SHUGART ASSOCIATES

PRODUCT SPECIFICATIONS

CODE	PART NO.	REV EC
PS	30134-0	A

SHEET 1 OF 22

TITLE SHUGART ASSOCIATES SYSTEM INTERFACE

SASI
SHUGART ASSOCIATES
SYSTEM INTERFACE

1.0 INTRODUCTION

This document defines the Shugart Associates System Interface. The interface was designed to provide an efficient method of communication between computers and peripheral input/output devices.

The Shugart Associates System Interface (SASI) is implemented with a bus that allows an economy or performance oriented system with the following key features:

- Single or multiple host computer system
- Multiple I/O device type compatibility
- Speed independent asynchronous communication
- Multiple overlap of I/O device operation
- Intelligent I/O device oriented
- Direct I/O device to I/O device copy

The practical application of this SASI hardware specification will require a compatible definition of software usage.

2.0 SHUGART ASSOCIATES SYSTEM INTERFACE (SASI)

Communication on the SASI bus is allowed between two SASI BUS PORTS. There is a maximum of eight (8) BUS PORTS. Each port is attached to a BUS DEVICE. Each BUS DEVICE has some associated equipment. The actual equipment required is determined by the system makeup.

When two BUS DEVICES communicate on the bus, one must perform the role of the bus INITIATOR and the other must perform the role of the bus TARGET. An INITIATOR is considered to be the BUS DEVICE that starts an operation. A TARGET is considered to be the BUS DEVICE that performs the operation. A BUS DEVICE may have a fixed role as an INITIATOR or TARGET, or the BUS DEVICE may be able to assume either role.

In a typical system, a computer's Host Adapter acts as the INITIATOR, and an I/O device's Control Unit acts as the TARGET.

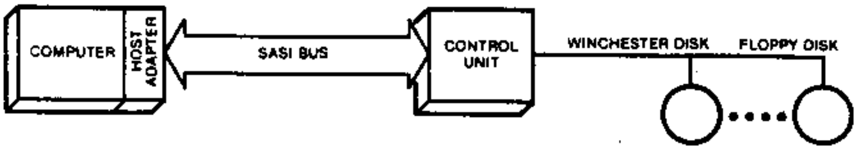
Certain bus functions are assigned to the INITIATOR and the TARGET. The INITIATOR may ARBITRATE for the bus and SELECT a particular TARGET. The TARGET may request the transfer of control or data information on the bus, and in some cases it may ARBITRATE for the bus and RESELECT an INITIATOR for the purpose of continuing some operation.

Data transfer on the bus is asynchronous and follows a defined Request/Acknowledge handshake protocol. One eight (8) bit byte of information may be transferred with each handshake.

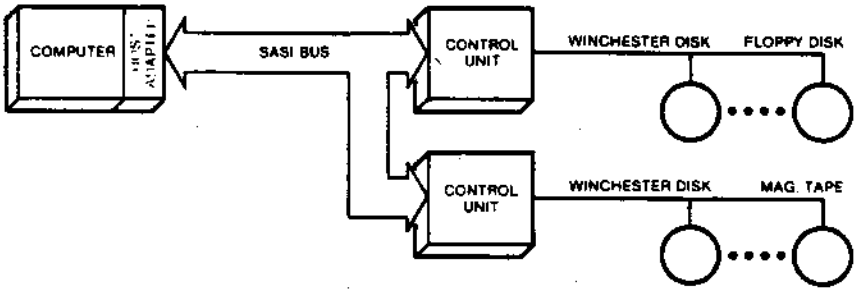
2.1 BUS DEVICE INTERCONNECTION

All BUS DEVICES are daisy chained together with a single fifty (50) conductor flat cable. Each end of the cable is terminated. All signal lines are common between all BUS DEVICES. The maximum cable length is 6 meters. Three example system configurations are shown in Fig. 2.0. A typical Intelligent Disk Controller block diagram is shown in Fig. 2.1.

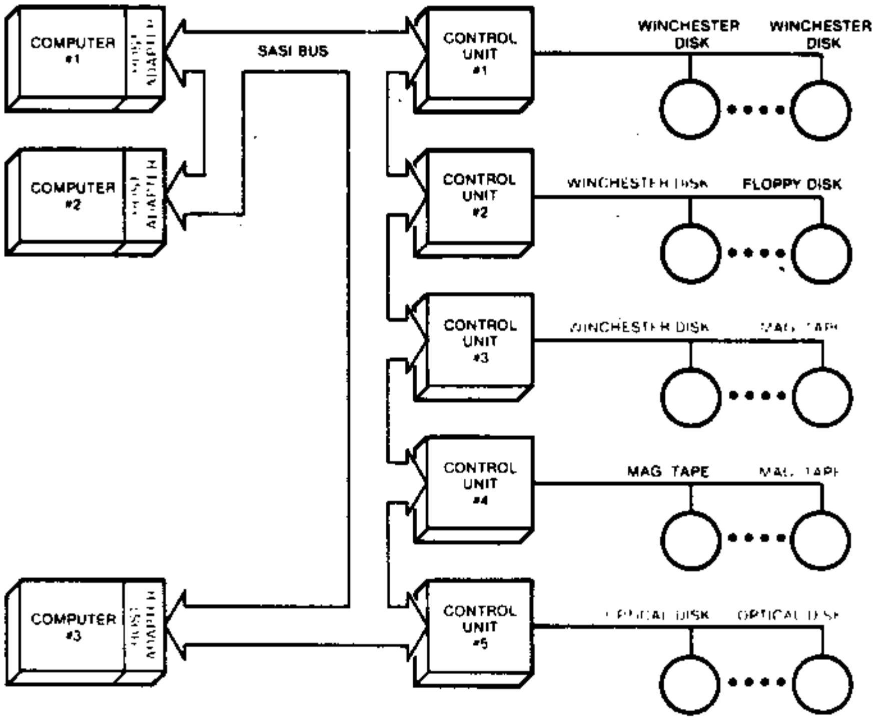
SAMPLE SASI CONFIGURATIONS



SIMPLE SYSTEM



BASIC TWO CONTROL UNIT SYSTEM



COMPLEX SYSTEM

Up to 8 devices can be supported by the SASI bus. The devices can be any combination of host CPUs and/or intelligent controllers.

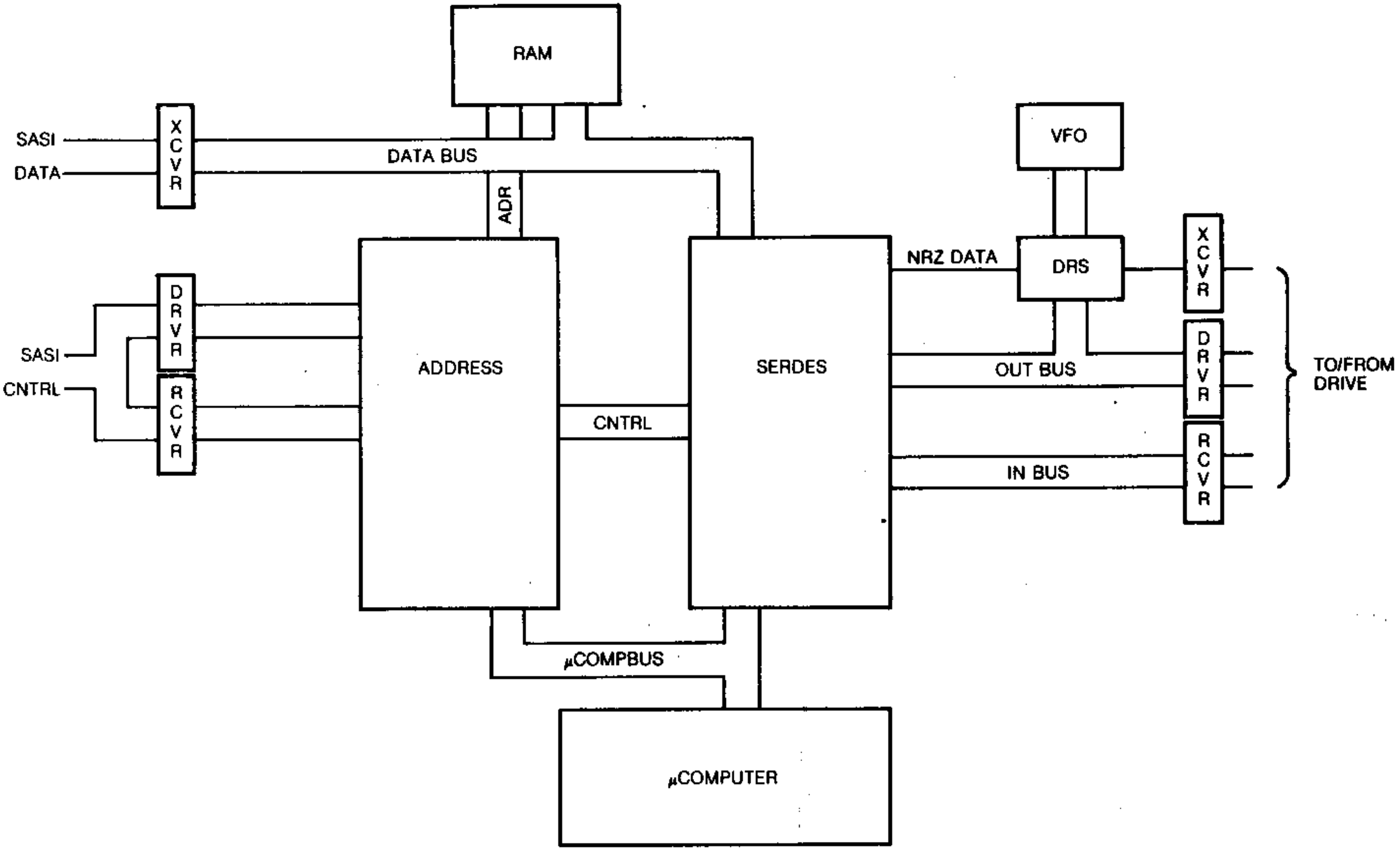


FIGURE 2.1 SASI INTELLIGENT INTERFACE COMPONENTS

2.2 BUS LINES

There are nine (9) control lines and nine (9) data lines (including parity).

-BSY

-SEL

-C/D

-I/O

-REQ

-ACK

-ATN

-MSG

-RST

-DB(7-0,P)

All bus lines are implemented Low True. Thus the ASSERTION of a control function or data bit will cause the corresponding line to assume a voltage typically less than 0.5 volts, while NON-ASSERTION will cause this line to assume a voltage typically greater than 2.4 volts. The Low True representation of a signal line is indicated on a schematic or timing diagram by the appendage of a minus sign to the front of the signal name (e.g. —BSY).

All control lines use open collector drivers. The data lines use open collector or three-state drivers as noted in their description.

The lines are described below:

-BSY (BUSY) A line which generally indicates when the bus is being used.
-SEL (SELECT) A line used by an INITIATOR to select a TARGET or by a TARGET to reselect an INITIATOR.

-C/D (CONTROL/DATA) A line driven by a TARGET which generally indicates whether Control or Data information is on the data lines.(ASSERTION = CONTROL).

-I/O (INPUT/OUTPUT) A line driven by a TARGET which indicates direction of data on the data on the data lines with respect to an INITIATOR. (ASSERTION = INPUT).

-REQ (REQUEST) A line driven by a TARGET to indicate a request for a REQ/ACK data transfer handshake.

-ACK (ACKNOWLEDGE) A line driven by an INITIATOR to indicate an acknowledgement for a REQ/ACK data transfer handshake.

-ATN (ATTENTION) A line driven by an INITIATOR to indicate the ATTENTION condition.

-MSG (MESSAGE) A line driven by a TARGET during the MESSAGE phase.

-RST (RESET) A line which indicates the RESET condition.

-DB(7-0,P) (DATA BITS) Eight data lines plus a parity line which form a DATABUS DB(7)is the most significant bit and is also the highest priority during arbitration. Bit number, significance and prioritydecrease down to DB(0).

Data parity DB(P) is odd. The use of parity is a system option. Parity is not valid during arbitration.

Each of the eight data lines is uniquely assigned as a BUS DEVICE's own BUS ADDRESS (i.e. BUS DEVICE ID). This line must be driven by an open collector during ARBITRATION. For all phases other than ARBITRATION, all nine data lines can be IMPLEMENTED with open collector or three-state drivers. Thus a DEVICE ID line may have two sources within a BUS DEVICE'S port.

2.3 BUS PHASES

The bus has eight (8) distinct operational phases:

- ARBITRATION Phase
- SELECTION Phase
- RESELECTION Phase
- COMMAND Phase
- DATA Phase
- STATUS Phase
- MESSAGE Phase
- BUS FREE Phase

The bus can never be in more than one phase at any given time. Unless otherwise noted, the following descriptions assume that bus lines which are not mentioned will not be asserted.

2.3.1 Arbitration Phase

Purpose:

The ARBITRATION phase allows one BUS DEVICE to gain control of the bus so that this device can assume the roll of an INITIATOR or TARGET.

Option:

Implementation of the ARBITRATION phase is a system option. Systems with no ARBITRATION phase can have only one INITIATOR. The ARBITRATION phase is required for systems which use the RESELECTION phase.

Bus Device Implementation Procedure:

After a BUS DEVICE detects the BUS FREE phase it waits a minimum of BUS FREE DELAY and a maximum of BUS SET DELAY in order to assert BSY and its onw DEVICE ID on the bus. (Detection time should be included in the delay calculations).

Note: The DEVICE ID is asserted on the DATA BIT line that corresponds to the device's unique BUS ADDRESS. All three-state DATA BUS drivers must be passive. Data parity is not guaranteed valid during ARBITRATION.

The BUS DEVICE will immediately clear itself from arbitration (within a BUS CLEAR DELAY time) by releasing its BSY and ID lines if SEL is asserted by any other BUS DEVICE.

After an ARBITRATION DELAY (measured from the assertion of BSY) the BUS DEVICE checks the data lines. If a higher priority ID is on the bus (DB(7) = highest) then the BUS DEVICE clears itself from ARBITRATION by releasing its BSY and ID lines. If the BUS DEVICE determines that its own ID is the highest asserted, then it wins arbitration and asserts SEL, (after the assertion of SEL the BUS DEVICE must wait a minimum of BUS SETTLE DELAY before changing any bus lines).

2.3.2 Selection Phase

Purpose:

The SELECTION phase allows an INITIATOR to select a TARGET for the purpose of initiating some TARGET function(s), (e.g. read or write data).

Bus Device Implementation Procedure:

Note: All during the SELECTION phase the I/O line is not asserted so that this phase can be distinguished from the RESELECTION phase.

In systems where the ARBITRATION phase is not implemented, the INITIATOR first detects the BUS FREE phase and then waits a BUS SETTLE DELAY. Then the INITIATOR asserts the DATA BUS with the desired TARGET's DEVICE ID and its own (INITIATOR) DEVICE ID. After two DESKEW DELAYS the INITIATOR asserts SEL.

In systems with ARBITRATION implemented, the BSY and SEL lines will be asserted by an INITIATOR when going from the ARBITRATION phase to the SELECTION phase. The INITIATOR then asserts the DATA BUS with the desired TARGET's DEVICE ID ~~and its own (INITIATOR) device ID.~~ After two DESKEW DELAYS the INITIATOR releases BSY.

The selected TARGET detects the simultaneous (within a DESKEW DELAY) condition of SEL and its own DEVICE ID asserted, and both BSY and I/O not asserted. ~~The select TARGET may sample the DATA BUS to determine the DEVICE ID of the INITIATOR that is doing the SELECTION~~. The selected TARGET then responds by asserting BSY.

After two DESKEW DELAYS the INITIATOR releases SEL and may change the DATA LINES.

Note: After an active INITIATOR releases SEL it is the only BUS DEVICE that can assert ACK and ATN (and the DATA LINES if the I/O signal is NOT asserted by the TARGET (indicating OUTPUT from the INITIATOR)).

Note: After an active TARGET detects the release of SEL, it is the only BUS DEVICE that can assert BSY, C/D, I/O, REQ and MSG (and the DATA LINES if I/O is asserted (indicating INPUT to the INITIATOR))

2.3.3 Reselection Phase

Purpose:

The RESELECTION phase allows a TARGET to reconnect to an INITIATOR for the purpose of continuing some operation that was previously started by the INITIATOR but was interrupted by the TARGET. (The TARGET DISCONNECTED by allowing a BUS FREE phase before the operation was complete.)

Option note:

RESELECTION can only be used in systems that have ARBITRATION implemented.

Bus Device Implementation Procedure:

After the TARGET has gone through the ARBITRATION phase it will be asserting BSY and SEL. The TARGET then asserts the I/O line and also asserts the DATA BUS with the desired INITIATOR's DEVICE ID ~~and its own (TARGET) DEVICE ID.~~

After these assertions the TARGET waits at least two DESKEW DELAYS and then releases BSY.

The reselected INITIATOR detects the simultaneous (within a DESKEW DELAY) condition of SEL, I/O and its own DEVICE ID asserted, and BSY not asserted. ~~The reselected INITIATOR may sample the DATA BUS to determine the DEVICE ID of the TARGET that is doing the RESELECTION.~~ The reselected INITIATOR then responds by asserting BSY.

The TARGET then also asserts BSY and will continue the assertion until it is done using the bus. After two DESKEW DELAYS the TARGET releases SEL and may change the I/O and DATA LINES.

The selected INITIATOR detects the release of SEL and releases its assertion of BSY.

2.3.4 Information Transfer Phases (COMMAND, DATA, STATUS and MESSAGE Phase)

Common notes:

The COMMAND, DATA, STATUS and MESSAGE phases can all be grouped together as the INFORMATION TRANSFER phases because they are all used to transfer data or control information via the DATA BUS.

The INFORMATION TRANSFER phases use one or more REQ/ACK hand shakes to control the data transfer. Each REQ/ACK allows the transfer of one byte of data. The REQ/ACK handshake starts with the TARGET asserting the REQ line. The INITIATOR responds by asserting the ACK line. The TARGET then releases the REQ line. The INITIATOR again responds by releasing the ACK line.

If the I/O line is asserted, data will be INPUT into the INITIATOR from the TARGET. The TARGET shall guarantee that valid data is available on the bus at the INITIATOR's port a DESKEW DELAY before the assertion of REQ is valid at the INITIATOR's port. The data shall remain valid until the assertion of ACK by the INITIATOR. It shall be the TARGET's responsibility to compensate for cable skew and the skew of its own drivers.

If the I/O line is NOT asserted data will be OUTPUT from the INITIATOR into the TARGET. The INITIATOR shall guarantee that it has placed valid data on the bus within a DESKEW DELAY after its assertion of ACK on the bus. Valid data shall remain on the bus until the TARGET releases REQ. It shall be the TARGET's responsibility to compensate for cable skew and the skew of its own receivers.

During each INFORMATION TRANSFER phase the BSY line shall remain asserted and the SEL line shall remain released. Additionally during each INFORMATION TRANSFER phase the TARGET shall continously envelope the REQ/ACK handshake(s) with the C/D, I/O and MSG lines in such a manner that these control lines are valid for a BUS SETTLE DELAY before the REQ of the first handshake and remain valid until the release of ACK at the end of the last handshake.

2.3.5 Command Phase

Purpose:

The COMMAND phase allows the TARGET to request command information from the INITIATOR.

Bus Device Implementation Procedure:

The TARGET asserts the C/D line and releases the I/O and MSG lines during the REQ/ACK handshake(s) of this phase.

2.3.6 Data Phase

The DATA phase is a term that encompasses both the DATA IN phase and the DATA OUT phase.

2.3.6.1 Data In Phase

Purpose

The DATA IN phase allows the TARGET to request that data be INPUT to the INITIATOR from the TARGET.

Bus Device Implementation Procedure:

The TARGET asserts the I/O line and releases the C/D and MSG lines during the REQ/ACK handshake(s) of this phase.

2.3.6.2 Data Out Phase

Purpose:

The DATA OUT phase allows the TARGET to request that data be OUTPUT from the INITIATOR to the TARGET.

Bus Device Implementation Procedure:

The TARGET release the C/D, I/O and MSG lines during the REQ/ACK handshake(s) of this phase.

2.3.7 Status Phase

Purpose:

The STATUS phase allows the TARGET to request that status information be sent from the TARGET to the INITIATOR.

Bus Device Implementation Procedure:

The TARGET asserts C/D and I/O and it releases the MSG line during the REQ/ACK handshake(s) of this phase.

2.3.8 Message Phase

The Message phase is a term that encompasses the CONTROL MESSAGE IN, CONTROL MESSAGE OUT, DATA MESSAGE IN and DATA MESSAGE OUT phases.

2.3.8.1 Control Message In Phase

Purpose:

The CONTROL MESSAGE IN phase allows the TARGET to request that CONTROL MESSAGES be INPUT to the INITIATOR from the TARGET.

Bus Device Implementation Procedure:

The TARGET asserts C/D, I/O and MSG during the REQ/ACK handshake(s) of this phase.

2.3.8.2 Control Message Out Phase

Purpose:

The CONTROL MESSAGE OUT phase allows the TARGET to request that a CONTROL MESSAGE be OUTPUT from the INITIATOR to the TARGET. The TARGET may invoke this phase at its convenience only in response to the ATTENTION condition created by the INITIATOR.

Bus Device Implementation Procedure:

In response to the ATTENTION condition, the TARGET asserts C/D and MSG and releases the I/O line during the REQ/ACK handshake(s) of this phase. (See the ATTENTION condition description).

2.3.8.3 Data Message In Phase

Purpose:

The DATA MESSAGE IN phase allows the TARGET to request that a DATA MESSAGE be INPUT to the INITIATOR from the TARGET.

Bus Device Implementation Procedure:

The TARGET asserts I/O and MSG and it releases the C/D line during the REQ/ACK handshake(s) of this phase.

2.3.8.4 Data Message Out Phase

Purpose:

The DATA MESSAGE OUT phase allows the TARGET to request that a DATA MESSAGE be OUTPUT from the INITIATOR to the TARGET.

Bus Device Implementation Procedure:

The TARGET asserts MSG and it releases the C/D and I/O lines during the REQ/ACK handshake(s) of this phase.

2.3.9 Bus Free Phase

Purpose:

The BUS FREE phase is used to indicate that no BUS DEVICE is actively using the bus and that the bus is available for subsequent users.

Bus Device Implementation Procedure:

The BUS FREE phase is created by the release of all bus lines. Note that any BUS DEVICE which has been asserting both SEL and BSY must release SEL at least two DESKEW DELAYS before the release of BSY.

BUS DEVICES shall detect the BUS FREE phase by the simultaneous (within a DESKEW DELAY) condition of both SEL and BSY not asserted while the RELEASE and RESET conditions are not active.

During the BUS FREE phase, all active BUS DEVICES shall immediately release all bus lines (within a BUS CLEAR DELAY) after the BSY and SEL lines are released from the bus.

2.3.10 Notes for Signal Timing:

Unless otherwise indicated, the delay time measurements for each BUS DEVICE are calculated from signal conditions existing at that device's own BUS PORT. Thus normally these measurements need not consider delays in the bus cable.

2.3.11 Signal Restrictions Between Phases:

When the Bus is between two phases, the following conditions shall apply to the bus lines:

- The BSY, SEL, REQ and ACK lines shall not change.
- The C/D, I/O, MSG and DATA LINES may change.
- The ATN, and RST lines may change as defined under the descriptions for the ATTENTION and RESET conditions.

2.4 BUS CONDITIONS

The bus has two asynchronous conditions:

- ATTENTION Condition
- RESET Condition

These conditions cause certain BUS DEVICE actions and can alter the bus phase sequence.

2.4.1 Attention Condition

Purpose:

The ATTENTION condition allows an INITIATOR to inform a TARGET that the INITIATOR has a CONTROL MESSAGE ready. The TARGET may get this message at its convenience by performing a CONTROL MESSAGE OUT phase.

Bus Device Implementation Procedure:

The INITIATOR creates the ATTENTION condition by asserting ATN at any time except during the ARBITRATION or BUS FREE phases.

The TARGET may respond with the CONTROL MESSAGE OUT phase.

The INITIATOR may keep ATN asserted if more than one byte is to be transferred.

The INITIATOR releases the ATN line during (1) the RESET condition, or when the bus goes to a BUS FREE phase or (2) while the REQ line is asserted and the ACK line is not yet asserted during the last REQ/ACK handshake of a CONTROL MESSAGE OUT phase.

2.4.3 Reset Condition

Purpose:

The RESET condition is used to immediately clear all BUS DEVICES from the bus and to reset these devices and their associated equipment (as required).

Bus Device Implementation Procedure:

Note: This condition takes precedence over all other phases and conditions.

The RESET condition can occur at any time.

Any BUS DEVICE (whether active or not) can create the RESET condition. The RESET condition should be used with caution because of its possible effects.

The RESET condition is created by the assertion of the RST line.

When the RESET condition exists, all BUS DEVICES will immediately (within a BUS CLEAR DELAY) release all bus lines except RST itself. In addition all BUS DEVICES and their associated equipment shall be reset to initial conditions (as required).

The RESET condition shall be on for a minimum of a RESET HOLD TIME.

During the RESET condition, no bus line except RST is guaranteed to be in a valid state.

Regardless of what bus phase may have been interrupted, following the RESET condition the bus shall go to a BUS FREE phase and then start a normal phase sequence.

2.5 PHASE SEQUENCING

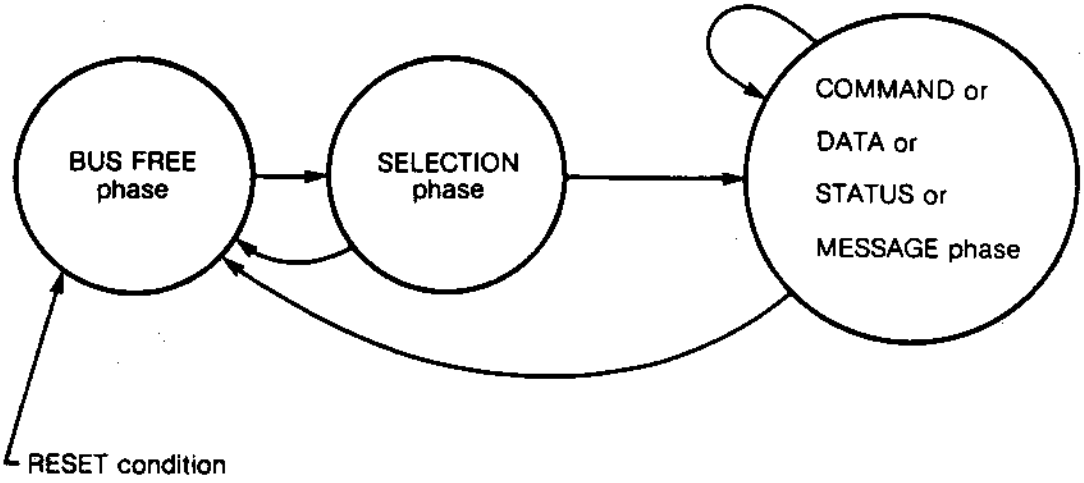
The order in which phases are used on the bus follow a prescribed sequence.

In all systems, the RESET condition can interrupt any phase and is always followed by the BUS FREE phase. Also any other phase can be followed by the BUS FREE phase.

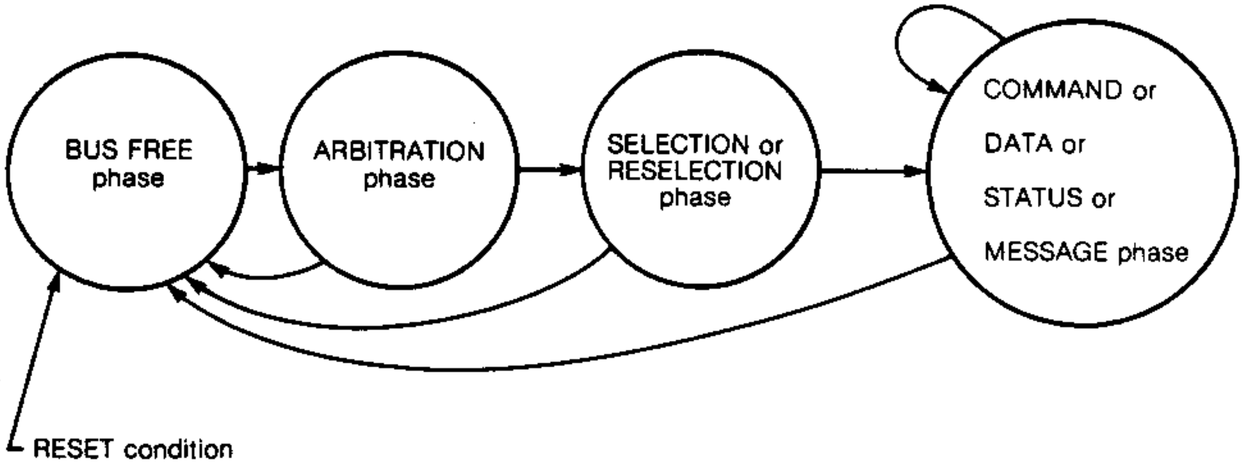
In sytems where the ARBITRATION phase in not implemented, the allowable sequencing is shown in the figure 2.2. The normal progression would be from the BUS FREE phase to SELECTION, and from SELECTION to one or more of the INFORMATION TRANSFER phases (COMMAND, DATA, STATUS or MESSAGE).

In systems where the ARBITRATION phase is implemented, the allowable sequencing is shown in figure 2.3. The normal progression would be from the BUS FREE phase to ARBITRATION, from ARBITRATION to SELECTION or RESELECTION, and from SELECTION or RESELECTION to one or more of the INFORMATION TRANSFER phases (COMMAND, DATA, STATUS, or MESSAGE).

There are no restrictions on the sequencing between INFORMATION TRANSFER phases. A phase may even follow itself (e.g. a DATA phase may be followed by another DATA phase).



PHASE SEQUENCING
(For systems with no arbitration)
FIGURE 2.2



PHASE SEQUENCING
(For systems with arbitration)
FIGURE 2.3

2.6 ELECTRICAL DESCRIPTION

All assigned lines are terminated with 220 ohms to +5 volts and 330 ohms to ground at each end of the cable.

All signal lines use open collector or three-state drivers as noted in section 2.2.

Each line driven by a BUS DEVICE shall have the following output characteristics when measured at the device's BUS PORT connection:

True = Signal Assertion = 0.0 VDC to 0.5 VDC @ 48 ma (max)

False = Signal Non-assertion = 2.5 VDC to 5.25 VDC

Note: For these measurements bus termination is assumed to be external to the port.

Each line received by a BUS DEVICE shall have the following input characteristics when measured at the device's BUS PORT connection:

True = Signal Assertion = 0.0 VDC to 0.8 VDC @ 0.8 ma (max)

False = Signal Non-assertion = 2.0 VDC to 5.25 VDC

Note: For these measurements bus termination is assumed to be external to the port.

2.7 PHYSICAL DESCRIPTION

Cable Requirements:

A fifty (50) conductor flat cable (or twisted pair flat cable) shall be used. The maximum cable length shall be 6.0 meters.

Each BUS PORT shall have a 0.1 meter maximum stub length of any conductor when measured from the bus cable.

Bus termination may be internal to the BUS DEVICES that are at the ends of the bus cable.

The cable pin assignment is shown in the Table 1.0.

Connector Requirements:

The connector shall be a fifty (50) conductor flat cable connector.

SIGNAL	PIN NUMBER	
-DB(0)	2	
-DB(1)	4	
-DB(2)	6	
-DB(3)	8	
-DB(4)	10	
-DB(5)	12	
-DB(6)	14	
-DB(7)	16	
-DB(P)	18	
—	20	for future usage
—	22	
—	24	
—	26	
—	28	
—	30	
-ATN	32	
SPARE	34	for future use (TERMINATE AS A SIGNAL LINE)
-BSY	36	
-ACK	38	
-RST	40	
-MSG	42	
-SEL	44	
-C/D	46	
-REQ	48	
-I/O	50	

Note:

All signals are low true. All odd pins are connected to ground.

3.0 COMMAND AND STATUS

The following list of commands and error status is presently defined. Note the number of commands and status available for expansion.

CLASS 0 COMMANDS

00	Test Drive Ready
01	Recalibrate Drive
02	Request Syndrome
03	Request Sense
04	Format Drive
05	Check Track Format
06	Format Track
07	Format Bad Track
08	Read
09	Write Protect Sector
0A	Write
0B	Seek
0C	Reserved
0D	Verify Restore
0E	Assign Alternate Disk Track
12	Reserve Drive
13	Release Drive
14	Write Protect Drive
15	Release Write Protect
16	Read, no seek, blocks
17	Search Data Equal
18	Search Data High
19	Search Data Low
1A	Read Diagnostic
1B	Verify Data

CLASS 1 COMMANDS

00	Copy
01	Restore
02	Backup
06	Set Block Limits

CLASS 2 COMMANDS

00	Load
01	Unload
02	Rewind
03	Space Forward
04	Space Forward File Mark
05	Space Reverse
06	Space Reverse File Mark
07	Track Select
08	Read
09	Read Verify
0A	Read Diagnostic
0B	Write
0C	Write File Mark
0D	Write Extended
0E	Write Extended File Mark
0F	Write Erase

CLASS 3 COMMANDS

00	Skip
01	Space
02	Return
03	Tab
04	Read Control
05	Write
06	Write Control

CLASS 4 COMMANDS

None defined

CLASS 5 COMMANDS

None defined

CLASS 6 COMMANDS

00	Define Floppy Disk Track Format
04	Format Drive (Error Map, Assign Alternates)
05	Read Error Map
06	Read Drive Type

CLASS 7 COMMAND

00	RAM Diagnostic
01	Write ECC
02	Read ID
03	Drive Diagnostic

ERROR STATUS TYPE 0

0	No Status
1	No Index Signal
2	No Seek Complete
3	Write Fault
4	Drive Not Ready
5	Drive Not Selected
6	No Track 0
7	Multiple Drives Selected

ERROR STATUS TYPE 1

0	ID Read Error (CRC)
1	Uncorrectable Data Error
2	ID Address Mark Not Detected
3	Data Address Mark Not Detected
4	Record Not Found
5	Seek Error
6	DMA Timeout Error
7	Write Protected
8	Correctable Data Check
9	Bad Block Found
A	Interleave Error
B	Command Buffer Full
C	Interface Timeout Error
D	Search Not Met

ERROR STATUS TYPE 2

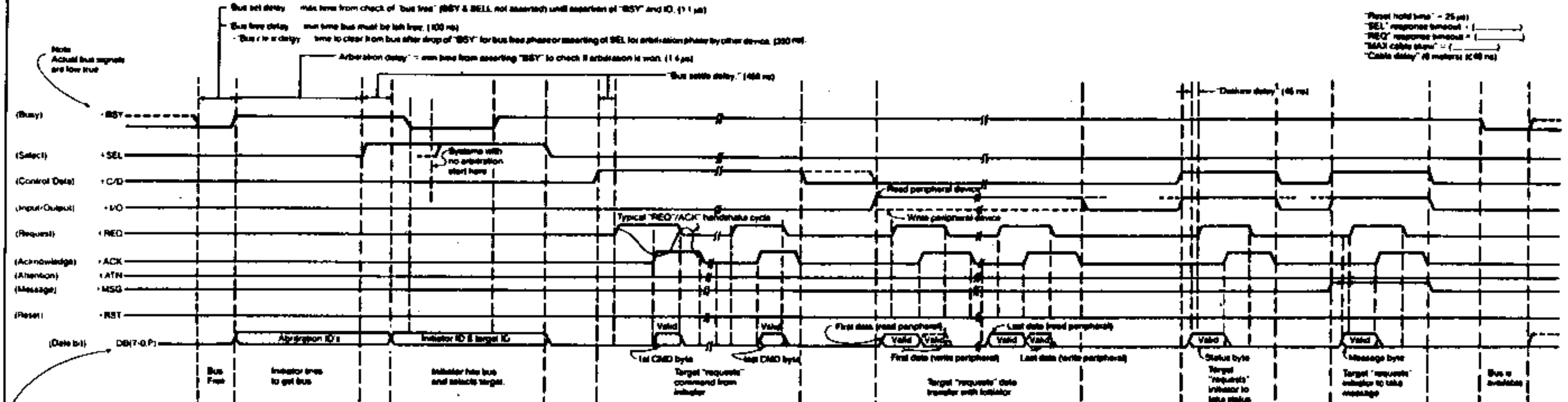
0	Invalid Command
1	Illegal Disk Address

ERROR STATUS TYPE 3

0	RAM Error

ERROR STATUS TYPE 4

0	Tape Transfer Incomplete
1	Drive Write Protected
2	Cartridge Not Loaded
3	Tape Full
4	Detected Unrestored Disk Block
5	Detected Unexpected Hole
6	Tape Drive Not Powered On
7	Tape Read/Write Error


"Bus set delay" max time from check of "bus free" (BSY & SELL not asserted) until assertion of "BSY" and ID. (1.1 μs)
"Bus free delay" min time bus must be left free. (100 ns)
"Bus clear delay" time to clear from bus after drop of "BSY" for bus free phase or asserting of SEL for arbitration phase by other device. (800 ns)
"Arbitration delay" = min time from asserting "BSY" to check if arbitration is won. (1.4 μs)
"Bus settle delay." (400 ns)
"Deskew delay" (45 ns)
"Reset hold time" = 25 μs)
"SEL" response timeout = (________)
"REQ" response timeout = (________)
"MAX cable skew" = (________)
"Cable delay" (6 meters) (<40 ns)
Note
Actual bus signals are low true
(Busy) +BSY
(Select) +SEL
(Control/Data) +C/D
(Input/Output) +I/O
(Request) +REQ
(Acknowledge) +ACK
(Attention) +ATN
(Message) +MSG
(Reset) +RST
(Data bit) DB(7-0,P)
Systems with no arbitration start here
Read peripheral device
Write peripheral device
Typical "REQ"/"ACK" handshake cycle
Arbitration ID's
Initiator ID & target ID
Valid
Valid
1st CMD byte
last CMD byte
First data (read peripheral)
Valid
Valid
First data (write peripheral)
Last data (read peripheral)
Valid
Valid
Last data (write peripheral)
Valid
Status byte
Valid
Message byte
Bus Free
Initiator tries to get bus
Initiator has bus and selects target.
Target "requests" command from initiator
Target "requests" data transfer with initiator
Target "requests" initiator to take status
Target "requests" initiator to take message
Bus is available

Note
DB(7) { = Most significant bit
= Highest priority ID for arbitration

Note
DB(P) = Data parity (odd)
Parity is not valid during arbitration

The use of parity is a system option

Note
In a typical system, a computer's host adapter will act as the "initiator," and an I/O device's control unit will act as the "target."

ARBITRATION PHASE

- After initiator sees that bus is free ("BSY" & "SEL" are not asserted) it waits a min. of "bus free delay" and a max. of "bus set delay" and asserts "BSY" and its own ID on the data bus.
- After the "arbitration delay," the initiator checks the data bus and clears itself from arbitration if a higher priority ID (DB(7) = highest) is on the bus.
- If "SEL" is asserted during arbitration by another device, the initiator will immediately clear itself from arbitration (within "bus clear delay" time)
- If the initiator determines that its own ID is the highest asserted, then it wins arbitration and asserts "SEL." (It then may not change any lines until after waiting a "bus settle delay")

SELECTION PHASE

- Initiator asserts data bus with desired target's ID & its own (initiator) ID.
- After two "deskew delays" the initiator drops "BSY."
- The selected target sees the condition of "BSY" not asserted while "SEL" & its ID are asserted. The target responds by asserting "BSY."
- After two "deskew delays" the initiator drops "SEL" and may change the data lines
- Note: After initiator drops "SEL," it is the only bus device that can assert "ACK" and ATN (and the data lines if "I/O" is not asserted (indicating output from initiator)
- Note: After the target sees drop of "SEL," the target is the only bus device that can assert "BSY," "C/D," "I/O," "MSG," "REQ," (and the data lines if "I/O" is asserted (indicating input to the initiator))
- Note: In systems in which the arbitration phase is not implemented, the initiator first detects the bus free phase ("BSY" & "SEL" not asserted) and then waits a "bus settle delay." Then the initiator asserts the data bus with the desired target's ID and its own (initiator) ID. Then after two "deskew delays" the initiator asserts "SEL." Then as described above, the target responds with "BSY" and the initiator drops "SEL."

COMMAND PHASE

- Target asserts "C/D and does not assert I/O" or "MSG."
- After a "bus settle delay" target asserts "REQ."
- Initiator puts data (command) on the bus and asserts "ACK."
- Target waits a "deskew delay," takes the data & then drops "REQ."
- Initiator drops "ACK" & data.

DATA PHASE

<Read peripheral device>

- Target asserts "I/O" and does not assert "C/D" or "MSG." Target then waits at least a "bus settle delay" before asserting "REQ."
- Target also puts data on the bus at least "deskew delay" before the assertion of "REQ."
- Initiator takes data and asserts "ACK."
- Data are no longer guaranteed valid.
- Target drops "REQ."
- Initiator drops "ACK."

<Write peripheral device>

- Target asserts "REQ" and does not assert "I/O," "C/D," or "MSG."
- Initiator asserts "ACK" & data.
- Target waits a "deskew delay," then takes the data & then drops "REQ."
- Data are no longer guaranteed valid
- Initiator drops "ACK."

STATUS PHASE (Ending status)

- Target asserts, "C/D" & "I/O" and does not assert "MSG." After a "bus settle delay," it asserts "REQ."
- Target also asserts data (status) at least a "deskew delay" before "REQ."
- Initiator takes data and asserts "ACK."
- Data are no longer guaranteed valid
- Target drops "REQ."
- Initiator drops "ACK" in response to the drop of "REQ."
- Note: In theory, more than one status byte could be sent

MESSAGE PHASE (Interrupt)

- Target asserts "C/D," "I/O," "MSG." After "bus settle delay" it asserts "REQ."
- Target also asserts data (message) at least a "deskew delay" before "REQ."
- Note: Message sent here could indicate command done interrupt.
- Initiator takes data and asserts "ACK."
- Data are no longer guaranteed valid
- Target drops "REQ."
- Then initiator drops "ACK" in response to the drop of "REQ."
- Note: In theory, more than one message byte could be sent

BUS FREE PHASE

- Target drops "BSY" to indicate that the bus is free
- Target and initiator get off the bus (within "bus clear delay" time) in preparation for a subsequent arbitration

Note: This is an example of a typical bus sequence. It is not a definition of bus protocol