ANSI/IEEE
Std 696-1983

An American National Standard

IEEE Standard 696 Interface Devices

Sponsor

Standards Committee of the
IEEE Computer Society

Approved June l0,1982

IEEE Standards Board

Approved September 8,1983

American National Standards Institute

The Institute of Electrical and Electronics Engineers, Inc
345 East 47th Street, New York, NY 10017, USA

Foreword

(This Foreword is not a part of IEEE Std 696-1983, IEEE Standard 696 Interface Devices.)

This standard represents over four and a half years of effort by numerous individuals to make the IEEE Std 696-1983 a truly workable and technically excellent standard. The original version of the standard was published in the July 1979 issue of COMPUTER magazine, and this standard represents a significant fine tuning of the original.

In the main, changes to the original consist of editorial changes to clear up ambiguities. The major technical change involves how we *think* about 8- and 16-bit bus transfers, rather than how the mechanism works electrically. To go along with this change in thinking, some new nomenclature was agreed upon.

To be more specific, the original standard described 16-bit transfers in terms of a *high* and low (or most and least significant) byte. This presented a problem because CPU chip manufacturers chose to order the bytes different from each other, that is, some transfer the high byte on the lower half of the data bus, and some do just the opposite. It was clear that the working group had to settle the issue of where to place each byte. A fundamental problem was that the group was split fairly evenly on this issue. The solution that evolved is both clever and unique in that it made everybody happy.

It was decided that the standard should not dictate a significance of the bytes at all. Instead, the standard concerns itself with making sure that bytes read or written in an 8-bit mode is read or written consistently in a 16-bit mode, and vice versa. Originally, data bits were called DATA16-DATA0 in a 16-bit mode, which inherently designates significance. Now the 16 data bits are thought of only as two bytes: an odd byte and an even byte-now called OD7-0 and ED7-0 (OD for Odd Data and ED for Even Data). The terms *high* and low have been replaced by *odd* and *even,* respectively.

Basically, the rule is as follows: Byte data that is written or read with AO=1 appears on the OD7-0 lines during a 16-bit transfer. Byte data that is written or read with AO=O appears on the ED7-0 lines during a 16-bit transfer. The nomenclature of OD and ED (odd data and even data) make it easy to remember the rule when looking at a schematic. All one has to do is think about the fact that any address with AO=1 is odd, and any address with AO=O is even.

There was one other change in the nomenclature, and that concerns the renaming of the term DMA (Direct Memory Access) to TMA (Temporary Master Access). This is the reason for the change. When a temporary master is accessing the bus, it may execute any type of cycle-memory or I/O. The term DMA implies that a memory access (as opposed to I/O) is the only type of cycle allowed. Since this implication is incorrect, the term TMA was substituted as it does not imply any particular type of cycle, and is also an accurate descriptor of the type of operation that is occurring.

As mentioned earlier, the only other changes to the standard were those required to clear up ambiguities that existed, and to specify parameters that had been implied by the standard, but not specifically stated.

The bulk of the standard remains unchanged, and has proven itself to be workable, practical, and quite viable in the real world, as well as on paper. As it stands, it is the culmination of the efforts of some of the brightest minds in the computing industry. It has turned out remarkably well, in view of the fact that it was started with a very arbitrary bus and ended up with an extremely flexible and usable bus structure, while preserving an extraordinary degree of compatibility with preexisting designs.

This standard was prepared by the 696 Working Group of the Microprocessor Standards Committee of the IEEE Computer Society. At the time this standard was approved the membership of the working group was as follows:

Mark Garetz, *Chairman*

Bob Davis	Howard Fullmer*	Sol Libes	Don Pannell
Kells Elmquist	David Gustavson	Dick Lowe	William Stark
Tim Erickson	Richard Jones	Ed Lupin	Bob Stewart
Gary Feirbach	Chuck Krause	George Morrow *	Michael Stolowitz
Kevin Fischer			John Terry

***Past Chairman**

Contents

SECTION PAGE

FIGURES

PAGE

TABLES

An American National Standard

IEEE Standard 696 Interface Devices

1. General

1.1 Scope. This standard applies to interface systems for computer system components interconnected by way of a 100-line parallel backplane commonly known as the S-100 bus.

It applies to microprocessor computer systems, or portions of them, where:

(1) Data exchanged among the interconnected devices is digital

(2) X maximum of 22 devices are interconnected

(3) The total transmission path length among interconnected devices is less than or equal to 25 in (63.5 cm)

(4) The maximum switching rate of any signal on the bus is less than or equal to 6 MHz

1.2 Object. This standard is intended:

(1) To define a rational, general-purpose interface system for designers of new computer system components that will ensure their compatibility with present and future IEEE Std 696 computer systems.

(2) To provide the microprocessor computer-system user with compatibie device families which will communicate in an unambiguous way without modification, from which a modularly expandable computer system may be constructed.

(3) To enable the interconnection of independently manufactured devices into a single system.

(4) To specify terminology and definitions related to the system.

(5) To define a system with the minimum number of restrictions on the performance characteristics of devices connected to the system.

(6) To define a system that, of itself, is of reiatively low cost, and allows the interconnection of low-cost devices.

(7) To define a system that is easy to use.

1.3 Definitions. The followlng definitions apply for the purpose of this standard. This section contains only general definitions. Detailed definitions are given in other sections as appropriate.

Within the context of this standard, the verbs *shall* and *should* are to be interpreted as follows:

Mandatory requirements will be characterized by the use of the verb *shall.*

Recommended practices will be characterized by the *use* of the verb *should.*

1.3.1 General-System Terms

compatibility. The degree to which devices may be interconnected and used without modification, when designed to conform to Sections 2, 3, and 4 of this standard.

device. A circuit or logical group of circuits resident on one or more boards capable of interacting with other such devices through the bus.

interface. A shared electrical boundary between parts of a computer system, through which information is conveyed.

interface system. The device independent functionai, electrical, and mechanical elements of an interface necessary to effect unambiguous communication among a set of devices. Driver and receiver circuits, signal line descriptions, timing and control conventions, data transfer protocols, and functional logic circuits are typical system elements.

kilobyte 1024 = 2_{10}

megabyte 1048 576 = 2_{20}

system. A set of interconnected elements constituted to achieve a given objective by performing specified functions.

1.3.2 Signals and Paths

active. A signal in its logically true state.

activate. Same as: assert.

assert. To cause a signal line to transition from its logically false (inactive) state to its logically true (active) state. The true or active state is either a high or low state, as specified for each signal.

bidirectional bus. A bus used by any individual device, or set of devices, for the two-way transmission of data, that is both input and output.

bit-parallel. A set of concurrent data bits present on a like number of signal lines used to carry information. Bit-parallel data bits may be acted upon concurrently as a group or independently as individual data bits.

bus. A set of signal lines used by an interface system, to which a number of devices are connected, and over which information is transferred between the devices.

bus cycle. The basic sequence of electrical events required to complete a transfer of data on the bus. A bus cycle shall contain at least three bus states.

bus state. A bus state is one clock cycle long and begins and ends just before the rising edge of ϕ. There shall be at least three bus states in every bus cycle.

byte. A set of bit-parallel signals corresponding to binary digits operated on as a unit. Connotes a group of eight bits where the most significant bit carries the subscript 7 and the least significant bit carries the subscript 0.

byte-serial. A sequence of bit-parallel data bytes used to carry information over a common bus.

deactivate. To cause a signal to transition from its logically true (active) state to its logically false (inactive) state. Opposite of assert.

high state. The electrically more positive signal level used to assert a specific message content associated with one of two binary logic states.

inactive. A signal in its logically false state.

low state. The electrically less positive signal level used to assert a specific message content associated with one of two binary logic states.

signal. The physical representation which conveys data from one point to another. For the purpose of this standard, this applies to digital electrical signals only.

signal level. The magnitude of a signal when considered in relation to an arbitrary reference magnitude (voltage in the case of this standard) .

signal line. One of a set of signal conductors in an interface system used to transfer messages among interconnected devices.

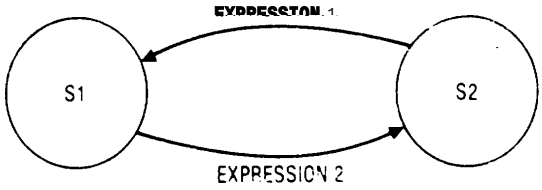
signal parameter. That parameter of an electrical quantity whose values or sequence of values convey information.

unidirectional bus. A bus used by a device for one-way transmission of messages, that is, either input only or output only.

word. A set of bit-parallel signals corresponding to binary digits and operated on as a unit. For this standard , word connotes a group of 16 bits where the most significant bit carries the subscript 15 and the least significant bit carries the subscript 0.

1.4 State Diagram Notation. Each state that an interface function can assume is represented graphically by a circle. A mnemonic is used within the circle to identify the state.

All permissible transitions between states of an interface function are represented graphically by arrows between them. Each transition between states may be qualified by an expression whose value must be either true or false. If a state transition is not qualified by an expression it is implied that transition from one state to another will occur after some time period, as indicated in the timing specifications. An interface function must enter the state pointed to if and only if the driving expression becomes true, or in the case of a time dependent transition: as soon as the minimum specified time has passed.



An expression consists of two parts, a driving expression and a driven expression, separated by a slash (/ j. The driving expression shall specify the conditions necessary for the state transition. The driven expression is optional and is used to indicate signal transitions as a result of the state transition. A signal transition is indicated by the signal name followed by an equal

sign (=), followed by an indication of the state attained by the signal as a result of the transition. A driving expression consists of one or more messages used in conjunction with the operators AND (a*b), OR (a+b), and NOT (-a). Precedence is defined by parentheses. An example expression is: (driving/driven)

A * (B+C) / D=F(ALSE), E=T(RUE)

If A AND (B OR C) is true, then D is forced false and E is forced true, and the state transition takes place.

1.5 Logical and Electrical State Relationships. This standard makes a distinction between the logical function of a signal and its electrical implementation. All equations in this standard **are** logic equations, not electrical equations (unless otherwise stated), and are written in terms of logic states.

There are two types of electrical implementation of the logic states as given in Tables 1 and 2.

In translating **a** logic equation into an electrical implementation, care must be taken to account for the active-high or active-low character of the electrical signal. For example, the logic equation

MWRT = pWR * -sOUT, **(logic equation) when** implemented electrically, becomes

MWRT = (- pWR*) * - sOUT, (electrical **equation**)

since

pWR* = electrical signal carrying the pWR information on the bus.

Note that this is equivalent to

MWRT = - (pWR* + sOUT), (electrical equation

by deMorgan's theorem; consequently, a single

Table 1
Active High Signals

Active high signals are represented without a suffix after the signal name mnemonic (that is, **ABCD).**

Logic State	Binary State	Electrical Signal Level	Electrical State
FALSE (F)	0	CORRESPONDS TO < = 0.8 V, CALLED THE LOW STATE.	L
TRUE (T)	1	CORRESPONDS TO > = 2.0 V, CALLED THE HIGH STATE.	H

Table 2
Active Low Signals

Active low signals are represented with an **asterisk** suffix after the mnemonic (that is, ABCD*).

Logic State	Binary State	Electrical Signal Level	Electrical State
FALSE (F)	0	CORRESPONDS TO > = 2.0 V, CALLED THE HIGH STATE.	H
TRUE (Tj	1	CORRESPONDS TO < = 0.8 V, CALLED THE LOW STATE.	L

two-input NOR gate is sufficient to implement MWRT, if it meets the loading and drive requirements.

The edge or change of electrical value of an electrical signal on a timing diagram which causes a transition change of the variable as a logic variable from false to true is:

Signal	Edge
active high	rising
active low	falling

Logic equations in the state diagrams are written in terms of logic state, not electrical state.

The suffix asterisk "*" is not a negation operator. It is a designator (like a comment or footnote) attached to a name, telling the reader what the relationship is between the truth state and the electrical state. That is, this variable is true when the line on the bus is low.

A prefix minus sign "-" represents the logical negation operator and is equivalent to the use of an overbar. Parentheses are used to enclose the negated variable when required for clarity.

1.6 Interface System Overview

1.6 .1 Interface System Objective. The overall purpose of the interface system is to provide an effective communication link over which messages are carried in an unambiguous way among a group of interconnected devices.

Messages in an interface system belong to either of two broad categories:

(1) Messages used to manage the interface system itself, called interface messages.

(2) Messages used by the devices interconnected by the interface system, and carried by that system, but not part of the interface system itself (that is, data). These are called device dependent messages.

The interface system herein described comprises the necessary functional and electrical specifications for interface messages to effect the objective of this standard, but it is beyond the scope of this standard to specify the nature or meaning (other than electrical signal level) of device dependent messages.

1.6.2 Fundamental Communicatio Capabilities. An effective communication link requires two basic functional elements to organize and manage the flow of information among devices :

(1) A device acting as a bus master

(2) A device acting as a bus slave

All data transfer communications between a bus master and **a bus** slave are carried out in terms of a generalized bus cycle generated by the bus master and responded to by the addressed **bus slave.**

In the context of the interface system described by this standard:

(1) A device acting as a bus master has the capability to address all bus slaves, or some portion of them, by generating all interface messages necessary to effect a bus cycle, and has the **capability** to transfer device dependent messages to or from the addressed bus **slave as** a part of that bus cycle.

(2) A device acting as a bus slave monitors all bus cycles, and has the capability, thus, to be addressed by a bus master and to transfer device dependent messages to or from a bus master.

Bus master and bus slave capabilities occur both individually and collectively in boards interconnected by way of the interface system defined by this standard.

1.6.3 Message Paths and Bus Structure. The IEEE Std 696 Bus interface system consists of a set of signal lines used to carry all information, interface messages, and device dependent messages among interconnected devices.

The bus structure is organized into eight sets of signal lines and one set of power lines as shown in Table 3.

Table 3
Bus Structures

(1) Data bus	16 signal lines
(2) Address bus	16 or 24 signal lines
(3) Status bus	8 signai lines
(4) Control output bus	5 signal lines
(5) Control input bus	6 signal lines
(6) TMA control bus	8 signal lines
(7) Veetored interrupt bus	8 signal lines
(8) Utility bus	16 signal lines
(9) Power Bus	9 power lines

2. Functional Specifications

2.1 Functional Partition. Devices interconnected by way of the interface system are divided into two broad classifications, bus masters and bus slaves, according to their relation-

ship to the generation and reception of interface messages.

Devices acting as bus masters shall be responsible for the initiation of all bus cycles, and for the generation of all signals necessary for the conduction of an unambiguous bus cycle. These signals are termed type M signals, and consist of the address, status, and control buses. Device-dependent messages are transmitted and received on the data bus.

Bus masters are subdivided into two classifications, permanent masters and temporary masters. A bus master (generally a CPU) is the highest priority master in the interface system. A temporary master may request the bus from the permanent master for an arbitrary number of bus cycles, and then returns control of the bus to the permanent master. The transfer of bus control from a permanent master to a temporary master and back to the permanent master is termed a TMA (Temporary Master Access) cycle.

The difference between a permanent bus master and a temporary bus master is that:

(1) Only one permanent master shall exist within the interface system, whereas up to 16 temporary masters may co-exist in a single system.

(2) A temporary master is not subject to a TMA cycle, that is, there are no nested TMA operations.

Devices acting as bus slaves are bus-cycle receptors. A bus slave monitors all bus cycles and, if addressed during a particular bus cycle, accepts or sends the requested device-dependent message on the data lines. While bus masters shall generate a specific set of signals in order to ensure an unambiguous bus cycle, a bus slave need only examine and generate that subset of bus signals necessary to communicate with bus masters.

2.2 Signal Lines

2.2.1 General. The bus is a collection of message paths defined relative to the current bus master. They are:

(1) Address bus
(2) Status bus
(3) Data input/output bus
(4) Control output bus
(5) Control input bus
(6) TMA control bus
(7) Vectored interrupt bus
(8) Utility bus

The nature and use of each bus is specified in the following sections. Except as otherwise specified, all bus signals are three-state lines.

2.2.2 Address Bus. The address bus consists of 16 or 24 bit-parallel signal lines used to select a specific location in memory or a specific input/output device for communication during the current bus cycle.

All bus masters shall assert A0 through ,415, but may assert lines Al6 through A23 if extended address capability is desired. Validity of the address bus is defined in 2.7.2.

All address lines are three-state lines.

Table 4 summarizes address usage for various bus cycles.

2.2.2.1 Standard **Memory Addressing.** The standard memory address bus consists of 16 lines specifying 1 of 64 kilobyte memory locations. These 16 lines are named Al 5 through A0, where Al5 is the most significant bit.

Table 4
Address Usage for Different Bus Cycles

Cycle Type	Standard Addressing	Extended Addressing
MEMORY READ		
MEMORY WRITE	Al 5-A0	A23-A0
Ml (OP-CODE FETCH)		
INPUT		
OUTPUT	A7-A0†	A15-A0
INTERRUPT ACKNOWLEDGE	NONE	NONE
HALF ACKNOWLEDGE	NONE	NONE

†See 2.2.2.4

2.2.2.2 Extended Memory Addressing. The extended memory address bus consists of 24 lines specifying 1 of 16 megabyte memory locations. These 24 lines are named A23 through AO, where A23 is the most significant bit.

2.2.2.3 PHANTOM*. The PHANTOM* signal provides the capability of over-laying two memory slaves at **a** common address location. When this line is activated phantom memory slaves are enabled and normal memory slaves are disabled. All normal memory slaves shall have the capability of being disabled in response to PHANTOM* being asserted. Normal memory slaves shall be disabled during PHANTOM* for both read and write cycles.

2.2.2.4 Standard Input/Output Device Addressing. The standard l/O device address bus consists of 8 lines, A7 through AO, specifying 1 of 256 I/O devices. A7 is the most significant bit.

NOTE: The I/O device address has traditionally been duplicated onto the high order address byte, A15-A8. While this is considered acceptable procedure, it is not recommended for new designs as it complicates expansion to extended I/O device addressing. Standard I/O slaves shall not rely solely on the high order address byte for address decoding.

2.2.2.5 Extended Input/Output Device **Addressing.** The extended I/O device address bus consists of 16 lines, Al5 through AO, specifying 1 of 64 kilobyte devices. Al5 is the most significant bit.

22.3 **Status Bus.** The status bus consists of eight lines which identify the nature of the bus cycle in progress, and qualify the nature of the address on the address bus.

The mnemonics for status lines **always** begin with a lowercase s.

The 8 status lines are :

(1) Memory read	sMEMR
(2) Op-code fetch	sM1
(3) Input	sINP
(4) output	sOUT
(5) Write cycle	swo*
(6) Interrupt acknowledge	sINTA
(7) Halt acknowledge	sHLTA
(8) Sixteen-bit data transfer request	sXTRQ*

The 8 lines on the status bus shall be generated by the current bus master.

All status lines are three&ate lines.

Validity of the status bus is given in 2.7.2.

2.2.3.1 **Status Memory Write.** One relevant status signal is not directly available on the bus, but may be created on individual slaves by the combination of two others. Status Memory Write is defined as:

sMemory Write = (- sOUT) * xWO, (logic equation

that is, status memory write is true when sOUT is false and s WO (write) is true.

2.2.3.2 Status Usage Chart. Table 5 gives the status word definitions for all possible bus cycles.

Table 5
Status Usage Chart

STATUS BITS		sMEMR	sM1	sWO*	sOUT	sINP	sINTA	sHLTA	sXTRQ*
CYCLE TYPE									
MEMORY READ	(B)	H	L	H	L	L	L	L	H
	(W)	H	L	H	L	L	L	L	L
OPCODE FETCH	(B)	H	H	H	L	L	L	L	H
	(W)	H	H	H	L	L	L	L	L
MEMORY WRITE	(B)	L	L	L	L	L	L	L	H
	(W)	L	L	L	L	L	L	L	L
OUTPUT	(B)	L	L	L	H	L	L	L	H
	(W)	L	L	L	H	L	L	L	L
INPUT	(B)	L	L	H	L	H	L	L	H
	(W)	L	L	H	L	H	L	L	L
INTERRUPT	(B)	L	X	H	L	L	H	L	H
ACKNOWLEDGE	(W)	L	X	H	L	L	H	L	L
HALT ACKNOWLEDGE		X	X	H	L	L	L	H	X
IDLE		L	L	H	L	L	L	L	X

Legend B = 8-BIT OPERATION; L = LOW STATE; X = DON'T CARE; H = HIGH STATE; W = 16-BIT OPERATION

(W) Refers to word (16-bit data path) operations
(B) Refers to byte (8-bit data path) operations
H = High state
L = Low state
X = Don't care

22.4 Data **Bus.** Data input and data output are always specified relative to the current bus master. Data transmitted by the current bus master to a bus slave is called data output. Data received by the current bus master from a bus slave is called data input.

The data bus consists of 16 lines grouped as two unidirectional 8-bit buses for byte operation and as a single bidirectional bus for 16-bit word operations.

2.2.4.1 Byte Operations. Two unidirectional 8-bit buses are used for byte data transfers. Data output appears on the data output bus (DO7-DO0), where DO7 is the most significant bit.

Data input appears on the data input bus (DI7-DI0), where D17 is the most significant bit.

22.4.2 Word Operations. For 16-bit data transfers the DI and the DO buses are ganged together, creating a single 16-bit bidirectional bus. Two signal lines control the ganging of the data buses, sixteen request (sXTRQ*) and sixteen acknowledge (SIXTN*). When both of these lines are true (in the low state), the data buses are ganged with the even addressed byte on the DO bus (which now becomes named ED7-ED0; ED for "even data"), and the odd addressed byte on the DI bus (which now becomes OD7-OD0; OD for "odd-data").

Complete specification of the 8/16-bit protocol is given in 2.6.

2.2.5 Control Output Bus. The 5 lines of the control output bus determine the timing and movement of data during any bus cycle. The mnemonics for the control output lines always begin with a lowercase p.

The five lines are:

(1) pSYNC, which indicates the start of a new bus cycle

(2) pSTVAL*, the active edge of which in conjunction with pSYNC indicates that stable address and status may be sampled from the bus in the current cycle.

(3) pDBIN, a generalized read strobe that gates data from an addressed slave onto the data bus.

(4) pWR*, a generalized write strobe that writes data from the data bus into an addressed slave.

(5) pHLDA, the hold acknowledge signal that indicates to the highest priority temporary master that the permanent master is relinquishing control of the bus.

The control output signals are subject to the functional and timing disciplines given in 2.7, 3.8, and 3.9.

2.2.6 Control Input Bus. The six lines of the control input bus allow bus slaves to synchronize the operations of bus masters with conditions internal to the bus **slave** (for example, data not ready), and to request operations of the permanent master (for example, interrupt or hold).

The six control input lines are:

(1) RDY
(2) XRDY
(3) INT*
(4) NMI*
(5) HOLD*
(6) SIXTN*

2.2.6.1 Ready Lines. The ready lines are used by bus slaves to synchronize bus masters to the response speed of the slave. Thus, cycles are suspended and wait states inserted until both are asserted. I

The RDY line is the general ready line for bus slaves. It is specified as an open collector line.

The XRDY line is a special ready line commonly used by front panel devices to stop and single step bus masters. It is specified as an active line, therefore it should not be used by other bus slaves, since a bus conflict may exist.

2.2.6.2 Interrupt Lines. The two interrupt lines, INT* and NMI*, are used to request service from the permanent bus master.

The INT* line may be masked off by the bus master, usually by way of an internal software operation. If the master accepts the interrupt request on the INT* line, it may respond with an interrupt acknowledge bus cycle, accepting vectoring information from the data bus. During vectored interrupts, INT* indicates the acceptance, by the vectored interrupt controller, of one or more vectored interrupt requests.

The NMI* line is a nonmaskable interrupt request line. that is, it may not he masked off by the bus master. Accepting an interrupt on the

Table 6
IEEE Std 696 Bus Fin List

Pin No	Signal and Type	Active Level	Description
1	+8 V (B)		Instantaneous minimum greater than 7 V, instantaneous maximum less than 25 V, average maximum less than 11 V.
2	+16 V (B)		Instantaneous minimum greater than 14.5 V, instantaneous maximum less than 35 V, average maximum less than 21.5 V
3	XRDY (S)	H	One of two ready inputs to the current bus master. The bus is ready when both these ready inputs are true. See pin 72.
4	VI0* (S)	L oc	Vectored interrupt line 0
5	VI1* (S)	L oc	Vectored interrupt line 1
6	VI2* (S)	L oc	Vectored interrupt line 2
7	VI3* (S)	L oc	Vectored interrupt line 3
8	VI4* (S)	L OC	Vectored interrupt iine 4
9	VI5* (S)	LOC	Vectored interrupt line 5
10	VI6* (S)	L oc	Vectored interrupt line 6
11	VI7* (S)	L oc	Vectored interrupt line 7
12	NMI* (S)	L oc	Non-maskabie interrupt
13	PWRFAIL* (B)	L	Power fail bus signal. (See 2.10.1 regarding pseudo open collector nature)
14	TMA3* (M)	L oc	Temporary master priority bit 3
15	Al8 (M)	H	Extended address bit 18
16	Al6 (M)	H	Extended address bit 16
17	Al7 (M)	H	Extended address bit 17
18	SDSB* (M)	L oc	The signal to disable the 8 status signals
19	CDSB* (M)	L oc	Tine signal to disable the 5 control output signals
20	0 V (B)		Common with pin 100
21	NDEF		Not to be defined. Manufacturer must specify any use in detail
22	ADSB* (M)	L oc	The signal to disable the address signals
23	DODSB* (M)	L oc	The control signal to disable the data output signals. (DO7-0 for 8 bit transfers, ED7-0 and OD7-0 for 16 bit transfers)
24	ϕ (B)	A	The master timing signal for the bus
25	pSTVAL* (M)	L	Status valid strobe
26	pHLDA (M)	H	A control signal used in conjunction with HOLD* to coordinate bus master transfer operations
27	RFU		Reserved for future use
28	RFU		Reserved for future use
29	A5 (M)	H	Address bit 5
30	A4 (M)	H	Address bit 4
31	A3 (M)	H	Address bit 3
32	Al5 (M)	H	Address bit 15 (most significant for non-extended addressing.)
33	Al2 (M)	H	Address bit 12
34	A9 (M)	H	Address bit 9
35	DO1 (M)/ED1 (M/S)	H	Data out bit 1, bidirectional even data bit i
36	DO0 (M)/EDO (M/S)	H	Data out bit 0, bidirectional even data bit 0
37	A10 (M)	H	Address bit 10
38	DO4 (M)/ED4 (M/S)	H	Data out bit 4, bidirectional even data bit 4
39	DO5 (M)/ED5 (M/S)	H	Data out bit 5, bidirectional even data bit 5
40	DO6 (M)/ED6 (M/S)	H	Data out bit 6, bidirectional even data bit 6

Table 6
IEEE *Std 696* Bus Pin List *(Continued)*

Pin No	Signal and Type	Active Level	Description
41	D12 (S)/OD2 (M/S)	H	Data in bit 2, bidirectional odd data bit 2
42	D13 (S)/OD3 (M/S)	H	Data in bit 3, bidirectional odd data bit 3
43	D17 (S)/OD7 (M/S)	H	Data in bit 7, bidirectional odd data bit 7
44	sM1 (M)	H	The status signal which indicates that the current cycle is an op-code fetch
45	sOUT (M)	H	The status signal identifying the data transfer bus cycle to an output device
46	sINP (M)	H	The status signal identifying the data transfer bus cycle from an input device
47	sMEMR (M)	H	The status signal identifying bus cycles which transfer data from memory to a bus master, which are not interrupt acknowledge instruction fetch cycle(s)
48	sHLTA (M)	H	The status signal which acknowledges that a HLT instruction has been executed
49	CLOCK (B)	A	2 MHz (±0.5%) 40-60% duty cycle. Not required to be synchronous with any other bus signal
50	0 V (B)		Common with pin 100
51	+8 V (B)		Common with pin 1
52	-16V (B)		Instantaneous maximum less than - 14.5 V, instantaneous minimum greater than -35 V, average minimum greater than -21.5 V
53	0 V (B)		Common with pin 100
54	SLAVE CLR* (B)	LOC	A reset signal to reset bus slaves. Must be active with POC* and may also be generated by external means
55	TMA0* (M)	LOC	Temporary master priority bit 0
56	TMA1* (M)	LOC	Temporary master priority bit 1
57	TMA2* (M)	LOC	Temporary master priority bit 2
58	sXTRQ* (M)	L	The status signal which requests 16-bit slaves to assert SIXTN*
59	A19 (M)	H	Extended address bit 19
60	SIXTN* (S)	LOC	The signal generated by 16-bit slaves in response to the 16-bit request signal sXTRQ*
61	A20 (M)	H	Extended address bit 20
62	A21 (M)	H	Extended address bit 21
63	A22 (M)	H	Extended address bit 22
64	A23 (M)	H	Extended address bit 23
65	NDEF		Not to be defined signal
66	NDEF		Not to be defined signal
67	PHANTOM* (M/S)	L oc	A bus signal which disables normal slave devices and enables phantom slaves-primarily used for bootstrapping systems without hardware front panels.
68	MWRT (B)	H	pWR* + sOUT (logic equation). This signal must follow pWR* by not more than 30 ns. (See NOTE, 2.7.5.3)
69	RFU		Reserved for future use
70	0 V (B)		Common with pin 100
71	RFU		Reserved for future use
72	RDY (S)	H OC	See comments for pin 3
73	INT* (S)	L oc	The primary interrupt request bus signal
74	HOLD* (M)	L oc	The control signal used in conjunction with pHLDA to coordinate bus master transfer operations

Table 6
IEEE Std 696 Bus Pin List *(Continued)*

Pin No	Signal and Type	Active Level	Description
75	RESET* (B)	L oc	The reset signal to reset bus master devices. This signal must be active with POC* and may also be generated by external means
76	pSYNC (M)	H	The control signal identifying BS_1
77	pWR* (M)	L	The control signal signifying the presence of valid data on DO bus or data bus
78	pDBIN (M)	H	The control signal that requests data on the DI bus or data bus from the currently addressed slave
79	AO (M)	H	Address bit 0 (least significant)
80	A1 (M)	H	Address bit 1
81	A2 (M)	H	Address bit 2
82	A6 (M)	H	Address bit 6
83	A7 (M)	H	Address bit 7
84	A8 (M)	H	Address bit 8
85	Al3 (M)	H	Address bit 13
86	Al4 (M)	H	Address bit 14
87	All (M)	H	Address bit 11
88	DO2 (M)/ED2 (M/S)	H	Data out bit 2, bidirectional even data bit 2
89	DO3 (M)/ED3 (M/S)	H	Data out bit 3, bidirectional even data bit 3
90	DO7 (M)/ED7 (M/S)	H	Data out bit 7, bidirectional even data bit 7
91	D14 (S)/OD4 (M/S)	H	Data in bit 4, bidirectional odd data bit 4
92	D15 (S)/OD5 (M/S)	H	Data in bit 5, bidirectional odd data bit 5
93	D16 (S)/OD6 (M/S)	H	Data in bit 6, bidirectional odd data bit 6
94	DI1 (S)/OD1 (M/S)	H	Data in bit 1, bidirectional odd data bit 1
95	DIO (S)/OD0 (M/S)	H	Data in bit 0 (least significant for 8 bit data) and bidirectional odd data bit 0
96	sINTA (M)	H	The status signal identifying the bus input cycle(s) that may follow an accepted interrupt request presented on INT*.
97	sWO* (M)	L	The status signal identifying a bus cycle which transfers data from a bus master to a slave.
98	ERROR* (S)	LOC	The bus status signal signifying an error condition during present bus cycle.
99	POC* (B)	L	The power-on clear signal for all bus devices; when this signal goes low, it must stay low for at least 10 μs.
100	0 V (B)		System ground

NMI* line need not generate an interrupt acknowledge bus cycle.

An interrupt request on the INT* line is asserted as a level, that is, the line is asserted until interrupt service is received. An interrupt request on the NMI* line, on the other hand, is asserted as a negative going edge, since no interrupt acknowledge cycle need be generated.

Both these lines are specified as open-collector lines.

2.2.6.3 Hold Request. The hold request line, HOLD*, is used by temporary bus masters to request control of the bus from the permanent bus master. The HOLD* line may be masked by the permanent bus master to prevent temporary masters from gaining bus control.

The HOLD* line is specified as an open collector line, and shall only be asserted at certain times. See 2.8.3.

2.2.6.4 Sixteen Acknowledge. The sixteen acknowledge line, SIXTN*, is a response to the status signal sixteen request (sXTRQ*), and indicates that the requested 16-bit data transfer is possible.

The SIXTN* line is specified as an open-collector line. Detailed specification of the use of this line is given in 2.6.

2.2.7 TMA Control Bus. The eight lines of the TMA control bus are used in conjunction with control bus signals HOLD* and pHLDA. They arbitrate among simultaneous requests for control of the bus by temporary masters and disable the signal drivers of the permanen t bus master, thus effecting an orderly transfer of bus control.

All eight lines of the TMA control bus are specified as open-coliector iines.

The eight TMA control lines are:

(1) TMA0*
(2) TMA1*
(3) TMA2*
(4) TMA3*
(5) ADSB*
(6) DODSB*
(7) SDSB*
(8) CDSB*

Detailed specification of the use of these lines is given in 2.8.

2.2.7.1 TMA Arbitration. The four lines that arbitrate among simultaneous requests for bus control by temporary masters are TMAO* through TMA3*. The encoded priority of requesters is asserted on these lines and, after settling, they contain the priority number of the highest priority requester.

Detailed specification of this process is given in 2.8.3.

2.2.7.2 **Bus Transfer** Signals. Four signals are available on the bus to disable the line drivers of the permanent bus master. They are:

(1) ADSB*, address disable
(2) DODSB*, data out disable
(3) SDSB*, status disable
(4) CDSB*, control output disable

NOTE: The signal DODSB* shall inhibit the eight DO bus drivers (DO7-DO0) during an eight bit transfer (sXTRQ* inactive) and shall inhibit the 15 bidirectional drivers (ED7-ED0 and OD7-OD0) from driving the bus during sixteen bit transfers (sXTRQ* and SIXTN* both active).

Use of these lines is tightly specified during the transfer of the bus from a permanent master to a temporary master, as given in 2.8.2., and any transfer involving the control output lines should follow a similar protocol.

The address, data, and status signals from the permanent master may be disabled and replaced using these signals as long as the contents of these buses is valid for the current bus cycle as though no replacement had occurred.

2.2.8 Vectored Interrupt Bus. The eight lines of the vectored interrupt bus are used in conjunction with the generalized interrupt request, INT*, to arbitrate among eight levels of interrupt request priorities. They are typically implemented as inputs to a bus slave which masks and prioritizes the requests, asserts the generalized interrupt request (INT*) to the permanent bus master, and responds to the interrupt acknowledge bus cycle with appropriate vectoring data. The interrupt controller need not assert INT* in response to vectored interrupts that are masked out.

The eight lines of the vectored interrupt bus are VI0* through V17*, where VIO* is the highest priority interrupt.

The vectored interrupt lines shall be implemented as levels, that is, they shall be held active until service is received.

The vectored interrupt lines are specified as open collector lines.

2.2.9 System Utilities

2.2.9.1 System Power. Power in IEEE Std 696 systems is distributed to bus devices as unregulated voltages. A total of nine bus lines are used:

(1) +8 V, 2 lines
(2) +16 V, 1 line
(3) -16 V, 1 line
(4) 0 V, 5 lines

0 V (ground) lines are distributed across the edge connector such that low-impedance grounds are available on both sides of the edge connector, and on both sides of the circuit cards.

Power lines are subject to the specifications given in 3.2.

2.2.9.2 System Clock (ϕ). The system clock, ϕ, is generated by the permanent master. The control timing for all bus cycles, whether they are cycles of the permanent master ***or cycles of*** temporary masters in control of the bus, shall be derived from this clock.

This signal is never transferred during a bus exchange operation.

2.2.9.3 Clock (CLOCK). This clock is specified as a 2 MHz (± 0.5% tolerance) signal with no relationship to any other bus signal. It is to be used by counters, timers, baud-rate generators, etc. It is specified as an active line.

2.2.9.4 **System** Reset Functions **(RESET*, SLAVE CLR* and POC*).** System reset functions are divided into three lines:

(1) RESET*, resets all bus masters
(2) SLAVE CLR*, resets all bus slaves
(3) POC*, power-on clear is active **at** power-on and following the rising edge of PWRFAIL*, and shall cause SLAVE CLR* and RESET* to be asserted.

The POC* signal is specified as having a minimum active period of 10 ms; RESET* and SLAVE CLR* are specified as having **a** minimum active period of 5 μs.

RESET* and SLAVE CLR* are specified as opencollector lines and POC* is specified as an active line.

2.2.9.5 Memory Write Strobe (MWRT). The memory write strobe, MWRT, shall be generated somewhere in the system. It has traditionally been generated by front-panel type devices, but is optionally generated by permanent masters or mother boards in systems without front panels. Care must be taken that it is generated at only one point in a given system. All boards that are capable of generating MWRT shall be provided with some means of disconnecting it from the bus.

Memory write is defined as:

MWRT = pWR* -sOUT (logic equation)

NOTE: MWRT shall be generated directly from the pWR* and sOUT bus signals, and not from the internal signals.

MWRT is specified as an active line.

2.2.9.6 Phantom Slaves (PHANTOM*). A line, PHANTOM*, is provided **for overlaying** memory slaves at a common address location. **When this line is activated phantom memory slaves are enabled and normal memory slaves are disabled. All normal memory slaves shall have the capability of being disabled** in response to PHANTOM*. Memory slaves shall be disabled during PHANTOM* for both read and write cycles.

This line is specified **as an opencollector line.**

2.2.9.7 Error (ERROR*). The line ERROR* is a generalized error line that is asserted when an error of some sort (for example, parity, write to protected memory) is occurring in the current bus cycle.

This line is specified as an open-collector line.

2.2.9.8 Manufacturer Specified Lines (NDEF). Three lines which can be specified by individual manufacturers are provided on the bus. These lines, termed NDEF (not to be defined), should only be implemented as options, and shall be provided with jumpers so that possible conflicts **may** be eliminated.

Any manufacturer SHALL specify in detail any **use** of these lines. Signals on these lines are limited to 5 V logic levels.

2.2.9.9 Power Fail (PWRFAIL*). The power-fail line indicates impending power failure, and remains true until power is restored. The rising edge of PWRFAIL* shall cause POC* to be asserted.

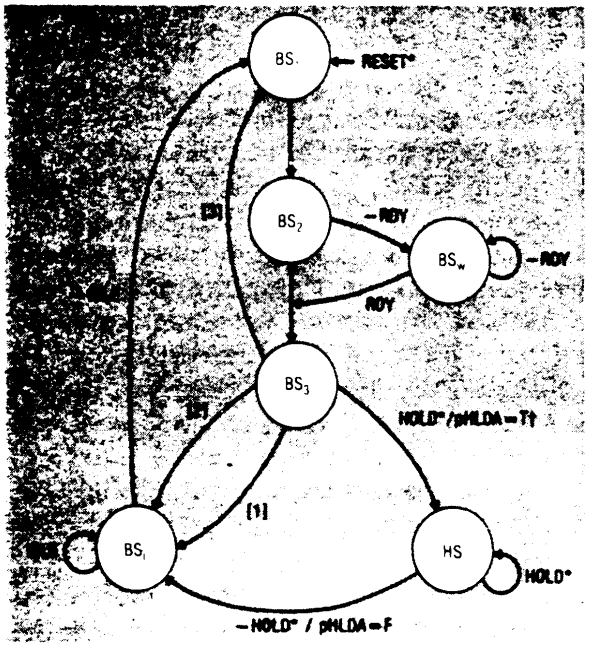
2.2.9.10 Reserved Lines (RFU). The four remaining lines are reserved for future use and shall not be used for any purpose.

2.2.10 **Pin List.** Pin connections to the card edge connector shall conform to the list given in Table 6.

2.3 The Permanent Master Interface

2.3.1 General. The permanent master interface provides the capability to transfer device dependent messages to and from all bus **slaves.** It is responsible for the generation and timing of all bus cycles while it has control of the bus, and is capable of generating all possible bus cycles.

The permanent master normally has control of the bus. It may relinquish 'bus control to a temporary bus master by way of a hold opera-



†There is a minimum specified time delay between hold and hold acknowledge
[1] Instruction execution complete INT enable . INT request/interrupt accept
[2] Instruction execution not complete + INT disabled + *no* interrupt request
[3] (Instruction execution complete) - HOLD* . - interrupt accept) + (Instruction execution not complete ◦ - IDLE)

Fig 1
Permanent Master State Diagram

tion for an arbitrary number of cycles. Upon completion of the hold operation, control of the bus is always returned to the permanent master.

2.32 Permanent Master State Diagram. The permanent master interface shall be implemented so as to conform to the state diagram given in Fig 1.

2.3.3 **Permanent Master State Descriptions**

2.3.3.1 Bus State 1 (BS1). The initial bus state, **BS1,** is the state in which the status and address buses are in transition to their values for the new bus cycle. pSYNC goes true during the **BS1** state, indicating the beginning of a new bus cycle.

2.3.3.2 Bus State 2 (BS2). Bus state 2, BS2, is the state during which the address and status lines become stable. When they are guaranteed stable the pSTVAL*, status valid strobe, is activated.

The ready lines and the sixteen acknowledge lines are sampled during the BS2 state (except during siave abort cycles, see 2.7.5 .4).

2.3.3.3 Wait State(BSw). The wait state, BSw, is entered if the ready line sampled in BS2 indicates that the addressed bus slave is not ready for data transfer. The ready line is sampled once every clock cycle until a ready condition is indicated (except during slave abort cycles, **see 2.7.5.4).** When the **ready** condition is indicated the BS2 state is completed and the BS3 state entered.

The BSw state is thus used to synchronize bus cycles generated by bus masters with the response speed of assorted bus slaves.

NOTE: The state of the ready lines shall be ignored if the current cycle is a slave abort (see 2.7.5).4

2.3.3.4 Bus **State 3(BS3). Bus** state 3, BS3, is the bus state during which the data transfer actually takes place between the master and the addressed slave.

2.3.3.5 Idle Bus States **(BSi).** After completion of the BS3 state, the master may enter one or more idle bus states.

NOTE: It is not required that masters generate idle bus states. Therefore, dynamic memory slaves cannot rely upon idle bus states for performing refresh operations.

While in an idle bus state the generalized data strobes pWR * and pDBIN must not be active, and pSTVAL* must not be asserted in conjunction with pSYNC active.

2.3.3.6 Hold Accept (HS). Permanent masters shall be configured to conditionally accept hold operations from temporary masters. This function may be disabled under hardware or software control, to allow indivisible test and set operations. If hold is enabled and active, the permanent master shall enter the hold state HS following a BS3 state, and pHLDA shall be asserted.

The permanent master remains in the hold state until the hold request HOLD* becomes false.

Hold operations shall take priority over interrupt operations.

2.3.3.7 Interrupt Accept. If hold request is not active, if execution of the current instruction is complete, and if interrupts are enabled and an interrupt is being requested, then the permanent master accepts the interrupt request at the end of the BS3 state. In the case of a vectored interrupt, the next bus cycle may be an interrupt acknowledge bus cycle. In the case of a nonmaskable interrupt, the response is usually a transfer to a predetermined location.

2.3.4 Required Signals for Permanent Masters

2.3.4.1 Output Signals. The following signals are output signals from permanent masters to bus slaves.

(1) A23-A0+
(2) All status signals
(3) All control output signals
(4) Data output signals (8 or 16 depending on processor type)
(5) ϕ, the system clock

NOTE: +A23 through Al6 are optional on permanent masters.

2.3.4.2 Input Signals. The following signals are required input signals to permanent masters:

(1) The control input signals, except NMI* and SIXTN*
(2) Data input signals (8 or 16 depending on processor type)
(3) The four disable signals ADSB*, DODSB*, SDSB*, CDSB*
(4) RESET*

2.3.5 Dummy Mastering. In cases where a number of processors co-exist in a single system as temporary masters, it **may** prove inefficient from a systems point of view to implement a permanent master.

In such a case it is permissible that the permanent master be implemented as a dummy, that is, as a device that conducts no bus cycles, but only supplies an arbitration interval so that the **TMA** control bus may settle.

The dummy master takes control of the bus between temporary masters, asserting the control output bus in the null state, and passes the bus to the next requester after a minimum arbitration interval of one clock cycle.

Required output signals for dummy masters are the control output signals, and the system clock ϕ. Input signals are HOLD* and CDSB*.

2.4 The Temporary **Master Interface**

2.4.1 General. The temporary master interface provides the capability to transfer device dependent messages to and from a selected set of bus slaves. The temporary master thus differs from the permanent master in that it need not generate all possible bus cycles.

The temporary master requests control of the bus from the permanent master. If the bus is granted, the temporary master is responsible for the generation and timing of all bus cycles until it returns control to the permanent master.

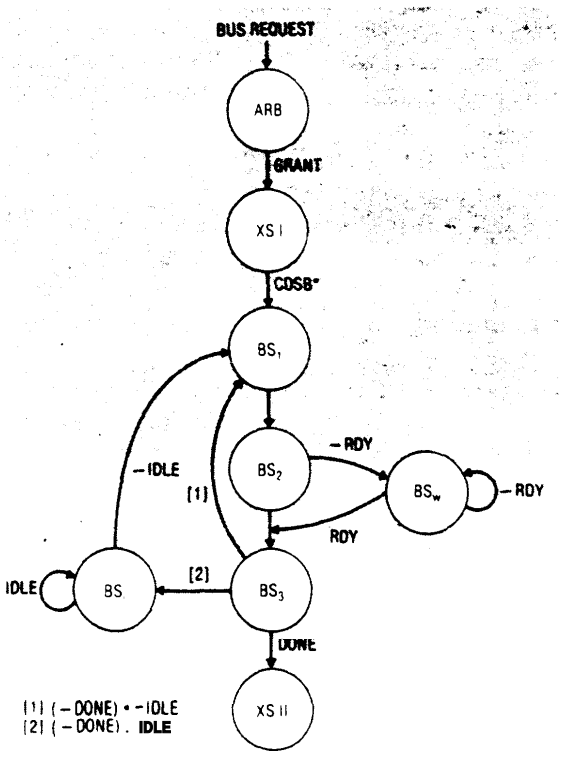


Fig 2
Temporary Master State Diagram

Since up to 16 temporary masters may coexist in a single system, a protocol has been developed to arbitrate among simultaneous bus requests. Detailed specifications of this protocol is given in 2.8.3. Any device that is capable of functioning as a temporary master shall be configured in such a way that other devices may also be used as temporary masters. That is to say that they shall arbitrate for priority.

2.4.2 Temporary Master State Diagram Temporary master interface shall be implemented so as to conform to the state diagram given in Fig 2.

2.4.3 Temporary Master State Descriptions

2.4.3.1 Arbitration (ARB). If more than one temporary master is present in the system, bus requesters shall arbitrate for the bus as given in 2.8.3.

During the arbitration sequence, bus requesters try to assert their priorities on the arbitration bus, and the contents of the arbitration bus are compared with each requester's priority.

If the contents of the arbitration bus is of higher priority than the locally attempted priority assertion, then a higher priority requester is present in the system, and the low

priority requester removes its low order bits from the arbitration bus. Thus, after some settling time, the priority of the highest priority requester is present on the arbitration bus. This requester is granted the bus on the rising edge of hold acknowledge.

2.4.3.2 Bus Transfer States (TSI and TSII). Since the bus has positive polarity control signals, extreme care must be taken in bus transfer operations to avoid erroneous pulses on the control lines.

In general terms, this is accomplished by specifying that both the permanent master and the temporary master drive the control lines in specified logic states during the bus transfer.

Detailed specification of this operation is given in 2.8.2.

2.4.3.3 Bus Cycle. The definition of bus-cycle states shall be the same as that for the permanent master interface, given in 2.3.3.1 through 2.3.3.5.

An arbitrary number of bus cycles may be performed by the temporary master before returning control to the permanent master.

2.4.4 Required Signals for Temporary Masters

2.4.4.1 Output Signals. Required output signals for a temporary master interface are as follows:

(1) Address lines A23-A0+
(2) All status signals
(3) All control output signals ++
(4) Data output lines
(5) TMA arbitration lines TMA0*-TMA3*
(6) Hold request, HOLD*

NOTES: +Temporary masters must generate A23-A16, however, they need only drive these signals low.

"Temporary masters should provide a jumper on the pSTVAL line, as older CPUs do not transfer this line with the control output lines. In this case, all bus masters use the same pSTVAL* signal.

2.4.4.2 Input Signals. Required input signals for a temporary master interface are as follows:

(1) The ready lines, RDY and optionally XRDY
(2) Hold acknowledge, pHLDA
(3) Data input lines
(4) The system clock, ϕ

2.5 The Slave Interface. A slave device responds to a bus cycle initiated by a bus master. Memory and input/output devices are examples of bus slaves.

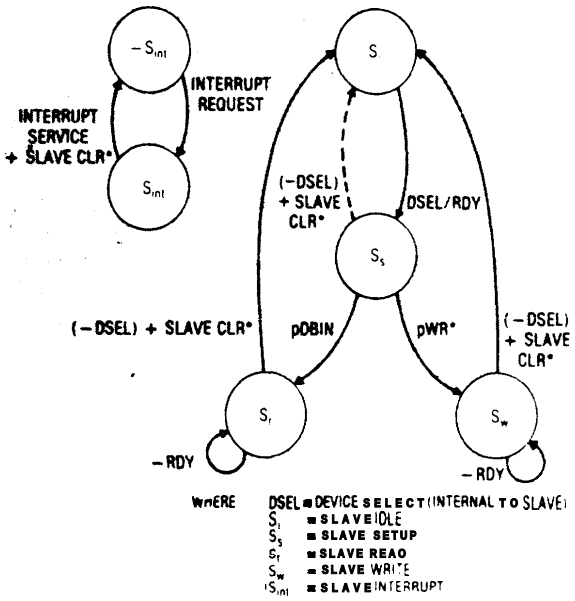


Fig 3
Slave Interface State Diagram

A slave device may request service from a bus master by generating an interrupt request.

2.5.1 Slave-Interface State Diagram. The slave interface shall conform, in general, to the state diagram given in Fig 3. Slave interfaces need not have both read and write capability.

2.5.2 Slave-State Definitions

2.5.2.1 Slave Idle State (Si). The slave idle state, Si, is a passive state with respect to the bus.

The slave monitors the stream of bus cycles to determine if it is selected for the current bus cycle.

The slave may be performing internal operations while in the idle state.

The assertion of SLAVE CLR* forces all slaves into the idle state.

2.5.2.2 Slave Setup (Ss). A slave moves from the slave idle state to the setup state, Ss, when it has been addressed by the current bus cycle. This is an operation internal to the slave which sets up a data transfer with a bus master. If a slave can tolerate spurious transitions from the idle state to the setup state, then the device select signal may be decoded statically from the address and status buses. If a device cannot tolerate spurious transitions, the device select line should be decoded in conjunction with the status valid strobe, pSTVAL*.

If synchronization is required by the slave before the data transfer may take place, the ready line is asserted false during this state until the device is ready for data transfer.

2.5.2.3 **Slave** Read **(Sr).** Data from the addressed **slave** is gated onto the data bus during the slave read state, Sr. The generalized read strobe governs the transition to this state.

When device select becomes false the slave returns to the idle state.

2.5.2.4 **Slave Write** (Sw). Data from the current bus master is written into the **slave** during the active period of the generalized write strobe, pWR*. When device select becomes false the slave returns to the idle state.

2.5.2.5 **Slave Abort.** When device select (DSEL) becomes false, the slave returns to the idle state even if no data read or write strobe has occurred. See 2.7.5.4.

2.5.2.6 Interrupt **Request State.** If a slave requires service by a bus master, an interrupt request may be generated by the slave. The interrupt should be held active until the slave is serviced, or until SLAVE CLR* is asserted.

2.5.3 Required Signals for Slave Interfaces. Slave interfaces need only receive and generate that subset of bus signals necessary for communication with masters.

2.6 8/16-bit Data Transfer Protocol

2.6 .1 **General.** Implementation of the 8/16-bit data transfer protocol allows both 8-bit and 16-bit parallel data transfers over the bus, and hence allows both 8-bit and 16-bit masters and slaves to co-exist in a single system. For 16-bit transfers the two unidirectional 8-bit data buses are ganged to form a single 16-bit bidirectional data bus.

Two iines are assigned to control the ganging of the data bus:

(1) sXTRQ*, status output from the master, which indicates a request for a 16-bit data transfer.

(2) SIXTN*, an acknowledge input to the master, which indicates that a 16-bit data transfer is possible.

Use of the sixteen acknowledge line SIXTN* permits the use of current design 8-bit memory boards without modification. When SIXTN* is false, a 16-bit transfer may be accomplished by two sequential single-byte transfers.

2.6.2 8-bit Data Paths. The current bus master requests an 8-bit transfer by not asserting sXTRQ*.

Byte data output from the master to the addressed slave is asserted on the data output bus, DO7 through DOO.

Byte data input from the addressed slave to the current bus master is asserted on the data input bus, D17 through DIO.

2.6.3 16-bit Data Paths. The current bus master requests a 16-bit transfer by asserting sXTRQ*.

If the addressed slave is capable of a 16-bit parallel data transfer, it asserts SIXTN*, as shown in the timing diagram (see Fig 12).

Sixteen-bit data transfer is then conducted by way of the ganged data buses, where DO0 = ED0 and D17 = OD7, with the DO bus carrying the even-addressed byte and the DI bus carrying the odd-addressed byte.

2.6.4 Memory Organization. Memory device capable of both 8-bit and 16-bit parallel data transfers are organized, as shown in Fig 4, as two banks of 8-bit memory, an odd-byte bank, and an even-byte bank. These **data** banks may be activated either together or separately, depending on the condition of the sixteen request status line, sXTRQ*.

2.6.4.1 Byte References. When sXTRQ* is not asserted, memory references are single-byte transfers.

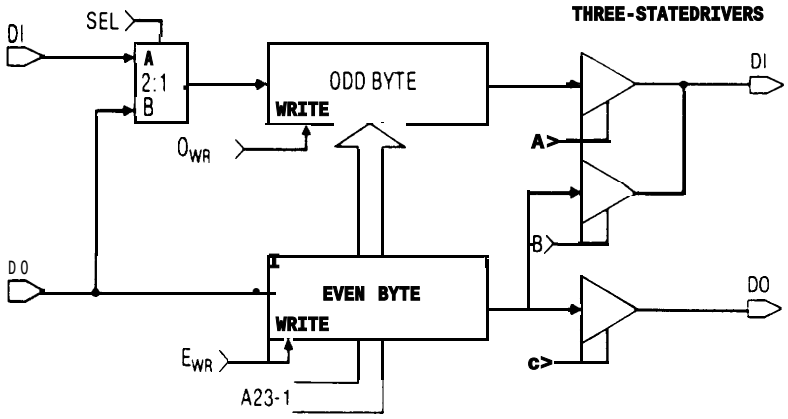
The proper location in memory is selected by the address output on address lines Al through Al5 (A23 for extended addressing systems), while the A0 line selects the even byte or the odd byte. A0 equals 0 selects the even byte of the 16-bit word, while A0 equals 1 selects the odd byte of the word.

See Fig 5 for address usage.

In the 8-bit mode, data output from the master, on the DO bus, is connected to the data input lines of both memory banks; the even-byte data input lines are connected directly to the DO bus, and the odd-byte data input lines are connected to the DO bus by way of a two-to-one multiplexer controlled by sXTRQ*.

Data output from the memory banks is routed to three-state bus drivers A and B in Fig 4. One of these drivers is enabled when the read strobe is activated, depending on the condition of AO. The selected byte is thus available to the master on the DI bus.

2.6.4.2 Word References. When sXTRQ* is asserted by the master, and SIXTN* is asserted by the slave, memory references are double-byte transfers.



SEL selects A for word references. B for byte references

Output enables
$A = 16_{rd} + (8_{rd} \cdot A\emptyset)$
$B = 8_{rd} \cdot -A\emptyset$
$C = 16_{rd}$

Write enables
$E_{wr} = 16_{wr} + (8_{wr} \cdot -A\emptyset)$
$O_{wr} = 16_{wr} + (8_{wr} \cdot A\emptyset)$

Where
16_{rd} = Device select . sXTRQ* . pDBIN
8_{rd} = Device select . -sXTRQ* . pDBIN
16_{wr} = Device select . sXTRQ* . pWR*
8_{wr} = Device select . -sXTRQ* . pWR*

Fig4
8/16 Bit-Memory Organization

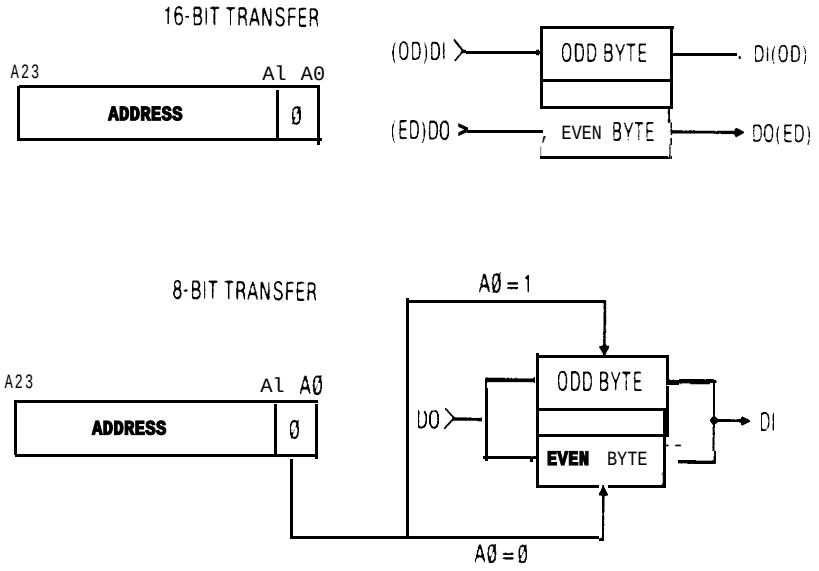


Fig5
8/16 Bit Address + Data Usage

Address lines Al through Al 5 (A23 in extended address systems) select the proper word from memory. The condition of the A0 bit does not enter into the decoding or addressing for word references, but A0 is specified as being 0 for word transfers.

See Fig 5 for address usage.

In the 16-bit mode, data output from the bus master is asserted on the 16 signal lines of the DO bus and the DI bus. The multiplexer on the data input lines now routes the odd-byte data, on the DI bus, to the data input lines of the odd-byte bank. Even-byte data, on the DO bus, is connected to the data input lines of the even-byte bank.

Data output from the memory banks is routed through buffers A and C to their respective data paths. Both A and C will be enabled by the read strobe.

2.6.5 Sixteen Acknowledge(SIXTN*). Implementation of the sixteen acknowledge line allows the use of 8-bit memory boards in a 16-bit system without modification, but with a reduction in maximum system bandwidth.

If a 16-bit master requests a 16-bit transfer, but the addressed slave is not capable of such a transfer, the sixteen acknowledge line shall not be asserted.

The master shall respond in one of two ways, by generating an error trap or by conducting the transfer in byte-serial fashion. It is recommended that the master be capable of performing byte-serial transfers.

2.6.5.1 Byte-serial Response. If the sixteen acknowledge line is not activated after a specified period, circuitry should be included on bus masters to conduct the requested 16-bit transfer as two consecutive byte operations, thus assembling the requested 16-bit word while holding the master in a wait state.

For this process to occur, the sixteen acknowledge line must meet the timing specifications for the ready line inputs, as shown in Fig 6.

2.6.5.2 Error Response. If circuitry does not exist on the master to conduct the requested 16-bit transfer as two consecutive byte operations, an error condition shall result immediately, with ERROR* asserted.

2.7 Fundamental Bus-Cycle Timing

2.7.1 General. This section deals with the fundamental timing concepts involved in the standard bus cycle. Detailed specification of the timing parameters discussed in this section is given in 3.8 and 3.9.

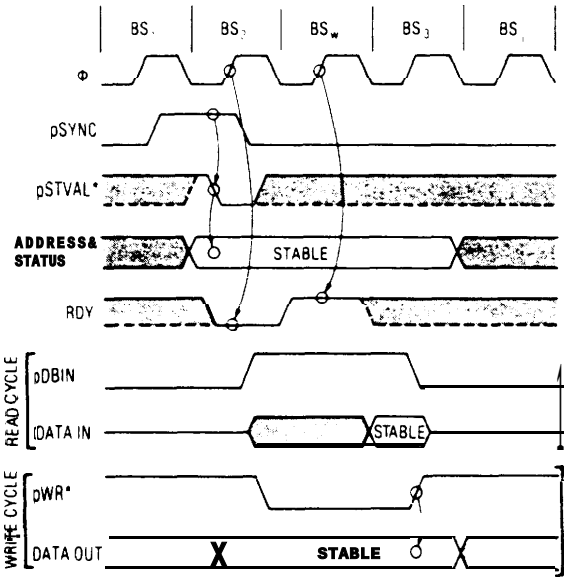


Fig 6
Bus-Cycle Fundamental Timing Relationships

The standard bus cycle is a pseudo-synchronous cycle, that is, the timing of the control signals bears a specified relationship to the master system clock ϕ.

All data transfers, including read or write cycles, 8- or 16-bit transfers, memory or input/output device transfers, and interrupt acknowledge are conducted on the bus as a standard bus cycle.

Figure 6 shows the fundamental timing for a standard bus cycle, with a single wait state inserted by the addressed slave.

2.7.2 Address and Status Buses. The beginning of a new bus cycle is indicated by the rising edge of the pSYNC signal, which closely follows the rising edge of the system clock, ϕ.

The address and status buses are changing to their values for the new cycle during the beginning of the pSYNC interval. Shortly after they can be guaranteed stable on the bus, the status valid strobe, pSTVAL*, is asserted. pSTVAL*, decoded in conjunction with pSYNC, indicates to all bus slaves that stable address and status may be sampled from the bus.

The position of the status valid strobe within the pSYNC interval is independent on the system clock, ϕ. This affords the designer of bus masters considerable flexibility in interfacing different processors to the bus. The status valid strobe should be positioned within the pSYNC

interval so that the delay between guaranteed status on the bus and the activation of the status valid strobe is as close to the minimum specifications as possible, thus maximizing memory and device access time.

In order to prevent false cycle starts in bus slaves, only one negative edge of the status valid strobe shall occur while pSYNC is asserted. If pSTVAL* is low at the rising edge of pSYNC, it must have been low for at least 30 ns.

Address and status information is thus stable on the bus from the negative transition of the status valid strobe during pSYNC, and is held stable until a specified period after the trailing edge of the data strobe (pDBIN in the read case, and pWR* in the write case). This hold time ensures that false decoding of the address and status information will not occur at the end of the bus cycle.

2.7.3 Ready and Sixteen Acknowledge Lines. The sixteen acknowledge line, since it may be used to place the bus master in a wait state while a requested 16-bit transfer is conducted in byte-serial fashion, is subject to the same timing constraints as the ready lines.

The ready lines are first sampled by the bus master on the rising edge of the system clock during the BS2 state, and if active, the master enters a wait state, sampling the ready line once every clock cycle on the rising edge of the system clock until the slave is ready for data transfer.

A minimum setup time before the rising edge of the system clock, and a minimum hold time after sampling must be met for the proper operation at the ready lines.

The time between the active edge of the status valid strobe and the sampling of the ready line may be very short. Hence, it is recommended practice not to make assertion of the ready line dependent on pSTVAL.*

Data output, address, status, and the read and write strobes are held stable during wait states.

2.7.4 Read Cycles

2.7.4.1 General. There are four types of read cycles: op-code fetch (M1), memory read, input, and interrupt acknowledge. These cycles are all similar with respect to timing, but make different use of the status bits and the address bus. See Tables 4 and 5.

2.7.4.2 The Read Strobe. The generalized read strobe pDBIN is used to gate data from an addressed slave onto the data bus during a read operation. The read strobe is asserted true by the bus master after a minimum specified time from the assertion of the status valid strobe. It is held true during any inserted wait states, and returns to the false state, returning the data bus to the high impedance state, shortly before the address and status buses are allowed to change.

2.7.5 Write Cycles

2.7.5.1 General. There are two possible types of write cycles on the bus, a memory-write cycle and an output cycle.

These two cycles are similar with respect to timing, but make different use of the status bits and address bus. A special write strobe, MWRT, is generated for memory cycles.

2.7.5.2 **The Write Strobe.** The generalized write strobe, pWR*, is used to write data from the data bus into the addressed bus slave. The write strobe is asserted true by the master after a minimum specified time from the assertion of the status valid strobe, pSTVAL*. It is held true during any inserted wait states and returns to the false state shortly before the address, status, and data buses are allowed to change.

Data out on the data bus shall be guaranteed valid for a specified period both before and after the activation of the write strobe. Hence, either the leading or the trailing edge of the write strobe may be used to strobe data into the addressed slave.

Address and status information must be held valid for a specified period of time from the trailing edge of the write strobe.

2.7.5.3 Memory-Write Strobe. While the generalized write strobe is activated for all write cycles, the memory-write strobe is activated for memory-write cycles only. The memory-write strobe is usually generated by front-panel devices, if they exist in the system, as a function of bus memory write or a front-panel deposit. Such front-panel deposit cycles do not comply with this standard. If front-panel devices do not exist the memory-write strobe must be generated somewhere in the system, but at only one point. This circuit should be designed so that it generates the memory-write strobe for all bus masters. Jumpers shall be provided to allow extra circuits to be disabled.

The memory-write strobe, MWRT, is defined as:

MWRT = pWR * - sOUT, (logic equation!

that is, memory write is true when pWR is true and sOUT is false.

The memory-write strobe must follow the pWR* strobe by not more than a specified period.

NOTE: The MWRT strobe shall be generated by the actual bus signals sOUT and pWR*, and not from internal board signals.

2.7.5.4 **Slave Abort.** It is permissable for the current bus master to suppress the assertion of the data strobes (pDBIN, pWR* and MWRT) so as to prevent a data transfer to a slave. This type of cycle is called the slave-abort cycle. If a master is capable of performing a slave-abort cycle, it shall ignore the state of the ready line (RDY) during the slave-abort cycle. This prevents slaves that never become ready (because a data strobe never occurred) from *hanging* the bus indefinitely.

2.8 **Special Bus** Operations

2.8.1 **General.** This section describes two special bus operations related to TMA operations, that is, the transfer of bus control from the permanent bus master to a temporary bus master for an arbitrary number of bus cycles, and the return of control to the permanent bus master.

These two operations are:

(1) The bus transfer protocol

(2) The arbitration protocol among simultaneous bus requesters

2.8.2 Bus Transfer Protocol

2.8.2.1 General. When a temporary bus master has been granted the bus by the permanent bus master, control must be transferred to the temporary master in such a way that spurious signals are not generated on the control output lines, causing false bus cycles. Since some of the control output signals are of positive polarity, extreme care must be taken in this operation. In general, the specified bus transfer protocol accomplishes this by having the permanent master and the temporary master drive the control output lines simultaneously at specified levels during the bus transfer.

2.8.2.2 Bus Transfer State Diagram. The bus transfer operation shall be implemented so as to conform to the bus transfer state diagram given in Fig 7.

2.8.2.3 Bus Transfer State Definitions

2.8.2.3.1 idle (IDLE). The idle state signifies that the temporary master is either involved in internal operations, and does not require the bus, or that it is waiting for the bus to become free so that it may assert its bus request.

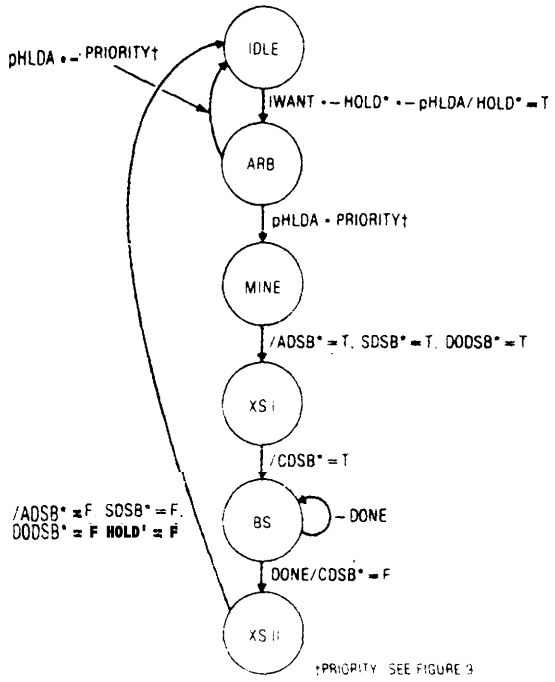


Fig 7
Bus-Transfer State Diagram

2.8.2.3.2 arbitration (ARB). If a temporary master desires the bus, and HOLD* is false and pHLDA is false, the temporary master enters the arbitration sequence, where it contests with other bus requesters for control of the bus.

Detailed specification of this process is given *in* 2.8.3.

2.8.2.3.3 **bus grant** (MINE). Priority assertions on the arbitration bus settle in the interval between the assertion of a hold request and a hold acknowledge. At the rising edge of the hold acknowledge signal the bus is granted to the highest priority requester, enabling the bus transfer operation for that requester.

If the bus is not granted to a requester, that requester returns to the idle state.

The bus grant state is termed MINE.

2.8.2.3.4 transfer state one (TSI). The bus transfer sequence begins with transfer state one, TSI. The bus transfer control circuit asserts the foliowing signals together:

(1) ADSB*

(2) SDSB*

(3) DODSB*

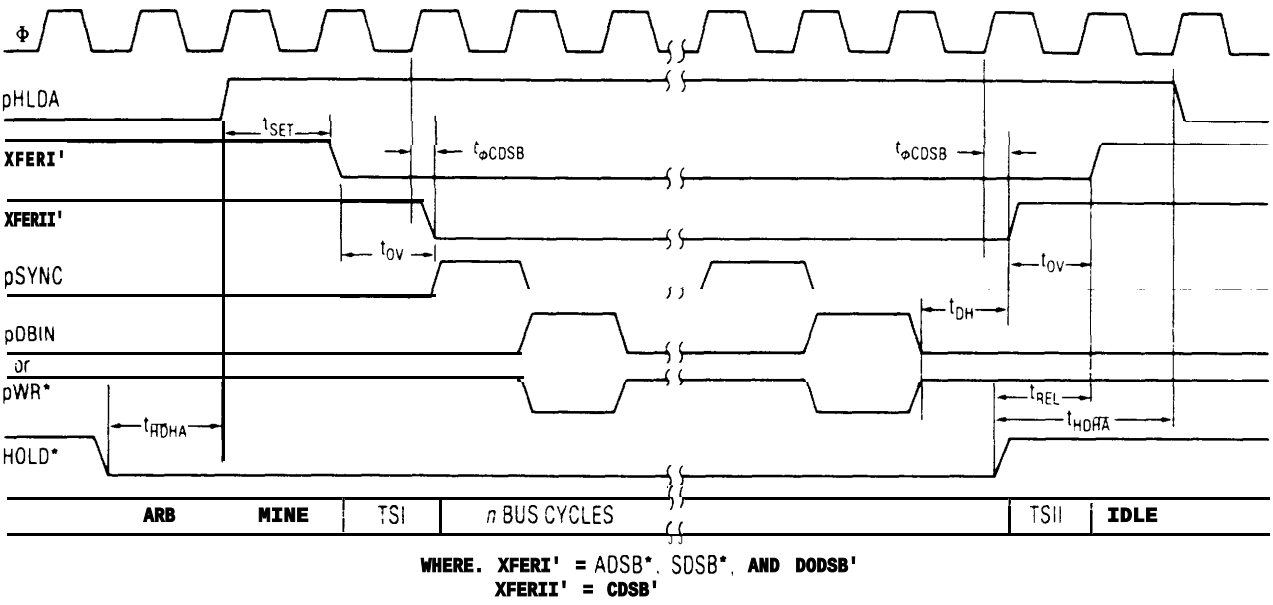


Fig 8
TMA Timing

disabling the address, status, and **data** output drivers of the permanent bus master and enabling the control output drivers of the temporary master. These lines are collectively **labelled** XFERI* in Fig 8. The permanent mas**ter and** the **temporary** master are now driving the control output lines. These lines are required to **have** the levels listed in Table 7.

The transfer state is terminated by the assertion (by the bus transfer control circuit) of the CDSB* line, called **XFERII*** in Fig 8, disabling the control drivers of the permanent master and enabling the address, status, and data out drivers of the temporary master. The temporary master now has complete control of the bus and begins its first bus cycle.

Table 7
Control Output Line Levels

Signal	Logic State	Electrical Level
(1) pSYNC	F	L
(2) pSTVAL*	F	H†
(3) pDBIN	F	L
(4) pWR*	F	H
(5) pHLDA	T	H

†See **NOTE in 2.4.4.1.**

2.8.2.3.5 **bus cycles** (BS). Any number of standard bus cycles are then conducted by the temporary bus master. Bus control is never transferred between cycles. When the temporary master is done, the **process** proceeds to TSII, transfer state two.

2.8.2.3.6 transfer state two (TSII). Transfer state two, TSII, is the mirror image of the sequence **in** TSI. **The state** begins with the release of the CDSB* and HOLD* signals, enabling the control-output drivers of the permanent master and disabling the address, status, and data-output drivers of the temporary master.

Both the temporary master and the permanent master drive the control-output lines for the remainder of TSII at the levels prescribed for TSI.

The state is ended by the release of other disable signals, enabling the address, status, and data-out drivers on the permanent master, and disabling the control-output drivers of the temporary master. The permanent master now has complete control of the bus and the temporary master returns to the idle state.

2 .8.2.4 Bus Transfer Timing Relationships

2.8.2.4.1 General. The fundamental timing relationships for a bus transfer and a single TMA bus cycle are given in Fig 8.

Relationship to the bus transfer states is

shown in boxes at the bottom of the figure.

Detailed specification of these times is given in 3.10 and Table 9.

2.8.2.4.2 **tSET.** A minimum time between the rising edge of the hold acknowledge signal and the assertion of the disable signals in TSI allows time for completion of the preceding bus cycle.

2.8.2.4.3 **tOV.** The time that the temporary master and the permanent master must drive the control-output signals has a specified minimum to ensure a smooth bus transfer.

Assertion of CDSB* is specified relative to the rising edge of the system clock, ϕ, so that the assertion of this signal may be used by the temporary master as a cycle start signal.

2.8.2.4.4 **tDH.** Hold time for data, addresses, and status from the end of the read or write strobes to the release of CDSB* and HOLD*.

2.8.2.4.5 **tREL.** The maximum time from the end of HOLD* to release of the DODSB*, SDSB* and ADSB* signals.

2.8.2.4.6 **tϕCDSB.** The maximum delay from the rising edge of the master bus clock ϕ to either the assertion or release of CDSB*.

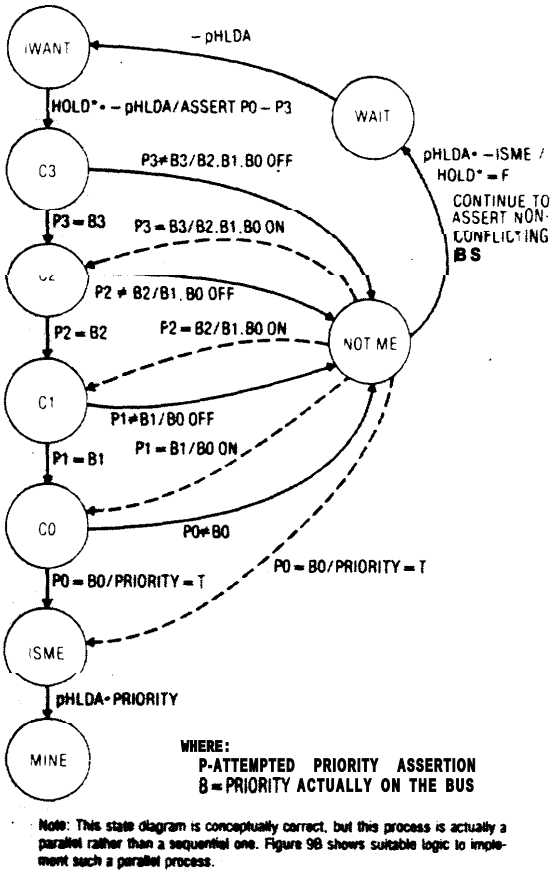
2.8.2.4.7 **tHDHA.** The minimum time from HOLD* being released by the temporary master until the permanent master can release pHLDA.

2.8.2.4.8 **tHDHA.** The minimum time between HOLD* being asserted by the temporary master and the permanent master asserting pHLDA.

2.8.3 Bus Arbitration Protocol. In a system which allows more than one master to use the system bus, for example a CPU permanent master and several temporary masters such as TMA controllers or muitiple CPUs, some means must be provided to determine which device will be allowed to control the bus at any given time.

The bus arbitration system uses four bus lines for arbitrating among 16 temporary masters. These lines are driven by open collector drivers, and are pulled high by pull-up resistors. Each temporary master has a unique priority number which it asserts on the arbitration but at an appropriate time. A higher binary number indicates a higher priority.

The temporary masters compare the priority appearing on the active-low open-collector bus with the priority they are asserting, starting with the most significant bit. If disagreement is detected by any temporary master at any given bit position, then another temporary master



NOTE: This state diagram is conceptually correct, but this process is actually a parallel rather than a sequential one. Figure 10 shows suitable logic to implement such a parallel process.

Fig 9
Bus-Arbitration Diagram

must be asserting that priority bit and thus must have a higher priority. In that case all less significant bits are removed by the detecting temporary master. All more significant bits agree, and thus need not be removed, and the bit which disagreed must have been a 0 and thus was not asserted. Leaving the agreeing bits asserted reduces system noise caused by the redistribution of driving currents in the bus, and speeds settling of the correct priority on the arbitration bus. This process is a continuous asynchronous parallel process, not a sequential bit-by-bit process as it may seem from the above description. Incorrect comparisons will occur and be removed as the bus lines settle for as long as four bus delays (not related to the choice of four bus lines) plus logic delays.

The four lines which comprise the arbitration bus are TMA0* through TMA3*, where TMA3* is the most significant bit. These iines.

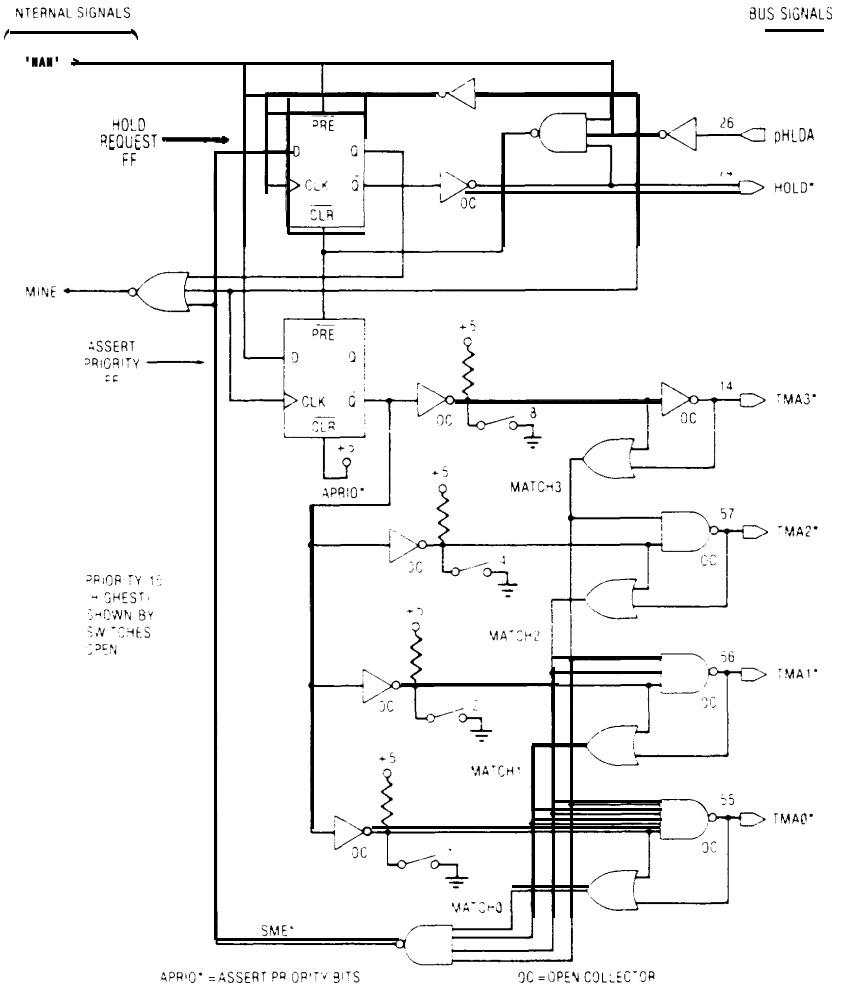


Fig 10
Bus-Arbitration Example

in conjunction with HOLD* and pHLDA, control the bus arbitration process.

2.8.3.1 Bus Arbitration Implementation. **An** implementation of the bus arbitration protocol is shown in Figs 9 and 10.

Any implementation shall obey the rules summarized in 2.8.4.

2.8.3.2 Bus Arbitration State Definitions.

2.8.3.2.1 IWANT (IWANT). The IWANT state is an internal state for a temporary master which has determined that a bus access is necessary and thus wishes to arbitrate for bus control.

Temporary masters may not assert their priorities nor remove them at arbitrary times, or the arbitration bus may be in transition when the result is needed. **A** temporary master may assert its priority and the HOLD* bus request only if

(1) pHLDA is not asserted (the permanent master has the bus), and

(2) HOLD* is not already asserted.

Furthermore, a master shall not assert HOLD* sooner than 30 ns after pHLDA goes false, and the permanent master shall not assert pHLDA sooner than a minimum specified delay after HOLD* is asserted. This guarantees that all

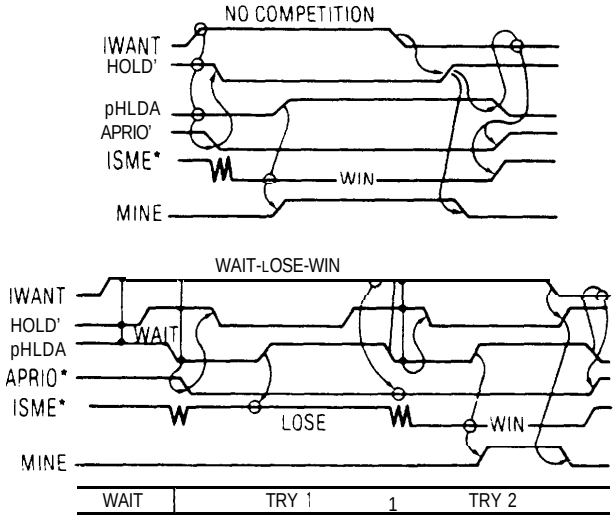


Fig 11
Bus-Arbitration Timing Diagrams

masters see the bus becoming available and allows ample time to settle the arbitration bus before the granting of the bus on the rising edge of pHLDA.

This scheme usually results in the first requester winning the bus. Only if simultaneous bus requests occur will the arbitration have any effect. This, however, is not improbable since multiple unsuccessful requesters will become synchronized by waiting for the failing edge of pHLDA.

2.8.3.2.2 priority comparison **states** (C3-CO). The priority comparison states, C3 through CO are the states where each requester compares the priority it is attempting to assert on the arbitration bus with the pricrity actually on the arbitration bus. Though C3 through CO are shown and described as sequential, they are actually parallel processes. While disagreement occurs at any bit position, less significant bits are removed from the arbitration bus. If no disagreement persists after the settling time the requester has the highest priority and will be granted the bus on the rising edge of pHLDA, proceeding to the state "MINE", where the bus transfer begins. All requesters continue to assert their priorities on the arbitration bus until the falling edge of pHLDA. Thus the priority number of the current bus master is available on the TMA bus while pHLDA is true. If the permanent master has the bus, pHLDA will be false.

A temporary master that wins the bus continues to assert its priority and HOLD* until its bus cycles are complete. A temporary master that loses the bus continues to assert its priority bits not turned off by the arbitration process, but must remove its assertion of the HOLD* line, so that the w-inner may indicate that it is finished by releasing HOLD*. A losing requester in this state is said to be in the "WAIT" state.

2.8.3.3 Bus Arbitration **Timing Relationships.** Figure 11 shows two possible cases of the bus arbitration procedure. The first of these is a case where the requester has no competition; it requests the bus and the bus is granted. The second case shows the requester waiting for the bus to be free, arbitrating for the bus and losing. and arbitrating for the bus and winning.

2.8.3.3.1 No Competition. When the temporary master determines **that** it requires the bus, it raises the internal signal IWANT. In this case, the rising edge of IWANT finds the **pHLDA** signal unasserted, meaning the permanent master has the bus, and the HOLD* signal unasserted, meaning that no other devices are requesting the bus. The temporary master may then assert the HOLD* signal and assert its priority on the arbitration bus. The ISME* signal is the result of the arbitration process, and is asserted if none of the bit-wise comparisons on the arbitration bus fail. This arbitration result is clocked by the rising edge of the pHLDA signal, creating the bus grant signal MINE.

When the temporary master is finished with the bus, the IWANT signal is released, releasing **the HOLD*** signal and resetting the bus grant signal, MINE. The permanent master releases the pHLDA signal, and all assertions are removed **from** the arbitration bus.

2.8.3.3.2 Wait-Lose-Win. In this example the requester raises its IWANT signal, but finds the bus already busy and must wait to assert its bus request and priority until the falling edge of pHLDA.

The requester arbitrates for the bus during try 1, but another requester has a higher priority and the arbitration result ISME* is false at the rising edge of pHLDA, indicating a loss in the arbitration process. The losing requester removes its assertion of the HOLD* signal, but continues to assert the nonconflicting high-**order** bits of its losing priority until the falling edge of pHLDA. At the falling edge of pHLDA, the process repeats, but this time results in a win for the requester.

2.8.4 Summary of Arbitration Protocol. Figures 9 and 10 represent an exmpie. not a required implementation. Any implementation which obeys the rules may be used. The rules which must be obeyed by **a** temporary master are:

(1) HOLD* shall be asserted only when it is not already asserted and pHLDA has been low for **at** least 30 ns.

(2) HOLD* shall be removed when pHLDA rises if another controller has asserted higher priority.

(3) HOLD* shall be removed when the controller no longer needs the bus.

(4) Prionty shall be asserted whenever HOLD* is asserted, and shall remain asserted until the next falling edge of pHLDA.

(5) The priority level shall be user-selectable and asserted by open-collector drivers on bus lines TMA3*-TMA0*.

(6) The most significant bit of the priority level (appearing on TMA3*) must be compared with the priority asserted. If the line is asserted low but not by this temporary master, all less significant priority bit assertions must be removed. Similarly, bits TMA2*, TMA1*, and TMA0* must be examined and possible less significant conflicting bits removed.

(7) If no lines are asserted low except those asserted by this temporary master after sufficient settling time, this temporary master has higher priority and may take the bus when pHLDA rises.

(8) Logic implementations shall be such that settling of the arbitration circuitry and bus will be completed between the assertion of HOLD* and the rise of pHLDA.

2.9 Interrupt Protocol. The purpose of an interrupt system is to allow peripheral devices to suspend the operation of a bus master in an orderly way and to request that the master service the requesting peripheral. When service is complete, the bus master returns to the operation from which it was interrupted.

The interrupt protocol is comprised of an 8-level vectored interrupt system and a nonmaskable interrupt. A complying master need only **be capable** of responding to INT* INT*, and the master's response to it, is the recommended way to request a TMA rearbitration. A temporary master which is capable of holding the bus indefinitely, shall be capable of responding to INT* in at least one of the following two ways:

(1) The temporary master shall be capable of relinquishing the bus in response to the assertion of INT*. This temporary master shall not participate in bus arbitration until INT* goes false.

(2) The temporary master shall be capable of servicing the interrupt request.

It is recommended that the permanent master ultimately service all interrupt requests.

2.9 .1 Vectored Interrupts

2.9.1.1 Vectored lnterrupt Requests. Eight levels of vectored interrupt requests are issued on the vectored interrupt lines. VI0* through VI7*, where VI0* is the most significant interrupt priority level. Vectored interrupt requests, however, may be rotated, masked individually,

or fenced out by the interrupt control slave, and hence the priority levels are not fixed. Requests on the VI lines should be asserted as levels; that is, they should be held active until service is received. **A slave** which asserts a VI line need take no further action to generate an interrupt. It is assumed that if interrupt acknowledge cycles occur, an interrupt controller somewhere in the system will respond appropriately .

The generalized interrupt request line, INT*, is implemented as **a** communication line between the interrupt controller and an interruptable master. Any **slave** or interrupt controller, using the INT* line, must respond appropriateiy to any interrupt acknowledge cycles. The interrupt controller (which may actually consist of multiple, nested, or intelligent interrupt **controllers)** shall **be capable of asserting** INT*, if a response is required by the bus master.

2.9.1.2 Interrupt Acknowledge. The interrupt acknowledge cycle is a standard bus-read cycle (except for status). The interrupt acknowledge cycle requests vectoring information from the interrupt controller to be asserted on the data bus during pDBIN.

Since no address information is asserted during an interrupt acknowledge cycle, if multiple interrupt controllers exist, they must either be ***daisy chained*** to avoid possible bus conflicts, or polled by the bus master.

2.9.2 Nonmaskable Interrupt (NMI*). The nonmaskable interrupt is an optional control input to **bus** masters. This interrupt is not maskable by a software instruction, and takes priority over other interrupt requests. The NMI* line may be used in the impiementation of the special condition lines, ERROR* and PWRFAIL*.

NMI* is an open-collector line. The bus master shall respond to negative going transitions on the NMI* line.

2.10 Special Condition Lines. Two special condition lines, PWRFAIL* and ERROR*, are available on the bus. Their use is optional.

2.10.1 Power-Fail Pending (PWRFAIL*). This line indicates an impending system power failure. It is specified that this line shall be activated at least 16 μs before the local voltage regulators drift out of specification.

The line stays low for at least 16 μs and its rising edge shall cause POC* to be asserted (which will cause RESET* and SLAVE CLR to be asserted). The circuit driving this line shall meet the electrical specifications for an open-collector line.

2.10.2 ERROR*. This is a generalized error line that indicates that the current bus operation is producing an error of some sort that is memory parity error, write to protected memory, inability to accommodate 8-bit slaves, etc.)

The ERROR* line should be implemented as **a trap. All relevant information about** the **error-causing cycle- address, data, status, device number** (for temporary master) — should be latched on the falling edge of ERROR*. The falling edge of error will generally cause an NMI*.

ERROR* is implemented as an open-collector line.

3. Electrical Specifications

3.1 Application. The electrical specifications for interface devices to be used in IEEE Std 696 bus systems are defined in this section. Proper operation of these devices also depends on two other fac tors :

(1) Short physical distance between devices
(2) Relatively low electrical noise

The electrical specifications for the bus driver and receiver circuits do not imply a particular technology, unless otherwise noted.

All specifications apply over the temperature range 0 to 70 °C.

3.2 Power Distribution. Power in IEEE Std 696 systems is distributed as unregulated dc power at three voltages, +8 V, +16 V, and - 16 V. Because these voltages are on adjacent lines it is relatively easy to short these lines on card removal. Therefore, bleeder resistors or other constant loads sufficient to discharge all three supplies rapidly are recommended.

3.2.1 + 8 V Specification. Instantaneous minimum must be greater than +7 V, instantaneous maximum less than 25 V, and average maximum less than 11 V.

3.2.2 + 16 V Specification. Instantaneous minimum must be greater than +14.5 V, instantaneous maximum less than 35 V, and average maximum less than 21.5 V.

3.2.3 -16 V Specification. Instantaneous maximum must be less than -14.5 V, instan-

taneous minimum greater than -35 V, and average minimum greater than - 21.5 V.

3.3 General Signal Discipline. Other than the power lines noted above, all signals on the bus are limited to positive signal levels between 0 V and +5 V, and may not have loaded rise or fall times less than 5 ns.

3.4 Driver Requirements

3.4.1 Driver Types. Three types of bus drivers are defined :

(1) An active driver, either in the high state or in the low state or in transition, which has the capability to accept current in the low state and to provide current in the high state.

(2) An open-collector driver, which will not accept or provide current in the high state. A 360 Ω ± 5% pull-up resistor to +5 V or equivalent must be provided somewhere in the system for open-collector lines. It is recommended that these pull-up resistors be provided on the bus. However, implementation on the permanent master is also acceptable.

(3) A three-state driver, which has the capability to be in the high-impedance state as well as in the high and low states.

3.4.2 Driver Specifications. Specifications for bus drivers shall be as follows :

Low state (Vol): Output voltage less than or equal to + 0.5 V at 24 mA sink current.

High state (Voh): Output voltage (for active and three-state drivers) greater than or equal to +2.4 V at 2 mA.

The leakage current for three-state drivers in the high-impedance state is specified as not greater than ±25 μA.

The internal capacitive load of a driver shall not exceed 15 pF at 25 °C whether in the active or the high-impedance state.

The rise and fall times of bus drivers should be minimized, subject to 3.3. In no case should the rise or fall times exceed 50 ns at rated capacitive load.

3.5 Receiver Specifications. The specifications for receivers on the bus shall be as follows:

Low state: A voltage less than or equal to +0.8 V shall be recognized as a low state.

High state: A voltage greater than or equal to +2.0 V shall be recognized as a high state.

Bus receivers shall have diode clamp circuits to prevent excessive negative voltage excursions.

Additional noise immunity is afforded by the use of Schmitt-type receiver circuits. Recommended hysteresis for such receivers should be ⩾0.4 v.

3.6 Bidirectional Signals. Some interface signals, such as the data bus, are combined three-state drivers and receivers. For each function these devices shall meet the same specifications as separate drivers and receivers.

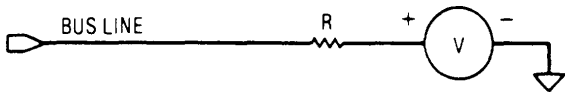
The total internal capacitive load for a line transceiver shall not exceed 20 pF at 25 °C.

3.7 Card-Level Bus Loading. At the card level, the following specifications apply:

(1) The total capacitive load on any bus input shall not exceed 25 pF.

(2) A card may not source more than 0.5 mA at 0.5 V nor sink more than 80 μA at 2.4 V on any signal line except for TMA0*, TMA1*, TMA2*, TMA3*, PHANTOM*, and PWRFAIL*. On these lines a card may not source more than 0.4 mA at 0.5 V.

3.7.1 Bus Termination. All bus lines except the power and ground lines may be terminated to reduce bus noise using a circuit equivalent to



where

V = 2.6 V ±0.2 v

R is no less than 180 Ω (±5%)

It is recommended that open-collector lines not be terminated as above, but rather pulled up to +5 V as stated in 3.4.1.

3.8 Read-Cycle Timing Specification. Figure 12 depicts the readcycle timing waveforms with the pertinent timing parameters shown. Table 8 specifies these parameters.

3.9 Write-Cycle Timing Specification. Figure 13 depicts the write-cycle timing waveforms with the pertinent timing parameters shown. Table 8 specifies these parameters.

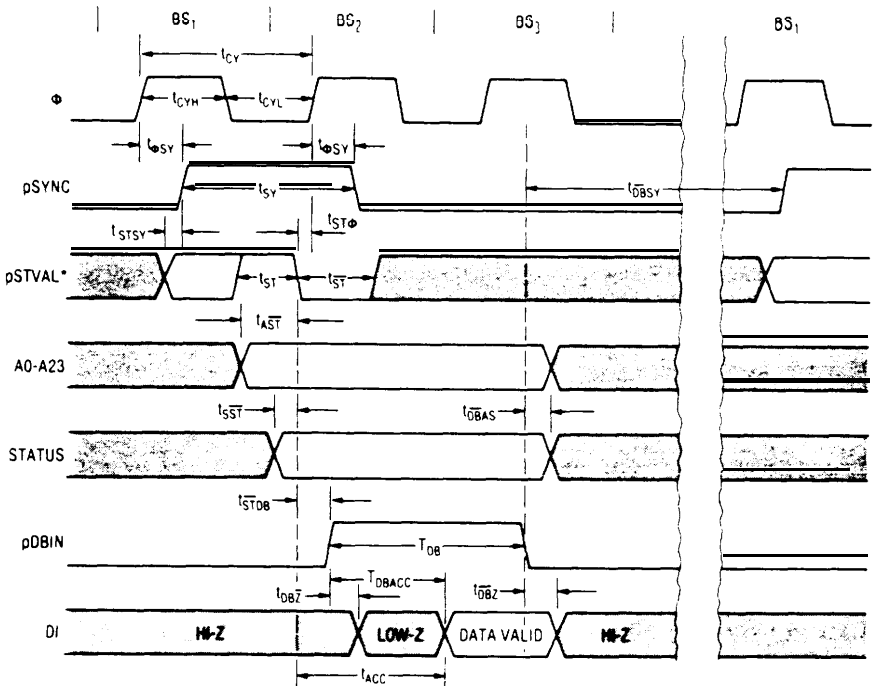


Fig 12
Read-Cycle Timing Diagram

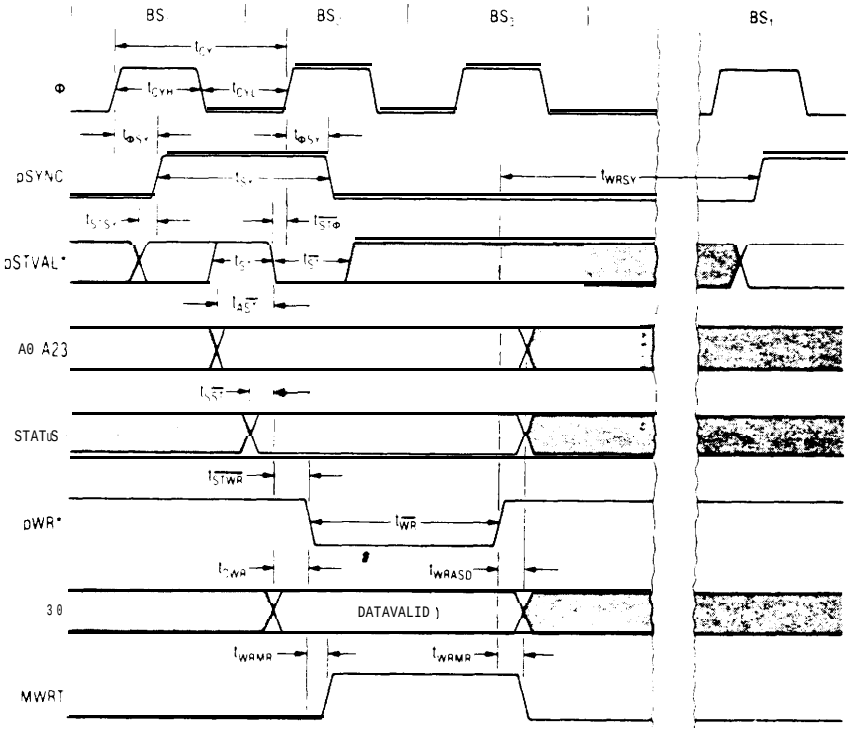


Fig 13
Write-Cycle Timing Diagram

Table 8
Read and Write Cycle Timing Parameters

		MIN (ns)	MAX (ns)
tCY	ϕ PERIOD	166	2000
tCYH	ϕ PULSE WIDTH HIGH	0.4tCY	
tCYL	ϕ PULSE WIDTH LOW	0.4tCY	
tϕSY	DELAY ϕ HIGH TO pSYNC HIGH; DELAY ϕ HIGH TO pSYNC LOW	10	0.4tCY
tSY	pSYNC PULSE WIDTH HIGH	0.7tCY	
t$\overline{ST}\phi$	pSTVAL* LOW PRIOR TO ϕ HIGH DURING pSYNC	0	
tST	pSTVAL* PULSE WIDTH HIGH	50	
t$\overline{ST}$	pSTVAL* PULSE WIDTH LOW	50	
tA$\overline{ST}$	ADDRESSES STABLE PRIOR TO pSTVAL* LOW DURING pSYNC HIGH	70	
tS$\overline{ST}$	STATUS STABLE PRIOR TO pSTVAL* LOW DURING pSYNC HIGH	40	
tDB	pDBIN PULSE WIDTH HIGH	0.9tCY	
t$\overline{ST}$DB	DELAY pSTVAL* LOW TO pDBIN HIGH	20	
tD$\overline{BS}$Y	DELAY pDBIN LOW TO pSYNC HIGH	0	
t$\overline{DB}$AS	HOLD TIME FOR ADDRESSES AND STATUS AFTER pDBIN LOW	50	
t$\overline{DB}$Z	DELAY pDBIN LOW TO SLAVE DI DRIVERS Hi-Z		70
tDBZ	DELAY pDBIN HIGH TO SLAVE DI DRIVERS ACTIVE	10	70
tACC	DELAY pSTVAL* LOW TO DATA VALID	SPECIFIED BY MANUFACTURER WORST CASE MAXIMUM FOR ALL SLAVES AND WORST CASE MINIMUM FOR ALL MASTERS	
t$\overline{WR}$	pWR* PULSE WIDTH LOW	0.9tCY	
t$\overline{ST}$WR	DELAY pSTVAL* LOW TO pWR* LOW	30	
tWRSY	DELAY pWR* HIGH TO pSYNC HIGH	0	
tD$\overline{WR}$	SETUP TIME DO VALID TO pWR* LOW	0.1tCY	
tWRASD	HOLD TIME ADDRESSES, STATUS, AND DO FROM pWR* HIGH	0.2tCY	
tWRMR	DELAY pWR* LOW TO MWRT HIGH; DELAY pWR* HIGH TO MWRT LOW		30
tRDYϕ	SETUP TIME RDY, XRDY, SIXTN* TO ϕ RISING	70	
tϕRDY	HOLD TIME RDY, XRDY, SIXTN* AFTER ϕ RISING	20	
tPOV	OVERLAP OF PHANTOM* AND pDBIN OR pWR*	30	
tSY$\overline{ST}$	DELAY FROM pSYNC HIGH TO pSTVAL* LOW	30	
tAϕ	ADDRESSES STABLE PRIOR TO ϕ HIGH DURING pSYNC HIGH	80	
t$\overline{ST}\phi$	STATUS STABLE PRIOR TO ϕ HIGH DURING pSYNC HIGH	50	

Table 9
Bus Transfer Timing Parameters

		MIN	MAX
tSET	DELAY pHLDA TO ADSB*, SDSB*, DODSB* LOW	0	
tOV	TIME BOTH TEMPORARY AND PERMANENT MASTER DRIVE THE CONTROL OUTPUT LINES	0.4TCY	
tDH	HOLD TIME ADDRESS, STATUS, AND DATA OUT FROM END OF STROBE TO CDSB* RISING	0.2tCY	
tREL	DELAY FROM HOLD* RISING TO ADSB*, SDSB* AND DODSB* HIGH		1.0tCY
$tHDH\overline{A}$	DELAY FROM HOLD* FALSE TO pHLDA FALSE	1.0tCY	
tφCDSB	DELAY FROM φ RISING TO CDSB* LOW DELAY FROM φ RISING TO CDSB* HIGH	0	0.3tCY
tHDHA	DELAY FROM HOLD* FALLING TO pHLDA RISING	1.0tCY	

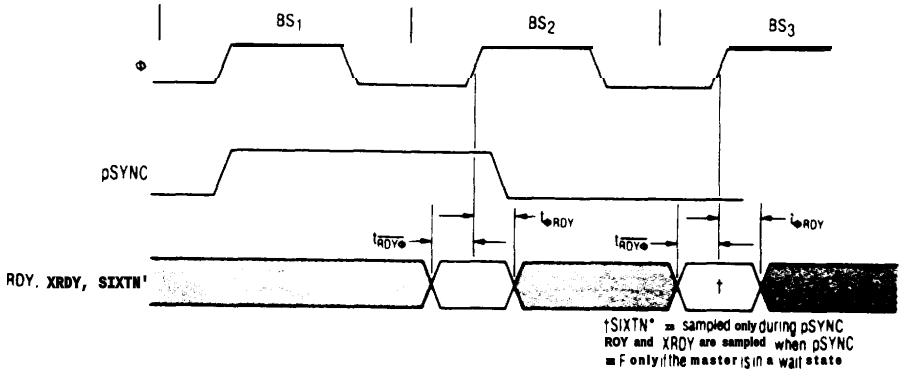


Fig 14
Timing of RDY, XRDY, and SIXTN'
During Read and Write Cycles

3.10 Ready and Sixteen Request Timing Specification. Figure 14 depicts RDY, XRDY, and SIXTN* timing waveforms during read and write cycles, with pertinent timing parameters shown. Table 8 specifies these parameters.

3.11 Bus Transfer Timing Specification. Figure 8 depicts bus-transfer timing waveforms with the pertinent timing parameters shown. Table 9 specifies these parameters.

3.12 PHANTOM* Timing Specification. Figure 15 shows the overiap of the PHANTOM* signal with respect to the read and write strobes. Table 8 specifies these parameters.

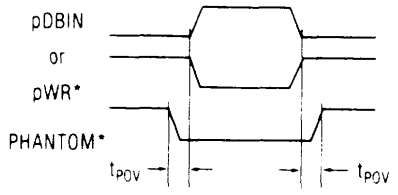


Fig 15
Overlap of PHANTOM* and
Read and Write Strobes

4. Mechanical Specifications

4.1 Application. The mechanical specifications for standard interface systems are described in this section.

4.2 **Connector Type.** The card edge connector is a 100-pin (dual 50) connector with contacts spaced on 0.125 in centers. It is nominally designed for printed circuit boards 0.062 in thick.

The connector is subject to the specifications in 4.2.1 and 4.2.2.

4.2.1 Electrical Considerations.

(1) ***Voltage rating: 200*V dc, minimum** pin to pin

***(2) Current rating: 2.5*A per contact**

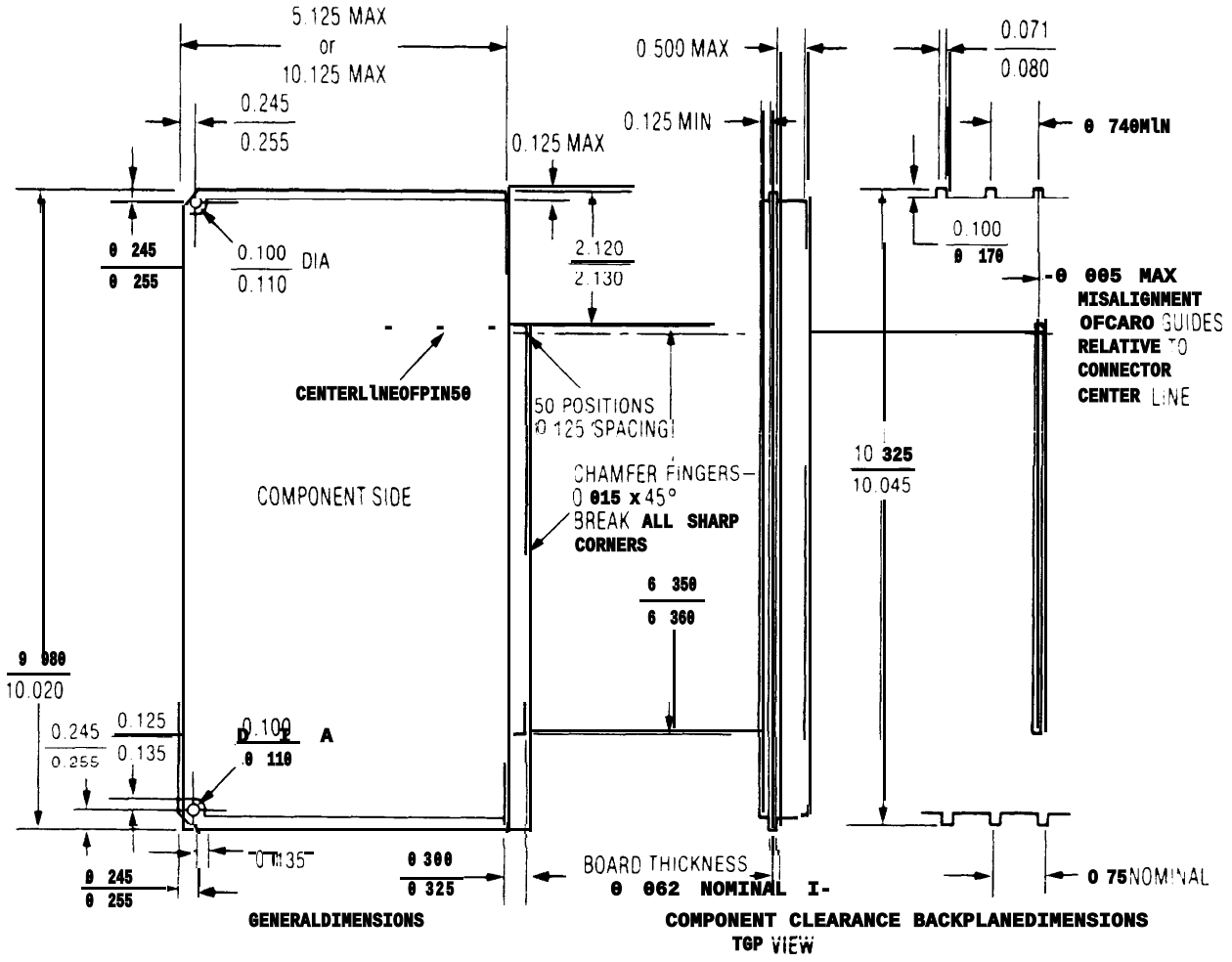
***(3) Contact resistance: 50*MΩ maximum at rated current after 100 insertions**

***(4) Insulation resistance:*1000 MΩ minimum**

4.2.2 **Connector Spacing.** Connectors should **be spaced 0.75** inches ± 0.01 inches center to center.

4.3 Board Size **Specification.** Circuit boards shall conform to the board **size specifications given in** Fig 16, with the **exception of the**

Fig 16
IEEE Std 696 Board
Mechanical Parameters



board height specifications. Boards may optionally be 10 in high. Manufacturers of double height boards should specify that a board is a double height board in all descriptive literature and advertisements. The edge connector pin shown in the figure is pin 50. Pin 100 opposes pin 50 on the back side of the board.

Total board depth shall not exceed 0.65 in.

Nominal board thickness is 0.062 in.

The unshaded areas shown shall be free of all components, connectors, or other protrusions. These areas may contain circuit traces. The holes in the corner of the board are to allow the use of card extractors. Connectors shall not extend more than 0.5 inches above the top of the board.

5. Quick Reference
IEEE Std 696 Bus Layout

pin 1	+8 V (B)	
pin 2	+16 V (B)	
pin 3	XRDY (S)	H
pin 4	VI0* (S)	L
pin 5	VI1* (S)	L
pin6	VI2* (S)	L
pin 7	VI3* (S)	L
pin8	VI4* (S)	L
pin9	VI5* (S)	L
pin 10	VI6* (S)	L
pin 11	VI7* (S)	L
pin 12	NMI* (S)	L
pin 13	PWRFAIL* (B)	L
pin 14	TMA3* (M)	L
pin 15	Al8 (M)	H
pin 16	Al6 (M)	H
pin 17	Al7 (M)	H
pin 18	SDSB* (M)	L
pin 19	CDSB* (M)	L
pin 20	o v	
pin 21	NDEF	
pin 22	ADSB* (M)	L
pin 23	DODSB* (M)	L
pin 24	ϕ (B)	H
pin 25	pSTVAL* (M)	L
pin 26	pHLDA (M)	H
pin 27	RFU	
pin 28	RFU	
pin 29	A5 (M)	H
pin 30	A4 (M)	H
pin 31	A3 (M)	H
pin 32	Al5 (M)	H
pin 33	Al2 (M)	H
pin 34	A9 (M)	H
pin 35	DO1 (M)/ED1 (M/S)	H
pin 36	DO0 (M)/EDO (M/S)	H
pin 37	A10 (M)	H
pin 38	DO4 (M)/ED4 (M/S)	H
pin 39	DO5 (M)/ED5 (M/S)	H
pin 40	DO6 (M)/ED6 (M/S)	H
pin 41	D12 (S)/OD2 (M/S)	H
pin 42	D13 (S)/OD3 (M/S)	H
pin 43	D17 (S)/OD7 (M/S)	H
pin 44	sM1 (M)	H
pin 45	sOUT (M)	H
pin 46	sINP (M)	H
pin 47	sMEMR (M)	H
pin 48	sHLTA (M)	H
pin 49	CLOCK (B)	
pin 50	o v	
pin 51	+8 V (B)	
pin 52	-16 V (B)	
pin 53	0 volts	
pin 54	SLAVE CLR* (B)	L
pin 55	TMA0* (M)	L
pin 56	TMA1* (M)	L
pin 57	TMA2* (M)	L
pin 58	sXTRQ* (M)	L
pin 59	A19	H
pin 60	SIXTN* (S)	L
pin 61	A20 (M)	H
pin 62	A21 (M)	H
pin 63	A22 (M)	H
pin 64	A23 (M)	H
pin 65	NDEF	
pin 66	NDEF	
pin 67	PHANTOM* (M/S)	L
pin 68	MWRT (B)	H
pin 69	RFU	
pin 70	o v	
pin 71	RFU	
pin 72	RDY (S)	H
pin 73	INT* (S)	L
pin 74	HOLD* (M)	L
pin 75	RESET* (B)	L
pin 76	pSYNC (M)	H
pin 77	pWR* (M)	L
pin 78	pDBIN (M)	H
pin 79	A0 (M)	H
pin 80	A1 (M)	H
pin 81	A2 (M)	H
pin 82	A6 (M)	H
pin 83	A7 (M)	H
pin 84	A8 (M)	H
pin 85	Al3 (M)	H
pin 86	Al4 (M)	H
pin 87	All (M)	H
pin 88	DO2 (M)/ED2 (M/S)	H
pin 89	DO3 (M)/ED3 (M/S)	H
pin 90	DO7 (M)/ED7 (M/S)	H
pin 91	D14 (S)/OD4 (M/S)	H
pin 92	D15 (S)/OD5 (M/S)	H
pin 93	D16 (S)/OD6 (M/S)	H
pin 94	DI1 (S)/OD1 (M/S)	H
pin 95	DIO (S)/OD1 (M/S)	H
pin 96	sINTA (M)	H
pin 97	sWO* (M)	L
pin 98	ERROR* (S)	L
pin 99	POC* (B)	
pin 100	o v	