

An American National Standard

**IEEE Standard
696 Interface Devices**

Sponsor

Standards Committee of the
IEEE Computer Society

Approved June 10, 1982

IEEE Standards Board

Approved September 8, 1983

American National Standards Institute

© Copyright 1983 by

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

*No part of this publication may be reproduced in any form,
in an electronic retrieval system or otherwise,
without the prior written permission of the publisher.*

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE which have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least once every five years for revision or reaffirmation. When a document is more than five years old, and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason IEEE and the members of its technical committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE Standards Board
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 A0=1 appears on the OD7-0 lines during a 16-bit transfer. Byte data that is written or read with A0=0 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 A0=1 is odd, and any address with A0=0 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 pre-existing 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
Kells Elmquist
Tim Erickson
Gary Feirbach
Kevin Fischer

Howard Fullmer*
David Gustavson
Richard Jones
Chuck Krause

Sol Libes
Dick Lowe
Ed Lupin
George Morrow*

Don Pannell
William Stark
Bob Stewart
Michael Stolowitz
John Terry

*Past Chairman

When the IEEE Standards Board approved this standard on June 10, 1982, it had the following membership:

I. N. Howell, Jr., *Chairman*

Edward Chelotti, *Vice Chairman*

Sava I. Sherr, *Secretary*

C. Y. R. Allen
J. J. Archambault
James H. Beall
John T. Boettger
J. V. Bonucchi
Edward J. Cohen
Len S. Corey

J. V. Fleckenstein
Jay Forster
Kurt Greene
Joseph L. Koepfinger
Irving Kolodny
John E. May
Donald T. Michael*

A. R. Parsons
J. P. Riganati
Frank L. Rose
Robert W. Seelbach
Jay A. Stewart
Clifford O. Swanson
Robert E. Weiler

*Member emeritus

Contents

SECTION	PAGE
1. General	8
1.1 Scope	8
1.2 Object	8
1.3 Definitions	8
1.3.1 General-System Terms	8
1.3.2 Signals and Paths	8
1.4 State Diagram Notation	9
1.5 Logical and Electrical State Relationships	10
1.6 Interface System Overview	11
1.6.1 Interface System Objective	11
1.6.2 Fundamental Communication Capabilities	11
1.6.3 Message Paths and Bus Structure	11
2. Functional Specifications	11
2.1 Functional Partition	11
2.2 Signal Lines	12
2.2.1 General	12
2.2.2 Address Bus	12
2.2.3 Status Bus	13
2.2.4 Data Bus	14
2.2.5 Control Output Bus	14
2.2.6 Control Input Bus	14
2.2.7 TMA Control Bus	18
2.2.8 Vectored Interrupt Bus	18
2.2.9 System Utilities	18
2.2.10 Pin List	19
2.3 The Permanent Master Interface	19
2.3.1 General	19
2.3.2 Permanent Master State Diagram	20
2.3.3 Permanent Master State Descriptions	20
2.3.4 Required Signals for Permanent Masters	21
2.3.5 Dummy Mastering	21
2.4 The Temporary Master Interface	21
2.4.1 General	21
2.4.2 Temporary Master State Diagram	21
2.4.3 Temporary Master State Descriptions	21
2.4.4 Required Signals For Temporary Masters	22
2.5 The Slave Interface	22
2.5.1 Slave-Interface State Diagram	22
2.5.2 Slave-State Definitions	22
2.5.3 Required Signals for Slave Interfaces	23
2.6 8/16-bit Data Transfer Control	23
2.6.1 General	23
2.6.2 8-bit Data Paths	23
2.6.3 16-bit Data Paths	23
2.6.4 Memory Organization	23
2.6.5 Sixteen Acknowledge (SIXTN*)	25
2.7 Fundamental Bus-Cycle Timing	25
2.7.1 General	25
2.7.2 Address and Status Buses	25
2.7.3 Ready and Sixteen Acknowledge Lines	26
2.7.4 Read Cycles	26
2.7.5 Write Cycles	26

SECTION	PAGE
2.8 Special Bus Operations	27
2.8.1 General	27
2.8.2 Bus Transfer Protocol	27
2.8.3 Bus Arbitration Protocol	29
2.8.4 Summary of Arbitration Protocol	32
2.9 Interrupt Protocol	32
2.9.1 Vectored Interrupts	32
2.9.2 Nonmaskable Interrupt (NMI*)	33
2.10 Special Condition Lines	33
2.10.1 Power-fail Pending (PWRFAIL*)	33
2.10.2 ERROR*	33
3. Electrical Specifications	33
3.1 Application	33
3.2 Power Distribution	33
3.2.1 +8 V Specification	33
3.2.2 +16 V Specification	33
3.2.3 -16 V Specification	33
3.3 General Signal Discipline	34
3.4 Driver Requirements	34
3.4.1 Driver Types	34
3.4.2 Driver Specifications	34
3.5 Receiver Specifications	34
3.6 Bidirectional Signals	34
3.7 Card-Level Bus Loading	34
3.8 Read Cycle Timing Specification	34
3.9 Write-Cycle Timing Specification	34
3.10 Ready and Sixteen Request Timing Specification	37
3.11 Bus Transfer Timing Specification	37
3.12 PHANTOM* Timing Specification	37
4. Mechanical Specifications	38
4.1 Application	38
4.2 Connector Type	38
4.2.1 Electrical Considerations	38
4.2.2 Connector Spacing	38
4.3 Board Size Specification	38
5. Quick Reference IEEE Std 696 Bus Layout	40

FIGURES

Fig 1 Permanent Master State Diagram	20
Fig 2 Temporary Master State Diagram	21
Fig 3 Slave Interface State Diagram	22
Fig 4 8/16 Bit-Memory Organization	24
Fig 5 8/16 Bit Address + Data Usage	24
Fig 6 Bus-Cycle Fundamental Timing Relationships	25
Fig 7 Bus-Transfer State Diagram	27
Fig 8 TMA Timing	28
Fig 9 Bus Arbitration Diagram	29
Fig 10 Bus Arbitration Example	30

	PAGE
Fig 11 Bus Arbitration Timing Diagrams	31
Fig 12 Read-Cycle Timing Diagram	35
Fig 13 Write-Cycle Timing Diagram	35
Fig 14 Timing of RDY, XRDY, and SIXTN* During Read and Write Cycles	37
Fig 15 Overlap of PHANTOM* and Read and Write Strokes	37
Fig 16 IEEE Std 696 Board Mechanical Parameters	38

TABLES

Table 1 Active High Signals	10
Table 2 Active Low Signals	10
Table 3 Bus Structures	11
Table 4 Address Usage for Different Bus Cycles	12
Table 5 Status Usage Chart	13
Table 6 IEEE Std 696 Bus Pin List	15
Table 7 Control Output Line Levels	28
Table 8 Read and Write Cycle Timing Parameters	36
Table 9 Bus Transfer Timing Parameters	37

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) A 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 compatible 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 relatively low cost, and allows the interconnection of low-cost devices.
- (7) To define a system that is easy to use.

1.3 Definitions. The following 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 functional, 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 $1\,048\,576 = 2^{20}$

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.

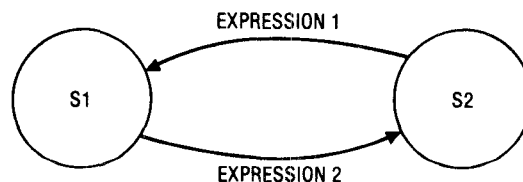
signal parameter. That parameter of an electrical quantity whose values or sequence of values convey information.

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 (/). 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 $A \text{ AND } (B \text{ OR } C)$ 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

$$MWRT = (-pWR*) * -sOUT, \text{ (electrical equation)}$$

since

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

Note that this is equivalent to

$$MWRT = -(pWR* + sOUT), \text{ (electrical equation)}$$

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 $\leq 0.8 \text{ V}$, CALLED THE LOW STATE.	L
TRUE (T)	1	CORRESPONDS TO $\geq 2.0 \text{ V}$, 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 $\geq 2.0 \text{ V}$, CALLED THE HIGH STATE.	H
TRUE (T)	1	CORRESPONDS TO $\leq 0.8 \text{ V}$, 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 Communication 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 signal lines
(4) Control output bus	5 signal lines
(5) Control input bus	6 signal lines
(6) TMA control bus	8 signal lines
(7) Vectored 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 A15, but may assert lines A16 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 A15 through A0, where A15 is the most significant bit.

Table 4
Address Usage for Different Bus Cycles

Cycle Type	Standard Addressing	Extended Addressing
MEMORY READ		
MEMORY WRITE	A15-A0	A23-A0
M1 (OP-CODE FETCH)		
INPUT		
OUTPUT	A7-A0†	A15-A0
INTERRUPT ACKNOWLEDGE	NONE	NONE
HALF ACKNOWLEDGE	NONE	NONE

†See 2.2.2.4