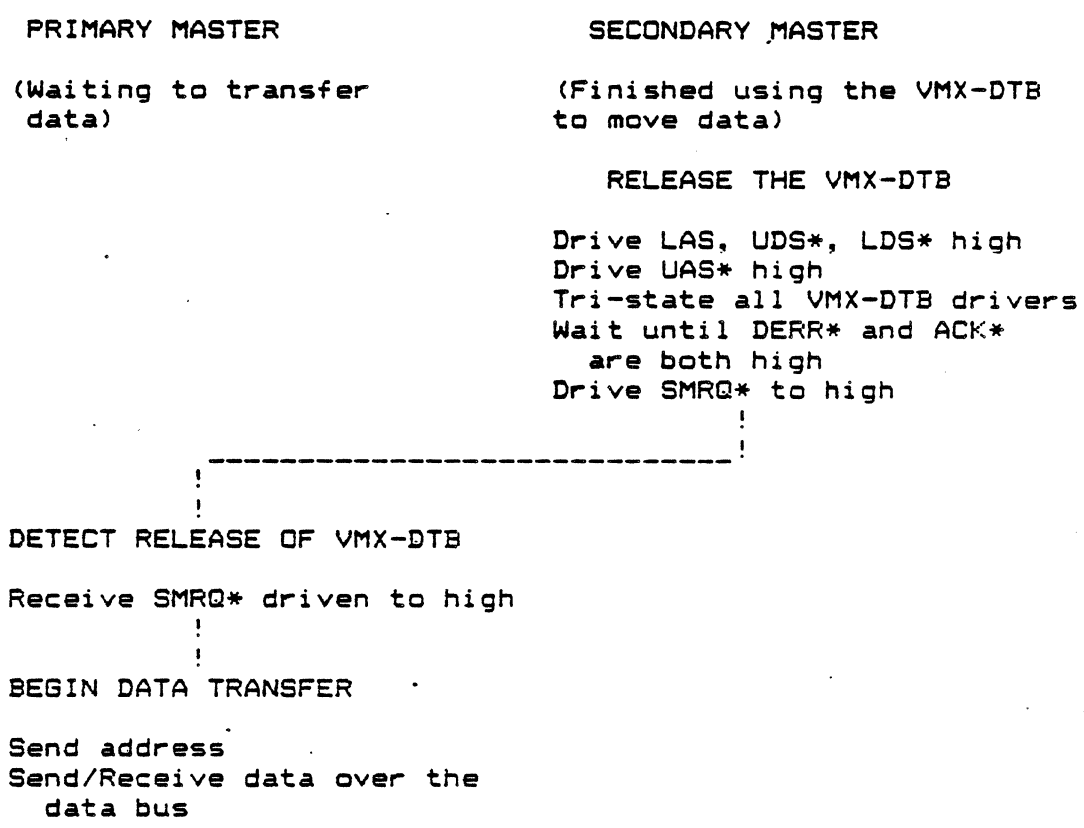
FIGURE 5-1


PRIMARY MASTER
(Using the VMX-DTB to move data)
SECONDARY MASTER
(Waiting to transfer data)
REQUEST THE VMX-DTB
Drive SMRQ* to low
DETECT REQUEST FOR VMX-DTB
Receive SMRQ* driven to low
GRANT THE VMX-DTB
(Finish data transfers)
Drive LAS, UDS*, LDS* high
Drive UAS* high
Tri-state all VMX-DTB drivers
Wait until DERR* and ACK* are both high
Drive SMACK* to low
BEGIN DATA TRANSFER
Receive SMACK* driven to low
Send address
Send/Receive data over the data bus

5.2.2 PRIMARY MASTER receives VMX-DTB From SECONDARY MASTER

Figure 5-2 shows the flow diagram for the case where the SECONDARY MASTER relinquishes control of the data transfer lines. When the sequence begins, the SECONDARY MASTER is transferring data between itself and VMXbus memory. When the transfer is complete, and it has tri-stated all of the VMX-DTB drivers, it drives SMRQ* high, signaling to the PRIMARY MASTER that it has finished using the VMX-DTB. The PRIMARY MASTER then drives SMACK* high, signalling to the SECONDARY MASTER that its high level on the SMRQ* line has been detected and that it may request the bus again when needed.

FIGURE 5-2


PRIMARY MASTER
SECONDARY MASTER
(Waiting to transfer data)
(Finished using the VMX-DTB to move data)
RELEASE THE VMX-DTB
Drive LAS, UDS*, LDS* high
Drive UAS* high
Tri-state all VMX-DTB drivers
Wait until DERR* and ACK* are both high
Drive SMRQ* to high
DETECT RELEASE OF VMX-DTB
Receive SMRQ* driven to high
BEGIN DATA TRANSFER
Send address
Send/Receive data over the data bus

5.3 ARBITRATION TIMING DIAGRAMS

The following two sections provide two pairs of timing diagrams that describe the detailed timing that the PRIMARY MASTER and SECONDARY MASTER must meet when passing control of the VMX-DTB between each other. These times guarantee that the two MASTERS never drive the same VMXbus lines simultaneously and that they don't drive the data lines simultaneously with the SLAVES. The timing diagrams describe the transfer the mastership as follows:

Fig. Nos.	Figure Title
5-3, 5-4	PRIMARY MASTER grants VMX-DTB
5-5, 5-6	PRIMARY MASTER receives VMX-DTB

5.3.1 PRIMARY MASTER grants VMX-DTB

TABLE 5-3
PRIMARY MASTER grants VMX-DTB: MASTER timing values

PARAMETER NUMBER	(NOTE A) MIN	MAX	NOTES
1	0		B
2	0		B
3	0		B
4	0		B
5	0		B
6	0		B
7	0		B
8	0		B
9	0		B
10	0		B
11	0		B
12	0		B
13	0		C
14	0		D
15	20		D, E
17	0		D
18	0		D
19	0		D
20	0		D
21	35		D, E
22	20		D, E

NOTES:

A. All times are given in nanoseconds unless otherwise noted.

B. The MASTER must guarantee this timing between two of its outgoing signal transitions.

C. The MASTER must wait for the incoming signal edge from the SLAVE before changing the level on its outgoing signal.

D. This is a guarantee that the SECONDARY MASTER will not start driving the VMX-DTB until the PRIMARY MASTER has stopped driving it.

E. This guarantees that, when mastership is transferred, the SLAVES are guaranteed a minimum time between cycles.

FIGURE 5-3
PRIMARY MASTER grants VMX-DTB: MASTER timing diagram

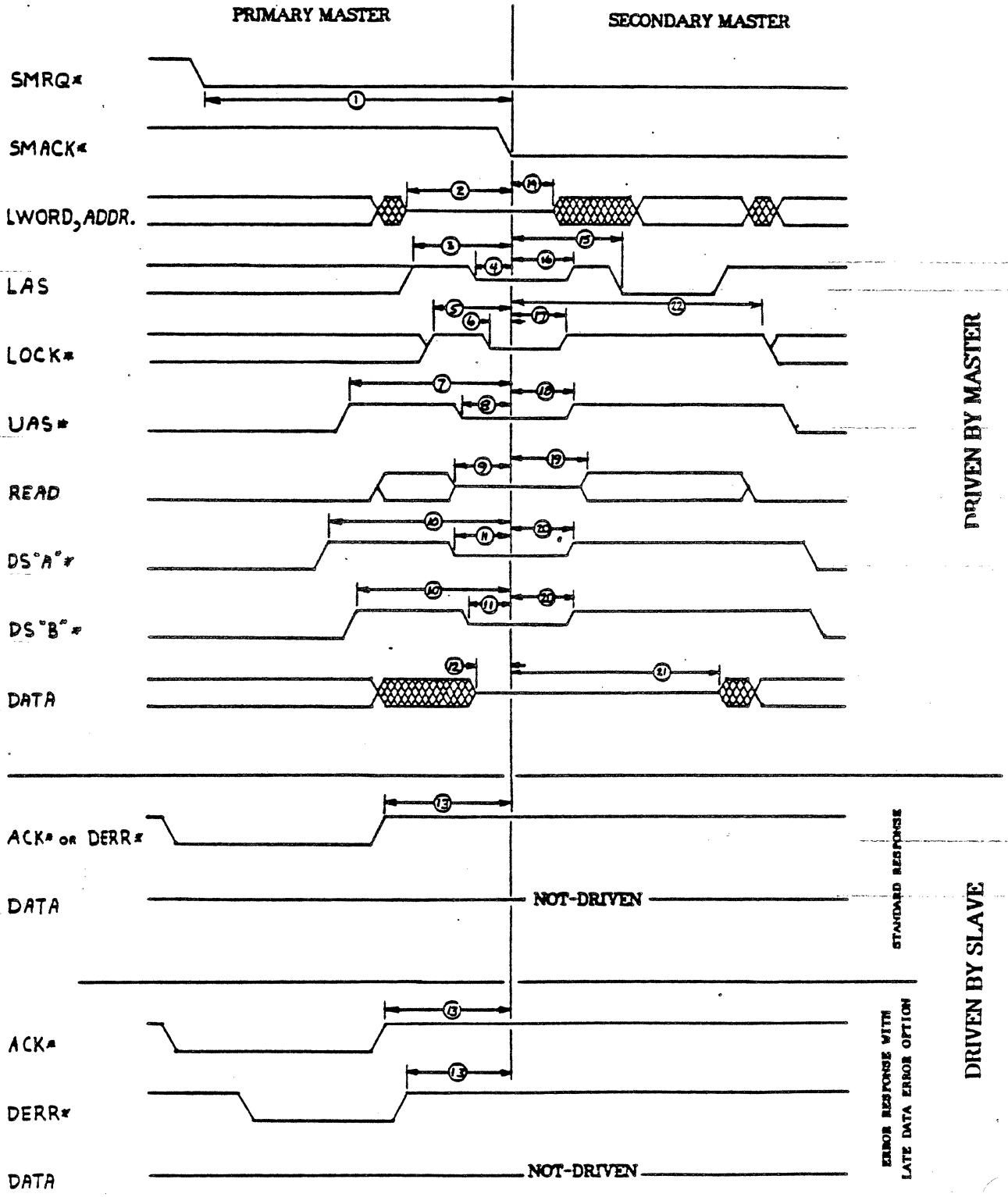


TABLE 5-4
PRIMARY MASTER grants VMX-DTB: SLAVE timing values

PARAMETER NUMBER	(NOTE A) MIN	MAX	NOTES
1	0		B
2	0		B
3	0		B
4	0		B
5	0		B
6	0		B
7	0		B
8	0		B
9	0		B
10	0		B
11	0		B
12	0		B
13	0		B
14	0		D
15	15		D, E
17	0		D
18	0		D
19	0		D
20	0		D
21	35		D, E
22	15		D, E

NOTES:

A. All times are given in nanoseconds unless otherwise noted.

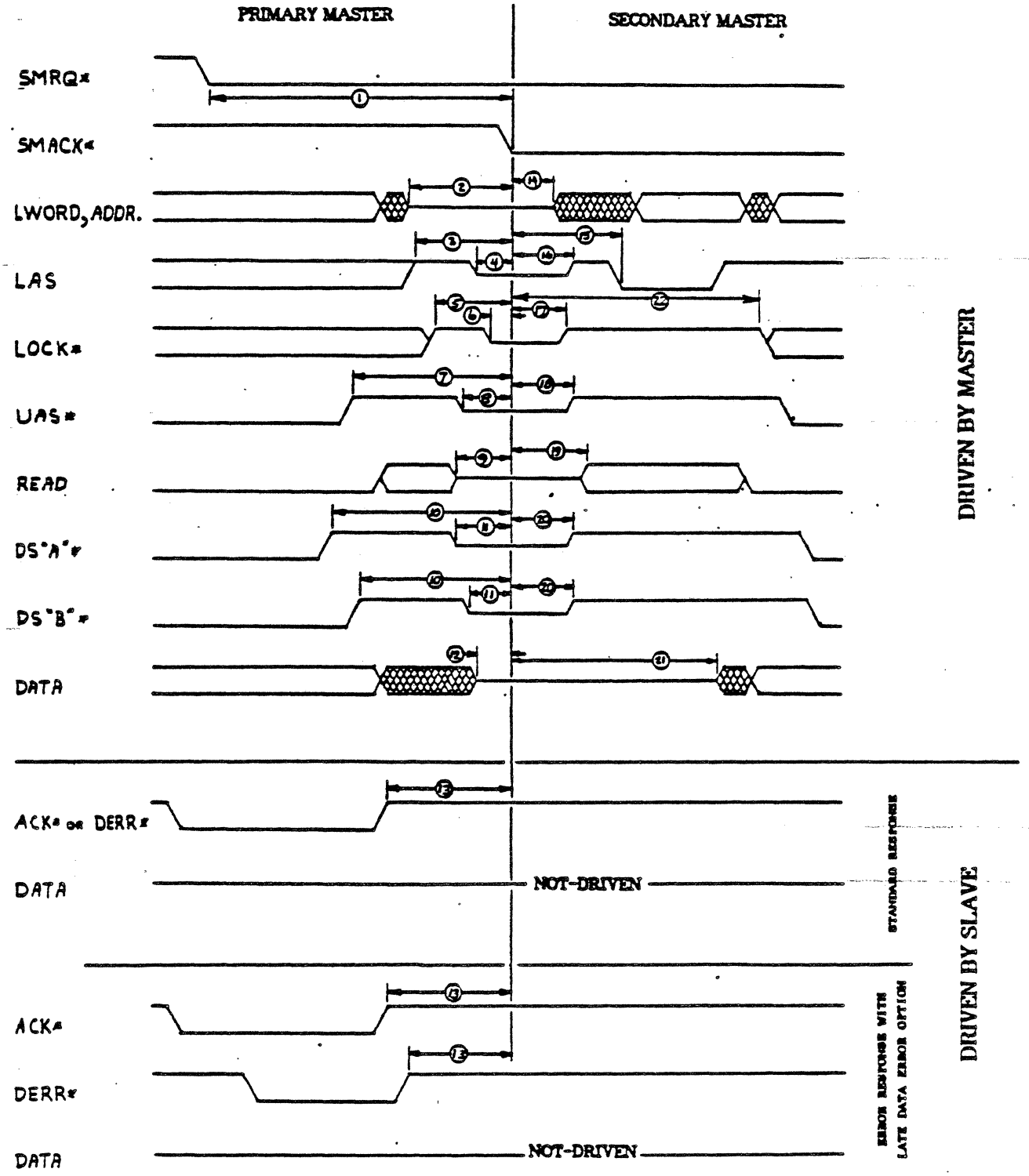
B. The SLAVE is guaranteed this timing between two of its incoming signal transitions.

C. This is a guarantee that the MASTER will not change the incoming signal until the SLAVE changes its outgoing signal.

D. This is a guarantee that the SECONDARY MASTER will not start driving the VMX-DTB until the PRIMARY MASTER has stopped driving it.

E. This guarantees that, when mastership is transferred, the SLAVES are guaranteed a minimum time between cycles.

FIGURE 5-4
PRIMARY MASTER grants VMX-DTB: SLAVE timing diagram



5.3.2 PRIMARY MASTER Receives VMX-DTB

TABLE 5-5
PRIMARY MASTER Receives VMX-DTB: MASTER timing values

PARAMETER NUMBER	(NOTE A) MIN	MAX	NOTES
1	0		D
2	0		D
3	0		D
4	0		D
5	0		D
6	0		D
7	0		D
8	0		D
9	0		D
10	0		D
11	0		D
12	0		D
13	20		C, E
14	0		B, E
15	0		B, E
16	0		B, E
18	0		B, E
19	0		B, E
20	0		B, E
21	15		B, E
22	0		B, E

NOTES:

A. All times are given in nanoseconds.

B. The PRIMARY MASTER must guarantee this timing between two of its outgoing signal transitions.

C. The PRIMARY MASTER must wait for the incoming signal edge from the SECONDARY MASTER before changing the level on its outgoing signal.

D. This is a guarantee that the SECONDARY MASTER will not start driving the VMX-DTB until the PRIMARY MASTER has stopped driving it.

E. This guarantees that, when mastership is transferred, the SLAVES are guaranteed a minimum time between cycles.

FIGURE 5-5
PRIMARY MASTER receives VMX-DTB: MASTER timing diagram

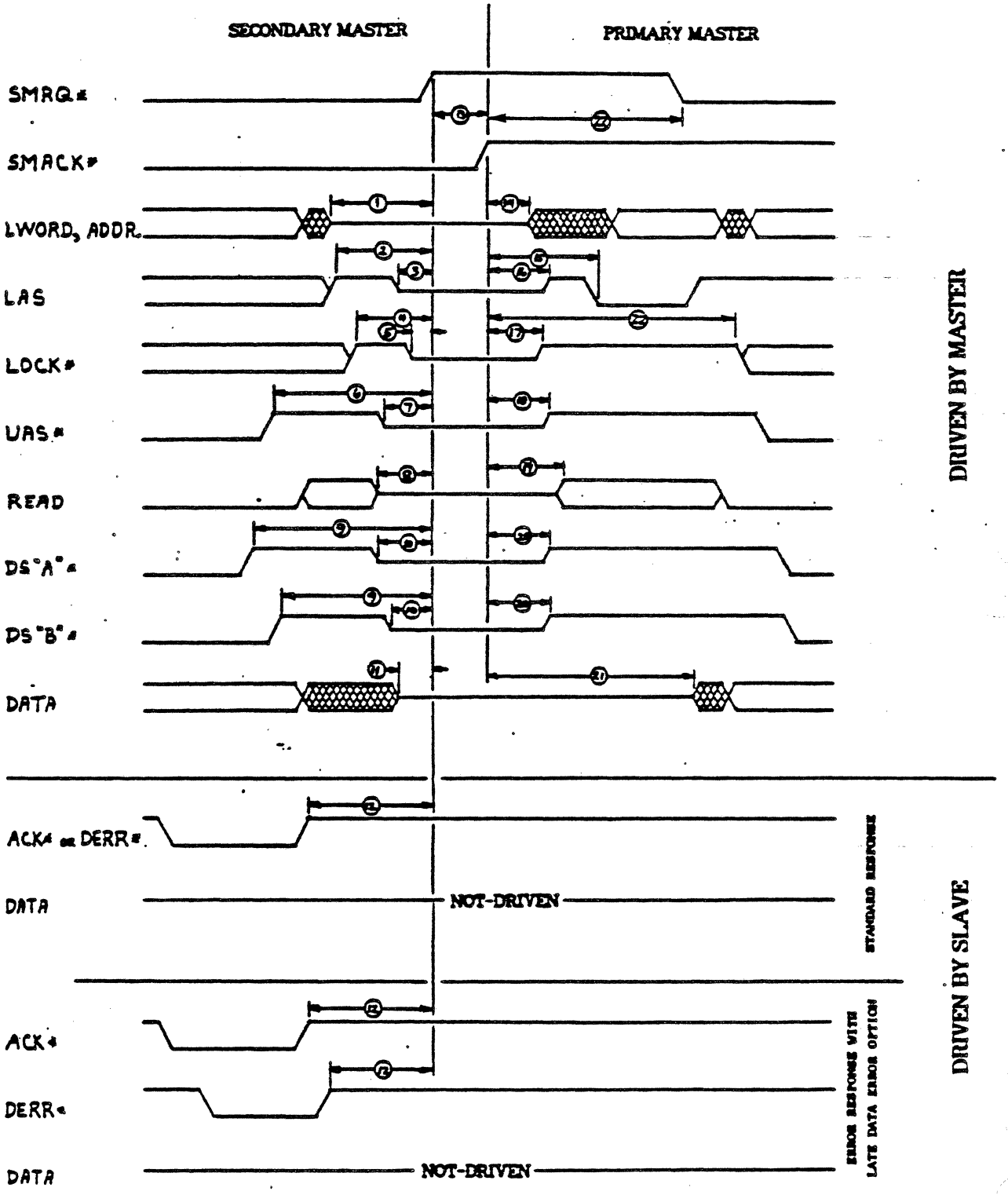


TABLE 5-6
PRIMARY MASTER Receives VMX-DTB: SLAVE timing values

PARAMETER NUMBER	(NOTE A) MIN	MAX	NOTES
1	0		D
2	0		D
3	0		D
4	0		D
5	0		D
6	0		D
7	0		D
8	0		D
9	0		D
10	0		D
11	0		D
12	0		D
13	0		C, E
14	0		B, E
15	0		B, E
16	0		B, E
18	0		B, E
19	0		B, E
20	0		B, E
21	15		B, E
22	15		B, E

NOTES:

A. All times are given in nanoseconds.

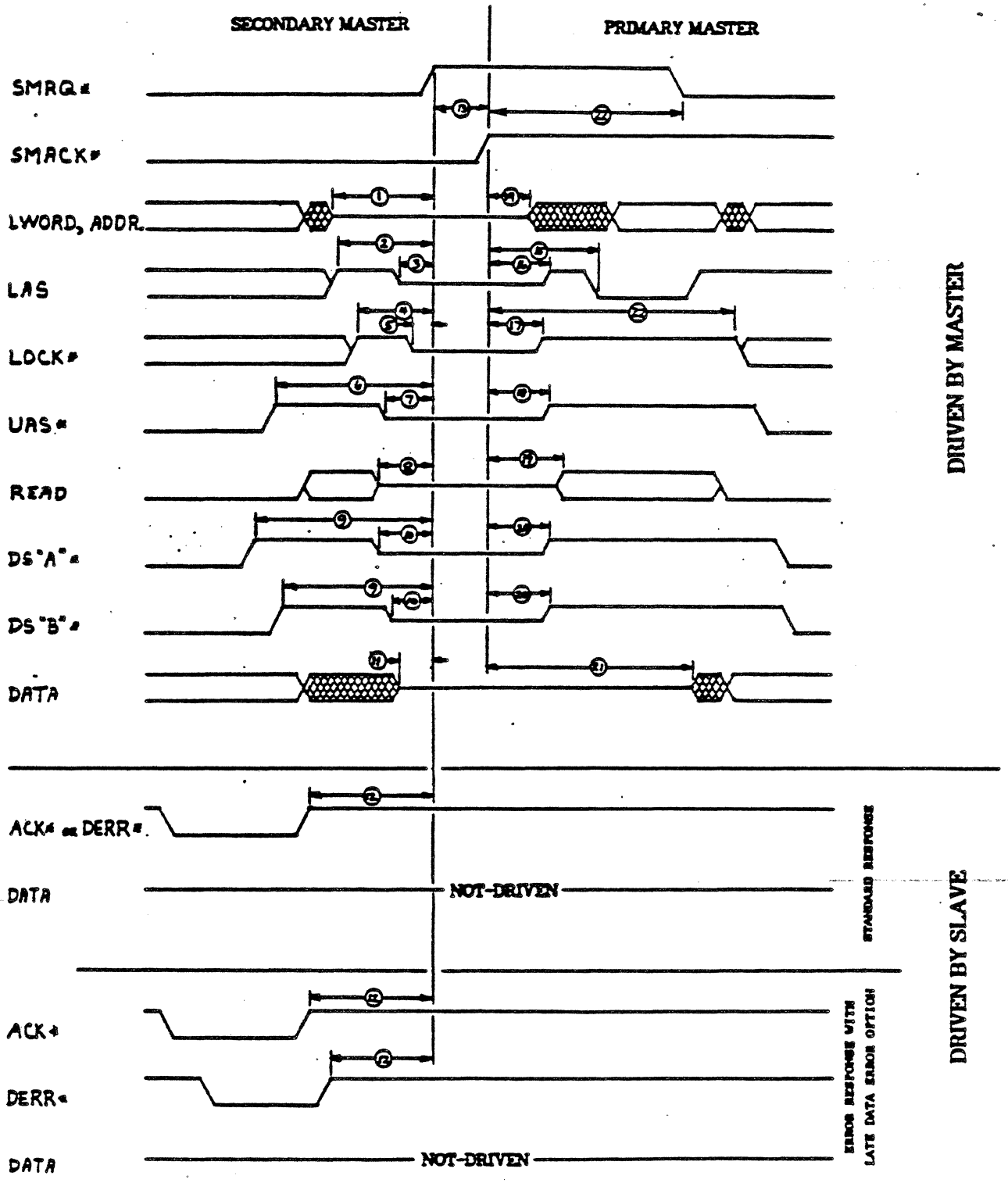
B. The SLAVE is guaranteed this timing between two of its incoming signal transitions.

C. The PRIMARY MASTER must wait for the incoming signal edge from the SECONDARY MASTER before changing the level on its outgoing signal.

D. This is a guarantee that the SECONDARY MASTER will not start driving the VMX-DTB until the PRIMARY MASTER has stopped driving it.

E. This guarantees that, when mastership is transferred, the SLAVES are guaranteed a minimum time between cycles.

FIGURE 5-6
PRIMARY MASTER receives VMX-DTB: SLAVE timing diagram



CHAPTER 6
ELECTRICAL SPECIFICATIONS

6.1 INTRODUCTION

All interface logic on the VMXbus boards must be designed in accordance with the rules laid out in this chapter to insure proper timing and to minimize noise and crosstalk. VMXbus signals are normally driven with TTL family drivers, although any technology which complies with this specification may be used. Recommendations and examples are included in this chapter to aid the designer in obtaining optimum system performance.

6.2 POWER DISTRIBUTION

Power and power-return distribution for the VMXbus is done through the VMEbus P1 and P2 backplanes. (This leaves all of the P2 ribbon cable lines available for signals and signal grounds.) The pins of the P1 and P2 connectors, which are used for power distribution, the voltage tolerances, the maximum ripple, and the current carrying capacity of each power pin are the same as those described in the VMEbus specification.

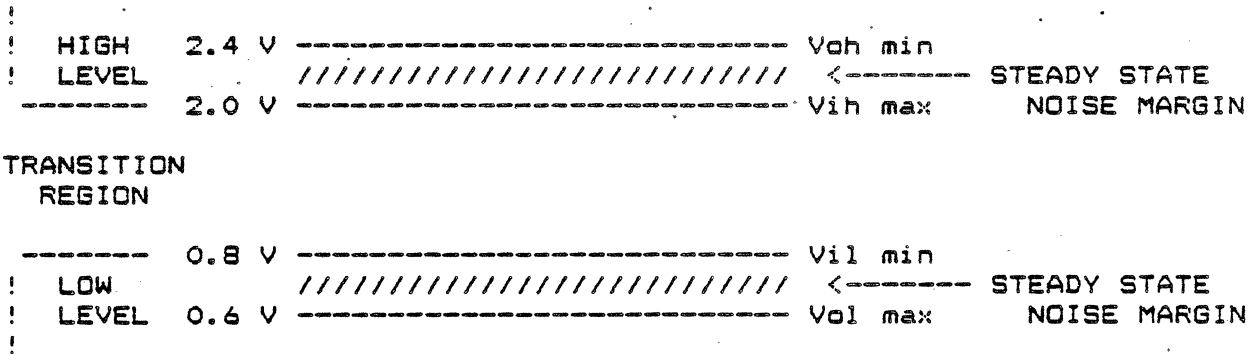
6.3 ELECTRICAL SIGNAL CHARACTERISTICS

Other than the power supply lines, all VMXbus signals are limited to positive levels between 0.0 and 5.0 volts. The signal levels are:

		Minimum	Maximum
Driver	low output level		0.6 V
Receiver	low input level		0.8 V
Driver	high output level	2.4 V	
Receiver	high input level	2.0 V	

Figure 6-3 gives a simple graphical representation of these levels.

FIGURE 6-3



6.4 DRIVER SPECIFICATIONS

There are three types of signal line drivers used on the VMXbus: totem-pole drivers, tri-state drivers, and open collector drivers.

Totem-pole drivers sink current when driving a signal line to its low level and source current when driving it to its high level. They are used to drive signal lines which are always driven by the same VMXbus board. SMRQ* and SMACK* are the two VMXbus lines driven by totem-pole drivers.

VMXbus tri-state drivers differ from totem-pole drivers in two respects: they can sink and source more current than the totem-pole drivers and they can go to a high impedance state (driver turned off) in addition to the low and high logic states. Tri-state drivers are used to drive lines that can be driven by several different boards on the VMXbus. (Only one of these drivers is ever driving a signal line at any one time.) The address bus, the data bus and most of the other signal lines on the VMXbus are driven by tri-state drivers.

Open collector drivers sink current when driving a signal line to its low level but sources no current to drive the signal line high. Pull-up resistors on the PRIMARY MASTER ensure that the signal line voltage rises to a high level whenever the open collector line is not driving it to a low level. ACK* and DERR* are the two VMXbus lines driven by open collector drivers.

The following sections give the specifications for each type of driver.

6.4.1 Totem-Pole Drivers

VMXbus totem-pole drivers shall provide, as a minimum, the following characteristics:

		Minimum	Maximum	Comments
DRIVER TURNED ON				
Low state sink current:	IOL	8 mA		
High state source current:	IOH	400 uA		
Low state output voltage:	VOL		0.5 V	sinking 8 mA
High state output voltage:	VOH	2.4 V		sourcing 400 uA
DRIVER SHORT CIRCUITED				
Low state source current:	IOS	20 mA	225 mA	0.0 V
DRIVER CAPACITANCE				
Capacitance at output pin:	COUT		18pf	

AVAILABLE CIRCUITS

These specifications allow the use of standard LS type drivers such as 74LS00, 74LS04, etc.

6.4.2 Tri-state Drivers

VMXbus tri-state drivers shall provide, as a minimum, the following characteristics:

		Minimum	Maximum	Comments
DRIVER TURNED ON				
Low state sink current:	IOL	20 mA		
High state source current:	IOH	1 mA		
Low state output voltage:	VOL		0.5 V	sinking 20 mA
High state output voltage:	VOH	2.4 V		sourcing 1 mA
DRIVER TURNED OFF				
Low state source current:	IOZL		50 uA	0.5 V
High state sink current:	IOZH		50 uA	2.4 V
DRIVER SHORT CIRCUITED				
Low state source current:	IOS	60 mA	225 mA	0.0 V
DRIVER CAPACITANCE				
Capacitance at output pin:	COUT		18pf	

AVAILABLE CIRCUITS

These specifications allow the use of 74F240, or 74F257 drivers.

6.4.3 Transceivers

VMXbus Transceivers shall provide, as a minimum, the following characteristics:

		Minimum	Maximum	Comments
DRIVER TURNED ON (low)				
Low state sink current:	IOL	20 mA		
High state source current:	IOH	1 mA		
Low state output voltage:	VOL		0.5 V	sinking 20 mA
High state output voltage:	VOH	2.4 V		sourcing 1 mA
TRASCEIVER SHORT CIRCUITED				
Low state source current	IOS	60 mA	225 mA	0.0 V
LOADING (DRIVER TURNED OFF)				
Low level source current	IIL		0.7 mA	0.5 V
High level sink current	IIH		70 uA	2.4 V
RECEIVER THRESHOLDS				
Low level threshold	VIL	0.8V		
High level threshold	VIH		2.0 V	
TRASCEIVER CAPACITANCE				
Capacitance at output pin	COUT		25pf	

AVAILABLE CIRCUITS

These specifications allow the use of a 74ALS245 transceiver.

6.4.4 Open Collector Drivers

VMXbus open collector drivers shall provide, as a minimum, the following characteristics:

		Minimum	Maximum	Comments
DRIVER TURNED ON (low)				
Low state sink current:	IOL	24 mA		
Low state output voltage:	VOL		0.4 V	sinking 24 mA
DRIVER TURNED OFF (high)				
High state source current	IOH		250uA	5.0 V
DRIVER CAPACITANCE				
Capacitance at output pin	COUT		20pf	

AVAILABLE CIRCUITS

Any open-collector driver in accordance with the electrical characteristics specified above may be used. As an example, a 74LS38 may be used.

6.5 RECEIVER SPECIFICATIONS

All VMXbus receivers should have a diode clamp which prevents negative voltage excursions from going below -1.5 V. Standard 74LS devices provide this clamping.

VMXbus receivers shall provide, as a minimum, the following characteristics:

		Minimum	Maximum	Comments
RECEIVER LOADING				
Low level source current	IIL		0.6 mA	0.5 V
High level sink current	IIH		20 uA	2.7 V
RECEIVER THRESHOLDS				
Low level threshold	VIL	0.8V		
High level threshold	VIH		2.0 V	
RECEIVER CAPACITANCE				
Capacitance at input pin	CIN		7pf	

AVAILABLE CIRCUITS

These specifications allow the use of most 74LS, 74F, and 74S type receivers with PNP inputs.

6.6 SIGNAL LINE TERMINATIONS

The following ten signal lines of the VMXbus are terminated on the PRIMARY MASTER board with 330 ohm pull-up resistors. Other boards must not pull these lines up.)

READ Tri-state
LOCK* Tri-state
LAS Tri-state
UAS* Tri-state
UDS* Tri-state
LDS* Tri-state
DERR* Open-collector
ACK* Open-Collector

These pull-up resistors pull their respective lines to a high level whenever a tri-state or open-collector driver turns off.

NOTE

It is recommended that decoupling capacitors be provided between the +5 volt source on the pull-up resistor and ground. Capacitors between 0.01 and 0.1 uF are appropriate and should be installed as close as possible to the Vcc pin of the pull-up resistor package and the VMXbus board's ground grid.

6.7 ISOLATION LINES

VMXbus includes ten signal lines called STatic Isolation Lines (STIL0-STIL9). The purpose of these lines is to prevent crosstalk between adjacent signal lines whose transitions take place at different times. All VMXbus boards must provide a 0.01 plus a 0.1 uF by-pass capacitor between each of these lines and its on-board ground grid. The length of the trace connecting this capacitor to the connector plus the length of the trace connnecting the other side of the capacitor to the ground grid must not exceed 2 inches.

6.8 BOARD SIGNAL LOADING

The following rules must be followed when designing VMXbus boards:

Circuit traces from the DIN connector pins to the on-board receivers and drivers shall not have a length of greater than 3 inches. (If the trace branches, the length of each branch is added to get a total length.)

No more than two drivers and two receivers (or two transceivers) per board may be connected to any VMXbus signal line.

6.9 BACKPLANE RIBBON CABLE SPECIFICATION

The ribbon cable that is used to bus the P2 connectors of a VMEbus card rack may have from two to six 64 pin DIN 41612 connectors. The nominal spacing of these connectors is 0.8 inches. In no case shall the total cable length exceed 5.0 inches.

The ribbon cable shall meet the following specifications:

PARAMETER	SPECIFICATION
Conductor Center-to-center Spacing	.050 inch
Conductor type	Stranded copper
Conductor size	28 AWG
Stranding	7 X 36
Number of Conductors	64
Impedence	>= 100 ohm
Capacitance	<= 20 pf/ft
Inductance	<= .25 uH/ft
Resistance	<= 70 ohms/1000 ft (at 20 degrees C)
Propagation Delay	<= 1.5 ns/ft

Recommended Cables: 3M 3365/64
Hitachi EG-2864

CHAPTER 7

MECHANICAL SPECIFICATIONS

7.1 DEFINITION OF TERMS

Information in this chapter is provided to assure that VMXbus boards will be mechanically compatible with the VMEbus systems into which they are installed. As in the case of the VMEbus, the following terminology is used:

Backplane - A board into which 96-pin connectors are installed. The backplane of primary interest to VMXbus users is the VMEbus "P2 backplane". This is the lower backplane in a double-high VMEbus card rack which buses only the center row pins of each "P2 connector".

Ribbon cable - A 64 wire cable which has installed on it two to six 64-pin connectors at 0.8" intervals. This cable is installed on the back of the "P2 backplane" to bus the two outside rows of pins on adjacent P2 connnectors together.

VMXbus board - A PC board which is plugged into the backplane and communicates with other VMXbus boards installed in the same backplane.

The "front" of a backplane is the side from which the VMXbus boards are inserted into the connectors.

The "rear edge" of a VMXbus board is the edge which provides the on-board connector for mating with the connector on the backplane.

7.2 VMXbus BACKPLANE DESCRIPTION

The VMXbus has been designed for use with a double high VMEbus card. This double high rack has two backplanes: an upper one which buses all three rows of pins on the P1 connectors and a lower one which buses only the center row of pins on the P2 connectors. (This VMEbus "P2 backplane" does not bus the two outer rows of the P2 connectors.) Since these two outer rows aren't bused by the P2 backplane they may be used as I/O pins. When a user wants to bus these two rows of pins for a secondary bus, such as VMXbus, a ribbon cable is installed to form a "private bus".

VMXbus is defined to provide just such a private bus. One or more 64 wire ribbon cables can be installed on the P2 connectors to "bus" the two outer rows of pins. Each ribbon cable may have from two to six connectors on it, and several of these cables can be used in a SINGLE CARD RACK to provide

several VMXbuses.

The ribbon cable provides all of the conductors needed to send 24 bit addresses from the MASTERS to the SLAVES and transfer up to 32 bits of data at a time. This ribbon cable connects a maximum of six VMXbus compatible boards which are installed at 0.8 inch intervals. The length of the cable used to connect these boards must not exceed 5.0 inches.

All VMXbus configurations assume the existance of a P2 backplane. Power for the VMXbus boards is provided by this P2 backplane through the center row (row "b") of its VMXbus connector. (Double-high VMXbus boards will typically also draw power through their P1 connector.)

7.3 VMXbus EUROCARD DESCRIPTION

VMXbus boards may be either single or double high, just like VMEbus boards. The dimensions for these boards must conform to those given for the single and double high boards in the VMEbus specification. All of the dimensions given there including maximum board warpage, maximum component height, etc apply to VMXbus boards also.

NOTE:

> All PRIMARY MASTER and SECONDARY MASTERS are double-high boards, since they monitor the level of SYSRESET* on the VMEbus.

7.4 VMXbus PIN ASSIGNMENT

VMXbus uses the two outside rows (rows a and c) of the P2 connector for all signal transmissions. VMXbus boards may draw power from the center row (row b) of P2 as shown. (These +5V power and GND pins are the same as the ones specified in the VMEbus specification.) In addition to these pins, a double-high VMXbus board may also draw power from its P1 connector. (See the VMEbus specification for the +5V and GND pins locations on P1.)

	a	b	c
1	DB00	+5V	DB01
2	DB02	GND	DB03
3	DB04	VMEbus signal	DB05
4	DB06	VMEbus signal	DB07
5	DB08	VMEbus signal	DB09
6	DB10	VMEbus signal	DB11
7	DB12	VMEbus signal	DB13
8	DB14	VMEbus signal	DB15
9	DB16	VMEbus signal	DB17
10	DB18	VMEbus signal	DB19
11	DB20	VMEbus signal	DB21
12	DB22	GND	DB23
13	DB24	+5V	DB25
14	DB26	VMEbus signal	DB27
15	DB28	VMEbus signal	DB29
16	DB30	VMEbus signal	DB31
17	~~WRITE*~~ Ref 1	VMEbus signal	STIL0
18	LOCK*	VMEbus signal	STIL1
19	LWORD/A12	VMEbus signal	STIL2
20	A02/A14	VMEbus signal	A01/A13
21	A04/A16	VMEbus signal	A03/A15
22	A06/A18	GND	A05/A17
23	A08/A20	VMEbus signal	A07/A19
24	A10/A22	VMEbus signal	A09/A21
25	STIL3	VMEbus signal	A11/A23
26	STIL4	VMEbus signal	LAS*
27	STIL5	VMEbus signal	UAS*
28	STIL6	VMEbus signal	UDS*
29	STIL7	VMEbus signal	LDS*
30	STIL8	VMEbus signal	DERR*
31	STIL9	GND	ACK*
32	SMRQ*	+5V	SMACK*