This proposed standard eliminates many of the problems in the S-100 bus and upgrades it for 16-bit microprocessors. It is offered here for public comment before submission to the IEEE Standards Board.

Standard Specification for S-100 Bus Interface Devices

(DRAFT)

IEEE Task 696.1 D2
Kells A. Elmquist, InterSystems Inc.
Howard Fullmer, Parasitic Engineering Inc.
David B. Gustavson, Stanford Linear Accelerator Center
George Morrow, Thinker Toys

Introductory comments by
Robert G. Stewart, Chairman,
IEEE-CS Computer Standards Committee

The following draft of a proposed standard for the S-100 bus is the culmination of over a year and a half of effort to eliminate many of the bus's problems and to upgrade it to be suitable for 16-bit microprocessors. The address bus has been extended to 24 bits, the data in and data out buses ganged to form a 16-bit wide data bus for 16-bit transactions, and two additional handshaking lines added to permit intermixing of 8- and 16-bit memory cards.

A binary encoded multiple master arbitration bus permits up to 16 masters on the bus. The necessary logic can be implemented in one chip. Additional ground lines, a power fail line, and an error line have been added. Three lines termed NDEF-for not to be defined-have been allotted to allow leeway to implementers for specialized use. Such use must be specified in all literature. Five lines are RFU-reserved for future use. Some lines formerly used for front panel purposes have been deleted, with the intention that such lines can best be handled by a jumper cable from the CPU card to the front panel. A DMA protocol is specified which provides overlap of the control lines at the beginning and end of the transition between permanent and temporary masters. This allows the address, data, and control buses to settle before information is transferred.

As a bit of personal testimony, I implemented the new DMA protocol on my own system, which includes a Digital Systems dual floppy disk interfaced to a MITS Altair 8800, using DMA for disk transfers.

The soft error rate, presumably due to glitching on the positive true logic lines, dropped from a situation where a file would be seriously munged in a few hours to the present situation where I can work for days on end without an observable error.

We have observed a new typographic convention in publishing the proposed standard. The use of an overbar to denote electrically low active or negative true logic lines has been replaced by a postfix asterisk to avoid confusion with Boolean negation and permit typing on word processing systems. This is verbalized by the word "star," replacing the prior word "bar." The Boolean negation overbar can be optionally replaced by a prefix minus sign, with parentheses if needed.

The named authors of the standard were evenly divided as to whether the asterisk should be included in logic equations and state diagrams as well as in electrical signal names and timing diagrams. Two authors believe that the asterisk, when thought of as a designator rather than as the negation operator, adds clarity and consistency, and lessens the need to remember or look up the electrically active level when converting from logic to electrical representations. Such use makes logic state diagrams more directly useful for interpreting oscilloscope or logic analyzer waveforms.

The other two authors feel that the inclusion of the asterisk in the name of a logic state or variable is likely to carry with it the implication of logical negation, thus causing the logic statements to be interpreted incorrectly. Furthermore, they assert that many designers think mostly in terms of electrical levels, with high being true, which again causes logic statements to be interpreted incorrectly. They propose to resolve this hazard by removing the electrical information, i.e., the asterisk, from the variable name when it is used in a logic context as opposed to an electrical or timing context.

A compromise has been reached where the asterisk is not used in the context of logic equations, but is included elsewhere in the document. We solicit feedback from the readers on these two points of view.

The S-100 bus subcommittee has been ably chaired by George Morrow and Howard Fullmer. Both of them provided invaluable technical insights which have been incorporated throughout the draft standard. John Walker of Marinchip Systems suggested the method of using 16-bit memory and interface cards interchangeably with 8-bit cards. David Gustavson and Leo Paffrath of SLAC suggested the bus arbitration scheme which has also been implemented on the Department of Energy's Fastbus. Howard Fullmer suggested the DMA overlap protocol which lowers glitching noise. Kells Elmquist of InterSystems offered a critique of the draft published in May 1978 in Computer and provided many useful suggestions for improvement. He carefully investigated numerous timing and electrical alternatives and resolved many open questions relating to the standard. Kells wrote the final version of the draft for submission to and revision by the subcommittee.

The IEEE Computer Society is publishing this standard in draft form to allow you to comment upon it prior to submission to the IEEE Standards Board for adoption as an IEEE standard. For example, should the data bus be extended to 32 bits, and if so, how? Your comments should be sent to George Morrow by August 15,1979, with copies to Gordon Force. Mr. Morrow's address is:

George Morrow
Thinker Toys
5221 Central Avenue
Richmond, California 94804

If you would like to participate in other standardization efforts of the Microprocessor Standards Committee, please contact its chairman:

Gordon Force
Logical Solutions
1128 Amur Creek Court
San Jose, California 95051

Finally, preparation of this proposed standard has benefited from the contributions of many individuals and companies. We indeed thank them all.

The Proposed Standard

1.0 General

1.1 Scope

This standard applies to interface systems for computer system components interconnected via 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 (as distinct from analog).
2) The total number of interconnected devices is small (22 or fewer).
3) The total transmission path length among interconnected devices is electrically short (25" or less). That is, transmission line propagation delays are not important.
4) The maximum data rate of any signal on the bus is low (less than or equal to 6 MHz).

1.0 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 S-100 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.

1.3.1 General system terms

Compatibility. The degree to which devices may be interconnected and used without modification, when designed as defined in Sections 2, 3, and 4 of this standard.

Interface. A shared 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, message transfer protocols, and functional logic circuits are typical interface system elements.

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

1.3.2 Signals and paths

Assert. To drive a signal line to the true state. The true 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 messages, 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 messages are carried.

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.

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

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.

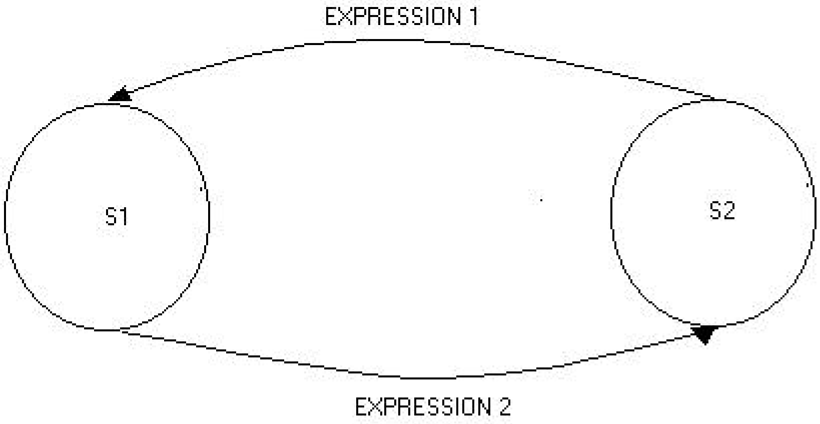
Unidirectional bus. A bus used by a device for oneway transmission of messages, that is, either input only or output only.

Word. A set of bit-parallel signals corresponding to binary digits and operated on as a unit. Usually 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 assumed that transition from one state to another will occur after a minimum 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 is mandatory and specifies 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. The use of the term "active" for the purpose of this standard is synonymous with the logic state true.

There are two types of electrical implementation of the logic states:

Active high signals. Active high signals are represented without a suffix after the signal name mnemonic (i.e. ABCD).

LOGIC STATE	BINARY STATE	ELECTRICAL SIGNAL LEVEL	ELECTRICAL STATE
FALSE (F)	0	Corresponds to < 0.8V. Called the LOW state.	L

TRUE (T)	1	Corresponds to > 2.0V. Called the HIGH state.	H

Active low signals. Active low signals are represented with an asterisk suffix after the mnemonic (i.e. ABCD*).

LOGIC STATE	BINARY STATE	ELECTRICAL SIGNAL LEVEL	ELECTRICAL STATE
FALSE (F)	0	Corresponds to > 2.0V. Called the HIGH state.	H
TRUE (T)	1	Corresponds to < 0.8V. Called the LOW state.	L

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*) • -BOUT, (electrical equation)

since pWR* is the 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 twoinput 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 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 (i.e. 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 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 the bus master and to transfer device dependent messages to or from the bus master.

Bus master and bus slave capabilities occur both individually and collectively in devices interconnected via the S-100 interface system.

1.6.3 Message paths and bus structure

The S-100 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:

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) DMA control bus- 8 signal lines.
7) Vectored interrupt bus- 8 signal lines.
8) Utility bus- 20 signal lines.

2.0 Functional specification

2.1 Functional partition

Functional devices interconnected via the interface system are divided into two broad classifications, bus masters and bus slaves, according to their relationship to the generation and reception of interface messages.

Devices acting as bus masters are 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 permanent 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 DMA cycle.

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

1) Only one permanent master may 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 DMA cy cle, that is, there are no nested DMA 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 must generate a specific set of signals in order to assure 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) DMA control bus.
7) Vectored interrupt bus.
8) Utility bus.

The nature and use of each bus is specified in the following sections.

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 must assert at least 16 address bits, but may assert 24 address bits if extended address capability is desired. Validity of the address bus is defined in 2.7.3.

Table 1 summarizes address usage for various bus cycles.

Table 1. Address usage for different bus cycles.

CYCLE TYPE STANDARD EXTENDED ADDRESSING ADDRESSING
MEMORY READ
MEMORY WRITE AO-A15 AO-A23
M1 (OP-CODE FETCH)
INPUT
OUTPUT , AO-Alt AO-A15
INTERRUPT ACKNOWLEDGE NONE NONE
HALT ACKNOWLEDGE NONE NONE
t see 2.2.2.3

2.2.2.1 Standard memory addressing

The standard memory address bus consists of 16 lines specifying 1 of 64K memory locations. These 16 lines are named AO through A15, where A15 is the most significant bit.

2.2.2.2 Extended memory addressing

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

2.2.2.3 Standard input/output device addressing

The standard 1/0 device address bus consists of 8 lines, AO through A7, specifying 1 of 2561/0 devices. A7 is the most significant bit.

NOTE: The 1/O device address has traditionally been duplicated onto the high order addre ss byte, A15-A8. While this is considered acceptable procedure, it is not recommended for new designs as it complicates expansion to extended 1/O device addressing.

2.2.2.4 Extended input/output device addressing

The extended 1/O device address bus consists of 16 lines, AO through A15, specifying 1 of 64K devices. A15 is the most significant bit.

2.2.3 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 lower-case 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 must be generated by the current bus master.

Validity of the status bus is given in 2.7.3.

2.2.3.1 Status memory write

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

sMemory Write = (-sOUT) • sWO, (logic equation)

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

2.2.3.2 Status usage chart

Table 2 gives the status word definition for all possible bus cycles. (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.

Table 2.

Status usage chart.

a
W O =
CL cc
J
STATUS BITS ° h h
CYCLE TYPE
MEMORY READ (B) H L H L L L L H

(W) H L H L L L L L
OP-CODE 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
WHERE: H = HIGH STATE L = LOW STATE X = DON'T CARE W = 16-BIT
OPERATION B = 8-BIT OPERATION

2.2.4 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 operations 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 (DO0-DO7), where D07 is the most significant bit.

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

2.2.4.2 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 DO0 corresponding to DATA 0 and DI7 corresponding to DATA 15, the most significant bit.

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 lower-case p.

The five lines are:

1) pSYNC, which indicates the start of a new bus cycle.
2) pSTVAL* 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 (e.g., data not ready), and to request operations of the permanent master (e.g., 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 ready lines are asserted.

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. As it is not specified as an open collector line, 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 via 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. The INT* line is often implemented as a "group interrupt" line in conjunction with the vectored interrupt bus. In this case, INT* indicates the presence of one or more vectored interrupt requests.

The NMI* line is a non-maskable interrupt request line, that is, it may not be masked off by the bus master. Accepting an interrupt on the 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 may 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 DMA control bus

The eight lines of the DMA 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 permanent bus master, thus effecting an orderly transfer of bus control.

All eight lines of the DMA control bus are specified as open collector lines.

The eight DMA control lines are:

1) DMA0*
2) DMA1*
3) DMA2*
4) DMA3*
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 DMA arbitration

The four lines that arbitrate among simultaneous requests for bus control by temporary masters are DMA0* through DMA3*. 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.

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 vectored 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 to the permanent bus master, and responds to the interrupt acknowledge bus cycle with appropriate vectoring data.

The eight lines of the vectored interrupt bus are VI0* through VI7*, where VI0* is considered the highest priority interrupt.

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

2.2.9 System utilities

2.2.9.1 System power

Power in S-100 systems is distributed to bus devices as unregulated voltages. A total of nine bus lines are used:

1) +8 volts, 2 lines.
2) +16 volts, 1 line.
3) -16 volts, 1 line.
4) GROUND, 5 lines.

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

The system clock, 0, 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, must be derived from this clock.

This signal is never transferred during a bus exchange operation.

2.2.9.3 CLOCK

This clock is specified as a 2-MHz (0.5 percent tolerance) signal with no relationship to any other bus signal. It is to be used by counters, timers, baud-rate generators, etc.

2.2.9.4 System reset functions

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 only on power-on, and asserts SLAVE CLR* and RESET*.

The POC* signal is specified as having a minimum active period of 10 msec.

RESET* and SLAVE CLR* are specified as open collector lines.

2.2.9.5 Memory write strobe

The memory write strobe, MWRT, must be generated somewhere in the system. It is usually 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.

Memory write is defined as:

MWRT = pWR • -BOUT (logic equation)

2.2.9.6 Phantom slaves

A line, PHANTOM*, is provided for overlaying bus slaves at a common address location. When this line is activated phantom bus slaves are enabled and normal bus slaves are disabled.

This line is specified as an open collector line.

2.2.9.7 Error

The line ERROR* is a generalized error line that is asserted when an error of some-sort (i.e., 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

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 MUST specify in detail any use of these lines. Signals on these lines are limited to 5 volt logic levels.

1111Power fail (PWRFAIL*)

The power fail line indicates impending power failure, and remains true until power is restored and POC* is true.

2.2.9.10 Reserved lines (RFU)

The five remaining lines are reserved for future use and may 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 3.

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.

Table 3. S-100 bus pin list.

PIN N0. SIGNAL & TYPE ACTIVE LEVEL DESCRIPTION

No.	Signal			Type	Description
1	+8V	B			Instantaneous minimum greater than 7 volts, instantaneous maximum less than 25 volts, average maximum less than 11 volts.
2	+16V	B			instantaneous minimum greater than 14.5 volts, instantaneous maximum less than 35 volts, average maximum less than 21.5 volts.
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	O.C.	Vectored interrupt line 0
5	VI1*	S	L	O.C.	Vectored interrupt line 1
6	VI2*	S	L	O.C.	Vectored interrupt line 2
7	VI3*	S	L	O.C.	Vectored interrupt line 3
8	VI4*	S	L	O.C.	Vectored interrupt line 4
9	VI5*	S	L	O.C.	Vectored interrupt line 5
10	VI6*	S	L	O.C.	Vectored interrupt line 6
11	VI7*	S	L	O.C.	Vectored interrupt line 7
12	NMI*	S	L	O.C.	Non-maskable interrupt.

13	PWRFAIL*	B	L		Power fail bus signal. (See Section 2.10.1 regarding pseudo open collector nature)
14	DMA3*	M	L	O.C.	Temporary master prioritybit 3.
15	A18	M	H		Extended address bit 18.
16	A17	M	H		Extended address bit 17.
17	A16	M	H		Extended address bit 16.
18	SDSB*	M	L	O.C.	The control signal to disable the 8 status signals.
19	CDSB*	M	L	O.C.	The control signal to disable the 5 control output signals.
20	GND	B			Common with pin 100.
21	NDEF				Not to be defined. Manufacturer must specify any use in detail.
22	ADSB*	M	L.	O.C.	The control signal to disable the 16 address signals.
23	DODSB*	M	L	O.C.	The control signal to disable the 8 data output signals.
24	PhiΦ	B	H		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	A15	M	H		Address bit 15 (most significant for non-extended addressing.)
33	A12	M	H		Address bit 12.
34	A9	M	H		Address bit 9.
35	D01 [DATA1]	M [M/S]	H		Data out bit 1, bidirectional data bit 1.
36	DO0 [DATA0]	M [M/S]	H		Data out bit 0, bidirectional data bit 0.
37	A10	M	H		Address bit 10.
38	D04 [DATA4]	M [M/S]	H		Data out bit 4, bidirectional data bit 4.
39	D05 [DATA5]	M [M/S]	H		Data out bit 5, bidirectional data bit 5.
40	DO6 [DATA6]	M [M/S]	H		Data out bit 6, bidirectional data bit 6.
41	DI2 [DATA10]	S [M/S]	H		Data in bit 2, bidirectional data bit 10.
42	DI3 [DATA11]	S [M/S]	H		Data in bit 3, bidirectional data bit 11.
43	DI7 [DATA15]	S [M/S]	H		Data in bit 7, bidirectional data bit 15.
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 HILT instruction

					has been executed.
49	CLOCK	B			2 MHz (0.5°/) 40-60% duty cycle. Not required to be synchronous with any other bus signal.
50	GND	B			Common with pin 100.
51	+8 VOLTS	B			Common with pin 1.
52	-16 VOLTS	B			Instantaneous maximum less than -14.5 volts, instantaneous minimum greater than -35 volts, average minimum greater than -21.5 volts.
53	GND	B			Common with pin 100.
54	SLAVE CLR*	B	L	O.C.	A reset signal to reset bus slaves. Must be active with POC* and may also be generated by external means.
55	DMA0*	M	L	O.C.	Temporary master priority bit 0.
56	DMA1*	M	L	O.C.	Temporary master priority bit 1.
57	DMA2*	M	L	O.C.	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	L	O.C.	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	O.C.	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, Section 2.7.5.3)
69	RFU				Reserved for future use.
70	GND	B			Common with pin 100.
71	RFU				Reserved for future use.
72	RDY	S	H	O.C.	See comments for pin 3.
73	INT*	S	L	O.C.	The primary interrupt request bus signal.
74	HOLD*	M	L	O.C.	The control signal used in conjunction with pHLDA to coordinate bus master transfer operations.
75	RESET*	B	L	O.C.	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 BS1.
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	A13	M	H		Address bit 13.
86	A14	M	H		Address bit 14.
87	A11	M	H		Address bit 11.
88	D02 [DATA2]	M [M/S]	H		Data out bit 2, bidirectional data bit 2.
89	D03 [DATA3]	M [M/S]	H		Data out bit 3, bidirectional data bit 3.
90	D07 [DATA7]	M [M/S]	H		Data out bit 7, bidirectional data bit 7.
91	DI4 [DATA12]	S [M/S]	H		Data in bit 4 and bidirectional data bit 12.
92	DI5 [DATA13]	S [M/S]	H		Data in bit 5 and bidirectional data bit 13.
93	DI6 [DATA14]	S [M/S]	H		Data in bit 6 and bidirectional data bit 14.
94	DI1 [DATA9]	S [M/S]	H		Data in bit 1 and bidirectional data bit 9.
95	DI0 [DATA8]	S [M/S]	H		Data in bit 0 (least significant for 8-bit data) and bidirectional data bit 8.
96	sINTA	M	H		The status signal identifying the bus input cycle(s) that may follow an accepted interrupt request presented on INT*.
97	sW0*	M	L		The status signal identifying a bus cycle which transfers data from a bus master to a slave.
98	ERROR*	S	L	O.C.	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 msecs.
100	GND	B			System ground.

The permanent master normally has control of the bus. It may relinquish bus control to a temporary bus master via a hold operation for an arbitrary number of cycles. Upon completion of the hold operation control of the bus is always returned to the permanent master.

2.3.2 Permanent master state diagram

The permanent master interface shall be implemented so as to conform to the state diagram given in Figure 1.

2.3.3 Permanent master state descriptions

2.3.3.1 Bus state 1

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 in the middle of the BS1 state, indicating the beginning of a new bus cycle.

2.3.3.2 Bus state 2

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 line are sampled during the BS2 state.

2.3.3.3 Wait state

The wait state, BS, 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. 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.

2.3.3.4 Bus state 3

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

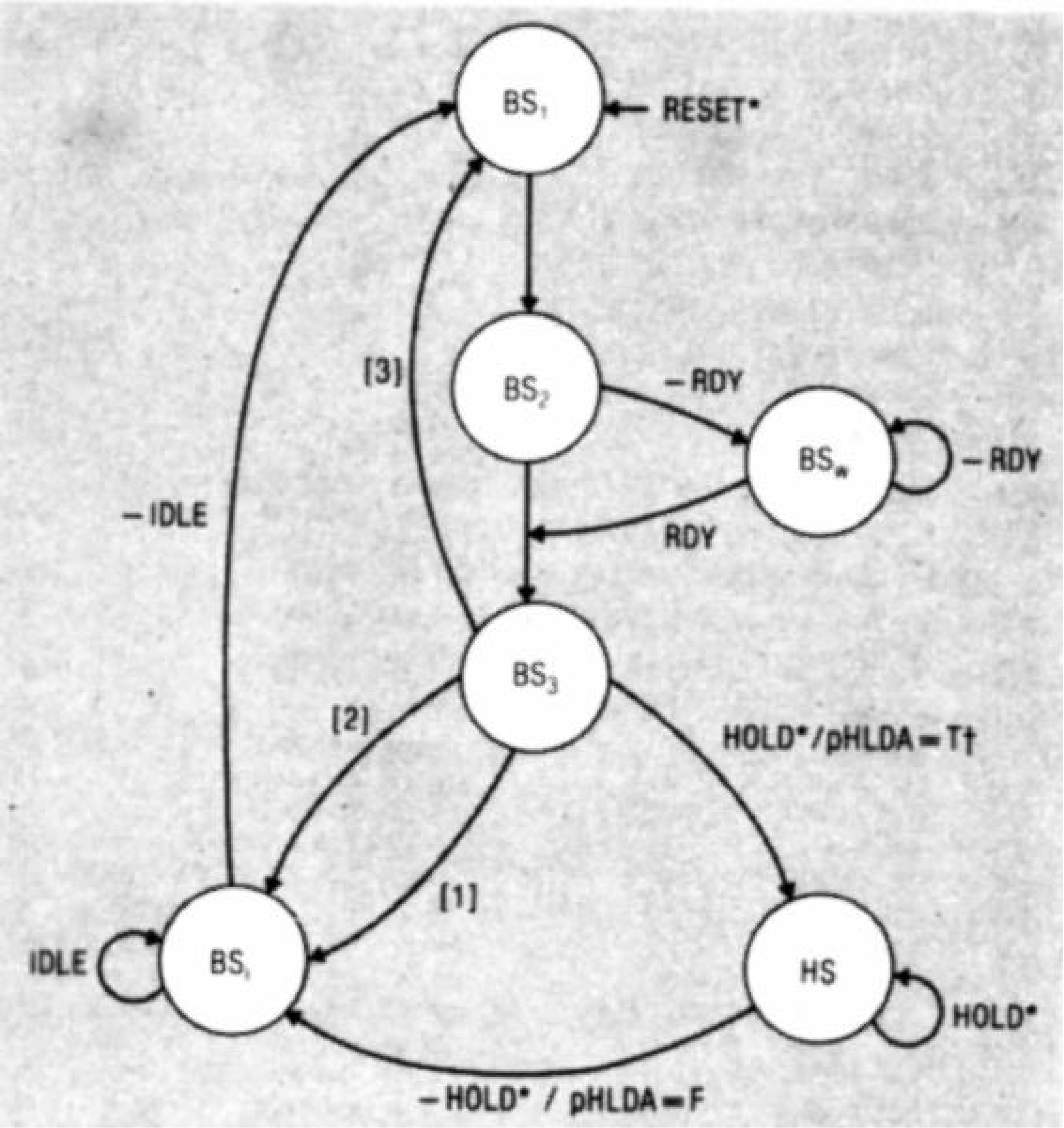


Figure 1. Permanent master state diagram.

t 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)

2.3.3.5 Idle bus states

After completion of the BS3 state, the master may enter one or more idle bus states.

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.

1111Hold accept
2

Permanent masters must be configured to conditionally accept hold operations from temporary masters. This function may be gated off under hardware or software control, to allow indivisible test and set operations. If hold is enabled and active, the permanent master will enter the hold state HS following a BS3 state, and pHLDA will be asserted.

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

Hold operations always 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 non-maskable 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) AO-A23t.
2) All status signals.
3) All control output signals.
4) Data output signals (8 or 16 depending on processor type).
5) (1), the system clock.

2.3.4.2 Input signals

The following signals are required input signals to permanent masters:

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

f A16 through A23 are optional on permanent masters.

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 DMA 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 an arbitration interval of one clock cycle.

Required output signals for dummy masters are the control output signals, and the system clock 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.

Since up to 16 temporary masters may co-exist in a single system, a protocol has been developed to arbitrate among simultaneous bus requests. Detailed specification of this protocol is given in 2.8.3.

2.4.2 Temporary master state diagram

The temporary master interface shall be implemented so as to conform to the state diagram given in Figure 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 must 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 (XS I and XS II)

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 is 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

The following are required output signals for a temporary master interface:

1) Address lines AO-A23t. 2) All status signals. 3) All control output signals tt. 4) Data output lines. 5) DMA arbitration lines DMA0*DMA3*. 6) Hold request, HOLD*.

2.4.4.2 Input signals

The following are required input signals for a temporary master interface:

t Note: Temporary masters must generate Alb-A23; they need only generate falses or lows on these 8 lines, however. ft Note: Temporary masters should provide a jumper on the pSTVAL* line, as 8080 CPUs of old design do not transfer this line with the control output lines. In this case all bus masters use the same pSTVAL* signal.

Figure 2. Temporary master state diagram.

1) The ready lines, RDY and XRDY.
2) Hold acknowledge, pHLDA.
3) Data input lines.
4) The system clock, 0.

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.

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

111Slave interface state diagram
2

The slave interface shall conform, in general, to the state diagram given in Figure 3. Slave interfaces need not have both read and write capability.

2.5.2 Slave state definitions

2.5.2.1 Slave idle state

The slide 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

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*.

Figure 3. Slave interface state diagram.

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.

1111Slave read

2

Data from the addressed slave is gated onto the data bus during the slave read state, r. 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

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 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.

111Required signals for slave interfaces

2

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

2.6 8116-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 masters and 16-bit masters and slaves to co-exist in a single system. For 16-bit transfers the two unidirectional 8bit data buses are ganged to form a single 16-bit bidirectional data bus.

Two lines 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, DO0 through D07.

Byte data input from the addressed slave to the current bus master is asserted on the data input bus, DI0 through DI7.

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 page 52).

Sixteen-bit data transfer is then conducted via the ganged data buses, where DO0 = DATA0, and D17 = DATA15.

2.6.4 Memory organization

Memory devices capable of both 8-bit and 16-bit parallel data transfers are organized, as shown in Figure 4, as two banks of 8-bit memory, a high-byte bank and a low-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.

Figure 4. 8/16-bit memory organization.

The proper location in memory is selected by the address output on address lines A1 through A15 (A23 for extended addressing systems, while the AO line selects the high byte or the low byte. AO equals 0 selects the high byte of the 16-bit word, while AO equals 1 selects the low byte of the word.

See Figure 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 low-byte data input lines are connected directly to the DO bus, and the high-byte data input lines are connected to the DO bus via a two-to-one multiplexer controlled by sXTRQ*.

Data output from the memory banks is routed to Tri-State bus drivers A and B in Figure 4. One of these drivers is enabled when the read strobe is activated, depending on the condition of A0. 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.

Address lines A1 through A15 (A23 in extended address systems) select the proper word from memory. The condition of the AO bit does not enter into the decoding or addressing for word references.

See Figure 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 high-byte data, on the DI bus, to the data input lines of the high-byte bank. Low-byte data, on the DO bus, is connected to the data input lines of the low-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.

*Tri-State is a trademark of National Semiconductor.

Figure 5. 8116-bit address and data usage.

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 lines will not be asserted.

The master will respond in one of two ways, by generating an error trap or by conducting the transfer in byte-serial fashion.

2.6.5.1 Byte-serial response

If the sixteen acknowledge line is not activated after a specified period, circuitry may 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.

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.

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 (D.

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

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, 4).

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.

t See possible exception, section 2.7.5.3.

The position of the status valid strobe within the pSYNC interval is independent of the system clock, (D. 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 such that the delay between guaranteed status on the bus and the activation of the status valid strobe is as close to the minimum specification 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 may occur while pSYNC is asserted.

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 BS 2 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 of 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, and status 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 1 and 2.

tNote: The ml signal output from current 8080 processor boards meets all the specifications as a status valid strobe. Hence, these boards meet the bus cycle specification without modification.

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.

1111The write strobe

2

The generalized write strobe, pWR*, is used to write data from the data bus into the addressed bus slave. The write strobe may be asserted by the master after the completion of the pSYNC interval.

Data out on the data bus must 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.

Figure 6. Bus cycle fundamental timing relationships.

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. 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 such 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 • -BOUT, (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."

tNote: Historically the MWRT strobe has been generated by frontpanel devices to accomplish the deposit function. In such a case, the MWRT strobe is asserted while the front-panel holds the CPU in a wait state during a memory read cycle. Note that the status will indicate a read cycle and the pDBIN strobe will be active.

While this is acceptable procedure for 8-bit systems, and 8-bit memories should respond to MWRT as well as pWR*, it is not permissible in 16-bit systems as a conflict will exist on the bidirectional data bus. A front-panel could be implemented either as a temporary master, or integrated into the CPU.

2.8 Special bus operations

2.8.1 General

This section describes two special bus operations related to DMA 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.

Figure 7. Bus transfer state diagram.

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 pt rmanent 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 Figure 7.

2.8.2.3 Bus transfer state definitions

2.8.2.3.1 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.

2.8.2.3.2 Arbitration

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

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, XS I

The bus transfer sequence begins with transfer state one, XS I. The bus transfer control circuit asserts the following signals together:

1) ADSB*
2) SDSB*
3) DODSB*

disabling the address, status, and data output drivers of the permanent bus master and enabling the control output drivers of the temporary master. Both the permanent master and the temporary master are now driving the control output lines. These lines are required to have the following levels during this time.

	Signal	Logic state	Electrical level
1)	pSYNC	F	L
2)	pSTVAL*	F	Ht
3)	pDBIN	F	L
4)	pWR*	F	H
5)	pHLDA	T	H

The transfer state is terminated by the assertion (by the bus transfer control circuit) of the CDSB* line, 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.

2.8.2.3.5 Bus cycles

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 XS II, transfer state two.

2.8.2.3.6 Transfer state two

Transfer state two, XS II, is the mirror image of the sequence in XS I. The state begins with the release of the CDSB * signal, 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 XS II at the levels prescribed for XS I.

The state is ended by the release of other disable signals and HOLD*, 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

28.2.4.1 General

The fundamental timing relationships for a bus transfer and a single DMA bus cycle are given in Figure 8.

tSee note in section 2.4.4.1.

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 5.

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 XS I allows time for completion of the preceding bus cycle.

2.8.2.4.3 Tov

The time that both the temporary master and the permanent master must drive the control output signals has a specified minimum to assure a smooth bus transfer.

Assertion of the XFER II signal, or CDSB*, is specified relative to the rising edge of the system clock,.), 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

The "done" signal is a signal internal to the temporary master. This signal should not be asserted until the hold time for data output, status, and address signals in the standard bus cycle has been met.

11111Trel
2

Completion of the reverse bus transfer XFER II shall precede the release of the pHLDA signal by a minimum specified time.

Figure 8. Bus transfer and single bus cycle.

Figure 9a. Bus arbitration diagram.

Figure 9b. Bus arbitration example.

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 DMA controllers or multiple 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 pullup resistors. Each temporary master has a unique priority number which it asserts on the arbitration bus 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 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 DMA0* through DMA3*, where DMA3* is the most significant bit. These lines, in conjunction with HOLD* and pHLDA, control the bus arbitration process.

28.3.1 Bus arbitration implementation

An implementation of the bus arbitration protocol is shown in Figures 9a and 9b.

Any implementation shall obey the rules summarized in section 2.8.4.

28.3.2 Bus arbitration state definitions

2.8.3.2.1 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. This guarantees an 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 falling edge of pHLDA.

2.8.3.2.2 Priority compare states

The priority comparison states, C3 through C0, are the states where each requester compares the priority it is attempting to assert on the arbitration bus with the priority 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 DMA 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 winner 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 10 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 low 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 non-conflicting 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 9A and 9B represent an example, 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* may be asserted only when it is not already asserted and pHLDA is low.
2) HOLD* must be removed when pHLDA rises if another controller has asserted higher priority.
3) HOLD* must be removed when the controller no longer needs the bus.
4) Priority must be asserted whenever HOLD* is asserted, and must remain asserted until the next falling edge of pHLDA.

Figure 10. Bus arbitration timing diagrams.

5) The priority level must be user-selectable by switches and asserted by open-collector drivers on bus lines DMA3*-DMA0*.
6) The most significant bit of the priority level (appearing on DMA3*) 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 DMA2*, DMA1*, and DMA0* 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 highest priority and may take the bus when pHLDA rises.
8) Logic implementations must 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 non-maskable interrupt. A complying master need only implement INT*.

2.9.1 Vectored interrupts

2.9.1.1 Vectored interrupt requests

Eight levels of vectored interrupt requests are issued on the vectored interrupt lines, VI0* thru VI7*, where VI0* is the most significant interrupt priority level. Vectored interrupt requests, however, maybe 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 appropriately to any interrupt acknowledge cycles. The interrupt controller is not required to use INT*. A vectored interrupt may occur without INT* ever being asserted.

2.9.1.2 Interrupt acknowledge

The interrupt acknowledge cycle is a standard bus read cycle. 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 acknowlege cycle, only one interrupt controller may exist on the bus. 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 Non-maskable interrupt (NMI*)

The non-maskable 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 implementation 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 50 msecs before the local voltage regulators drift out of specification.

The line stays low until the power-on clear signal is activated. This implies that either a normally closed relay or a battery powered circuit drive the power fail line. The circuit driving this line must 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 (i.e., 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 masters)-should be latched on the falling edge of ERROR*.

ERROR* is implemented as an open collector line.

3.0 Electrical specifications

3.1 Application

This section defines the electrical specifications for interface devices to be used in S-100 bus systems. Proper operation of these devices also depends on two other factors:

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 Ta = O'C to 70'C.

3.2 Power distribution

Power in S-100 systems is distributed as unregulated DC power at three voltages, +8 volts, +16 volts, and -16 volts. 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 volt specification

Instantaneous minimum must be greater than +7 volts, instantaneous maximum less than 25 volts, and average maximum less than 11 volts.

3.2.2 +16 volt specification

Instantaneous minimum must be greater than 14.5 volts, instantaneous maximum less than 35 volts, and average maximum less than 21.5 volts.

3.2.3 -16 volt specification

Instantaneous maximum must be less than -14.5 volts, instantaneous minimum greater than -35 volts, and average minimum greater than -21.5 volts.

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 volts and +5 volts, and may not have loaded rise or fall times less than 5 nsecs.

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 100052 ± 5% pullup resistor to +5 volt or equivalent must be provided somewhere in the system for open collector lines. It is recommended that these pullup resistors be provided on the bus. However, implementation on the permanent master is also acceptable.
3) A Tri-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 volts at 24 mA sink current. High state (VOH): Output voltage (for active and Tri-State drivers) greater than or equal to +2.4 volts at 2 mA.

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

The internal capacitive load of a driver shall not exceed 15 pF at 25 °C whether in the active or the highimpedance 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 nsec at rated capacitive load.

11 Receiver specifications
2

The specifications for receivers on the bus shall be as follows:

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

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

Bus receivers shall source no more than 0.5 mA at 0.5 volts and sink no more than 50 MA at 2.4 volts.

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 greater than or equal to 0.4 volts.

3.6 Bidirectional signals

Some interface signals, such as the data bus, are combined Tri-State drivers and receivers. For each function these devices must 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 volts nor sink more than 80 uA at 2.4 volts on any signal line exc ept for DMA0* DMA1 *, DMA2*, DMA3*, PHANTOM*, and PWRFAIL*. On these lines a card may not source more than 0.4 mA at 0.5 volts.

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 volts ± 0.2 volts and R is no less than 18052(± 5%).

Open collector lines may have a combination pullup and termination scheme using a circuit equivalent to

where V =2.6 volts ± 0.2 volts, R1 = 1.5Kohm± 5%, and R2 should be no less than 18052 (±5%).

3.8 Read cycle timing specification

Figure 11a depicts the read cycle timing waveforms with the pertinent timing parameters shown. Table 4 specifies these parameters.

3.9 Write cycle timing specification

Figure 11 b depicts the write cycle timing waveforms with the pertinent timing parameters shown. Table 4 specifies these parameters.

3.10 Ready and sixteen request timing specification

Figure 12 depicts RDY, XRDY, and SIXTN* timing waveforms during read and write cycles, with pertinent timing parameters shown. Table 4 specifies these parameters.

3.11 Bus transfer timing specification

Figure 8 depicts bus transfer timing waveforms with the pertinent timing parameters shown. Table 5 specifies these parameters.

4.0 Mechanical specifications

4.1 Application

This section defines the mechanical specifications for standard interface systems.

4.2 Connector type

The card edge connector is a 100-pin (dual 50) connector with contacts spaced on 0.125" centers. It is nominally designed for printed circuit boards 0.062 " 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 volts DC, minimum pin to pin.
2) Current rating: 2.5A per contact.
3) Contact resistance: 50 mohm maximum at rated current after 100 insertions.
4) Insulation resistance: 1000 Mohm minimum.

4.2.2 Connector spacing

Connectors should be spaced 0.75 inches -0.01 inches center to center.

Figure 11 a. Read cycle timing diagram.

Figure 11b. Write cycle timing diagram.

Table 4. Read/write cycle timing parameters.

MIN.(NSECS) MAX.(NSECS)

tcy 4) PERIOD 166 2000
tCYH 4) PULSE WIDTH HIGH 0.4tcy
tCYL (D PULSE WIDTH LOW 0.4tcy
tphiSY DELAY m HIGH TO pSYNC HIGH; 10 0.4tcy DELAY 4) LOW TO pSYNC LOW
tsy pSYNC PULSE WIDTH HIGH 0.7tcy
t!STphi pSTVAL* LOW PRIOR TO 4) LOW DURING pSYNC 0
tST pSTVAL* PULSE WIDTH HIGH 50
t!ST pSTVAL* PULSE WIDTH LOW 50
tSTSY pSTVAL* FALLING EDGE PRIOR TO pSYNC HIGH 0
t!AST ADDRESSES STABLE PRIOR TO pSTVAL* LOW DURING pSYNC HIGH 70
t!SST STATUS STABLE PRIOR TO pSTVAL* LOW DURING pSYNC HIGH 40
tDB pDBIN PULSE WIDTH HIGH 0.9tCY
t!STDB DELAY pSTVAL* LOW TO pDBIN HIGH 20
t!DBSY DELAY pDBIN LOW TO pSYNC HIGH 0
t!DBAS HOLD TIME FOR ADDRESSES AND STATUS AFTER pDBIN LOW 50
t!DBZ DELAY pDBIN LOW TO SLAVE DI DRIVERS Hi-Z 25+0.1tcy
tDB!Z DELAY pDBIN HIGH TO SLAVE DI DRIVERS ACTIVE 10 25+0.1tcy
tACC DELAY pSTVAL* LOW TO DATA VALID SPECIFIED BY MANUFACTURER.

WORST CASE MAXIMUM FOR ALL SLAVES AND WORST CASE MINIMUM FOR ALL MASTERS.

tS!DB DATA VALID SETUP TIME TO pDBIN LOW
t!WR pWR* PULSE WIDTH LOW 0.9tcy
t!STWR DELAY pSTVAL* LOW TO pWR* LOW 30
tWRSY DELAY pWR* HIGH TO pSYNC HIGH 0

tD!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

t!RDYphi SETUP TIME ROY, XRDY, SIXTN* TO m RISING 80

tphiRDY HOLD TIME ROY, XRDY, SIXTN* AFTER m RISING 70

Figure 12. Timing of RDY, XRDY, and SIXTN' during read and write cycles.

Table 5.

Bas transfer timing parameters.

tSET DELAY pHLDA TO ADSB', SOW, DODSB' SD

IRS TIME BOTH TEMPORARY AND PERMANENT MASTER DRIVE THE CONTROL OUTPUT LINES 0 Stay

tOH HOLD TIME ADDRESS, STATUS, AND DATA OUT DURING DMA CYCLE 0 2tcy

WET SETUP TIME, END OF BUS TRANSFER TO pHLDA RISING EDGE 20

4.3 Board size specification

Circuit boards shall conform to the board size specifications given in Figure 13. The edge connector pin shown in the figure is pin 60. Pin 100 opposes pin 50 on the back side of the board. Total board depth shall not exceed 0.66 ". Nominal board thickness is 0.062 ".

Figure 13.5-100 111 mechanical parameters.