pdp11 bus handbook

digital

CONTENTS

APPENDICES

PREFACE

Since the introduction of the PDP-11 in 1970, engineers have found PDP-11 bus structures easy to understand and easy to interface. This book describes the technical specifications of the PDP-11 UNIBUS and LSI-11 bus in a convenient reference document. We hope the book will assist users in configuring large, complex PDP-11 systems and aid in the design of interfaces for PDP-11's.

The reader is expected to be familiar with PDP-11 architecture. For further information on the PDP-11 architecture and instruction set, see the various processor and other handbooks available from Digital.

The UNIBUS and LSI-11 bus are protected by Digital patents. However, licenses are available under these patents for peripheral and memory devices to be connected to these buses. The PDP-11 family of computers includes other buses whose specifications are not available to customers. These include the high speed Massbus, the multidrop DECdataway, and the network interconnecting PCL-11 bus. Each of these is briefly characterized in an appendix to this book.

An excellent discussion of buses in general, and the UNIBUS and LSI-11 bus in particular, can be found in the book **COMPUTER ENGINEERING, A DEC View of Hardware Systems Design,** written by C. Gordon Bell, J. Craig Mudge, and John E. McNamara, and published by Digital Press; Bedford, Massachusetts; copyright 1978.

part 1 unibus specification

PART 1

UNIBUS SPECIFICATION

INTRODUCTION

Part 1 of this handbook defines the terminology and specifies the minimum requirements of the PDP-11 Unibus. All products designed at DIGITAL for use on the Unibus must conform to the Unibus Specification before being considered for production release. Users are advised to follow the technical specifications carefully in designing custom interfaces.

Scope

This specification applies to the PDP-11 family of central processor units that use a single high-speed bus, known as the Unibus. Address, data, and control information are sent along the 56 lines of this bus. Each processor with a Unibus uses the same set of signals for communication on the bus.

This specification applies to PDP-11 Unibus central processor units (CPU) and options which use the Unibus.

Content

The Unibus specification is divided into the following sections.

1. Introduction—scope and contents.
2. Unibus Description—system architecture, definitions, priority structure, protocol introduction, and types of data transfers.
3. Unibus Signal Details—types of Unibus lines and interrelationship among signals.
4. Unibus Protocol—priority arbitration, data transfers, and initialization.
5. Unibus Interface Design Guidelines—bus drivers and receivers, placement of these on an interface, wiring backplanes, deskewing signals, and a discussion of the master device.
6. Unibus Configuration—definitions, rules, and guidelines to configure a system; bus elements, bus loads, and bus length.
7. Unibus Hardware—cables, jumpers, terminators, and backplane pin assignments.

UNIBUS DESCRIPTION

This section defines the Unibus components in general terms, the arrangement of these components into a system, and the various types of operations that are transacted in a system. Greater technical detail follows in later sections.

A bus is an electrical conductor that carries current or signals to several other electronic circuits. The Unibus is a set of electrical circuits defined for the PDP-11 family of central processors. The Unibus is defined by its architecture, its protocol, and its electrical characteristics. The word "bus" when used in Part 1 of this book is synonymous with the word "Unibus."

Architecture—The Unibus is a linear bus. It consists of a multi-conductor transmission medium to which the components of a system are attached at various points, as shown in Figure 1-1.

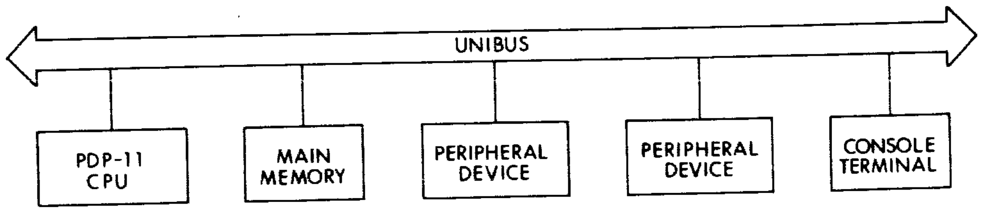


Figure 1-1 Unibus System Simplified Block Diagram

Unibus Transmission Medium—The Unibus transmission medium interconnects the component "bus devices" of a system. The transmission medium consists of the following elements.

1. A flat flexible cable containing 56 signal lines and 64 corresponding ground lines.
2. Drivers and receivers for the signals transmitted on the 56 signal lines. A "driver" (or bus driver) is a circuit used by a device to transmit signals to the Unibus; a "receiver" (or bus receiver) is a circuit used by a device to receive signals from the Unibus. These drivers and receivers are physically located in the devices but are electrically part of the Unibus. A device contains drivers and/or receivers only for the bus signals that it uses.
3. The interconnect between the flat cable and the output of the drivers or the input to the receivers.

The 56 signal lines are grouped logically into three types.

1. Initialization (3 lines): Signals on these lines control power-up, power-down, or initialization sequences of the bus devices;
2. Data (38 lines): These lines are used for data transfer between devices; and
3. Priority Arbitration (15 lines): Signals on these lines decide which device will next be allowed to control the data section.

Bus Terminator—The Unibus, as a transmission medium or transmission line, has a characteristic impedance and must be properly terminated. A Unibus is terminated at both ends by a bus terminator.

Bus Segment—A bus segment is that portion of a Unibus system between two terminators. A system may consist of one or more segments. The number of devices that may be connected to a segment is limited, as is the length of its cable.

Bus Repeater—A bus repeater is a device used to interconnect two segments of a multi-segment Unibus system. A repeater receives the signals from one segment and retransmits them onto the other segment. Its purpose is twofold:

1. It prevents signal levels from becoming degraded due to the connection of too many devices to a bus.
2. By receiving and then retransmitting all signals going between one segment and the next, it ensures that the proper timing relationship between signals is maintained.

Bus Master—The bus master is the device or processor that is currently permitted to use the data section of the Unibus. Only one device may be master at a given time. Typically, a master uses the data section of the bus to transfer data between itself and another device which is called "slave." A typical example of this relationship is the processor, as master, fetching an instruction from memory (which is always a slave).

Bus Slave—The bus slave is the device that communicates with the bus master. Only one device may be the bus slave at a given time.

Some devices may become both master and slave (at different times), while other devices may become only master or only slave.

Bus Arbitrator—The bus arbitrator is a logic circuit that compares priorities from devices requesting the use of the data section of the bus in order to determine which device is to be granted control next (i.e., become next bus master). The processor, as bus arbitrator, may pass bus control to a Direct Memory Access (DMA) device. The DMA device, as bus master, could then communicate with memory (always a slave) without processor intervention.

The arbitrator may or may not be part of a processor. There must be one and only one arbitrator on a Unibus. A system may have more than one Unibus, each with its own arbitrator.

Bus Request—A request by a master device to use the data section of the bus. The request is sent to the arbitrator on one of five signal lines: BR4, BR5, BR6, BR7, and NPR.

Bus Grant—A grant is a signal to the requesting device that it may become next bus master. The grant is sent by the arbitrator on one of five signal lines: BG4, BG5, BG6, BG7, and NPG.

Processor—A processor is a bus device that includes the circuits that control the interpretation and execution of instructions. A processor does not include the Unibus, main memory, or peripheral devices. A processor may become master or slave.

Interrupt Fielding Processor—There may be more than one processor connected to a Unibus; however, there is normally only one "interrupt fielding processor." An interrupt fielding processor is a processor that has special connections to the arbitrator.

These special connections permit the interrupt fielding processor to service interrupt transactions on the Unibus.

UNIBUS SYSTEMS

The elements of the Unibus are interconnected to constitute a computing system. The function of the Unibus in the system is to allow data to be exchanged between devices as directed by the program. A program is a sequence of instructions as interpreted by a processor.

Data is transmitted on the bus either as 16-bit words or as 8-bit bytes. The data is exchanged between a master and a slave.

There can be only one master and one slave on the bus at any given time. The master determines which device will become slave by putting the address of the desired slave device on the bus. In order to become bus master, a device must request and obtain the use of the data section. This request may be made at any time the device is ready for a data transfer. Any number of devices may be asserting a request at the same time. The priority scheme implemented by the arbitrator determines which of these requests is honored (i.e., which device will obtain the use of the data section when it becomes free).

Several devices may be ready to transfer data at the same time. Since only one of these devices can obtain the use of the data section, the other requesting devices have to wait before being allowed to transfer data. If the wait is too long, some devices may lose data. This wait period is known as latency.

The concepts of signal lines, priority structure, address space, and latency are discussed in the following paragraphs. These concepts must all be taken into consideration when arranging devices on a Unibus.

Signal Lines

Previous literature has defined Unibus signal lines as being either "bidirectional" or "unidirectional." These terms are ambiguous and are not used in this description. They are only mentioned in this paragraph because of the previous usage.

Unibus signals may be divided into two general categories with respect to the manner in which they are transmitted. The majority of signals use lines that are, in effect, wired-OR circuits to which the inputs to the bus receivers and the outputs of the bus drivers are connected. These lines are thus available along the length of the Unibus to any device which needs to receive or to assert/negate the signals transmitted on the lines. These lines were in the past called "bidirectional lines" because a signal asserted or negated at any point on the line may be received at any other point on the bus. It should be noted, however, that some of the signals transmitted on lines of this type are logically, if not electrically, "unidirectional," For example: requests for permission to use the data section of the bus are asserted on a line of this type by devices that need to become bus master, but are received only by the arbitrator. This request signal is thus in effect "unidirectional," although transmitted on a "bidirectional" line.

The drawing in Figure 1-2 shows these wired-OR lines schematically.

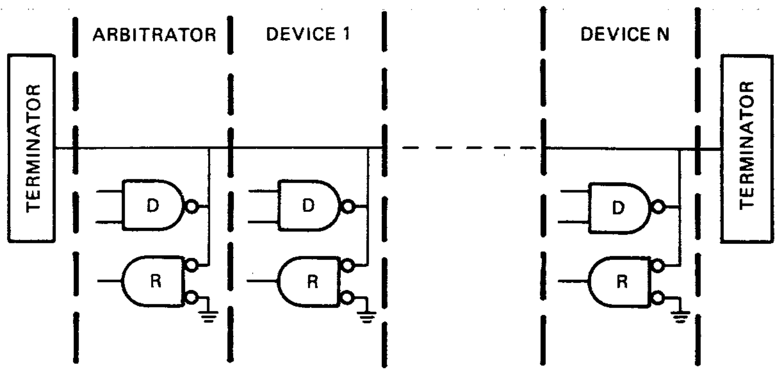


Figure 1-2 Illustration of Unidirectional Lines

Five Unibus signals use lines where the signal is received only by the device that is closest on that line to the origin of the signal; this receiving device, in turn, either retransmits the signal to the next device on the line ("passes" the signal) or does not retransmit it ("blocks" the signal). The transmission process continues until either a device blocks the signal or the end of the line is reached. These lines have been called "unidirectional lines." They are used only by the arbitrator to grant bus access permission to devices requesting the use of the data section of the bus. Figure 1-3 shows this type of line schematically.

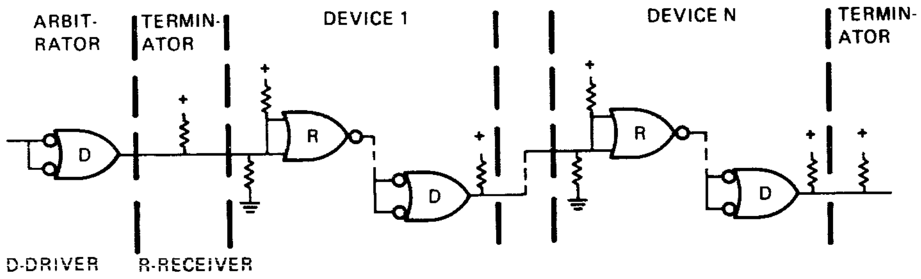


Figure 1-3 Requesting Bus Access

Priority Structure
A priority structure determines which requesting device will use the data section of the bus. The priority of a device is a function of (1) the priority level assigned to the device and (2) its position on the bus with respect to other devices of the same priority level.

All devices, with the exception of the interrupt fielding processor, may be assigned to one (or more) of five hardware priority levels. A signal line is dedicated to each of these levels. Each signal line is driven by all bus devices assigned to the priority level. These five lines are referred to as "request lines" and are monitored by the arbitrator. A device that requires the use of the data section of the bus asserts a request on one of these lines. This request is received by the arbitrator.

The arbitrator also monitors the priority level of the interrupt fielding processor. The arbitrator issues a grant to the highest priority request active if the interrupt fielding processor is not at a higher priority level.

A grant signal informs the requesting device that it may become next bus master. This signal is sent on a separate line than the request, either: NPG, BG7, BG6, BG5, or BG4. The level of the grant is the same as that of the request.

The arbitrator issues a grant to the first device on the bus assigned to the same priority level as the grant. If this device is requesting the use of the data section of the bus, it accepts and acknowledges receipt of the grant. It also blocks further transmission on the grant line. If the device is not requesting the use of the data section, it passes the grant to the next device on the same grant line. This procedure is repeated until a device accepts the grant or until the end of the bus is reached. In this last case, the grant is cancelled by the arbitrator and the arbitration process is restarted. Each device on a Unibus is assigned a discrete position in the priority structure. This position is determined by:

1. the priority level assigned to the device, and

2. the position of the device on the grant line with respect to the other devices of the same priority level.

All devices assigned to a given priority level have higher priority than any device at a lower level. Within a given priority level, the device closest to the arbitrator (origin of the grant signal) has the highest effective priority.

Address Space

An address is the name of a location in memory of a register. A number specifies the location. A Unibus device may contain one or more locations. A simple device such as a line clock may contain only one; a memory device may contain thousands. A device may be able to store many words and yet have only three or four address locations. An example of this is a disk or tape controller. An address location may be used for storage of data (i.e., data register) or for control of a device (i.e., peripheral address).

A PDP-11 16-bit word is divided into a low byte (bits 0-7) and a high byte (bits 8-15). A 16-bit word used for byte addressing can address a maximum of 32K words (or 64K bytes). However, the top 4,096 word locations are reserved for peripheral register addresses, and the user therefore has 28K words for programs. Systems with Memory Management options provide an 18-bit effective memory address which permit addressing up to 124K words of memory for programs.

A device must be bus master to use the address lines. An address put on the address lines by a master is received by all bus devices that are capable of becoming slaves: one of these devices recognizes its address and becomes the slave; the slave does not know which device is master, nor where the master is physically located on the bus. The master does not know the physical location of its slave, nor does it need to know.

Each external I/O device in a computing system has an external page address assigned to it. The use of this address on the Unibus is defined as part of the PDP-11 I/O page.

Latency

Latency is the delay between the time that a device initiates a request and the time that it receives a response. Nonprocessor request (NPR) or Bus request (BR) latency is the total time consumed by all the operations that occur between the instant a device makes a request and the moment it becomes bus master. (NPR and BR device requests are further defined in later sections.) The delay encountered is a function of current bus activity, the types of other devices in the system, and the arrangement or configuration of the equipment along the bus.

Maximum tolerable latency is the longest time that a device can wait for service before losing data. The service time is measured from the assertion of a request by the device to the time that the requested data transfer is complete.

Arrangement of Devices on the Unibus

The arrangement of devices on the Unibus is a function of the following four factors.

1. Any device can communicate with any other device on the data section of the bus. Physical position of the device is not important to its ability to communicate on the Unibus.
2. Grants are issued by the arbitrator and received and reissued by each device of the same level. All devices capable of becoming bus master must be on the same side of the arbitrator. In reality, the arbitration is on the CPU module. Most bus devices are located on one side of the CPU module.
3. The maximum tolerable latency for each device must not be exceeded. This depends on:
   a. the priority level assigned to the device,
   b. its position on its grant signal line relative to other devices of same priority level,
   c. the effect on the bus of all other devices at their priority levels, and
   d. the length of the bus.
4. The maximum length of the Unibus cable should not exceed 15.24 m (50 ft). A bus repeater can drive an additional 50 ft. More information on determining bus length is found in the section on Unibus Configuration.

NOTE

The interrupt fielding processor is typically next to the arbitrator on the bus. An arbitrator is built into each PDP-11 processor module.

A typical Unibus configuration is shown in Figure 1-4. The bus is terminated at both ends. The processor module terminates one end of the bus; a terminator module terminates the other end. Various devices (such as memories, peripheral controllers, and input/output devices) are placed between these modules.

Expanding the Unibus System

If a system exceeds the limitations set forth in this document, the bus must be divided into bus segments that are connected by a bus repeater. The limitations apply to the maximum number of devices on a segment (bus loading) and to the length of the bus cable. Figure 1-5 shows a two-segment Unibus configuration.

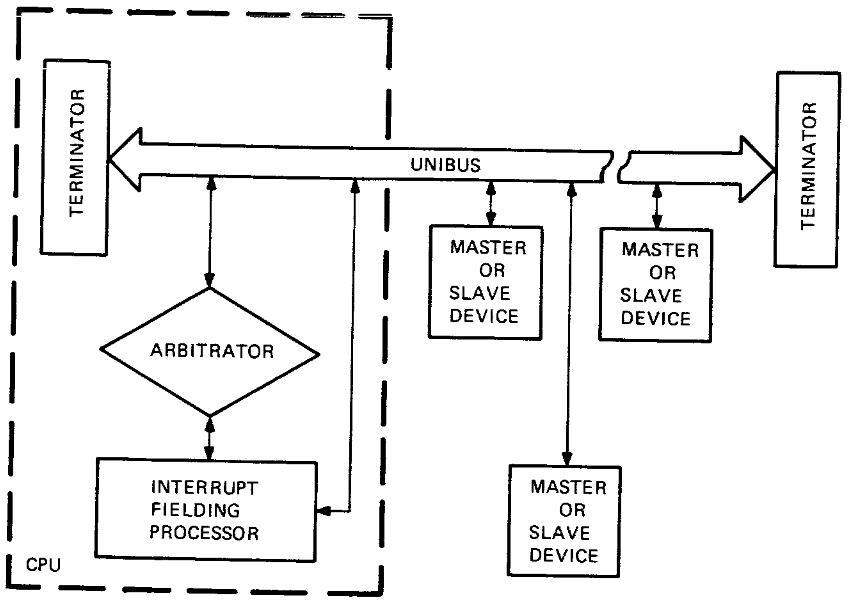


Figure 1-4 Typical Unibus Configuration

A device that can only be a slave, never a master, (such as a memory built into the CPU module) never requests nor receives grants. It may be physically located on either side of the arbitrator, as shown in Figure 1-5b.

PROTOCOL

The Unibus protocol defines the procedures that are used for communication on the bus. Communication between devices is in the form of transactions. A transaction is a sequence of signals that complete a logical unit of activity on the Unibus.

Types of Transactions

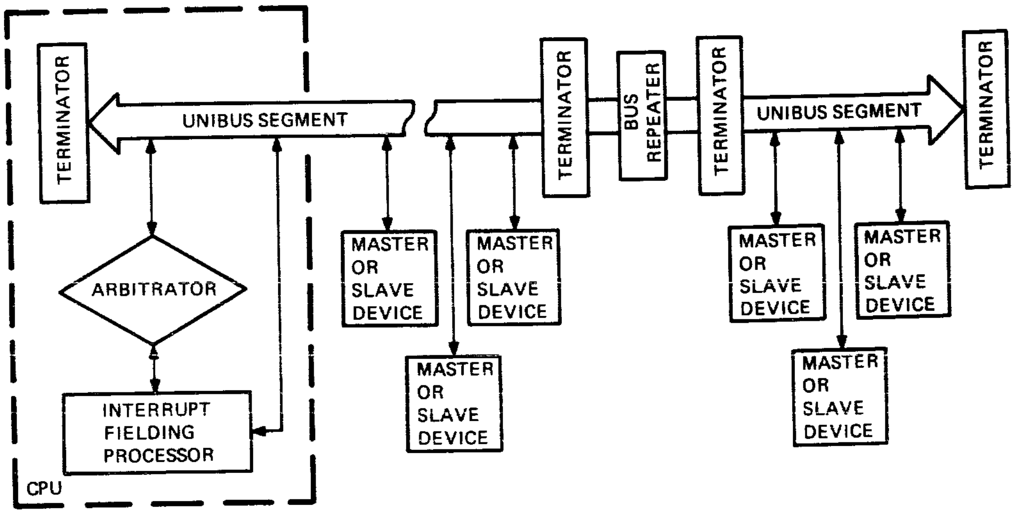
There are three types of Unibus transactions.

1. priority arbitration,
2. data transfer, and
3. initialization.

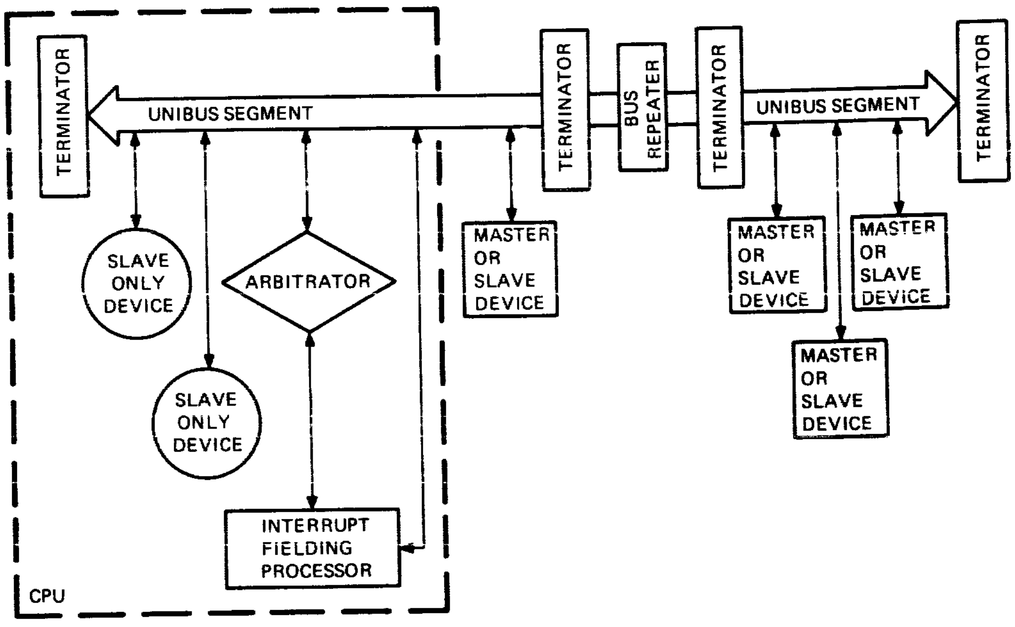
Each of these occurs on a separate section of the Unibus. This allows simultaneous priority arbitration and data transfer transactions. For example, the next master can be selected while the current master executes its data transfer.

Priority Arbitration Transactions

Priority arbitration determines which device will obtain control of the bus. In order to transfer data on the data section of the bus, a device must become bus master. No change of bus master occurs during the priority arbitration sequence. The sequence selects the device to become the next master when the current



(A) TWO-SEGMENT UNIBUS



(B) TWO-SEGMENT UNIBUS WITH SLAVE DEVICES ON BOTH SIDES OF ARBITRATOR

Figure 1-5 Two-Segment Unibus System

master releases the data section of the bus. Any number of devices may request the use of the data section at one time. The one device that is selected to be bus master is the one with the highest effective priority.

The highest priority device grant is called a non-processor grant (NPG). It is used by a device that does not require processor time. It may be used by devices **only** for data transfers that do not require interrupts. When requested, an NPG grant is issued on this level.

NOTE

An NPG is issued by the arbitrator for an NPR request. NPR is the highest level request; NPG is the highest level grant.

More than one device may be connected to the Unibus at the NPR level of priority. In the case of a tie for an NPR request, the device closest to the arbitrator is granted the NPG (on the NPG line).

When no request or grant is active at the NPG priority level and if there is no BG issued, the interrupt fielding processor may become bus master. The priority arbitration is affected by the priority state of the CPU.

The CPU program execution priority (PRI) varies from 0 to 7. The Unibus arbitrator grants use of the bus to non-CPU devices by the following rules:

1. At any time when an NPR is received, assert NPG. (A controller may do direct memory access data transfers at any time.)
2. When the CPU is between instructions, then:
   a. If PRI <7 and BR7 is received, assert BG7;
   b. If PRI <6 and BR6 is received, assert BG6;
   c. If PRI <5 and BR5 is received, assert BG5;
   d. If PRI <4 and BR4 is received, assert BG4.

   (The CPU will accept interruptions from a controller whose priority is greater than the current program execution priority of the CPU.)

Data Transfers

A data transfer is defined as the transmission of data between a master and a slave. This transfer can occur without processor intervention or supervision.

The device that has been granted next bus master waits for the Unibus to be released by the current bus master. When a device becomes bus master, it causes one or more words of data to be transferred between itself and a slave device. The particular slave and the direction of transfer are determined by the information issued by the bus master onto the Unibus. When the transfer is complete, the master releases the bus, at which time a new device may assume mastership. A device in control of the bus can transfer data at the maximum rate allowed by the combination of the master, the slave, and the bus.

An interrupt is a special type of data transaction to which only the interrupt fielding processor can respond as slave.

There are four types of data transfer transactions on the Unibus: data-in, data-out, read/modify/write, and interrupt.

1. **Data-in** is the transfer of one word from a slave to a master.
2. **Data-out** is the transfer of one word of data from a master to a slave; **data-out, byte** is the transfer of one byte of data from a master to a slave.
3. **Read/modify/write** is a transaction in which data is transferred from a slave location to a master, modified by the master, and transferred back to the same slave location. A read/modify/write consists of a **data-in, pause** followed by a **data-out** or by a **data-out byte.** The **data-in, pause** is identical to the **data-in;** however, it informs the slave that a **data-out** transaction to the same location will follow the **data-in.** If the slave is a destructive readout device (e.g., a core memory), it automatically restores the data after a **data-in.** After a **data-in, pause,** the slave might not restore the data but may wait for the modified data from the following **data-out.**
4. An **interrupt** transaction is the transfer of one word, the "interrupt vector," from the master to the interrupt fielding processor (CPU).

The fact that data transfers may be executed without processor intervention or supervision permits operations such as a disk directly refreshing a CRT display or transfers between the memory and a mass storage device.

Processors may use data transfers for instruction fetch and for control of other devices by modification of their control registers, as well as for the manipulation of information.

Initialization

The initialization section of the Unibus continuously monitors the AC power input to the bus power supplies. It also controls the orderly power-up and power-down of all bus devices.

Individual devices may be initialized under program control. Bus initialization stops all bus operations and puts all bus devices in a known well defined state.

UNIBUS SIGNAL DETAILS

The Unibus consists of 56 signals. Standardized control logic makes it possible to use separate dedicated lines for all signals. For example, a bus master provides the address of a location it wishes to access. The slave device responds. Control and timing signals are provided. Address, control, data, and timing functions each use a distinct set of bus lines.

All bus activity is asynchronous and depends on interlocked control signals. In every case, a control signal transmitted is acknowledged by the receiver of that signal.

Unibus Sections and Signal Lines

Although the Unibus is a single communication path for all devices in a PDP-11 computer system, the bus actually consists of three interrelated sections:

the priority arbitration section,
the data transfer section, and
the initialization section.

These sections and their signal lines are shown in Table 1-1.

Table 1-1
Unibus Sections and Signal Lines

Name	**Mnemonic**	**No. of Lines**	**Function**	**Assertion Level**
Data Transfer Section				
Address	A <17:00>	18	Selects slave device and/or memory address	Low
Data	<D 15:00>	16	Information transfer	Low
Control	C0, C1	2	Type of data transfer	Low
Master Sync	MSYN	1	Timing control for data transfer	Low
Slave Sync	SSYN	1		Low
Parity	PA, PB	2	Device parity error	Low
Interrupt	INTR	1	Interrupt	Low
Priority Arbitration Section				
Bus Request	BR4, BR5, BR6, BR7*	4	Requests use of bus (usually for interrupt)	Low
Bus Grant	BG4, BG5, BG6, BG7*	4	Grants use of bus (usually for interrupt)	High
Non-Processor Request	NPR	1	Requests use of bus for data transfer	Low
Non-Processor Grant	NPG	1	Grants use of bus for data transfers	High
Selection Acknowledge	SACK	1	Acknowledges grant	Low
Bus Busy	BBSY	1	Indicates that the data section is in use	Low
Initialize Section				
Initialize	INIT	1	System reset	Low
AC Low	AC LO	1	Power monitoring	Low
DC Low	DC LO	1		Low

* "BRn" and "BGn" are abbreviations used to designate that one pair of Bus Request (BR) and Bus Grant (BG) lines is used.

All transactions on the priority arbitration section and on the data transfer section are interlocked dialogs between devices. The requesting devices and the arbitrator communicate on the priority arbitration section. The bus master and the bus slave communicate on the data transfer section.

The signals that delimit or establish the boundaries for data operations are:

Data Transfer: MSYN, SSYN
Interrupt: INTR, SSYN

The signals that delimit priority arbitration are:

Priority Arbitration: [NPR, NPG] or [BRn, BGn], SACK, BBSY

In all bus transactions, the assertion and the negation of a signal used in the interlocking dialog have different meanings. A device may assert a signal to indicate the initiation of a process and negate the same signal to note the completion of the same process. The important event is the transition from one state to the other. For example, in a **data-in** operation, the bus master asserts MSYN to request data from the slave, and negates MSYN to acknowledge receipt of the data from the slave. The assertion and the negation of MSYN can be considered two different signals issued by a bus master on a single line. The following definitions apply to this specification.

"The assertion of..." or "at the assertion of..." signifies the transition of a signal from the negated state to the asserted state.

"While... is asserted" refers to the interval between the assertion and the negation of a signal. While asserted, a signal is in the logically true state.

"The negation of..." or "at the negation of..." signifies the transition from the asserted state to the negated state.

"While... is negated" refers to the interval between the negation and the assertion of the signal. While negated, a signal is in the logically false state.

In tables and diagrams, a 0 may be used in place of "while negated" and a 1 in place of "while asserted."

Signal Transmission
Bus transmission delay and skew affect signal transmission on the Unibus.

Bus Transmission Delay—Bus transmission delay is the length of time taken by a signal to travel from one device on the bus to another. The delay is measured from the time of the assertion or negation of the signal at the input to the driver of the sending device to the time at which the assertion or negation arrives at the output of the receiver of the receiving device.

Skew—Skew is the propagation delay due to variations in electronic circuits. Two signals starting at the same time, from one device to another, experience a time difference on arriving at the second device, even if similar circuitry and transmission media are used. This time difference (or time uncertainty) is called skew. It is caused mainly by the variation in the transmission delay time through the Unibus drivers and receivers, and by different loading of the bus by devices connected to the Unibus. Figure 1-6 illustrates this time differential.

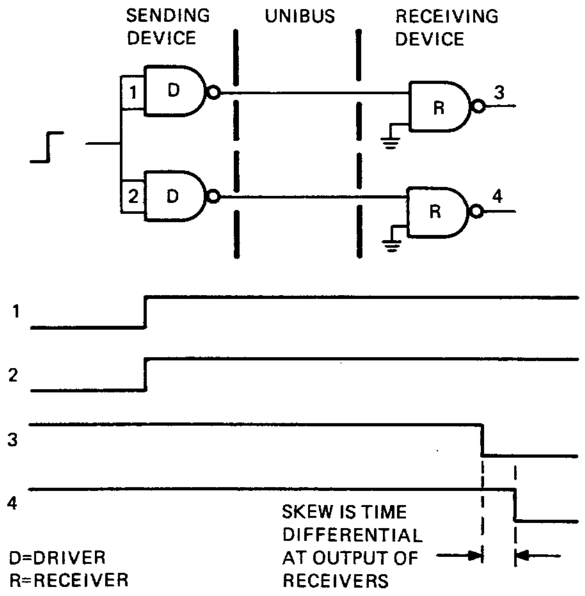


Figure 1-6 Example of Skew Time Differential

Maximum skew is the greatest difference in arrival time of two signals on the Unibus. This is measured from the inputs of two bus drivers in the first device to the output of receivers in the second device. Maximum skew on the Unibus is 75 ns. This includes the effects of drivers, receivers, and cables.

Deskew—To deskew is to introduce a delay in a circuit to compensate for skew.

A master is responsible for deskewing on all data transfer transactions with the exception of interrupt. The arbitrator and interrupt fielding processor are responsible for deskewing on interrupt transactions. The arbitrator is responsible for deskewing on priority arbitration transactions. By assigning these responsibilities, the amount of logic elements required throughout the system is minimized. The number of master devices on a system are generally fewer than slave devices. Also, there is only one arbitrator and one interrupt fielding processor on a bus.

All delays in this Unibus specification refer to assertions or negations at the input to the Unibus drivers. Figure 1-7a shows an example of correct gating of data lines and a delayed control signal (MSYN); Figure 1-7b is an example of incorrect gating because the inverters do not all have the same propagation delay, thus introducing additional skew.

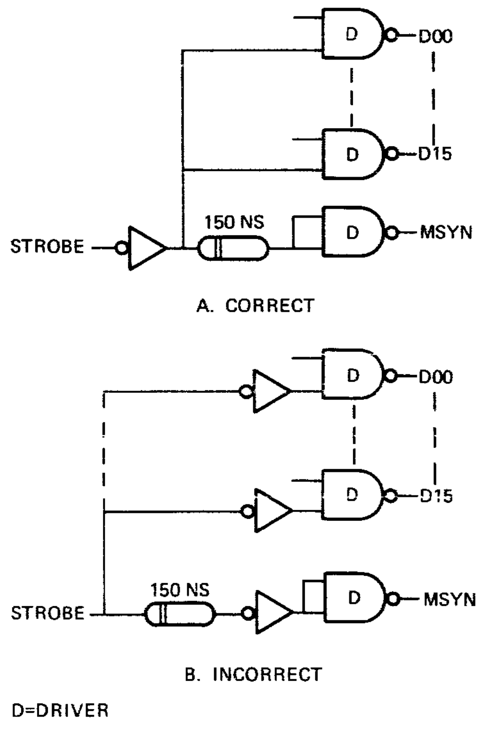


Figure 1-7 Example of Deskewing

Types of Unibus Signal Lines

Three types of signal lines are used on the Unibus. These are defined in the following paragraphs and illustrated in Figure 1-8.

NOTE

The drivers and receivers are considered part of the Unibus and NOT part of the device or processor in which they are physically located.

Type 1 Line—A type 1 line is a transmission line that has drivers or receivers connected in a logic OR configuration. If one device asserts a signal on this type line, all other devices receive the sig-

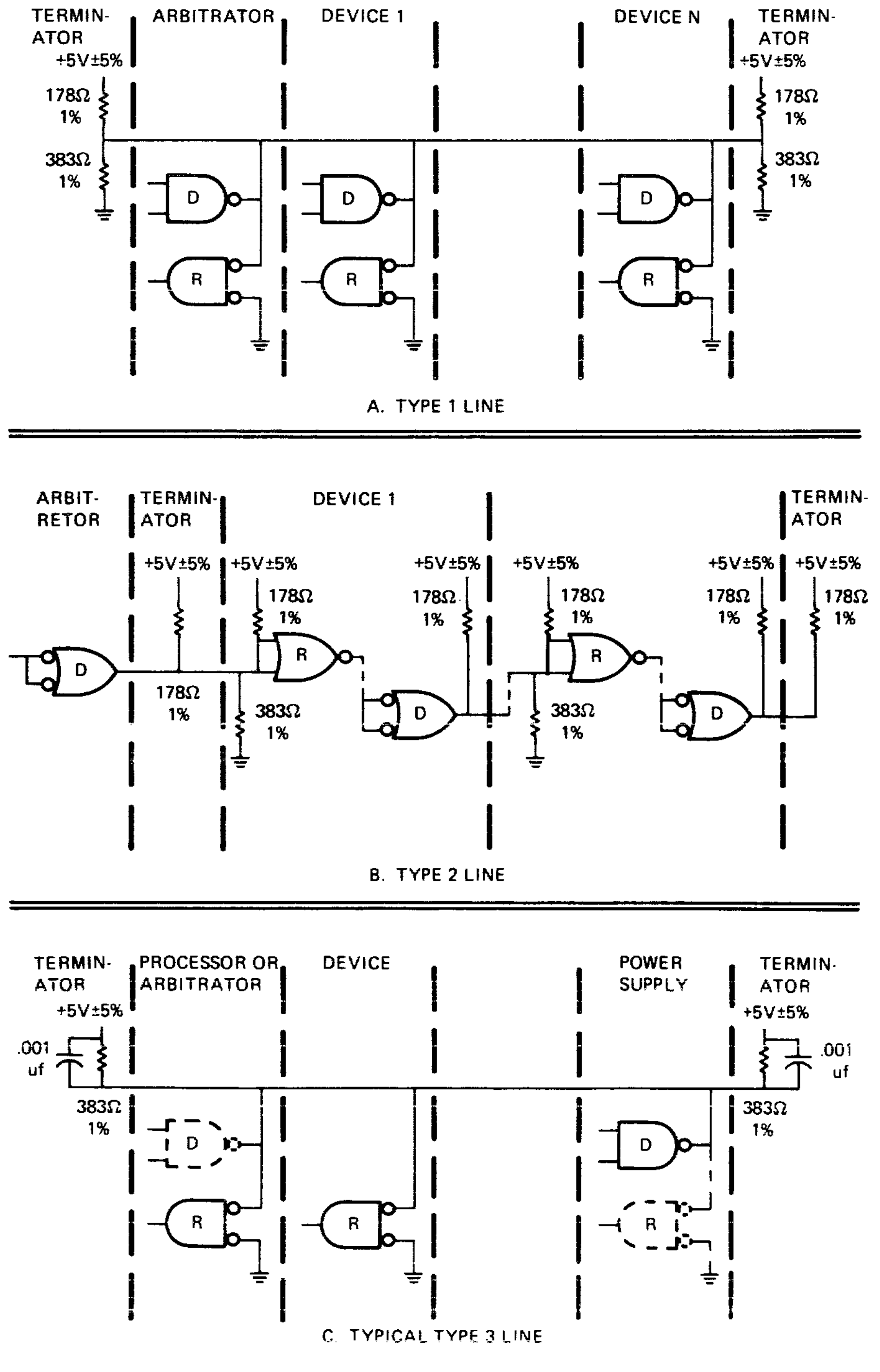


Figure 1-8 Types of Unibus Lines

nal, for example: bus busy. A line of this type is terminated at both ends of the Unibus by:

1. a resistor to + 5 Vdc and
2. a resistor to ground.

A device or processor may have drivers or receivers (or both drivers and receivers) connected to a type 1 line. Type 1 lines are used by all Unibus signals with the *exception* of NPG, BG7, BG6, BG5, BG4, AC LO and DC LO.

Type 2 Line—A type 2 line is a transmission line in which a signal is received only by the device electrically closest to it on that line. This device either transmits or does not transmit the signal to the next device on the same line. Each segment of a type 2 line is terminated at the receiver end by resistors to ground and to +5 Vdc and at the driver end by a resistor to +5V. Type 2 lines are used by Unibus grant signals: NPG, BG7, BG6, BG5, and BG4.

Type 3 Line—A type 3 line is a transmission line used for monitoring purposes. These lines are constantly at one state unless a failure occurs. These lines are terminated at both ends of the Unibus by a resistor and a capacitor in parallel to +5 Vdc. Type 3 lines are used for power monitor signals: AC LO and DC LO.

Priority Arbitration Section
Twelve signal lines are used on the priority arbitration section of the Unibus.

Transactions on this section of the bus occur between the priority arbitration network and devices that need to become bus master. A request (NPR or BRn) is asserted by one or more of these devices. The arbitrator issues a grant (NPG or BGn) for the highest priority device requesting the bus, and the requesting device electrically closest to the arbitrator at this priority level accepts and acknowledges the grant by asserting SACK. When the current data transaction is ended, the master releases the bus by negating BBSY, the new master asserts BBSY and starts its data cycle(s).

Requests are transmitted on type 1 lines and grants on type 2 lines.

A device that receives a grant and does not require the use of the data section of the bus retransmits the grant to the next device on the same grant line. This is known as "passing the grant."

NOTE

Lines used to pass a grant are also used to negate the grant.

A device that receives a grant and does require the use of the data section of the bus accepts the grant and does not retransmit it. This is called "blocking the grant."

If a grant is not acknowledged by any device on the bus, the grant is cancelled. This is done either by the arbitrator after a time out delay, or by a terminator at the opposite end of the bus from the arbitrator. This terminator asserts SACK if it receives a grant. Upon receipt of SACK, the arbitrator negates the grant. SACK is negated by the terminator upon receipt of this negation. In both cases, a new arbitration cycle is started. If a grant is cancelled due to the assertion of INIT (as part of a bus initialization procedure), arbitration is not resumed until receipt of the negation of INIT.

Non-Processor Request (NPR)—NPR is an asynchronous signal requesting the use of the data section of the bus. A device that requires the use of the bus to execute data transfers sends NPR to the arbitrator. The data transfers are made without active participation by the processor. The NPR signal may be asserted any time the device is ready to start a data transfer. An NPR has the highest priority. (A controller may do DMA transfers at any time.)

NPR is transmitted on a type 1 line.

NOTE

PDP-11/15 and PDP-11/20 data transfers are completed prior to NPR or BR service.

Non-Processor Grant (NPG)—The arbitrator generates the non-processor grant signal. Each device capable of asserting a non-processor request (NPR) receives and optionally retransmits this grant.

NPG is transmitted on a type 2 line.

One or more data transfers (commonly called "NPR transfers") may be executed while the device is bus master.

NOTE

An interrupt must not be attempted while doing NPR transfers by the device that became bus master under the authority of an NPG.

Assertion—The assertion of NPG by the arbitrator informs the first device on the NPG line (that has NPR active) that it may become next bus master. This occurs after the new master determines the current user of the data section of the bus is done. (The next master monitors $\overline{\text{BBSY}} \cdot \overline{\text{SSYN}} \cdot \overline{\text{NPG}}$ to determine when to assert BBSY.

If a device intends to assert Selection Acknowledge (SACK), it blocks the grant. If a device does not intend to assert SACK, it passes the grant.

NOTE

Priority arbitration is disabled while NPG is asserted.

Negation—The following rules apply to negate NPG.

1. The arbitrator acknowledges receipt of the assertion of SACK by negating NPG.
2. A device that is asserting SACK may not negate SACK before it has received the negation of NPG.
3. A device that is not asserting SACK passes the negation of NPG to the next device on the NPG line.

Bus Request (BR4, BR5, BR6, BR7)—A Bus Request is a signal requesting use of the data section of the bus. A device that requires use of the bus for the purposes of executing data transfers, executing an interrupt transaction, or both of these, sends a Bus Request to the arbitrator. The request is an interrupt. The request, the grant, and the data transfer together represent an interrupt transaction. Only one interrupt transaction may be executed under a single grant.

BR4, BR5, BR6, and BR7 are transmitted on type 1 lines.

Bus Grant (BG4, BG5, BG6, BG7)—A Bus Grant is a signal by the arbitrator. Each device capable of asserting the corresponding BR line (BRn) receives and optionally retransmits this bus grant. The arbitrator guarantees that a BG is never asserted unless the processor that receives interrupts is ready to accept an interrupt vector at that level.

BG4, BG5, BG6, and BG7 are transmitted on type 2 lines.

Assertion—The assertion of BGn by the arbitrator informs the first device on the BGn line (that has BRn active) that it may become next bus master. This occurs after the device determines the current user of the data section of the bus is done. (The next master monitors $\overline{\text{BBSY}} \cdot \overline{\text{SSYN}} \cdot \overline{\text{BGn}}$ to determine when to assert BBSY.)

If a device intends to assert Selection Acknowledge (SACK), it blocks the grant. If a device does not intend to assert SACK, it passes the grant.

NOTE

Priority arbitration is disabled while BGn is asserted.

Negation—The following rules apply to negate Bus Grant.

1. The arbitrator acknowledges receipt of the assertion of SACK by negating BGn.
2. A device that is asserting SACK may not negate SACK until it has received the negation of BGn.
3. A device that is not asserting SACK passes the negation of BGn to the next device on the BGn line.

Selection Acknowledged (SACK)—SACK is a signal sent to the arbitrator by a device that has received a grant.

SACK is transmitted on a type 1 line.

Assertion—The assertion of SACK is the acknowledgement by a device that it has accepted a grant. Priority arbitration is disabled while SACK is asserted.

Negation—The negation of SACK allows the start of a new priority arbitration cycle. The negation of SACK signifies that the master has almost finished using the data section of the bus. SACK is negated before the last data transfer by the current master. This allows the next bus master to be determined and ready when the current master is finished.

Bus Busy (BBSY)—BBSY is a signal sent by a bus master to all other bus devices.

BBSY is transmitted on a type 1 line.

Assertion—While asserted, BBSY informs all devices on the Unibus that a master exists. During this time no device other than the master may assert BBSY, MSYN, INTR, or use the address or control lines. The data lines may be used only by the master or by the slave designated by the address lines. SSYN may be asserted only by the slave; although it may or may not be negated before the master negates BBSY. Other devices may use signal lines in the priority arbitration section of the bus.

Negation—While negated, BBSY means there is currently no bus master.

Data Transfer Section
Forty-one signal lines are used for data transfer. In a data transfer, one device is a bus master and controls the transfer of data to or from a slave device.

All signals in the Data Transfer Section are transmitted on type 1 lines.

Data Lines, D<15:00>—The 16 data lines contain the word of information that is being transferred between the master and the slave devices. A word consists of two eight-bit bytes as shown in Figure 1-9. The low order byte contains bits 00 through 07; the high order byte, bits 08 through 15.

Address Lines, A<17:00>—The 18 address lines carry the 18 address bits from the master during a data transfer transaction. These bits specify a location. The device contains the specified location responds as the slave for this data transaction.

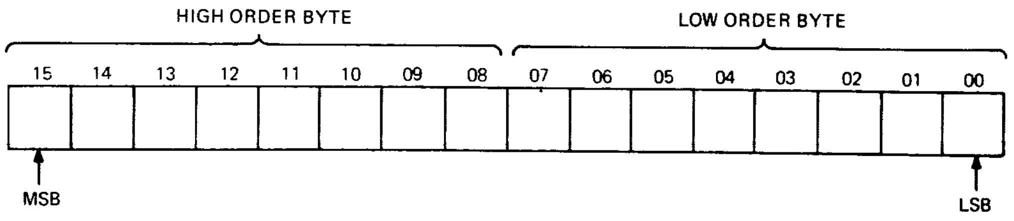


Figure1-9 High and Low Byte Format

The address format is shown in Figure 1-10. The 17 address lines A<17:01> specify a unique location. All locations contain a 16-bit word which is at an even address. A byte is half a word. In byte operations, bit A00 specifies which byte is being addressed. All words are located at an even address X (i.e., its A00 = 0), the low order byte is addressed at X and the high order byte at X plus 1.

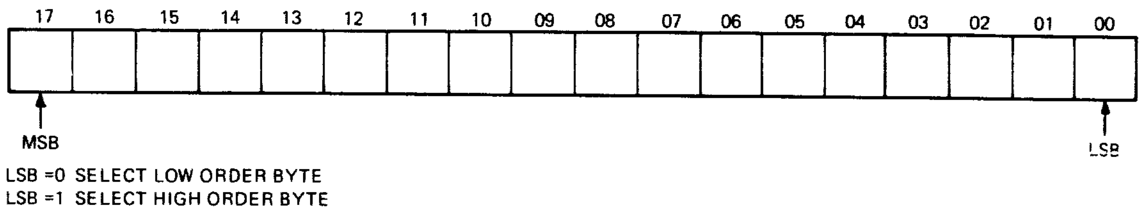


Figure 1-10 Address Format

Control Lines, C0, C1—The control lines are used by the master to indicate one of four possible data transfer operations. The operations are listed in Table 1-2 and defined in the following paragraghs.

NOTE

The direction of data transfer is always specified with reference to the master device; data-in is from slave to master, and data-out is from master to slave.

Data-In Transactions—The DATI and DATIP transactions request transfer of data from a slave to a master. Both transactions use the D lines to carry the data. These transactions are always a full word transfer; i.e., the slave places the data on D<15:00>. If the master wants only one byte, it must retrieve the data from the proper lines. For a low-order byte use D<07:00>; for a high-order byte use D<15:08>. For byte operations, the master can not assert bit A00 and the slave must ignore this bit.

DATIP Transaction—The DATIP operation is identical to the DATI, except DATIP informs the slave device that the present transfer is the first part of a read/modify, write cycle.

Table 1-2

Control Operations (C Lines)

Name	Mnemonic*	Value C1	Value C0	Function
Data In	DATI	0	0	One word of data from slave to master
Data In, Pause	DATIP	0	1	Same as DATI, but inhibits restore cycle in destructive read-out device. Must be followed by DATO or DATOB to the same location.
Data Out	DATO	1	0	One word of data from master to slave.
Data Out, Byte	DATOB	1	1	One byte of data from master to slave. Data is transferred on: D <15:08> for A00 = 1 D <07:00> for A00 = 0

* The notations DATI/P and DATO/B are equivalent to DATI (or DATIP) and DATO (or DATOB).

Example:
A pause flag is set in a destructive read-out device (e.g., core memory) that inhibits the restore cycle. The DATIP must be followed by a data-out cycle (DATO or DATO/B) to the same word address.

Since address bit A00 may change between a DATIP and a DATO/B, the slave must check the bus address at the beginning of the DATO/B. The master must retain bus control until this DATO/B is completed; i.e., it must remain bus master (assert BBSY) without interruption from the start of the DATIP cycle to the end of the DATOB cycle. No other data transfer may be executed between the DATIP and the DATO/B cycles.

In nondestructive readout devices (i.e., flip-flops), the DATI and DATIP are treated identically by the slave.

NOTE

In the case of locations which can be accessed by more than one Unibus or other bus (e.g., the PDP-11/45 semiconductor memory) a DATIP on one bus must prevent the slave from responding on any other bus until the DATO/B cycle has been completed. This is necessary to avoid problems in multiple processor systems.

If a DATIP is followed by a DATO/B, and the device performs a destructive readout, then the device takes the responsibility for restoring the other byte.

Data-Out Transactions—The DATO and DATOB operations transfer data from the master to the slave. A DATO is used to transfer a word to the address specified by A<17:01>. The slave ignores A00 and the master places data on D<15:00>.

A DATOB is used to transfer a byte of data to the address specified by A<17:00>. Line A00 = 0 indicates the low-order byte. The master places the data on lines D<07:00>. A00 = 1 indicates the high order byte. The master now places the data on lines D<15:08>.

Parity Error Indicators (PA, PB)—PA and PB are generated by a slave and received by a master. They indicate parity error internal to a device as follows.

PA	PB	Condition
0	0	No error in a slave in DATI/P
0	1	Error in slave in DATI/P
1	X	Reserved for future expansion

The protocol for PA and PB is the same as that for D<15:00>.

PA and PB are not defined in a DATO transaction. PA and PB may be used by the bus master's parity error logic.

Master Sync (MSYN)—MSYN is a signal issued by a bus master and received by a slave. MSYN has two functions depending on whether it is being asserted or negated.

Assertion—The assertion of MSYN requests the slave, defined by the A lines, to perform the function required by the C Lines.

Negation—The negation of MSYN indicates to the slave that the master considers the data transfer concluded.

Slave Sync (SSYN)—SSYN is a signal issued by a slave and received by a master. SSYN has two functions depending on whether it is being asserted or negated.

NOTE

In an interrupt transaction, the interrupt fielding processor is the slave and the interrupting device is the master.

Assertion—In a master-slave data transfer, the assertion of SSYN informs the bus master that the slave has concluded its part of the data tranfer. For a DATI or DATIP, the requested data is placed on the D lines; for a DATO or DATOB, the data on the D lines is accepted.

In an interrupt operation, SSYN is asserted by the interrupt fielding processor. In this case, SSYN signifies that the interrupt vector is accepted by the interrupt fielding processor.

Negation—The negation of SSYN informs all bus devices that the slave has concluded the data transfer. For *data-in* (DATI/P), the negation of SSYN signifies the negation of Master Sync (MSYN) is received and the data removed from the D lines. For *data-out* (DATO/B), the negation of SSYN means that the negation of Master Sync (MSYN) is received. For an interrupt, the negation of SSYN signifies that the negation of the Interrupt Request (INTR) is received by the processor.

Interrupt Request (INTR)
Assertion—INTR is a signal asserted by an interrupting device, after it becomes bus master. The signal informs the interrupt fielding processor that an interrupt is to be performed and that the interrupt vector is present on the D lines. INTR may only be asserted by a device that obtains bus mastership under the authority of BG4, BG5, BG6, or BG7.

Negation—INTR is negated upon receipt of the assertion of the Slave Sync (SSYN) from the interrupt fielding processor at the end of the transaction.

Initialization Section
Three signals are used in the initialization section of the Unibus: INIT, AC LO, and DC LO.

Initialize (INIT)—INIT may be generated by processors or arbitrators. A console operation may cause a processor to assert INIT. INIT may be received by any UNIBUS device. The purpose of INIT is to stop all bus operations and initialize the bus devices by placing each in a well defined state.

INIT is transmitted on a type 1 line.

AC Low (AC LO)— AC LO is a signal generated by all power supplies whose failure may affect Unibus transactions. This signal is received by processors, arbitrators and bootstrap devices.

AC LO is transmitted on a type 3 line.

Assertion—The assertion of AC LO informs Unibus devices that the AC power input to a power supply is not within specifications. This failure may make the bus inoperable.

Negation—The negation of AC LO informs Unibus devices that all power supplies can maintain DC power within specifications long enough for a complete power-down and power-up sequence.

DC Low (DC LO)—DC LO informs bus devices that dc power is about to fail.

DC LO may be received by any bus device. It is typically received by core memories that use it to disable their internal operations so as not to lose data. DDC LO may also be used to initialize devices.

DC LO is transmitted on a type 3 line.

Assertion—The assertion of DC LO informs the bus devices that DC power is about to fail.

DC LO may be generated by any Unibus device or power supply. It is generally issued by a processor, or by a power supply, or by both of these. No device should issue DC LO except in correct sequence with AC LO.

Negation—The negation of DC LO informs the bus devices that DC power is within specifications and bus operations can therefore resume.

UNIBUS PROTOCOLS

Definitions and Concepts

Arbitration—The goal of arbitration is to decide which device is to have the use of the bus. Data transfers can originate from more than one source in the Unibus. Arbitration is the process of deciding which source is to use the bus next.

Centralized Arbitration—A signal must pass from a requesting device to a common arbitration point, and a response signal must return to the requesting device before it may transfer data. The Unibus uses centralized arbitration.

Priority Arbitration—In case of an apparent tie to request use of the data bus, rules are established to let one device go ahead of another. A priority arbitration sequence is a transaction during which a device is selected as next bus master. No actual bus transfer is performed. Priority arbitration is controlled by the arbitrator.

Data Transfer (Bus Cycle)—A data transfer, or bus cycle, is the transfer of one word between a master and a slave. The cycle starts when the master puts the address and control bits on the A and C lines; it ends when the master removes these bits from the lines.

Transaction—A transaction is a sequence of signals which completes a logical unit of activity on the Unibus. A transaction may consist of one or more bus cycles. For example: A data-in (DATI) is a single cycle transaction; a read/modify/write sequence (DATIP-DATO/B) is a two bus cycle transaction.

Protocol—The Unibus protocol is a set of procedures that must be used for a bus transaction.

Unibus Operation Concepts

Three concepts must be understood about the operation of the Unibus.

1. A device that has been designated as next bus master can turn the arbitrator on or off. The bus master asserts SACK to turn the arbitrator off; no further arbitration takes place. The master negates SACK to turn the arbitrator back on.
2. A bus master controls the data section of the bus. No other device may control this section until the master releases it. The master asserts BBSY to prohibit any other device (with the exception of the designated slave) from using the data section of the bus.
3. Priority arbitration and data transfer may be executed simultaneously on the Unibus. This is shown in Figure 1-11. Device N requests the use of the data section of the bus. A device may do this at any time. No action is taken on this request until device N-1, the current bus master, releases the arbitrator. When this is done, if no requests of a priority level higher than that of device N are pending, device N is selected as next master. At this time, device N controls the arbitrator and device N-1 still controls the data section of the bus.

   The arbitration sequence begins at the release of the arbitrator by device N-1 and ends with the selection of device N as next bus master. The arbitration sequence can take place while the data transfer by device N-1 is being executed.

   Device N-1 releases the data section of the bus when its data transfer is complete. Device N then becomes bus master and starts its own data transfer. Device N now controls both the arbitrator and the data section of the bus.

NOTE

To ensure optimal operation of the Unibus, the next master should be selected before the current master releases the data section. The current master should release the arbitrator before it releases the data section. Typically the bus master releases the arbitrator just before its last data transfer. The arbitrator waits 75 ns then resumes arbitration.

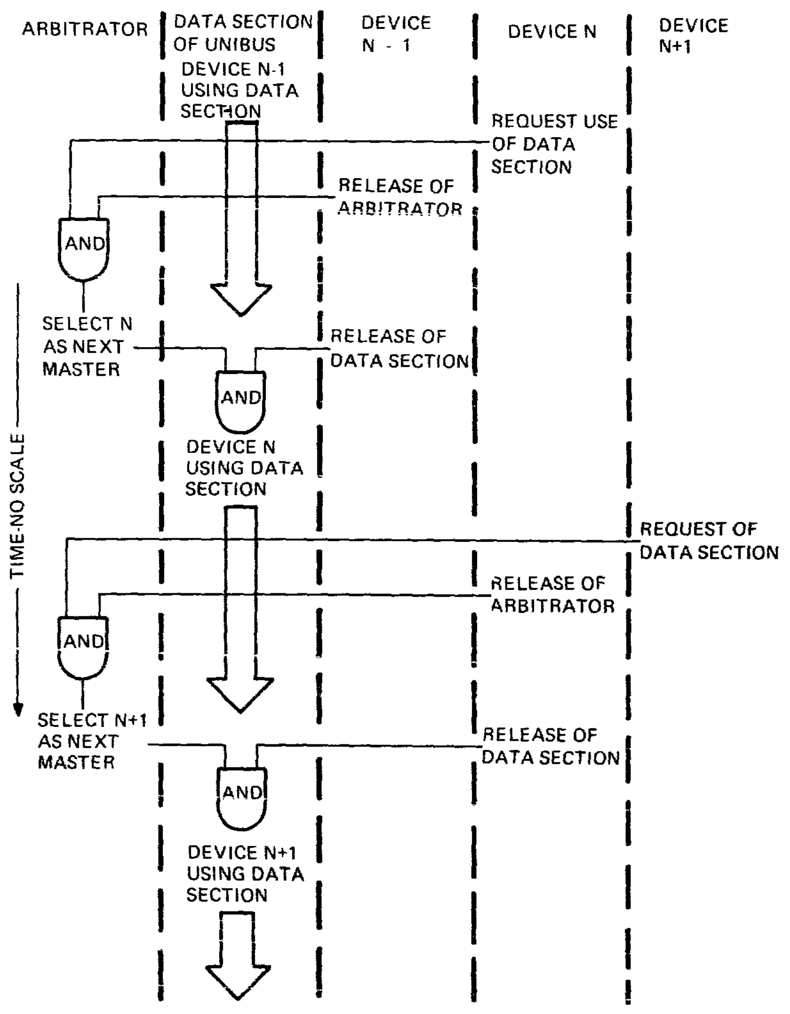


Figure 1-11 Arbitrator Request Sequence

Priority Protocol

The use of the data section of the bus is granted to requesting devices in accordance with the priority assigned to each one. The priority of a device on the Unibus is determined by two factors: (1) the priority level assigned to the device, and (2) its electrical position on the bus with respect to other devices of the same priority level.

Bus Device Priority Levels—Five priority levels are available for assignment to all bus devices capable of becoming bus master, with the exception of the interrupt fielding processor. These levels in descending order are:

NPR
Level 7

Level 6
Level 5
Level 4

The PDP-11/15 processor has only two priority levels: NPR and BR. An option (KF11) allows this processor to use all four BR priority levels.

Five bus request lines correspond to these priority levels: NPR, BR7, BR6, BR5, and BR4. These are type 1 signal lines. A device that requires the use of the data section of the bus asserts a request on one of these lines. This request is received by the arbitrator.

A device may be assigned to more than one priority level. A typical combination is a device which uses NPR to transfer data and a BR to interrupt the processor at the end of a data block.

Interrupt Fielding Processor CPU Priority Levels—The arbitrator monitors the priority state of the CPU. The CPU program execution priority (PRI) varies from 0 to 7. Five of these levels are relevant to the arbitration process.

PRI 7 (highest CPU priority level)
PRI 6
PRI 5
PRI 4
PRI <4 (below 4)

Grant—A grant signal informs a requesting device that it may become bus master after the current master releases the data section of the bus. The following two rules apply.

1. An NPR has the highest priority, assert NPG (non-processor grant). (A controller may do DMA transfers at any time.)
2. Whenever the CPU is between instructions (interruptable), the arbitrator may issue a grant at the level of the highest bus request if the CPU PRI level is not equal to or higher. The order of priorities is as follows and is shown in Figure 1-12.

Bus Request Priority Level	**CPU Priority Level**	**Bus Grant**
NPR (Highest)		NPG
BR7	PRI 7	BG7
BR6	PRI 6	BG6
BR5	PRI 5	BG5
BR4	PRI 4	BG4
	PRI <4 (lowest)	

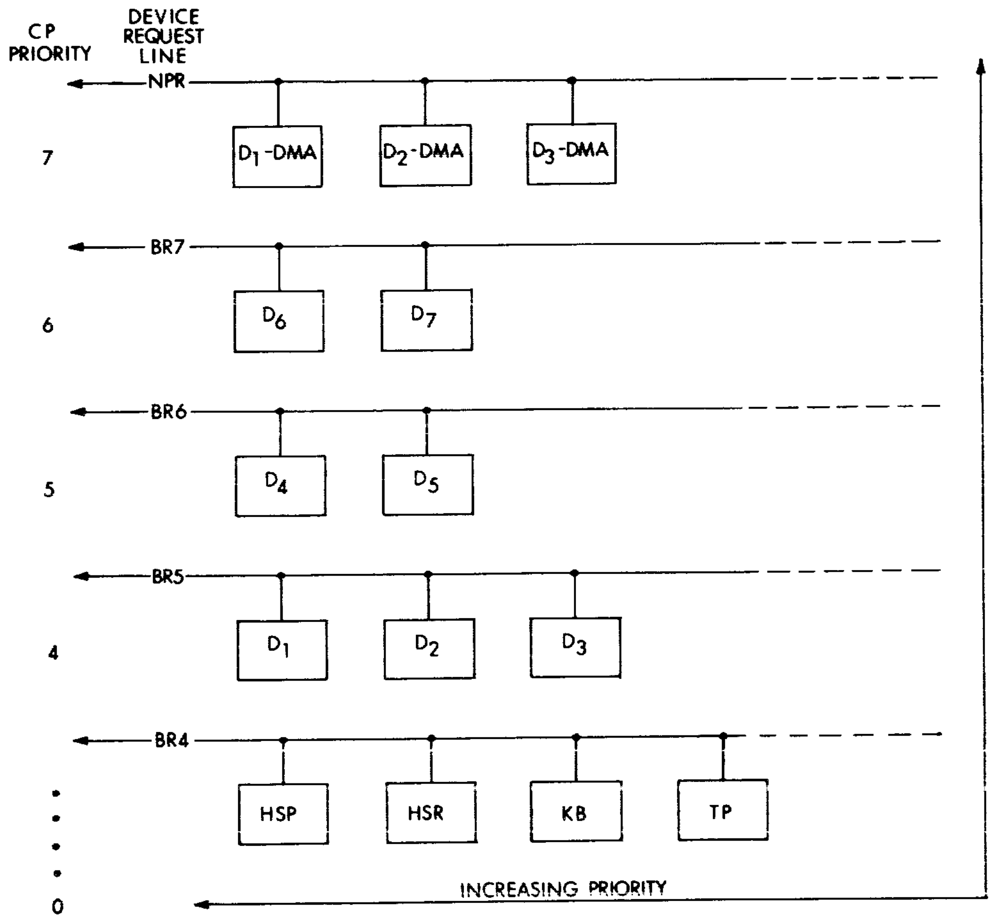


Figure 1-12 Priority Structure for Issuing Grants

Example 1:
If the CPU is operating under its highest program execution priority (PRI 7) the only grant that can be issued is an NPG in response to an NPR. A controller may do DMA transfers at any time.

Example 2:
If the CPU is operating at PRI 6 and a BR6 bus request is received, ignore that request. The processor is operating at an equal to or higher priority level.

NOTES

No other grant (BG or NPG) may be issued by the arbitrator while a BG is asserted.

PDP-11/15 and PDP-11/20 data transfers occur prior to NPR or BR service.

Assertion—The arbitrator asserts the grant. The first device on the bus assigned to the same priority level as the grant receives this grant. If this device is requesting the use of the data section of the bus at that level, it acknowledges receipt of the grant by asserting SACK and blocks the grant. If the device is not requesting the use of the data section, it passes the grant on to the next device of the same priority level. This procedure is repeated until a device accepts the grant or until the end of the bus is reached. In this last case, the grant is cancelled. This cancellation is affected by either a "NO SACK TIMEOUT" circuit in the arbitrator or by a "SACK TURNAROUND" bus terminator at the far end of the bus.

Negation—A grant is cancelled when it reaches the end of the bus without being acknowledged.

If a device asserts a request then negates it before receiving a grant, the grant is cancelled by the arbitrator. This is only possible if no other device at the same priority level accepts the grant.

A program may enable and then disable a request, thus cancelling the grant. A hardware interface that does not latch a request may lose the request and unintentionally cancel the grant. These two methods of cancelling a grant are not recommended as they slow down the Unibus operation.

Priority Structure Rules—The following rules define the Unibus priority structure.

1. The priority of a device on the Unibus is determined by two factors: (1) the priority level assigned to the device, and (2) its electrical position relative to the other devices of same priority level on the bus.
2. The priority level assigned to a device is the primary factor in determining whether or not this device obtains the use of the bus.
3. No device may use the bus until it has received a grant issued by the arbitrator.
4. A grant is issued only to the highest priority level request line that is asserting a request for the use of the bus. Only the devices at the priority level of this grant may obtain the use of the bus.
5. The requesting device electrically closest to the arbitrator on the specific grant line (NPG, BG7, BG6, BG5, or BG4) accepts, acknowledges, and blocks the grant, thus preventing the other devices on the grant line from using the grant signal. Electrical position is the secondary factor in determining the priority of a device.
6. The arbitrator does not know which devices are requesting grants, nor their physical position on the bus. The five request lines may be asserted by more than one device on each line at the same time.

7. The arbitrator does not know which of the devices at the level at which a grant is issued is using this grant, nor the physical position of this device on the bus.
8. In order to receive grants, all devices capable of becoming bus master must be physically located on the same side of the arbitrator on the bus. These devices must be able to receive and retransmit grants at their priority level.

Priority Arbitration Example—In Figure 1-13, devices A through E are assigned to priority levels as follows.

Device	Priority Level
A	Level 6
B	Level 4
C	Level 5
D	Level NPR and Level 5
E	Level 5

Their physical position on the bus is as shown in Figure 1-13. Device A is closest to the arbitrator. The effective priority order of devices A through E is as follows .

Priority Order by Device	Priority Level
D (highest)	NPR
A	6
C	5, 1st in line
D	5, 2nd in line
E	5, 3rd in line
B (lowest)	4

NOTE

Device D has the highest priority at its NPR level, but it is fourth at its level 5. Device B is in the second physical position on the bus, but it has the lowest priority.

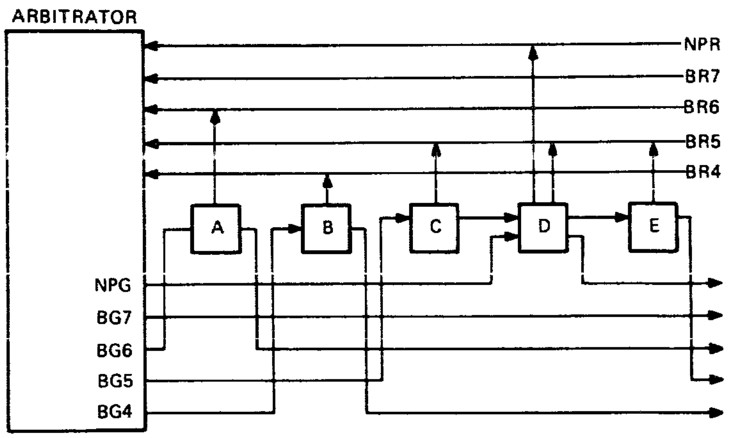


Figure 1-13 Priority Arbitration Example

Priority Arbitration Transactions—The priority arbitration of the Unibus is directly affected by the priority state of the CPU. The CPU priority (PRI) can range from 0 to 7. The arbitrator grants use of the bus to non-CPU devices by the following rules.

1. A controller may do direct memory access (DMA) data transfers at any time. To do this it asserts an NPR. The arbitrator then asserts an NPG. No other grant (NPG or BG) may be issued while an NPG is asserted.
2. Whenever the CPU is between instructions, it accepts an interrupt from a controller whose priority is greater than that of the current program being executed. The guidelines for interrupting the CPU are in this order.
   a. If PRI is less than 7 and BR7 is asserted, then assert BG7.
   b. If PRI is less than 6 and BR6 is asserted, but not BR7, then assert BG6.
   c. If PRI is less than 5 and BR5 is asserted, but not BR7 or BR6, then assert BG5.
   d. If PRI is less than 4 and BR4 is asserted, but not BR7, BR6 or BR5, then assert BG4.

The arbitrator responds to: (1) signals from bus devices requesting the use of the data section of the Unibus, and (2) enabling signals from the interrupt fielding processor.

The interrupt fielding processor prohibits the arbitrator from issuing bus grants: (1) during an interrupt transaction, and (2) after this transaction when the processor is determining its new priority level. The interrupt fielding processor must save the old level and establish the new level. During this time it cannot service a bus grant. Also, the arbitrator cannot issue any more bus grants. This sequence of establishing levels typically requires four bus cycles. When complete, the arbitrator may again provide grants at a higher level than that of the new level of the interrupt fielding processor.

Typical Arbitration Example:

NOTE

When describing arbitration transactions in the following paragraphs, the arbitrator is allowed to issue a grant at the level of the request. This implies: (1) No device has a request at a higher priority level at the arbitrator. (2) The priority level of the interrupt fielding processor is lower than that of the request.

A typical arbitration sequence is described in Figure 1-14. Starting at the top, the following steps are observed and correspond to the numbers in the figure.

1. Device 1 is current bus master. It is using the data section of the bus and asserting Bus Busy (BBSY).
2. Before the last data transfer, device 1 negates SACK. This enables the arbitrator, and a new priority arbitration sequence starts.
3. Device 2 requests a bus grant from the arbitrator.
4. The arbitrator issues a grant at the priority level of the request. At the same time the grant disables the arbitrator.
5. Device 3 requests a grant; however, with the arbitrator disabled, this request is ignored.
6. Device 2 acknowledges the grant by asserting SACK. This keeps the arbitrator disabled. Device 2 is now bus master.
7. The arbitrator acknowledges receipt of SACK by negating the grant. This signals the end of the arbitration sequence.
8. Device 1 ends its data transfer and relinquishes the bus by negating BBSY. Then device 2 becomes bus master, asserts BBSY, and starts its data transfer.

NOTE

Requests are not honored by the arbitrator while a grant is asserted, nor while the assertion of SACK is seen at the arbitrator.

A priority arbitration sequence may or may not occur at the same time as a data transfer cycle. In the case of devices 1 and 2 above, it does. However, the arbitration sequence for device 3 does not start until the data transfer by device 2 is almost ended.

All Unibus signals in the above sequence use type 1 lines, with the exception of the grants. The grants use type 2 lines. Thus, the arbitrator asserts a grant to the first device on the bus wired to a particular grant line. If this device requires the use of the data section of the bus, it blocks the grants from progressing further and asserts SACK. If the device does not require the use of the data section of the bus, it passes (asserts) the grant to the next device of the same priority level on the bus. A device may not accept a grant (assert SACK) after it passes the grant.

Notes on the Timing Diagrams—
Arrows on the timing diagrams in this section show the act or cause that establishes the new condition of a signal. A timing specification may be indicated with the arrow. An arrow with the notation "75 ns" means 75 nanoseconds elapse between the start of the arrow and its end point.

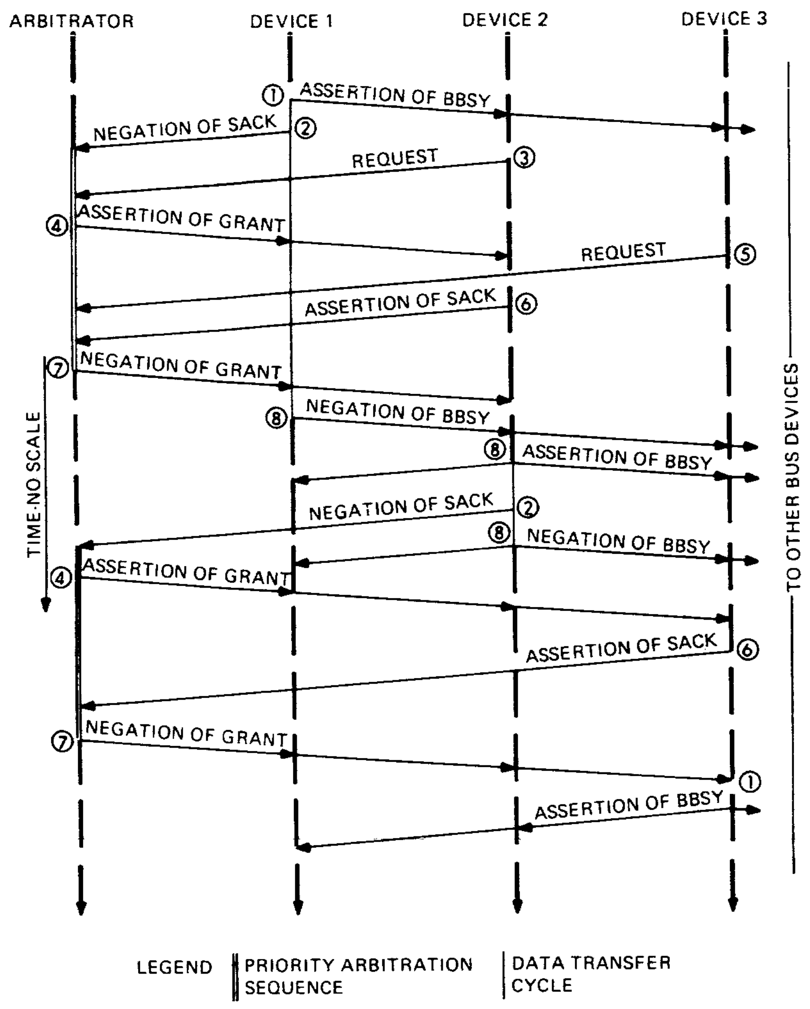


Figure 1-14 Typical Arbitration Sequence

Asserted is to be in the "true" state.

Defined refers to the A, C and D lines received by a device after they are asserted in a valid configuration by another device.

Irrelevant signifies that the state of the line is of no concern at the time.

Negated is to be in the "false" state.

Undefined means that the line may be either asserted, negated, or in some intermediate state because of low dc power. Power may be coming up or going down. Slanted lines at the beginning or end of an undefined period represent **skew** time.

Grant Status lines on the timing diagrams indicate the types of grants that may be issued by the arbitrator during the arbitration sequence.

All the timing diagrams that follow in this section show timing at more than one device. With the exception of the Typical Power Up/Down Sequence, these device timing diagrams are independent of each other.

NPR Arbitration Sequence—Figure 1-15 shows an NPR arbitration timing diagram. The step numbers that follow correspond to the numbers in the figure.

1. The requesting device asserts NPR.
2. After a propagation delay, the arbitrator receives the NPR.

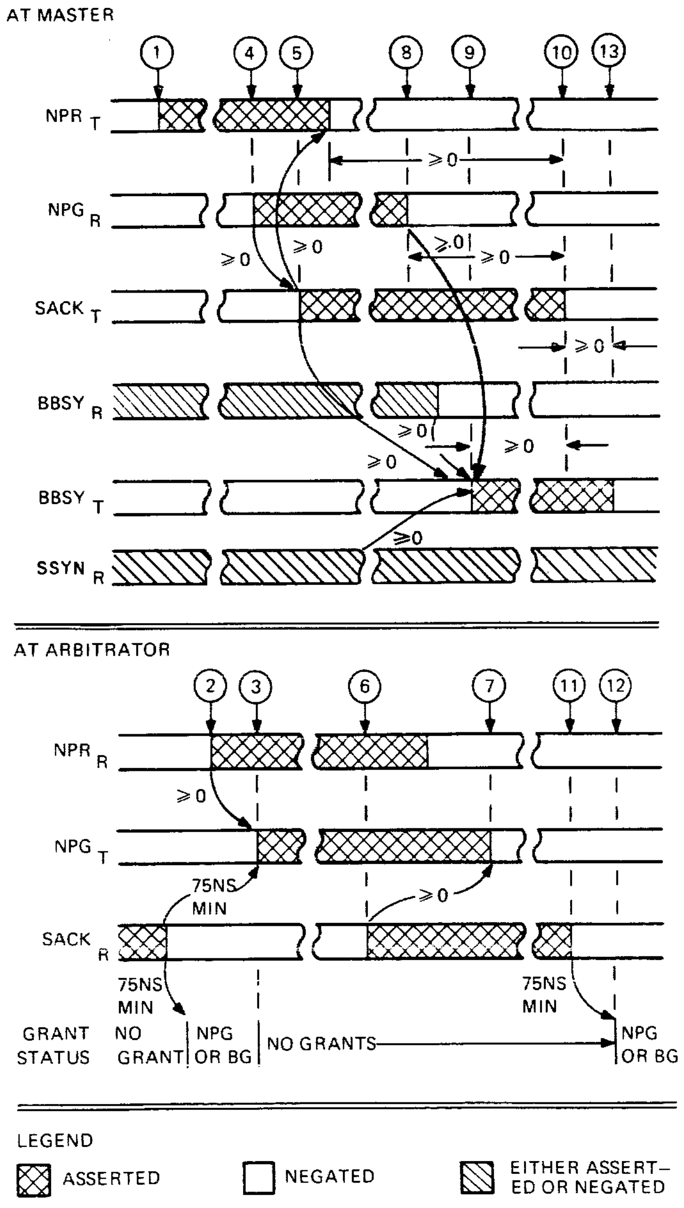


Figure 1-15 Typical NPR Arbitration Sequence

3. If SACK has been negated for at least 75 ns, the arbitrator asserts NPG, and the arbitration process stops.

NOTE

The arbitrator may not issue a grant while SACK is asserted, nor for a minimum of 75 ns after SACK is negated. This delay ensures that the negation of NPR or BR from the previous arbitration sequence arrives at the arbitrator before arbitration resumes. This prevents issuing a new grant in response to the previous request in case the request is negated at the same time as SACK.

In a single word transfer, the master typically negates SACK immediately after asserting BBSY. The SACK delay ensures BBSY is sensed before negating SACK. This prevents the interrupt fielding processor from asserting BBSY when the bus is free.

4. After a propagation delay, NPG is received at the requesting device.
5. The requesting device then asserts SACK. In a single word transfer, the requesting device must then negate NPR before SACK is negated. In multiple data transfers, NPR may remain asserted.
6. After a propagation delay, the arbitrator receives SACK.

NOTE

If the arbitrator does not receive SACK within a specified time after it asserts NPG, the grant is negated and arbitration resumes. This time delay is typically 5 to 10 microseconds.

Systems may avoid the timeout delay by having a terminator that asserts SACK when it receives an NPG. The terminator must be located at the opposite end of the bus from the arbitrator. The terminator must also negate SACK upon receipt of the negation of NPG. After a propagation delay, the arbitrator receives the negation of SACK. The arbitrator waits a minimum of 75 ns then resumes arbitration.

7. The arbitrator then negates NPG.

8. After a propagation delay, the requesting device receives the negation of NPG.
9. After receiving the negation of BBSY, NPG and SSYN, the requesting device asserts BBSY. The requesting device becomes bus master at this time and starts its data transfer.
10. If the bus master receives the negation of the bus grant, it may negate SACK before finishing its data transfer.

NOTE

For a single word transfer, a device typically asserts BBSY and negates SACK at the same time. The master must not negate SACK prior to receiving the negation of NPG. This ensures that the arbitrator receives the assertion of SACK.

11. After a propagation delay, the arbitrator receives the negation of SACK.
12. The arbitrator waits a minimum of 75 ns then resumes arbitration.
13. At the end of its last data transfer cycle, the master waits at least 75 ns after negating MSYN, then it removes any A, C, or D bits it placed on the bus. It then negates BBSY releasing the bus. SACK must be negated before BBSY is negated.

A bus master may issue an interrupt command to the interrupt fielding processor. The bus master forces the interrupt fielding processor into a subprogram by means of a vector address. The vector is asserted on the D lines.

Figure 1-16 shows the interaction between the bus master, the interrupt fielding processor, and the arbitrator for a typical interrupt transaction. Upon gaining bus mastership the bus master puts the vector on the D lines, asserts INTR, and negates SACK.

Upon receipt of INTR, the interrupt fielding processor first delays to deskew the D lines; then it strobes the vector and asserts SSYN.

Upon receipt of the assertion of SSYN, the master removes the vector from the D lines and negates INTR and BBSY.

When the interrupt fielding processor receives the negation of INTR, it negates SSYN.

Upon receipt of the assertion of INTR, the arbitrator ceases to issue BGs. It grants no BGs until authorized to do so by the interrupt fielding processor. NPGs, however, may be granted during this time.

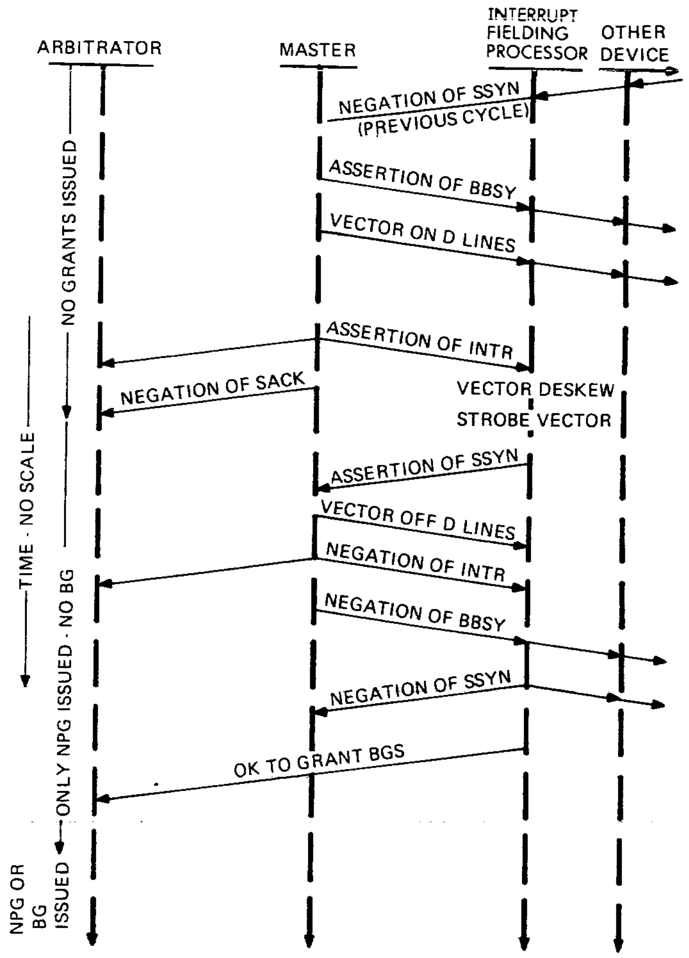


Figure 1-16 Typical INTR Cycle

BR Interrupt Arbitration Sequence—Figure 1-17 shows a bus request (BR) interrupt timing diagram. The numbers of the steps in this paragraph correspond to the numbers on this timing diagram.

1. The requesting device asserts BRn.
2. After a propagation delay, the arbitrator receives the assertion of BRn.
3. If the negation of SACK from the previous priority arbitration sequence has been received by the arbitrator for at least 75 ns and if the interrupt fielding processor is ready to accept an interrupt vector at the level of the interrupting device, the arbitrator asserts BGn and the arbitration process is stopped.

NOTE

The arbitrator may not issue a grant while SACK is asserted, nor for a minimum of 75 ns after SACK is negated. This delay ensures that the negation of NPR or BR from the previous arbitration sequence arrives at

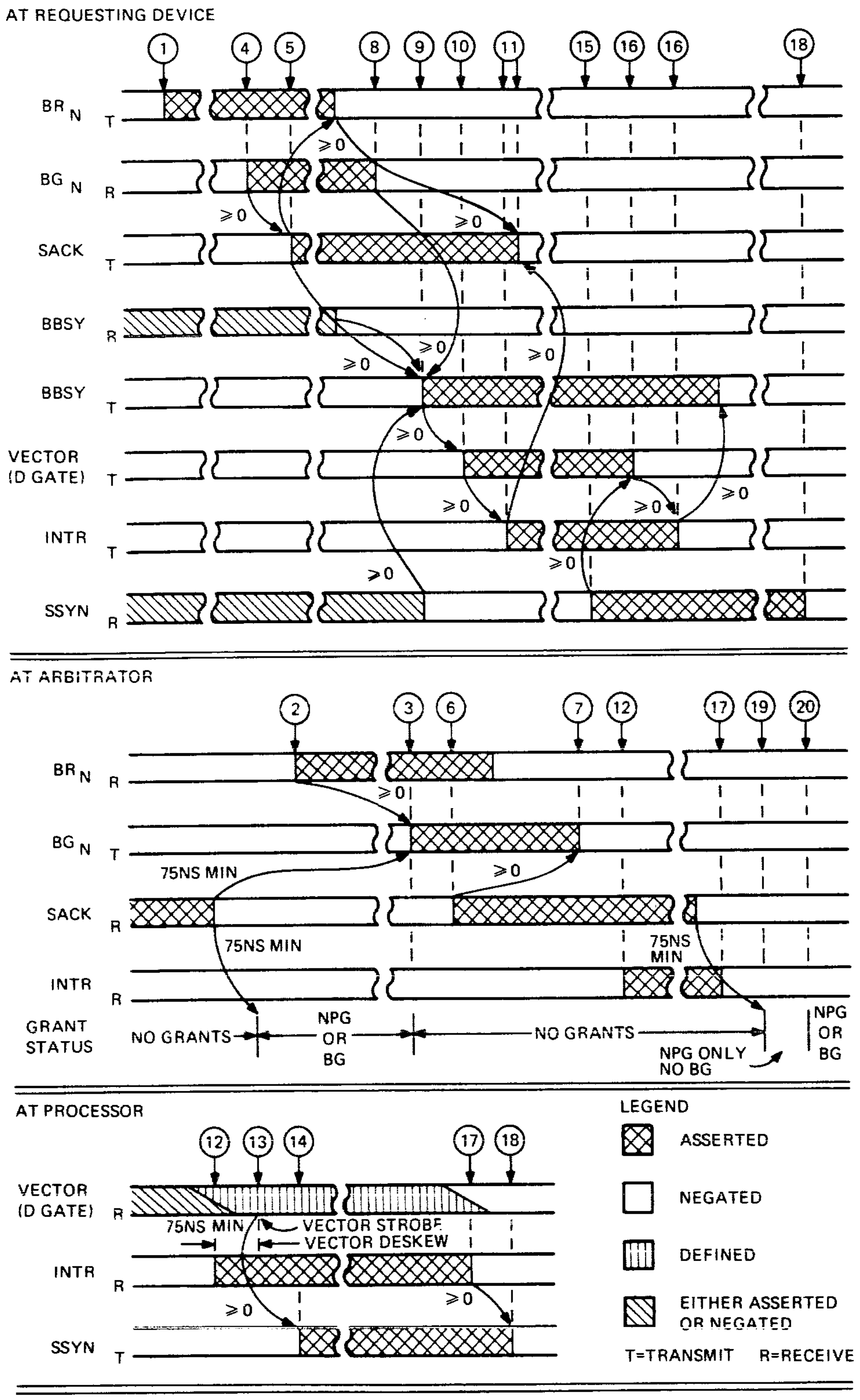


Figure 1-17 Typical Interrupt Timing Diagram

the arbitrator before arbitration resumes. This prevents issuing a new grant in response to the previous request in case the request is negated at the same time as SACK.

In a single word transfer, the master typically negates SACK immediately after asserting BBSY. The SACK delay ensures BBSY is sensed before negating SACK. This prevents the interrupt fielding processor from asserting BBSY when the bus is free.

4. After a propagation delay, the requesting device receives BGn. A bus grant (BG), not an NPG, designates the next bus master for an interrupt arbitration sequence.
5. The requesting device then asserts SACK. For a single transaction, the requesting device must then negate BRn before SACK is negated. For multiple transactions BRn may remain asserted.
6. After a propagation delay, the arbitrator receives SACK.

NOTE

If the arbitrator does not receive SACK within a specified time after it asserts BGn, the grant is negated and arbitration resumes. This time delay is typically 5 to 10 microseconds.

System throughput may be improved by the use of a terminator that asserts SACK when it receives a BGn. The terminator must be located at the opposite end of the bus from the arbitrator. The terminator must also negate SACK upon receipt of the negation of BGn. After a propagation delay, the arbitrator receives the negation of SACK. The arbitrator waits a minimum of 75 ns then resumes arbitration.

7. The arbitrator then negates BGn.
8. After a propagation delay, the requesting device receives the negation of BGn.
9. After receiving the negation of BBSY, SSYN and BGn, the requesting device asserts BBSY. The requesting device becomes bus master at this time.
10. The bus master puts the interrupt vector on the D lines.
11. The master asserts INTR, then negates SACK.

NOTE

> The interrupt (INTR) must be asserted before SACK is negated. This ensures the receipt of the INTR before the end of the SACK delay at the arbitrator. The SACK delay compensates for skew between INTR and SACK at the arbitrator.
>
> The master must not negate SACK prior to receiving the negation of BGn. This ensures that the arbitrator receives the assertion of SACK.

12. After a propagation delay, the arbitrator and the interrupt fielding processor receive the assertion of INTR.
13. The interrupt fielding processor waits for at least 150 ns (vector deskew), then it strobes the vector from the D lines.

NOTE

> The vector deskew compensates for the skew between INTR and the D lines at the interrupt fielding processor.

14. The interrupt fielding processor asserts SSYN.
15. After a propagation delay, the master receives the assertion of SSYN.
16. The master then removes the vector from the D lines and then negates INTR. The master then typically negates BBSY. With this action the master releases the data section of the bus.
17. After a propagation delay, the arbitrator and the interrupt fielding processor receive the negation of INTR.
18. The interrupt fielding processor then negates SSYN.
19. After receiving the negation of SACK (step 11 above), the arbitrator waits 75 ns (SACK delay); then it may resume issuing NPGs but not BGs.

NOTE

> Typically, the interrupt fielding processor reads a new program counter and status word from memory locations designated by the interrupt vector. This is done immediately following the interrupt transaction. From this the interrupt fielding processor determines its new priority level.

20. The interrupt fielding processor informs the arbitrator that it may start issuing BGs.

NOTE

Data may be transferred by a device that has become bus master through a Bus Request/Bus Grant sequence. The procedure is the same as that described for a nonprocessor request (NPR).

A master may only execute one INTR transaction per bus grant. A master may release the data section of the bus after a data transfer(s) but no interrupt transaction. This release constitutes passive release of the data section of the bus.

Data Transfer Protocol

Data-In Transaction (DATI or DATIP)—Data-in is defined as a data transfer from a slave to a master. DATI and DATIP are similar data-in operations. Figure 1-18 shows hte interaction between master and slave for a typical DATI or DATIP.

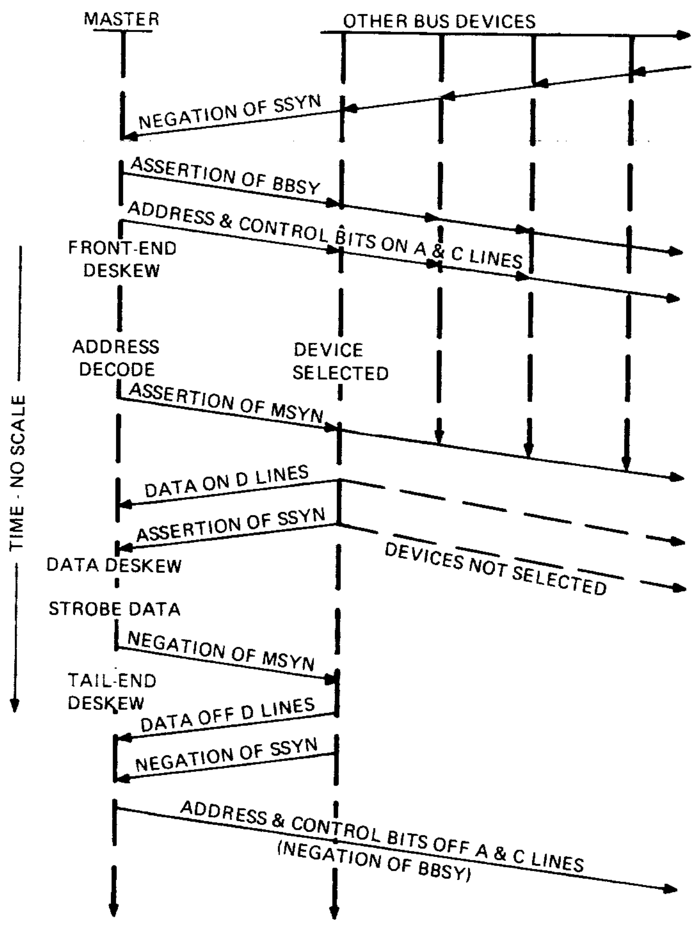


Figure 1-18 Typical Data-In Cycle (DATI or DATIP)

A bus master (BBSY asserted) places the slave address and the required control bits on the A and C Unibus lines. All devices decode the A and C lines to see if they are selected as the slave for this transaction.

The master asserts MSYN after two conditions are met.

1. A 150 ns delay is allowed for:
   a. deskewing of the A and C lines, and
   b. decoding the address and control information by the slave.
2. A 75 ns delay is allowed after the receipt of the negation of slave sync (SSYN) to ensure that the previous slave is no longer driving the D lines.

The new slave receives MSYN, places the requested data on the D lines, and asserts SSYN.

The master receives SSYN, deskews the D lines, strobes the data, and negates MSYN.

Negation of MSYN informs the slaves that the master accepts the data. The slave then removes the data from the D lines and negates SSYN. This ends the slave's part of the data transfer cycle.

After negating MSYN, the master deskews the A and C lines. This ensures that the negation of MSYN is received by all devices before the A and C lines become invalid, and thus prevents false selection by another device. After the deskew, the master removes the address and control bits from the A and C lines ending its part of the data transfer.

If the master is not immediately going to use the bus for another data transfer, it negates BBSY. This releases the data section of the bus for use by another device. If there is to be another transfer (i.e., a DATO or DATOB after a DATIP), BBSY is held asserted by the current master.

Detailed Description, DATI and DATIP—Figure 1-19 shows a typical DATI transaction. The numbers of the steps in this paragraph correspond to the numbers in this figure.

1. With bus busy (BBSY) asserted, the bus master puts the address and the control bits on their respective Unibus lines.
2. After a propagation delay, each device on the bus receives the address and control bits and decodes them.
3. Meanwhile, the master waits for at least 150 ns; this is called front-end deskew. Then, if SSYN is negated, the master asserts MSYN. The master may not assert MSYN at the driver input until 150 ns elapse since the A, C, and enable lines become valid at the A and C driver inputs.

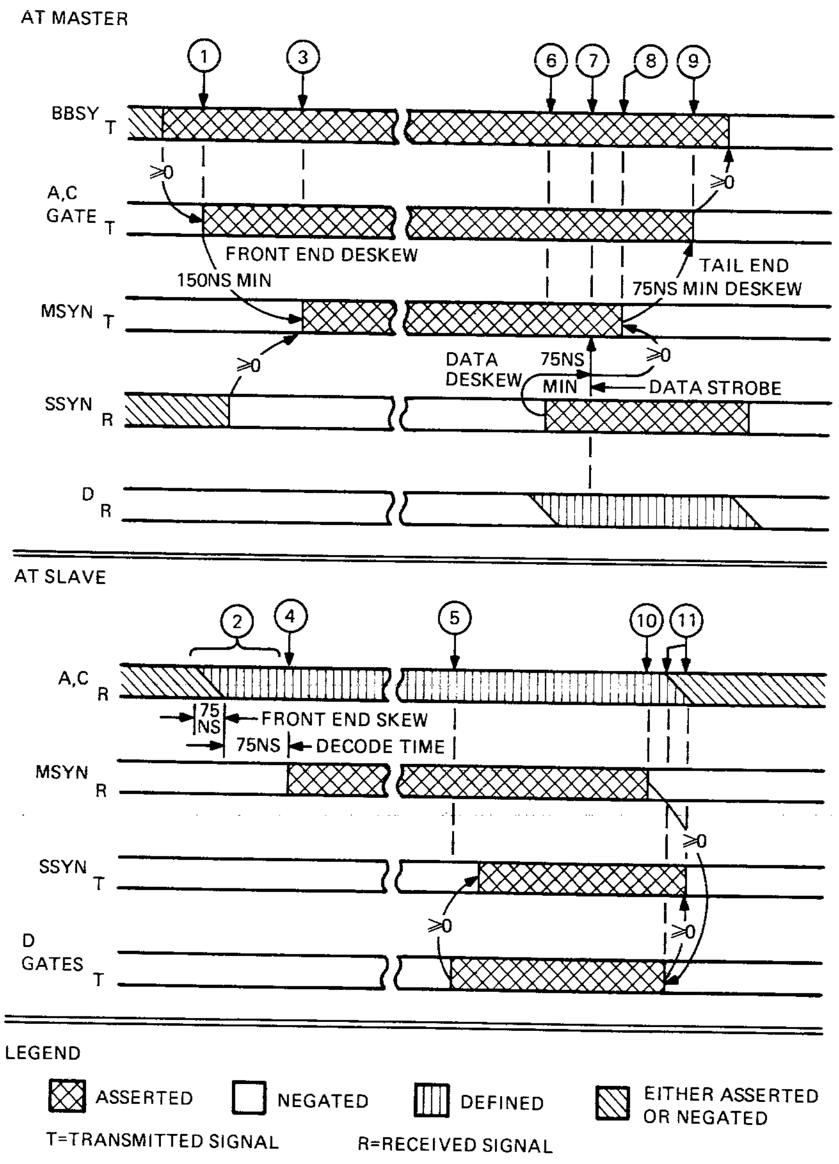


Figure 1-19 Typical DATI Transaction

NOTE

The front-end deskew consists of 75 ns to compensate for the skew of the A and C lines at the slave, plus 75 ns to allow the slave to decode these lines.

4. After a propagation delay, each device on the bus receives the assertion of MSYN. One decides, after decoding the address, that it is the slave for this transaction.
5. After receiving the assertion of MSYN, the slave puts the requested data on the D lines, then asserts SSYN. The slave must not assert SSYN at the driver input before the data and enable lines are valid at the D driver inputs.

NOTE

SSYN must not be asserted before the data is put on the D lines. This ensures that the master deskews the data with respect to SSYN and strobes the data while it is valid.

6. After a propagation delay, the master receives the assertion of SSYN.

NOTE

If the master does not receive SSYN during a specified time after it asserts MSYN (time-out delay), step 7 below may be executed. The master must then execute steps 8 and 9. An error bit may be set.

The timeout delay is typically 10 to 20 microseconds in processors. The use of some devices [e.g., bus window (DA11), data link (DL10)] require much longer times which can be milliseconds. These devices are used in multi-processor or multi-bus systems.

7. After waiting for at least 75 ns and after receiving SSYN (data deskew), the master strobes in the data.

NOTE

The data deskew compensates for the skew of the D lines at the master.

8. The master negates MSYN.
9. After a 75 ns minimum wait (tail-end deskew), the master removes the address and control bits from the A and C lines. If this is the last data transfer under the current grant, the master then negates BBSY.

NOTE

The tail-end deskew guarantees that the A lines do not change at any bus device while the device is receiving the assertion of MSYN. This prevents false selection of a device due to changing A lines while MSYN is asserted.

10. After a propagation delay, the slave receives the negation of MSYN.
11. The slave removes the data from the D lines and negates SSYN.

NOTE

SSYN must not be negated before the data is removed from the D lines. This ensures that the negation of SSYN is a valid indication that the data bits are removed from the D lines.

Data-Out Transaction (DATO or DATOB)—Data-out is defined as a data transfer from a master to a slave. DATO and DATOB are data-out operations. The timing and protocol for both of these operations are identical. Figure 1-20 shows the interaction between master and slave for a typical DATO or DATOB.

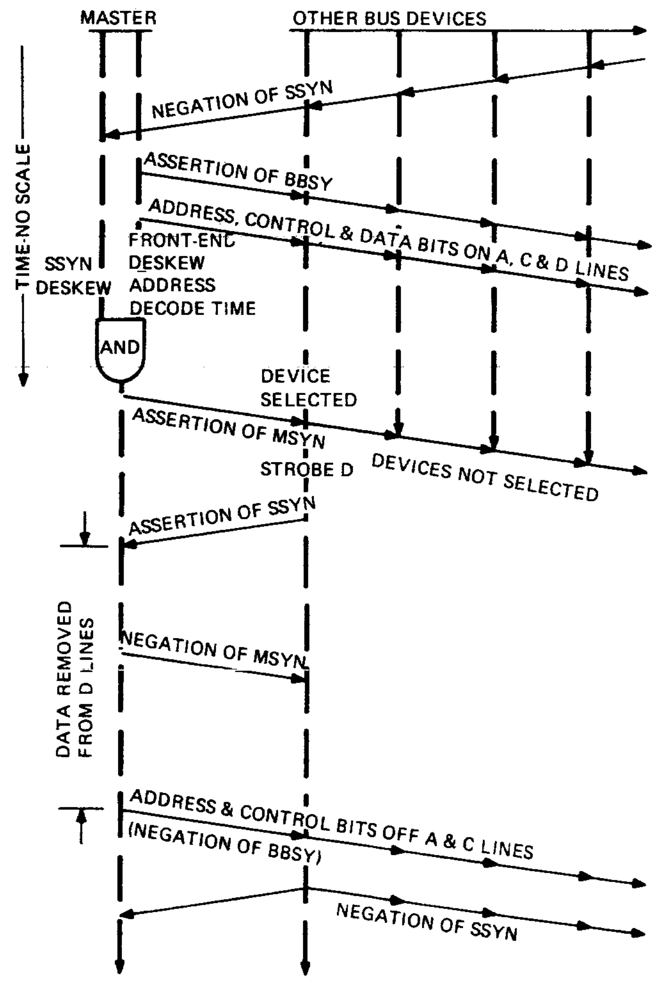


Figure 1-20 Typical Data-Out Cycle (DATO or DATOB)

A bus master (BBSY asserted) places the slave address, the required control bits, and the data on the A, C, and D Unibus lines. All devices decode the A and C lines to see if they are selected as the slave for this transaction.

The master asserts MSYN after two conditions are met.

1. A 150 ns delay is allowed for:
   a. deskewing of the A, C, and D lines, and
   b. decoding the address and control information by the slave.
2. A 150 ns delay is allowed after the receipt of the negation of slave sync (SSYN) to ensure that the previous slave is no longer driving the D lines and to set up new devices.

The new slave receives MSYN, strobes the data on the D lines, and asserts SSYN.

The master receives SSYN, negates MSYN, then deskews the A and C lines. This ensures that the negation of MSYN is received by all devices before the A and C lines become invalid, and thus prevents false selection by another device. After the deskew, the master removes the address and control bits from the A and C lines ending its part of the data transfer.

The master may remove data from the D lines any time after it receives the assertion of SSYN, but no later than it removes the address and control bits from the A and C lines.

The slave, upon receipt of the negation of MSYN, ends its part of the data transfer cycle by negating SSYN.

If the master is not immediately going to use the bus for another data transfer after removing the address and control bits from the A and C lines, it negates BBSY. This releases the data section of the bus for use by another device. If there is to be another transfer, BBSY is held asserted by the current master.

Detailed Description, DATO and DATOB—Figure 1-21 shows a typical DATO transaction. The number of steps in this paragraph correspond to the numbers in this figure.

1. With BBSY asserted, the bus master puts the address, control, and data bits on their respective Unibus lines.
2. After a propagation delay, each device on the bus receives the address and control bits and decodes them.
3. After putting the address, control and data bits on the A, C, and D lines, the master waits for at least 150 ns (front end deskew). The master may not assert MSYN at the driver input until 150 ns have elapsed since the A, C, D, and enable lines become valid at the A, C, and D driver inputs.

NOTE

The front-end deskew consists of 75 ns to compensate for the skew of the A and C lines at the slave, plus 75 ns to allow the slave to decode these lines.

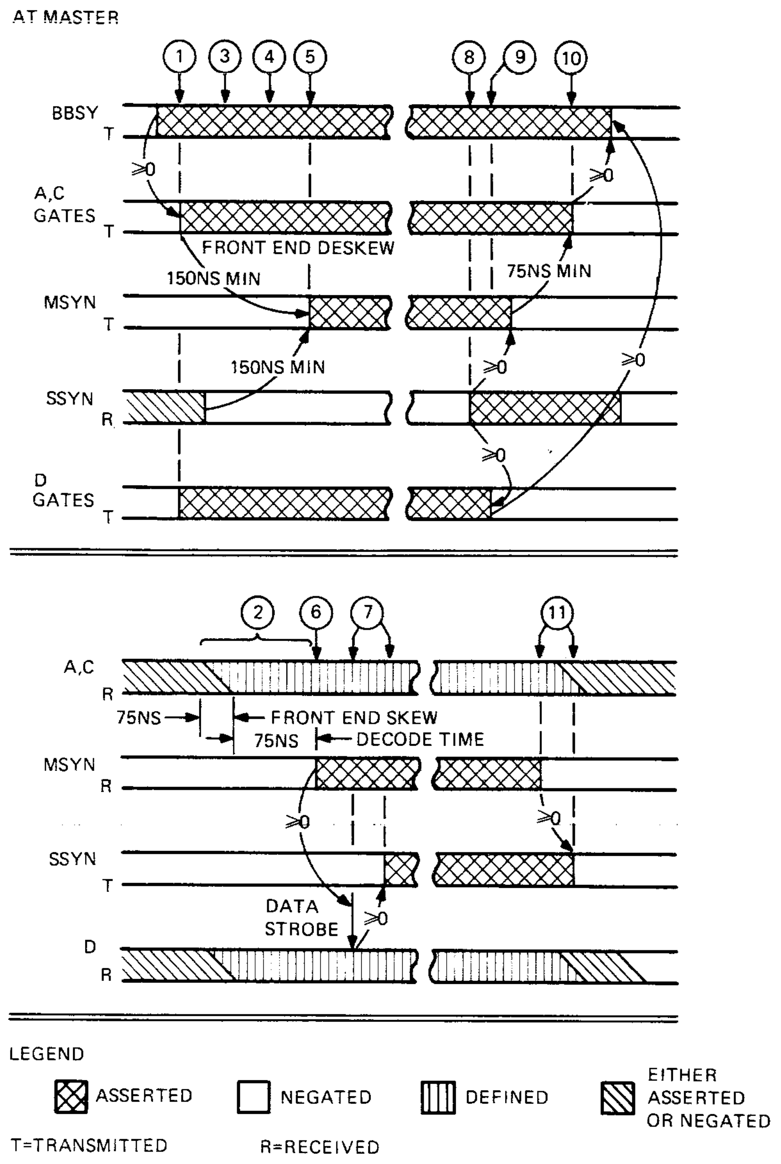


Figure 1-21 Typical Data-Out Transaction (DATO)

4. The master waits for a minimum of 150 ns after receiving the negation of SSYN (SSYN deskew).

NOTE

The 150 ns SSYN deskew consists of: (1) 75 ns to ensure that the data from any previous DATI or DATIP transaction is removed from the D lines and (2) 75 ns for devices that may require set-up time.

5. Having met the conditions in steps 3 and 4 above, the master asserts MSYN.

6. After a propagation delay, each device on the bus receives the assertion of MSYN. One decides, after decoding the address, that it is the slave for this transaction.
7. Upon receiving the assertion of MSYN, the slave strobes the data from the D lines and asserts SSYN.

NOTE

The data must be strobed by the slave either at the same time as, or previous to, the assertion of SSYN. This is required because the master may remove the data from the D lines upon receipt of the assertion of SSYN.

8. After a propagation delay, the master receives the assertion of SSYN.

NOTE

If the master does not receive SSYN during a specified "timeout delay" after it asserts MSYN, steps 10 and 11 below are executed. An error bit may be set.

The "timeout delay" is typically 10 to 20 microseconds in processors. The use of some devices in multi-processor or multi-bus systems require much longer times (milliseconds).

9. Upon receipt of the assertion of SSYN, the master negates MSYN, and may remove the data from the D lines.
10. After a 75 ns minimum wait (tail-end deskew), the master removes the address and control bits from the A and C lines. If this is the last data transfer under the current grant, the master then negates BBSY. If the data has not previously been removed from the D lines, it must be removed: (a) no later than the removal of the A and C bits from the bus if another transfer will be done under the current grant; or, (b) before the negation of BBSY if this is the last transfer under the current grant.

NOTE

The tail-end deskew guarantees that the A lines do not change at any bus device while the device is receiving the assertion of MSYN. This prevents false selection of a device due to changing A lines while MSYN is asserted.

11. After a propagation delay, the slave receives the negation of MSYN and then negates SSYN.

Read/Modify/Write Transaction (DATIP-DATO/B)—A read/modify/write transaction consists of a DATIP followed immediately by a DATO or DATOB. Figure 1-22 shows a typical DATIP-DATO/B transaction. The following rules must be followed.

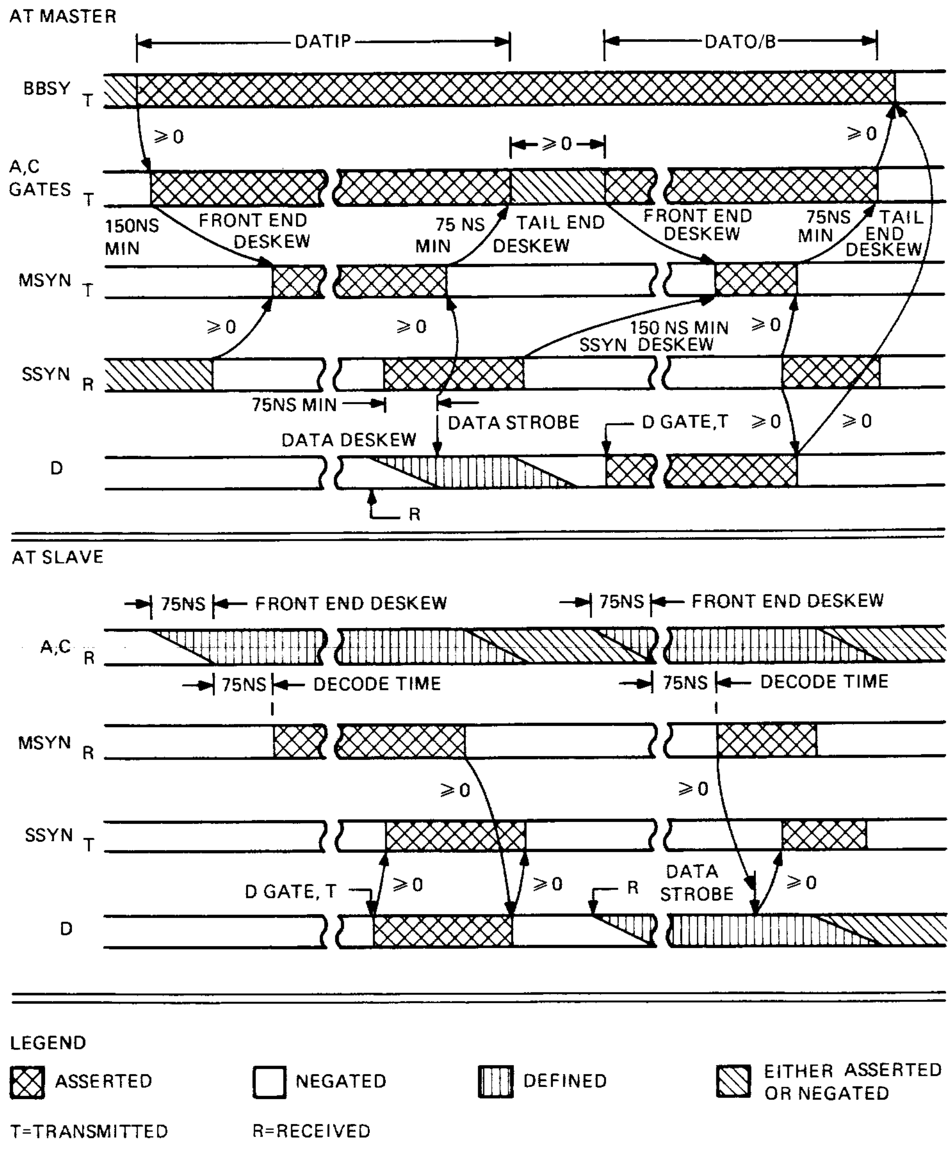


Figure 1-22 Typical DATIP-DATO/B Transaction

1. All protocol rules previously set forth for DATIP and DATO/B must be followed.
2. The master must ensure that no other device becomes bus master from the start of the DATIP to the end of the DATO/B. BBSY must be held asserted during this period.
3. The same word location must be accessed during both data transfer cycles; i.e., address bits A<17:01> must not change.

4. The DATO/B must follow the DATIP, immediately; no other data transfer cycle may be executed between them.

Multiple Word Transfers—A multiple word transfer has more than one word (or byte) transferred between master and slave during a single grant. The bus is not released by the master between word transfers. Several types of data tansfers may be executed, in any order and to various locations if required, providing that all rules for each type are obeyed; (e.g., a DATIP-DATO/P may have more than one location addressed).

NOTE

Multiple word transfers are used by high speed devices that may lose data because of bus latency.

INITIALIZATION SECTION

The Initialization section of the Unibus controls initializing or resetting the system, and the power-up and power-down sequences of all bus devices. Three Unibus signals are used: INIT, AC LO, and DC LO.

Initialization (INIT)

INIT is caused by some console operations, the RESET instruction, and DC LO. Only a processor or the arbitrator may assert INIT.

Processor Requirements—A processor must become bus master, then wait for 5 microseconds before it may assert INIT. No bus cycles may be executed during these 5 microseconds. This delay ensures that all memory cycles in progress are complete before asserting INIT.

However, if a processor tries to assert INIT but cannot obtain the use of the data section of the bus after 100 microseconds, it may then assert INIT without becoming bus master.

A processor, as bus master through a grant sequence, may not negate SACK until after it asserts INIT. This ensures that the arbitrator receives the assertion of INIT before the negation of SACK. This prevents the arbitrator from starting arbitration until it receives the negation of INIT.

Any processor, after negating INIT, must wait 75 ns before asserting any signal, except AC LO, DC LO, or INIT. This ensures that the negation of INIT reaches all bus devices before the processor asserts any signals on the data section of the bus.

Arbitrator Response—Upon receipt of the assertion of INIT, the arbitrator negates all grants and may not issue any for events that occurred before INIT. No grants may be issued while INIT is asserted.

Master/Slave Device Response—When a master/slave device receives the assertion of INIT, it responds as follows.

1. It completes any bus cycle in process. If the device is bus master, then it negates BBSY. If the bus cycle in progress is a DATIP, the master completes the DATO/B. The memory must be capable of completing the DATO/B, or it must restore itself and treat any following DATO/B as a new transaction.
2. It negates any of the following signals that it may be asserting: SACK, NPR, BR4, BR5, BR6, BR7. And it passes all grants.
3. It clears the Interrupt Enable bit. It may assert AC LO, DC LO, and any signals required by step 1 above. It may not assert NPR, BR, SACK, or BBSY.

After receipt of the negation of INIT, a device must be reset for normal programming. If the device is not ready, it may set a busy bit until its internal initialization sequence is finished. The device may have an error condition set. Some of the device registers may contain new or old values. The content of these registers after receipt of the negation of INIT should be defined in the device specification. Devices should retain as much status information as possible in order to make error analysis easier.

A device is not required to buffer commands received during its internal initialization sequence, provided it sets a bit indicating that it is not ready to accept commands (busy bit).

Power-Up and Power-Down Sequences
The purpose of the Power-up and Power-down sequences is to guarantee time to store (on power-down) and retrieve (on power-up) the program parameters required for continued operation.

A dc power failure, as used here, means that dc power has dropped below its minimum operating level.

Figure 1-23 shows a typical power-up/power-down sequence. The numbers in the following paragraphs refer to that figure. These numbers are also reflected in the flowchart in Figure 1-24.

1. When power is off in any Unibus device, AC LO and DC LO are asserted and all other Unibus signals are undefined.
2. When the dc voltage to the processor rises to a level at which the logic elements operate, DC LO initializes the processor with BBSY and INIT asserted.
3. (*Power-up*) DC LO is negated by the power supply 5 microseconds after dc power is within specifications.

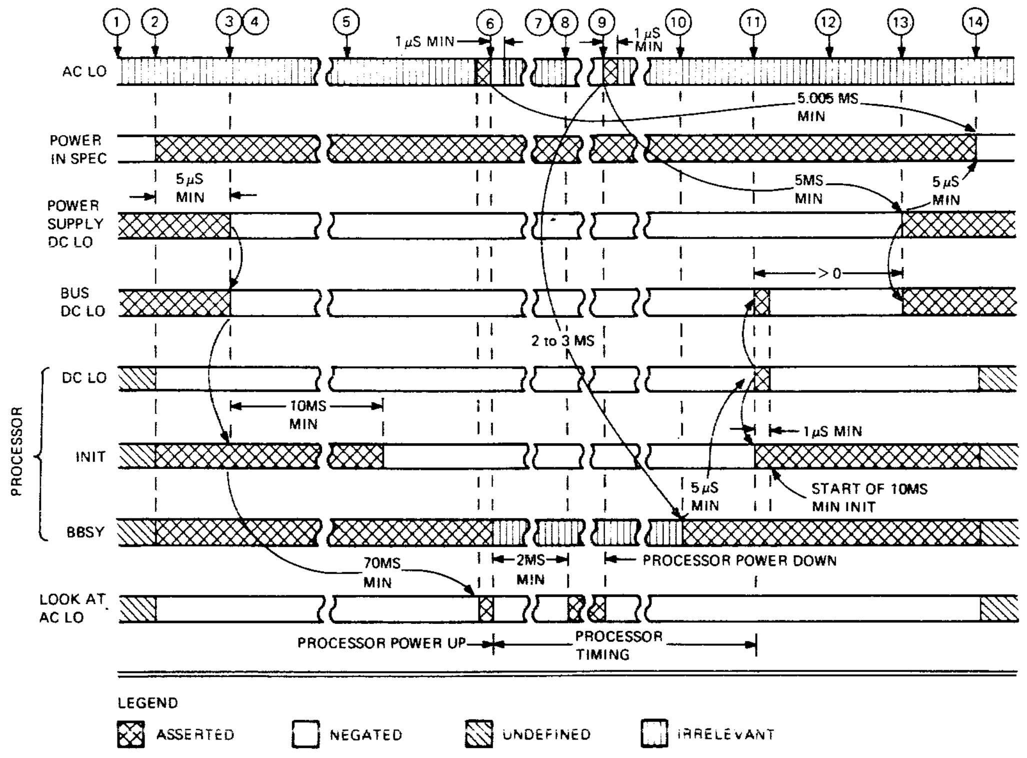


Figure 1-23 Typical Power-Up/Power-Down Sequence

4. INIT remains asserted for a minimum of 10 ms after receipt of the negation of DC LO. This provides time for initialization of most bus devices.
5. The processor waits a minimum of 70 ms after the negation of DC LO to allow all bus devices to complete their internal initialization operations.
6. Before or at the end of this 70 ms delay, INIT is negated. The processor then tests AC LO. When it senses the negation of AC LO, the processor starts its power-up sequence; and the arbitrator is enabled. AC LO must not be negated by the power supply for less than 1 microsecond.

NOTE

While AC LO is negated, dc power is guaranteed to be within specifications for a minimum of 5 ms plus 5 microseconds.

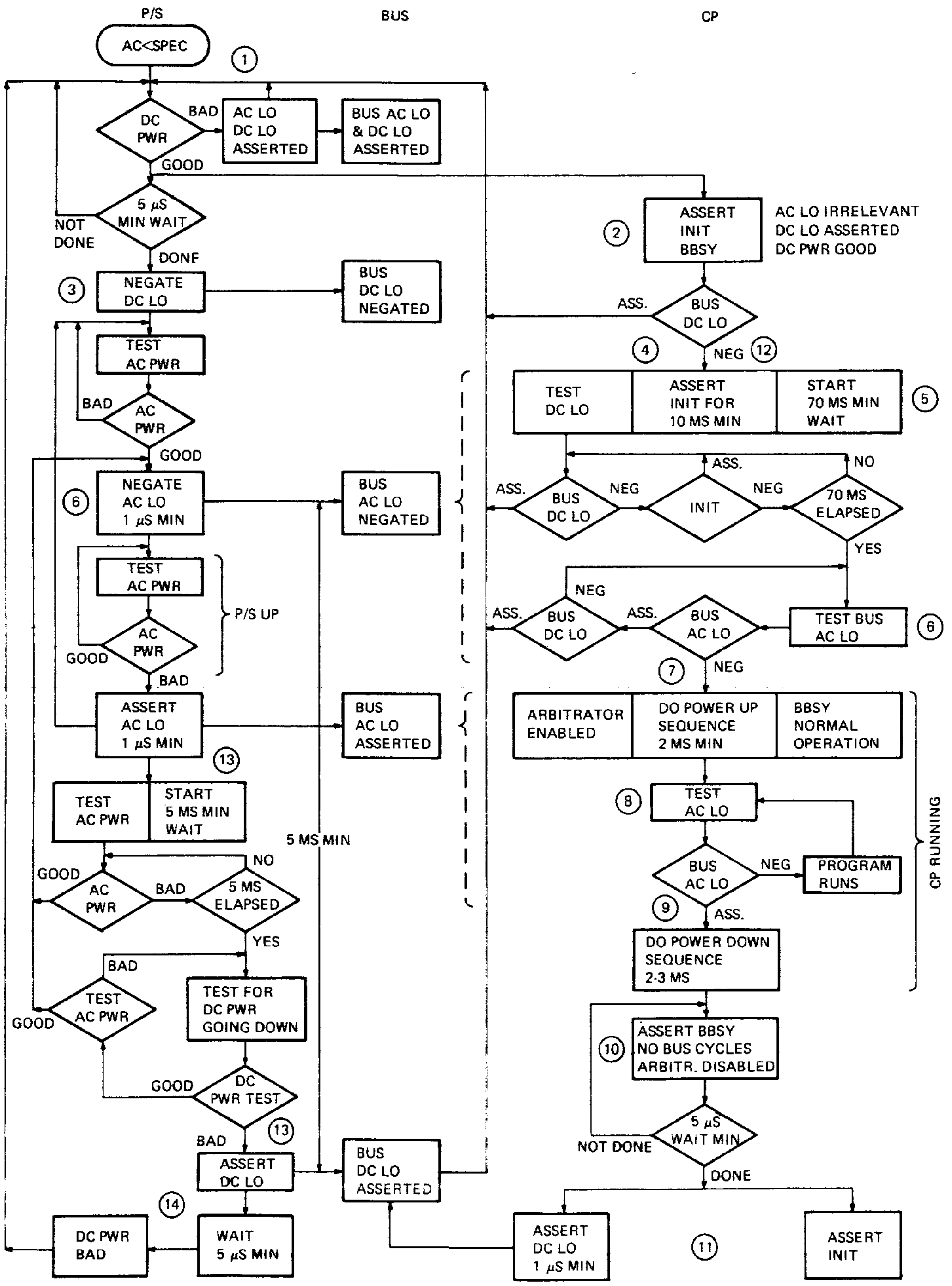


Figure 1-24 Power-Up/Power-Down Flowchart

7. The processor waits a minimum of 2 ms before testing AC LO again. These 2 ms are used by the processor for its power-up sequence.
8. Having completed its power-up sequence, the processor continuously monitors AC LO.
9. (*Power-down*) Upon receipt of the assertion of AC LO, a processor starts its power-down routine. (AC LO must not be asserted by the power supply for less than 1 microsecond.) The processor does not test AC LO again until its next power-up sequence.
10. For a minimum of 2 ms and a maximum of 3 ms, the processor asserts BBSY and does not use the data section of the bus. It stops execution of programs.
11. A minimum of 5 microseconds later, the processor asserts DC LO for at least 1 microsecond. This, in turn, asserts INIT for the same time.
12. The subsequent negation of DC LO by the processor is the beginning of a power-up sequence as in step 3 above. DC LO may be held asserted by a power supply or by another bus device.
13. DC LO must not be asserted by the power supply for a minimum of 5 ms after it asserts AC LO. This ensures enough time is available for a complete power-up and power-down cycle.
14. DC power must be within specifications for a minimum of 5 microseconds after the assertion of DC LO by the power supply.

NOTE

The power-up sequence starts at the negation of AC LO 70 ms after the negation of DC LO. AC LO is not tested during the 2 ms alloted to the power-up sequence. It follows that if AC LO is re-asserted before the end of the power-up sequence (2 ms), the power-down sequence (2-3 ms) must be performed immediately following the power-up sequence. This requires a 5 ms minimum of guaranteed DC power at the negation of AC LO. This also implies that 5 ms of guaranteed DC power is available at any time while AC LO is negated.

A *brown out* occurs when AC LO is asserted, but dc power is within specifications. (DC LO is negated.) In this case, the processor waits for the negation of AC LO as in step 6 above.

UNIBUS INTERFACE DESIGN GUIDELINES

This section presents Unibus interface guidelines. They are not restricted to a specific type of Unibus device but are presented for general reliability and compatibility with the Unibus. The circuit examples in this section demonstrate concepts only; they are not intended to be directly implementable logic designs.

Preferred Interface Integrated Circuit (IC) Chips

The following three IC chips are recommended for use in new Unibus interface designs.

IC Type	DEC OPTION*	Function
8640	00956	Quad NOR gate (receiver). Pin-compatible replacement for DEC380.
8641	00964	Quad transceiver (receiver/driver). Pin-compatible replacement for DEC8838.
8881	00957	Quad NAND gate (driver).

* Package of 10.

Customers may purchase these chips from DIGITAL using the option number. These three chips are the only ones that DIGITAL approves for customer-designed Unibus interfaces at this time. Figures 1-25, 1-26, and 1-27 shows the circuit schematics of these chips.

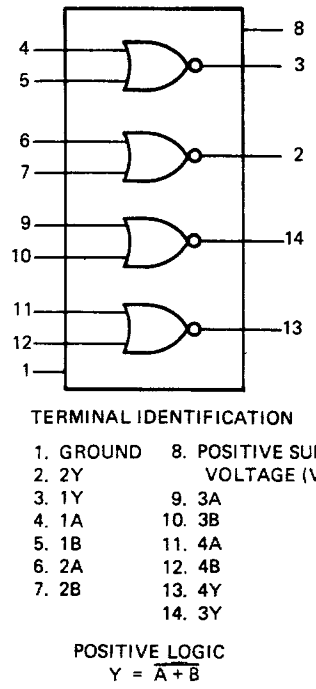
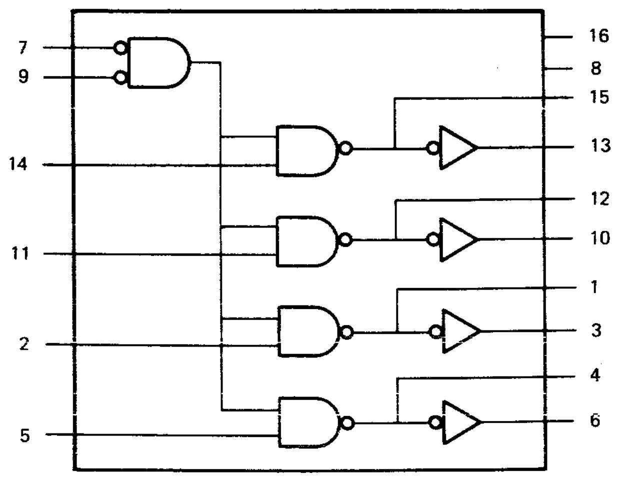


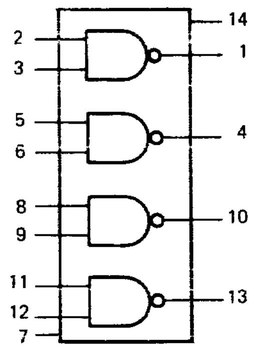
Figure 1-25 8640 Bus Receiver (NOR Gates)



TERMINAL IDENTIFICATIONS

1. BUS 3	9. ENABLE A
2. INPUT 3	10. OUTPUT 2
3. OUTPUT 3	11. INPUT 2
4. BUS 4	12. BUS 2
5. INPUT 4	13. OUTPUT 1
6. OUTPUT 4	14. INPUT 1
7. ENABLE B	15. BUS 1
8. GROUND	16. POSITIVE SUPPLY VOLTAGE (VCC)

Figure 1-26 8641 Bus Transceiver



POSITIVE LOGIC
$Y = \overline{AB}$

TERMINAL IDENTIFICATION

1. 1Y	8. 3A
2. 1A	9. 3B
3. 1B	10. 3Y
4. 2Y	11. 4A
5. 2A	12. 4B
6. 2B	13. 4Y
7. GROUND	14. POSITIVE SUPPLY VOLTAGE (VCC)

Figure 1-27 8881 Bus Driver (NAND Gates)

Unibus Transmitter (Driver)

An element of the 8881 transmitter is shown in Figure 1-28. If both inputs to the transmitter are logic 1 (high), this places a logic 1 (low) on the Unibus. Logically the transmitter is an AND gate; that is, there is voltage inversion but no logic inversion. If an 8881 transmitter has both inputs at logic 1, there is a logic 1 on the Unibus.

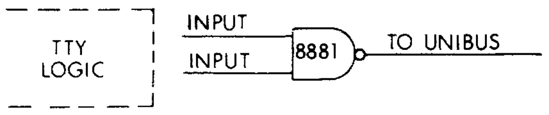


Figure 1-28 Unibus Transmitter

Unibus Receiver

An element of the 8640 receiver is shown in Figure 1-29. If the Unibus line is a logic 1 (low), the output of the receiver is a logic 1 (high). Logically the receiver has no effect on the signal; that is, there is voltage inversion but no logic inversion. The voltage inversion cancels the voltage inversion of the Unibus transmitter.

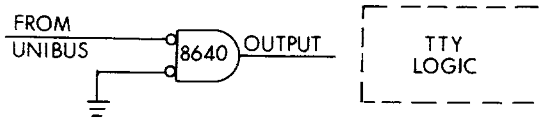


Figure 1-29 Unibus Receiver

NOTE

If the inputs in Figure 1-29 were connected to separate Unibus signals, the receiver is logically a 2-input OR gate. The other input can also be connected to a TTL output within the module.

Bus Receiver and Transmitter Equivalent Circuits

The equivalent circuits of the standard Unibus receivers and transmitters are shown in Figure 1-30. The characteristics of the ICs used by DIGITAL are listed in Table 1-3.

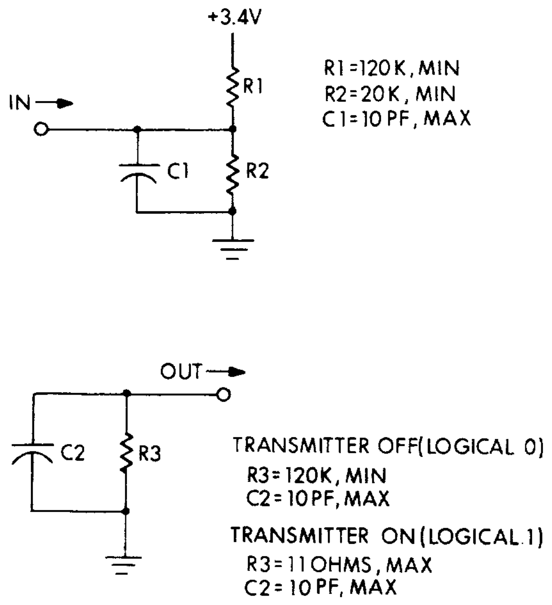


Figure 1-30 Transmitter and Receiver Equivalent Circuits

Table 1-3

Bus Driver/Receiver Specifications

	Characteristic		Specifications	Notes
Receiver (DEC 8640, DEC 8641)	Input high threshold	VIH	1.7 V min	1
	Input low threshold	VIL	1.3 V max	1
	Input current at 2.5 V	IIH	80 μA max	1, 3
	Input current at 0 V	IIL	10 μA max	1, 3
	Output high voltage	VOH	2.4 V min	2
	Output high current	IOH	(16 TTL loads)	2, 3
	Output low voltage	VOL	0.4 V max	2
	Output low current	IOL	(16 TTL loads)	2, 3
	Propagation delay to high state	TPDH	10 ns min 35 ns max	4, 5
	Propagation delay to low state	TPDL	10 ns min 35 ns max	4, 5
Driver (DEC 8881, DEC 8641)	Input high voltage	VIH	2.0 V min	6
	Input low voltage	VIL	0.8 V max	6
	Input high current	IIH	60 μA max	6
	Input low current	IIL	−2.0 mA max	6
	Output low voltage at 70 mA sink	VOL	0.8 V max	1
	Output high leakage current at 3.5 V	IOH	25 μA max	1, 3
	Propagation delay to low state	TPDL	25 ns max	5, 7
	Propagation delay to high state	TPDH	35 ns max	5, 8

NOTES

1. This is a critical parameter for use on the I/O bus. All other parameters are shown for reference only.
2. This is equivalent to driving 16 unit loads of standard 7400 series TTL integrated circuits.
3. Current flow is defined as positive if it goes into the terminal.
4. Conditions of load are 390 ohms to −5.0 V and 1.6K ohms in parallel with 15 pF to ground for 10 ns min and 50 pF for 35 ns max.

5. Times are measures from 1.5 V level on input to 1.5 V level on output.
6. This is equivalent to 1.25 standard TTL unit loading of input.
7. Conditions of 100 ohms to +5V, 15 pF to ground on output.
8. Conditions of 1K ohms to ground on output.
9. Bus transceivers (DEC 8641) meet the above specifications.

NOTE

The preceding specifications must be met, but are not necessarily sufficient in all cases to insure compatability with the Unibus.

DC Bus Load

DC loading is the amount of dc leakage current a module presents to a bus signal line. DC loading is expressed in terms of dc bus loads. One dc load is approximately 105 microamps. This is equivalent to having a maximum of one transmitter and one receiver (or one transceiver) per Unibus line as shown in Figure 1-31. Therefore, if a new design is to be rated as one bus load, each Unibus line may be loaded with only one of the following:

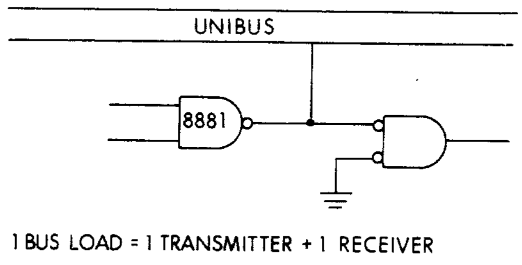


Figure 1-31 One DC Bus Load

1. One receiver
2. One transmitter (or driver)
3. One transmitter and one receiver
4. One transceiver (preferred over 3 above)
5. None of the above

The Unibus is limited to a maximum of 20 dc bus loads. This limit is set to maintain a sufficient noise margin. For more than 20 dc bus loads, a Unibus repeater option (DB11-A) is used.

Module Layout

The interface IC chips should be kept as close to the module fingers as possible, preferrably in rows one and two (where row one contains the ICs located nearest the module fingers). The etch runs from the fingers to the chips should be made as short as possible. In particular, etch runs on BUS SSYN L, BUS BBSY L and BUS MSYN L should not exceed two inches if the module is double-layer; or one inch if the module is multi-layer.

If there is a choice between placing either a driver or a receiver closer to the module fingers, the driver should be chosen. This provides the driver with a shorter ground return path.

Backplanes
Backplanes for non-SPC (small peripheral controller) modules should have the Unibus signals routed between the Unibus-in and Unibus-out slots with either printed circuit (PC) etch or number thirty (30) wirewrap. The signals required by modules should be obtained from rows A and B, as shown in Figure 1-32.

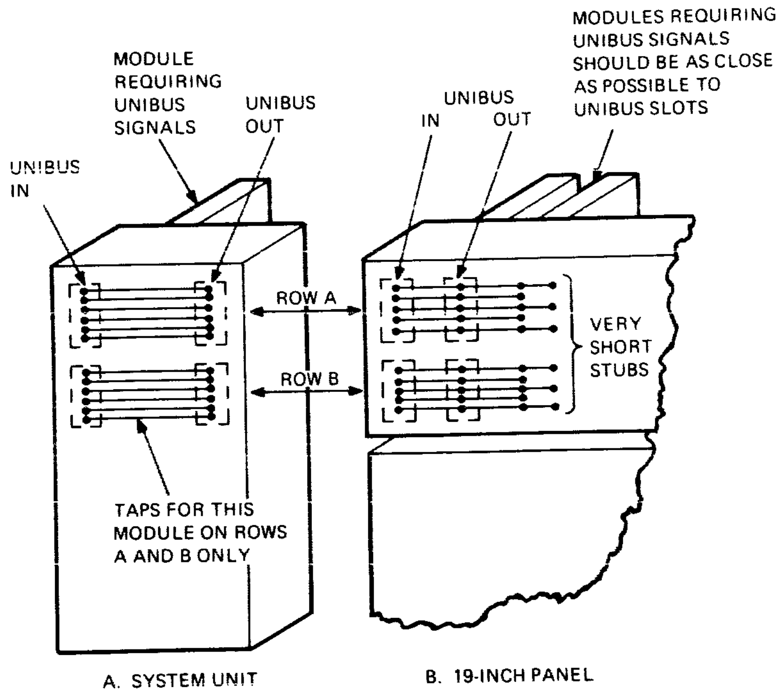

UNIBUS IN
MODULE REQUIRING UNIBUS SIGNALS
UNIBUS OUT
ROW A
ROW B
TAPS FOR THIS MODULE ON ROWS A AND B ONLY
UNIBUS IN OUT
MODULES REQUIRING UNIBUS SIGNALS SHOULD BE AS CLOSE AS POSSIBLE TO UNIBUS SLOTS
VERY SHORT STUBS
A. SYSTEM UNIT
B. 19-INCH PANEL

Figure 1-32
Design of Backplanes That Do Not Accommodate SPCs

Critical signals on backplanes for SPC modules should not have stubs. (These are short wire runs or etch runs that are connected to the main signal line at only one end, as shown in Figure 1-33a.) To prevent stubs, signal runs are routed so that Unibus signals travel through interfaces, as shown in Figure 1-33b. For example, the BUS SACK L must also be routed in this manner so that its run length is the same as that of BUS BBSY L. This prevents skew buildup between BUS SACK L and BUS BBSY L signals. The following critical signals should be routed from Unibus in to Unibus out connectors.

BUS SSYN L
BUS BBSY L
BUS MSYN L
BUS SACK L
BUS A<00:17> L
BUS C0 L
BUS C1 L

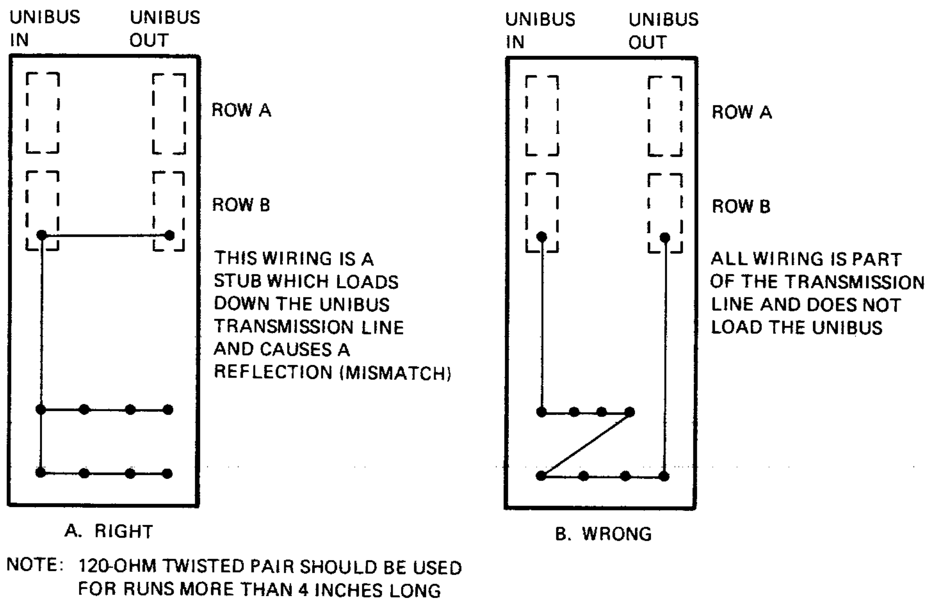

UNIBUS IN
UNIBUS OUT
ROW A
ROW B
THIS WIRING IS A STUB WHICH LOADS DOWN THE UNIBUS TRANSMISSION LINE AND CAUSES A REFLECTION (MISMATCH)
A. RIGHT
NOTE: 120-OHM TWISTED PAIR SHOULD BE USED FOR RUNS MORE THAN 4 INCHES LONG
UNIBUS IN
UNIBUS OUT
ROW A
ROW B
ALL WIRING IS PART OF THE TRANSMISSION LINE AND DOES NOT LOAD THE UNIBUS
B. WRONG

Figure 1-33 Design of Critical Signal Runs in Backplanes That Accommodate SPCs

Excessive backplane cross talk onto the Unibus signals may cause system failures. For this reason, it is recommended that the following Unibus signals use 120-ohm twisted pair (DIGITAL part number 91-07773) for any backplane wire run that exceeds four (4) inches in length:

BUS INIT L	BUS NPR L	BUS SSYN L
BUS INTR L	BUS NPG H	BUS MSYN L
BUS BBSY L	BUS BR<4:7> L	BUS AC LO L
BUS SACK L	BUS BG<4:7> H	BUS DC LO L

PC etch or number thirty (30) wirewrap may be used for other wire runs. Twisted pairs, other than the type mentioned above, should not be used to route Unibus signals because their impedance may contribute significantly to unwanted signal reflections.

System units (SU) should be designed to accommodate no more than four (4) dc bus loads. This restriction limits the maximum capacitive loading at any point on the Unibus. This allows the use of an M9202 (Unibus jumper with cable) to distribute capacitance and to resolve failures caused by reflections. Each new Unibus interface should be designed with all of the above rules kept in mind to maintain the integrity of the bus.

Grounding

Noise on backplane and module ground return paths may cause system failures. These failures may be avoided by providing all ground returns with a low-impedance path to a common ground plane.

Multi-layer modules with internal V_{CC} and ground planes solve this problem automatically. The double-layer modules should have mutually perpendicular ground and V_{CC} runs with 0.01 mfd decoupling capacitors at each intersection (normally at each IC). This forms a low-impedance ground reference plane.

Module areas with many Unibus drivers and receivers should have a wider than normal etch path on V_{CC} and ground because of the very large, high-speed switching currents in those areas. The ground path connecting these Unibus drivers and receivers should tie to a dedicated ground pin, separated from all other grounds on the module. The same applies to V_{CC} of the same areas.

The PS etch on backplanes should have one side devoted to a ground plane. Backplanes without a PC-etch ground plane are not recommended for new designs.

Logic Design Guidelines for Unibus Interfaces

A good design philosophy is to keep the Unibus "clean." In general, try to prevent placing any signal on the Unibus that is not needed for the transaction in progress. For example, prevent unasserted data line drivers from spiking the Unibus before the assertion of slave sync (SSYN) in a data-in transaction. Even though the spike does not violate the Unibus specification, reflections from it may cause a failure on a heavily loaded Unibus.

NOTE

It is very important to adhere rigidly to the timing restrictions in this specification, particularly with respect to skew timing considerations.

Table 1-4 lists Unibus events that require deskewing. Excessive skew on these events can cause a failure. The table summarizes the critical situations where a transmitted signal is dependent on the timing of another transmitted signal of the same device. This list provides designers with a convenient checklist to verify the interface timing. The worst-case propagation delays in the circuits used should not violate the specification. For example, ensure that the interrupt vector is placed at the driver inputs at the same time as, or before, the BUS INTR signal.

Table 1-4
Unibus Events that Require Deskewing

NOTE

Events or signals are being asserted unless otherwise specified, i.e., (n) = negation.

Event that may cause a failure by occurring too soon	Event that may cause a failure by occurring too late	Device transmitting events	Device receiving events	Device which must deskew	Name of deskew
SSYN (n)	D (releasing)	Slave	Master of next bus cycle (for SSYN) and next device (master or slave) which receives D lines	Master of next bus cycle	SSYN deskew
A and C (releasing)	MSYN (n)	Master	Slave	Master	Tail-end deskew
BBSY (n) (DATI only)	A and C (releasing)	Master	Next master for BBSY; next slave for A and C	Next master	—
A and C (gating)	BBSY	Master	Slave	Master	—
DATO/B Transaction					
MSYN	A and C (gating)	Master	Slave	Master	Front-end deskew
MSYN	D (gating)	Master	Slave	Master	—
A and C (releasing)	MSYN (n)	Master	Slave	Master	Tail-end deskew
BBSY (n)	A and C (releasing)	Master	Next master for BBSY; next slave for A and C	Next master	—
A and C (gating)	BBSY	Master	Slave	Master	—
A and C (releasing)	D (releasing)	Master	Next slave	This master during next bus cycle	—
BBSY (n)	D (releasing)	Master	Next master for BBSY; next slave for D	Next master	—
Initialization					
Assertion of any signal other than AC LO, DC LO, or INIT	INIT (n)	CPU	All Unibus devices	CPU	—

Table 1-4
Unibus Events that Require Deskewing (Cont)

Event that may cause a failure by occurring too soon	**Event that may cause a failure by occurring too late**	**Device transmitting events**	**Device receiving events**	**Device which must deskew**	**Name of deskew**
NPR Arbitration Sequence					
NPR (n)	SACK	Master	Arbitrator	Arbitrator	—
BBSY	SACK	Master	Arbitrator	Arbitrator	—
SACK (n)	NPR (n)	Master	Adbitrator	Arbitrator	—
SACK (n)	BBSY	Master	Interrupt fielding processor	Interrupt fielding processor	—
BBSY (n)	SACK (n)	Master	Arbitrator	Arbitrator	—
Interrupt Transaction					
SACK (n)	INTR	Requesting device	Interrupt fielding processor	Interrupt fielding processor	—
INTR	D<02:08> (gating)	Requesting device	Slave (CPU)	Slave (CPU)	Vector deskew
BBSY (n)	D<02:08> (releasing)	Requesting device	Next master (for BBSY) and next device (master or slave) which receives D lines	Next master	—
BBSY (n)	INTR (n)	Requesting device	Next master (for BBSY) and CPU receives INTR	CPU	—
INTR (n)	D<02:08>, (releasing)	Requesting device	CPU	CPU	—
D<02:08> (gating)	BBSY	Requesting device	CPU	CPU	—
DATI/DATIP Transaction					
MSYN	A and C (gating)	Master	Slave	Master	Front-end deskew
SSYN	D (gating)	Slave	Master	Master	Data deskew

Master Devices

A master device is a device which is capable of becoming bus master; that is, a device that can command the data section of the bus. A master device must obtain the bus through the arbitration process, assert BUS BBSY L, and execute one or more data transfers. The master must also be able to receive commands relating to these functions. These commands are generally receievd from a processor.

The design of master devices is described in this section. The following topics are presented:

1. the Unibus priority transfer control logic,
2. an interface for a device that requires a BR interrupt at one priority level,
3. an interface for a device that requires two BR interrupts at 2 priority levels, and
4. the Unibus interface for an NPR device.

The examples in this section are typical of existing Unibus interface designs; they are presented here so that the implementation of the protocol described in this specification may be understood. They are not "model" or "ideal" designs; they do not "stretch" the protocol to its limits. They do, however, conform to the requirements of this specification.

Unibus Control Logic

A control logic diagram is presented in Figure 1-34. Items noted in the following paragraphs refer to that diagram. This diagram gives a logical sequence of events; it is not intended to be implemented in any circuit.

In this example:

1. The following signals are connected to their respective Unibus signal lines.

BUS NPR L	BUS SACK L
BUS REQUEST L	BUS BBSY L
BUS GRANT IN H	BUS SSYN L
BUS GRANT OUT H	

2. These signals are connected as described in the following paragraphs.

REQUEST L	INIT H
MASTER CLR H	CLR SACK ENB L
STEAL GRANT L	SACK H
MASTER L	

REQUEST L (item 1 in Figure 1-34) is generated by the master device. It is asserted to initiate a priority transfer sequence. It becomes the D input to the Take Grant flip-flop.

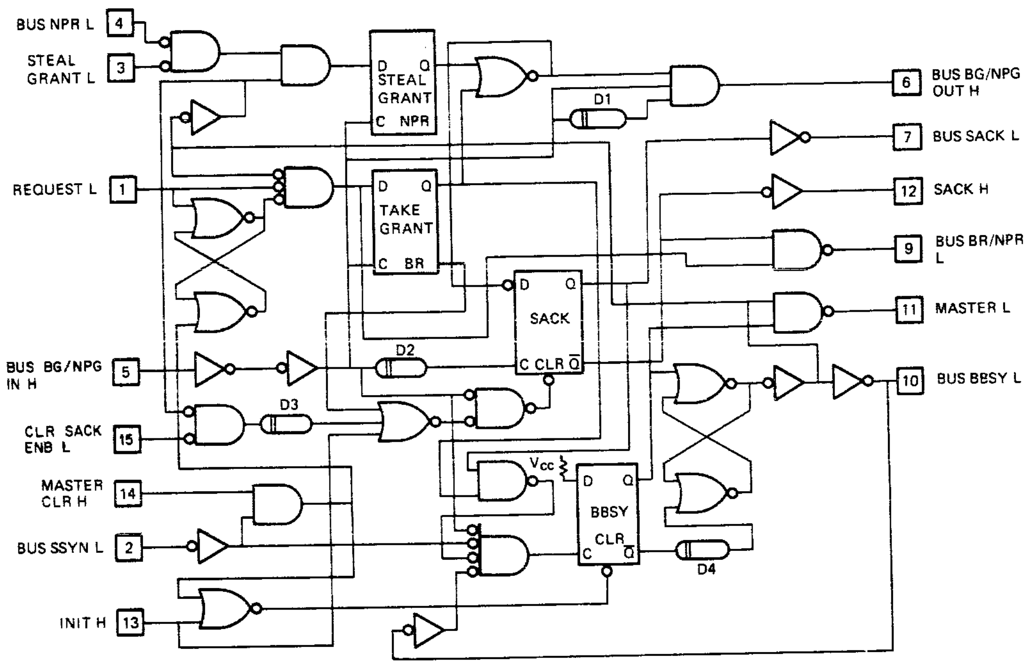


Figure 1-34 Unibus Control Logic Example

The latch ensures that only one priority transfer sequence is initiated for each REQUEST L. Once the latch is set, REQUEST L must be negated and asserted again before another cycle begins. MASTER CLR H (item 14) must not be grounded. The device must not already be bus master and must not be asserting BUS BBSY L (item 10).

With the SACK flip-flop reset, BUS REQUEST L (item 9) is asserted.

NOTE

The device cannot initiate a priority transfer sequence if it is already bus master since both BUS BBSY L and BUS SACK L must be negated before BUS SACK L is asserted. The device negates BUS REQUEST L when it asserts BUS SACK L.

BUS SACK L must be asserted at the same time as, or before, BR or NPR is negated. The approved Unibus drivers have a maximum skew of 35 ns. This timing requirement is met if the assertion of BUS SACK L is skewed at the driver output no more than 35 ns from the negation of BUS REQUEST L.

The state of BUS BBSY L (item 10) determines whether the received BUS GRANT IN H (item 5) is passed or blocked. If BUS

BBSY L is low, the device accepts the grant and blocks it from passing to another device. In this case, it desires to become next bus master. If BUS BBSY L is high, the grant passes to another device of the same priority level.

BUS GRANT IN H clocks the Take Grant and Steal Grant flip-flops. With BUS REQUEST L asserted, the Take Grant flip-flop is set; or if another device is asserting BUS NPR L and STEAL GRANT L (item 3) is asserted, the Steal Grant flip-flop is set. Either flip-flop being set disables the Bus Grant driver; the grant is accepted and the SACK flip-flop is set (after delay D2).

NOTE

> Delay D2 ensures that the Take Grant and Steal Grant flip-flops have time to respond to BUS GRANT IN H before the SACK flip-flop is clocked.

BUS SACK L (item 7) is asserted. This allows the arbitrator to negate the grant after a minimum of 75 ns and BUS REQUEST L is negated.

The BBSY flip-flop is set when its clock input conditions are satisfied; that is, the Take Grant flip-flop is set; the SACK flip-flop is set; and BUS GRANT IN H, BUS SSYN L, and BUS BBSY L are all negated.

BUS BBSY L (item 10) and MASTER L (item 11) are asserted. BUS BBSY L maintains the bus for the master to perform data and/or interrupt transfers. MASTER L is a signal that may be used by the master to initiate a data transfer or an interrupt sequence; it may be used to enable the appropriate data lines and bus interrupt.

When the requesting device completes the transfer, it asserts MASTER CLR H (item 14). When both MASTER CLR H and BUS SSYN L (item 2) are asserted (signifying transfer complete), the BBSY flip-flop is reset. This negates MASTER L and after 80 ns (delay D4) negates BUS BBSY L.

NOTE

> Delay D3 ensures that the external BUS INTR (driven by MASTER L) is asserted before the SACK flip-flop is cleared. The SACK flip-flop may be reset when BUS GRANT IN H is negated.
>
> CLR SACK ENB L is typically grounded (asserted).
>
> Delay D4 ensures that the negation of BUS BBSY L occurs at least 80 ns after the negation of MASTER L.

INIT H (item 13) will directly clear BBSY and SACK flip-flops if BUS GRANT IN H is negated.

This system can improve NPR latency. A device not requesting the bus may assert STEAL GRANT L. It receives NPR L followed by a bus grant. It blocks the bus grant and asserts BUS SACK L. The grant, intended for a device further down the bus, is intercepted. BUS SACK L causes the arbitrator to negate the BG and stop arbitrating. BUS GRANT IN H, now negated, resets the SACK flip-flop.

A device closer to the arbitrator may assert an NPR before a grant is issued to the original BR device. The arbitrator first honors the NPR and issues an NPG. When the NPR device has completed its transaction, the arbitrator issues a BG to the original device.

Bus Request (BR) Device—One Vector

Figure 1-35 shows a circuit schematic for a Unibus interface that requires one interrupt vector. This interface uses the control logic in Figure 1-34 and demonstrates how this can be used in a typical application. The circuit shows how the interrupt vector should be multiplexed with the device's data lines. By multiplexing before the Unibus drivers, the total Unibus loading of the device can be significantly reduced. Also, the gating shown ensures operation within the timing constraints specified in this specification.

In this circuit, a bus request is initiated by the device whenever an interrupt is necessary. INT ENB is normally a bit in the control and status register of the device. When INT ENB H and INT H become enabled, REQUEST L is asserted. This initiates a priority transfer sequence on the Unibus.

When the device receives the grant, MASTER L is asserted and begins the interrupt sequence.

The master negates both REG SELECT L and IN L. The Unibus transceivers for data lines (D<00:01> and D<10:15>) are disabled.

When the multiplexer select input (SEL) is disabled, the A inputs determine the output. When SEL is enabled, the B inputs determine the output. Cutting the appropriate jumpers on the B inputs selects the desired interrupt vector.

The assertion of MASTER L:

1. enables one input to Bus Interrupt driver (8881),
2. activates SEL inputs of the multiplexors,
3. activates ENB 2 lines of the transceivers that drive D<02:09>, and
4. after a delay, enables other input of Bus Interrupt INTR driver.

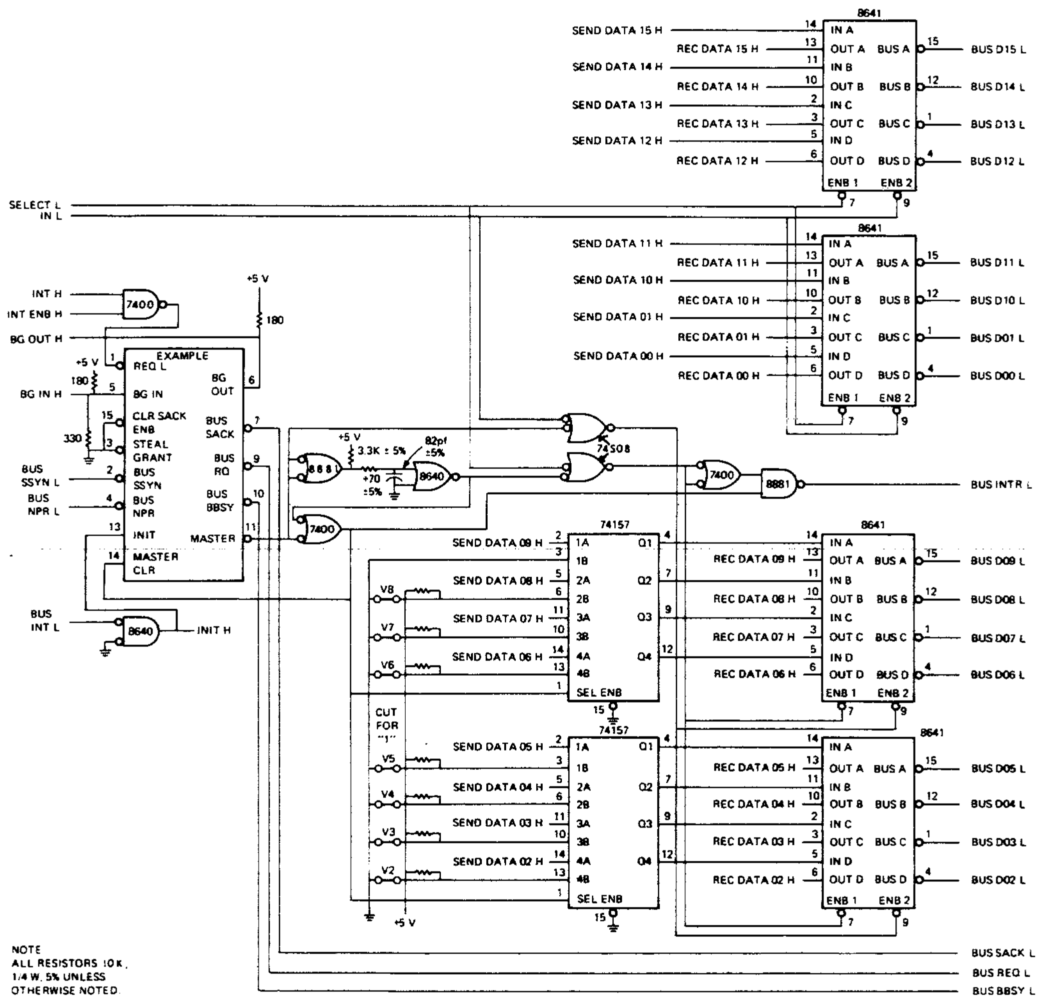


Figure 1-35 Bus Request (BR) Device—One Vector Circuit

BUS INTR L is asserted after the transceivers are enabled. Thus, the interrupt vector is asserted before BUS INTR L in compliance with timing constraints specified in this specification.

When the processor receives BUS INTR L, it strobes the vector from the D lines and issues a BUS SSYN L.

The device:

1. receives BUS SSYN L,
2. negates MASTER L terminating the vector and BUS INTR L, and
3. after a delay of at least 80 ns, negates BUS BBSY L.

The use of Schottky-series logic ensures that both BUS INTR L and D<02:09> are negated within 80 ns of MASTER L negation; thus, all asserted lines have been negated when BUS BBSY L is negated. This ensures that the termination of the interrupt sequence is in compliance with timing constraints specified in this specification.

Note that the CLR SACK ENB L and the STEAL GRANT L signals, shown in Figure 1-34, are grounded (remain asserted) for this example.

Bus Request (BR) Device—Two Vectors
Figure 1-36 shows a circuit schematic for a Unibus interface that requires two interrupt vectors. This example uses two circuits like control logic shown in Figure 1-34. A typical application of this circuit may be for separate "send" and "receive" vectors. By using two complete circuits, an interrupt request can be initiated for either vector regardless of the state of the other vector.

NOTE

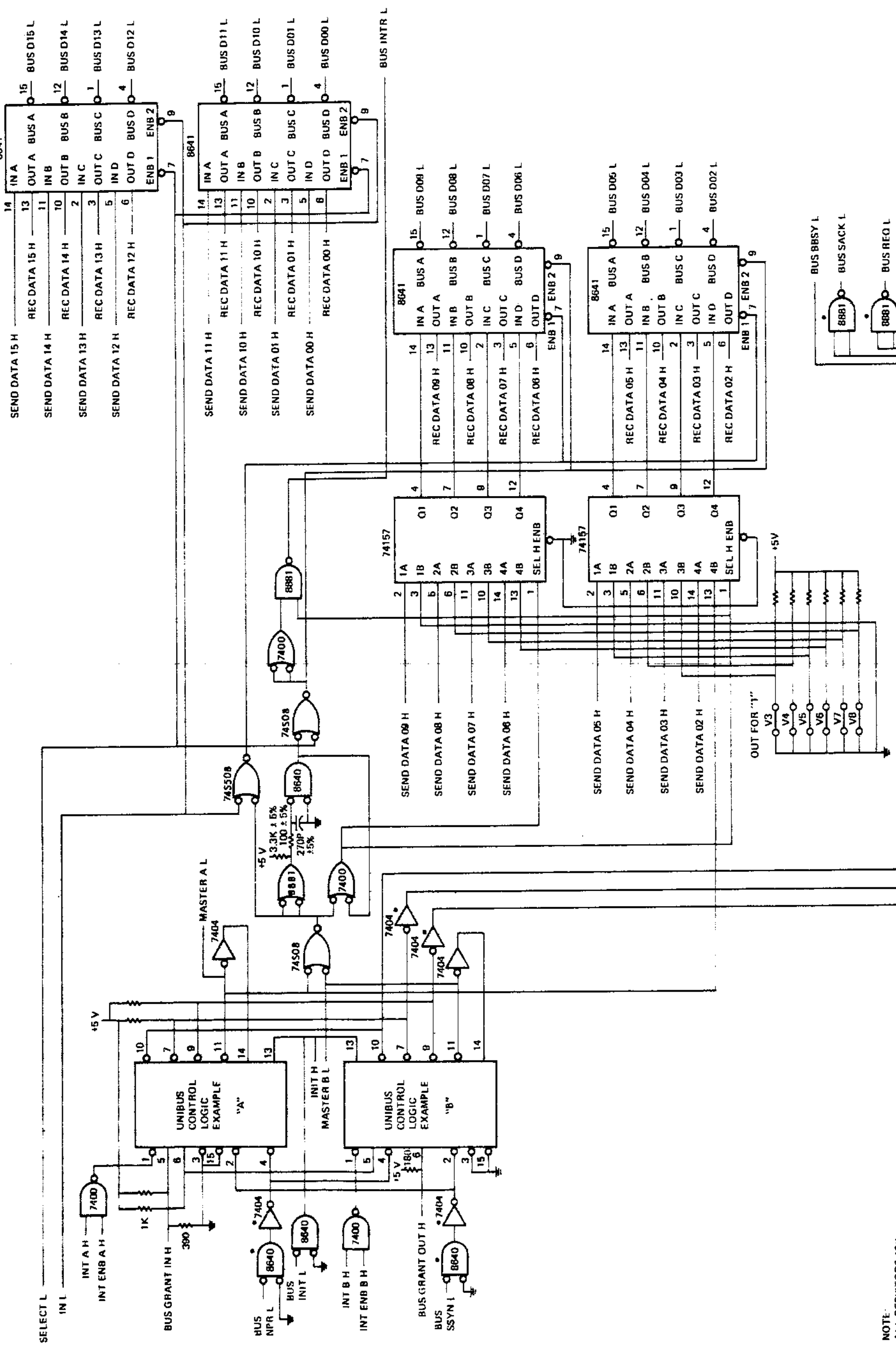
Signals are labelled "A" for the top circuit and "B" for the bottom circuit. The signals CLR SACK ENB L and STEAL GRANT L are grounded on both A and B circuits.

MASTER A L is directly connected to the DATA 02 input of the multiplexer. Two independent vectors are implemented: vector XX0 for circuit A and XX4 for circuit B. The value of XX is determined by cutting appropriate jumpers V3-V8.

Either circuit A or B may initiate an interrupt sequence. For example, assert INT A H while INT ENB A H is enabled; a BR sequence is initiated by circuit A.

Both circuits have the same BR level. Note that:

1. The BUS GRANT IN H input of circuit A is connected to the Unibus,



NOTE:
ALL RESISTORS 10K,
1/4 W, ±5% UNLESS
OTHERWISE NOTED.

* GATES ADDED OR REQUIRED TO ACHIEVE ONE UNIT LOAD

Figure 1-36 Bus Request (BR) Device—Two Vector Circuit

2. the BUS GRANT OUT H output of A is connected (with a pull-up resistor) to the BUS GRANT IN H input of B, and
3. the BUS GRANT OUT H output of B drives the Unibus grant line of the same level.

Thus, A and B form two links in the grant chain for whatever priority level is selected. Because the two circuits are daisy-chained, they must both request at the same BR level.

When the device becomes bus master, MASTER L of one of the circuits is asserted. The state of MASTER L determines which vector is placed on the data lines during the interrupt sequence.

NOTE

It is possible to use two different BR levels. The signals BUS BG IN and BUS BG OUT and the BR level must be configured the same for each circuit. Circuit A cannot be daisy-chained to circuit B for these signals.

Non-Processor Request (NPR) Device
Figure 1-37 illustrates a typical NPR Unibus interface using the control logic shown in Figure 1-34.

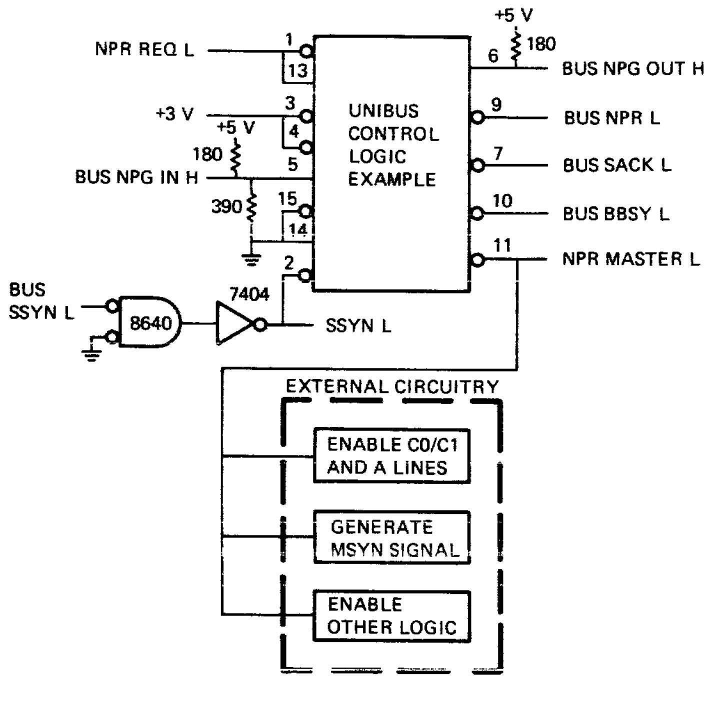


Figure 1-37 NPR Device Circuit Schematic

When an NPR transfer is required, the requesting device asserts NPR REQ L. The arbitrator recognizes this request and issues an

NPG. The requesting device blocks the grant from passing through to the next device and acknowledges with a BUS SACK. With BUS SACK asserted, the arbitrator negates the grant and stops arbitrating. The requesting device receives the negations of BUS BBSY L, BUS SSYN L and BUS NPG H. It then asserts BUS BBSY L and becomes bus master. As bus master it starts the data transfer(s).

NPR MASTER L is asserted at the same time as BUS BBSY L. NPR MASTER L is used to drive external circuitry (not shown); for example, it can enable the data and address lines or generate MSYN L and other control logic.

When the data transfer is completed, the bus mastership is terminated by negating the NPR REQ L signal. This signal must be asserted again if another transfer is required. Note the NPR REQ L is directly connected to INIT H signal. Once the request is asserted, it must remain asserted until the data transfer is completed; otherwise, bus mastership is terminated prematurely.

NOTE

The Steal Grant flip-flop, in Figure 1-34, is disabled by connecting BUS NPR L and STEAL GRANT L to a logic "1" level.

The Control Logic example can prevent the negation of BUS SACK for devices that require more than one data cycle. CLR SACK ENB is held high until the beginning of the last bus cycle; for example, if 100 NPR data cycles are to be transferred, this signal is held to a logic "1" until the completion of 99 data cycles. The bus is then given to the highest priority requesting device at the end of the 100 transfers.

The CLR SACK ENB signal is grounded (asserted), in Figure 1-34, so only one bus cycle is done for each request.

An interrupt cannot be done by a device that becomes bus master under an NPG. In most NPR applications, the completion of the current set of NPRs is usually followed by an interrupt (not shown). This interrupt may be used to notify the processor that the NPR transfers are complete, or an error has occurred during the data transfers.

UNIBUS CONFIGURATION

The definitions, rules, and guidelines in this section provide an aid to configure a reliable system on the Unibus. First, the various units such as CPU, memory, peripherals, disks, magtapes, display, printer, terminals, etc. must be mounted in a sensible and workable arrangement within the equipment cabinets. Some equipment must be located near other units for speed reasons; some equipment must be located at certain heights for ease of operator use; some equipment must be spread out so as not to overload an individual power supply; and some equipment must be located at various points along the Unibus because of timing restrictions.

The Unibus is a transmission line for the high-speed signals traveling along it. To preserve the electrical properties of the Unibus, the rules that follow must be observed. These are explained in further detail in this section.

1. The total length of the Unibus cable should not exceed 15.24 m (50 ft).
2. The Unibus is rated to drive 20 dc bus loads.
3. One segment on the Unibus should contain no more than 20 ac bus loads.
4. For additional loads or lengths, a bus repeater (DB11-A) is used, dividing the Unibus in sections. The DB11-A can drive an additional 19 bus loads and up to 50 feet.
5. Each bus section must be terminated at each end by a Unibus terminator, or its equivalent. Normally a CPU module contains terminators for one end of the bus.

To optimize the system, the total cable length should be kept as short as possible, and bus repeaters should be kept to a minimum. Bus repeaters slow down the system and add to the cost. If the number of bus repeaters is excessive, total cable length can sometimes be reduced by rearranging the order of options on the bus. Then, reapply the rules in this configuration section.

Definitions
The following definitions help to understand the configuration rules.

Unibus Segment—A Unibus segment is a portion of a Unibus system between and including two terminators. A bus segment consists of a terminator, a 120-ohm transmission path (cable) with options having drivers and receivers, and another terminator (in that order). A single bus system has one bus segment. A multiple bus system contains more than one bus segment, usually separated by bus repeaters or bus switches. For example, a DT03 bus switch contains bus repeaters.

Unibus Cable—A Unibus cable is a 120-conductor ribbon cable connecting two Unibus backplanes or system units. It is considered part of a Unibus element. The 120 lines include all 56 Unibus signals plus 64 ground lines. Various cable lengths are available from DIGITAL; refer to the later section on Unibus Hardware.

Unibus Element—A Unibus element is any module, backplane, cable or group of these with one or more Unibus signal lines (other than AC LO or DC LO). For example, a DD11 backplane or a BC11A cable is a Unibus element; an H720 power supply or a BA11 expander box is not a Unibus element .

AC Unit Load—An ac unit load is defined as a number related to the impedance that a Unibus element presents to a Unibus signal line (due to backplane wiring, PC etch runs, receiver input loading, and driver output loading). This impedance load on a transmission line causes a "reflection" to occur when a step is sent down the line. This reflection shows up on an oscilloscope as a spike occurring shortly after asserting or unasserting edge. AC loads must be distributed on the Unibus in order to provide bus operation with reflections guaranteed to be at or less than a tolerable level.

DC Unit Load—A dc unit load is defined as a number related to the amount of dc leakage current that a Unibus element presents to a Unibus signal line which is high (undriven). A dc unit load is nominally 105 μA (80 μA—receiver plus 25 μA—driver). However, the dc unit load rating of a bus element is not strictly based on the element's signal line that has the greatest leakage, (e.g., dc leakage is less important on D lines than it is on SSYN).

Lumped Load—A lumped load is a group of Unibus elements (other than cables or jumpers) that are interconnected via the Unibus (backplane wire or PC etch). The group is not a lumped load if it uses a cable to interconnect the elements, or if the elements are separated by a bus repeater.

Example 1:
Figure 1-38a shows a system with 2 lumped loads as follows.

1. M930, 11/05 CPU, and MM11-L
2. RK11-D, DD11-B, DL11-A, and M9301

If the M920 is replaced by an M902 cable, then the system has 3 lumped loads, as shown in Figure 1-38b.

1. M930, 11/05 CPU, and MM11-D
2. RK11-D
3. DD11-B, DL11-A and M9301

Example 2:

Figure 1-39 shows a system with two Unibus segments separated by a bus repeater; this system has two lumped loads as follows.

1. M930, 11/45 CPU, DB11-A (left side)
2. DB11-A (right side), DD11-B, four DL11-As, M9301

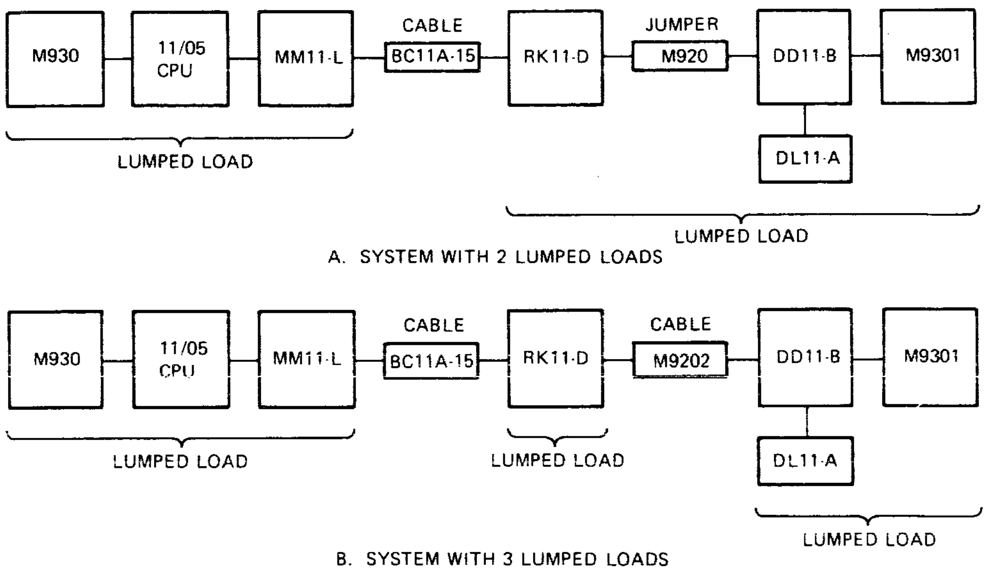


Figure 1-38 Lumped Loads (Example 1)

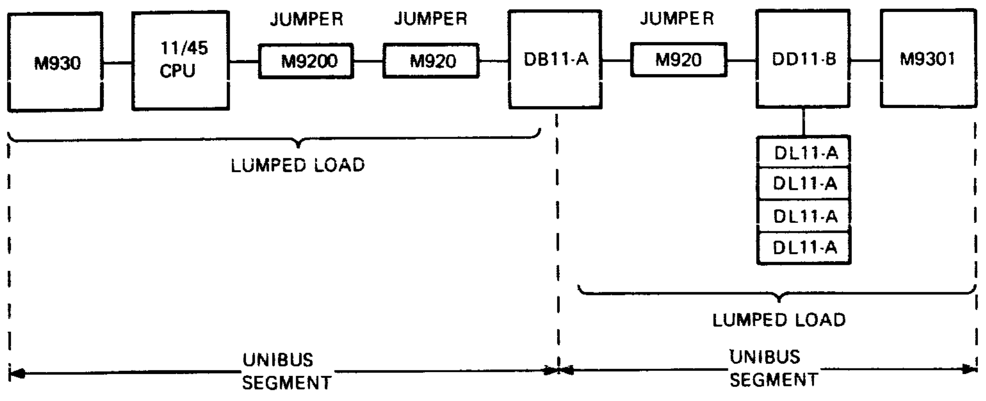


Figure 1-39 Lumped Loads (Example 2)

Bus Terminator—A bus terminator is a bus element (or part of an element) containing a resistive network that connects to the end of a bus segment and matches the 120 ohm-characteristic impedance of the bus transmission path. The following bus elements contain Unibus terminators.

M930	bus terminator
M981	jumper/terminator
M9300	Unibus B terminator (M930 + NPR logic)
M9301	bootstrap/terminator
M9302	M930 with SACK return
M9306	11/04 Near End Unibus Terminator
M9312	Bootstrap/Terminator/ROM Option
DT03	bus switch
DB11-A	bus repeater
CPU	CPUs also contain terminators for one end of the bus.

A Unibus segment must always have a Unibus terminator at each end of its 120-ohm transmission path.

Semi-Lumped Load—A semi-lumped load is a group of lumped loads interconnected by 91.44 cm (3 ft) or less of cable and not separated by a bus repeater. This cable may be one of the following:

Cable	**Length**
BC11-A-0F	15.24 cm (0.5 ft)
BC11-02	60.96 cm (2 ft)
BC11A-03	91.44 cm (3 ft)
M9202	60.96 cm (2 ft)

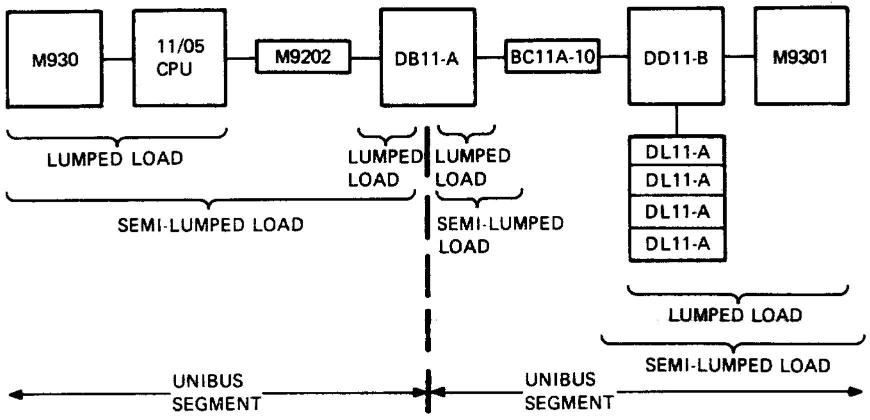


Figure 1-40 Semi-Lumped Load Example

Example 3:

Figure 1-40 shows a system with two Unibus segments having a total of four lumped loads and three semi-lumped loads.

The lumped loads are:

1. M930, 11/45 CPU
2. DB11-A (left side)
3. DB11-A (right side)
4. DD11-B, four DL11s, M9301

The semi-lumped loads are:

1. M930, 11/45 CPU, DB11-A (left side)
2. DB11-A (right side)
3. DD11-B, four DL11s, M9301

Data-In Transactions—The DATI and DATIP transactions request transfer of data from a slave to a master. Both transactions use the D lines to carry the data. These transactions are always a full word transfer; i.e., the slave places the data on D<15:00>. If the master wants only one byte, it must retrieve the data from the proper lines. For a low-order byte use D<07:00>; for a high-order byte use D<15:08>. For byte operations, the master should not assert bit A00 and the slave should ignore this bit.

DATIP Transaction—The DATIP operation is identical to the DATI, except DATIP informs the slave device that the present transfer is the first part of a read/modify/write cycle.

Example:
A pause flag is set in a destructive read-out device (e.g., core memory) that inhibits the restore cycle. The DATIP must be followed by a data-out cycle (DATO or DATO/B) to the same word address.

Since address bit A00 may change between a DATIP and a DATO/B, the slave must check the bus address at the beginning of the DATO/B. The master must retain bus control until this DATO/B is completed; i.e., it must remain bus master (assert BBSY) without interruption from the start of the DATIP cycle to the end of the DATO/B cycle. No other data transfer may be executed between the DATIP and the DATO/B cycles.

In nondestructive readout devices (i.e., flip-flops), the DATI and DATIP are treated identically by the slave.

NOTE

In the case of locations which can be accessed by more than one Unibus or other bus (e.g., the PDP-11/45 semiconductor memory) a DATIP on one bus must prevent the slave from responding on any other bus until the DATO/B cycle has been completed. This is necessary to avoid problems in multiple processor systems.

If a DATIP is followed by a DATO/B, and the device performs a destructive readout, then the device takes the responsibility for restoring the other byte.

Data-Out Transactions—The DATO and DATOB operations transfer data from the master to the slave. A DATO is used to transfer a word to the address specified by A<17:01>. The slave ignores A00 and the master places data on D<15:00>.

A DATOB is used to transfer a byte of data to the address specified by A<17:00>. Line A00=0 indicates the low-order byte. The master places the data on lines D<07:00>. A00=1 indicates the high order byte. The master now places the data on lines D<15:08>.

Parity Error Indicators (PA, PB)—PA and PB are generated by a slave and received by a master. They indicate parity error in a device as follows:

PA	PB	Condition
0	0	No error in a slave in DATI/P
0	1	Error in slave in DATI/P
1	X	Reserved for future expansion

The protocol for PA and PB is the same as that for D<15:00>. PA and PB are not defined in a DATO transaction. PA and PB may be used by the bus master's parity error logic.

Master Sync, (MSYN)—MSYN is a signal issued by a bus master and received by a slave. MSYN has two functions depending on whether it is being asserted or negated.

Assertion—The assertion of MSYN requests the slave, defined by the A lines, to perform the function required by the C lines.

Negation—The negation of MSYN indicates to the slave that the master considers the data transfer concluded.

Slave Sync, (SSYN)—SSYN is a signal issued by a slave and received by a master. SSYN has two functions depending on whether it is being asserted or negated.

NOTE

In an interrupt transaction, the interrupt fielding processor is the slave and the interrupting device is the master.

Assertion—In a master-slave data transfer, the assertion of SSYN informs the bus master that the slave has concluded its part of the data transfer. For a DATI or DATIP, the requested data is placed on the D lines; for a DATO or DATOB, the data on the D lines is accepted.

In an interrupt operation, SSYN is asserted by the interrupt fielding processor. In this case, SSYN signifies that the interrupt vector is accepted by the interrupt fielding processor.

Negation—The negation of SSYN informs all bus devices that the slave has concluded the data transfer. For **data-in** (DATI/P), the negation of SSYN signifies the negation of Master Sync (MSYN) is received and the data removed from the D lines. For **data-out** (DATO/B), the negation of SSYN means that the negation of Master Sync (MSYN) is received. For an interrupt, the negation of SSYN signifies that the negation of the Interrupt Request (INTR) is received by the processor.

Interrupt Request (INTR)
Assertion—INTR is a signal asserted by an interrupting device, after it becomes bus master. The signal informs the interrupt fielding processor that an interrupt is to be performed and that the interrupt vector is present on the D lines. INTR may only be asserted by a device that obtains bus mastership under the authority of BG4, BG5, BG6, or BG7 .

Negation—INTR is negated upon receipt of the assertion of the Slave Sync (SSYN) from the interrupt fielding processor at the end of the transaction.

AC Bus Load—AC loading is the amount of capacitance a module presents to a bus signal line. AC loading is expressed in terms of ac loads. A Unibus segment can support up to 20 ac loads. For more than 20 ac loads, an M9202 (jumper with cable) is used to separate a large load into two Unibus segments.

A number is assigned to each module indicating its ac load. The number is derived using reflection techniques in the time domain. The ac load causes a reflection to occur on the bus line shortly after a signal is asserted. This reflection shows up on an oscilloscope as a spike shortly after the asserting edge of the signal. The ac load rating is usually based on worst case reflections on BBSY, SSYN, and MSYN signal lines.

DC Bus Load—DC bus loading is the amount of leakage current a module presents to a bus signal line. DC loading is expressed in terms of dc loads; one dc load is approximately 105 μA. This represents 80 μA for the receiver plus 25 μA for the transmitter (driver). One dc bus load can also be thought of as one bus driver and one bus receiver (or one transceiver).

Most modules present 1 dc bus load to the Unibus. A Unibus segment can support a maximum of 20 dc bus loads. This limit is set to maintain a sufficient noise margin. For more than 20 dc bus loads, a Unibus repeater option (DB11-A) or equivalent is used. (For example, a DT03 bus switch also contains a bus repeater.) The DB11-A imposes 1 dc bus load on the first Unibus section, but then it can drive an additional 19 dc bus loads and 50 feet of cable. DC loading is more important on SSYN than on data lines.

Unibus Configuration Rules
Five rules are listed below for quick reference. A more detailed description and suggestions are presented in the paragraphs that follow these rules.

1. Maximum cable length of a Unibus segment should not exceed 15.24 m (50 ft).
2. Maximum dc loading should not exceed 20.

3. Maximum ac loading should not exceed 20 for any lumped load (unless the entire segment consists of one lumped load).
4. Different cable lengths should be selected when:
   a. a lumped load requires cables longer than 2.59 m (8.5 ft) for its bus-in and bus-out connections.
   b. the sum of the ac loads in the two lumped loads connected by the cables exceeds 18.
   c. the sum of the ac loads in the two semi-lumped loads connected by the cables exceeds 36.

   The length of cables selected should differ by 1.52 m (5 ft) or more with the longer cable connected to the greater ac load (if possible); or the longer cable should be connected to the semi-lumped load with the greater ac load if possible without increasing the total cable length more than 1.52 m (5 ft).
5. Voltage margin tests should be completed for system acceptance:
   a. when the system is originally configured, and
   b. when any Unibus element is added, deleted, or swapped (including a defective module or backplane).

Maximum Cable Length (Rule 1)—The maximum Unibus cable length should not exceed 15.24 m (50 ft) for a Unibus segment. In calculating the total cable length, add the length of cable from the CPU to the terminator or bus repeater whichever occurs first. Be sure to include:

1.	BC11A-0F	= 115.24 cm (6 in)
2.	M920	= 0 cm
3.	M9202	= 60.96 cm (2 ft)

If the length of cable is longer than specified, the dc drop across the far end of the bus may be excessive and crosstalk may be excessive. Changing the order of bus elements may reduce the length. If that fails, add a DB11-A bus repeater.

Maximum DC Loading (Rule 2)—The maximum dc bus loading for a Unibus segment should not exceed 20 dc loads. The dc load for each option can be found in the *PDP-11 Peripherals Handbook*. Most modules have 1 dc load (1 driver and 1 receiver).

If too many dc loads are placed on a Unibus segment, the voltage on the undriven bus line (normally high) may be lowered. Reflections from lumped loads may turn on bus receivers, and the noise margin may become too small. If more than 20 dc loads are used, divide the bus into smaller segments by adding a DB11-A bus repeater.

Maximum AC Loading (Rule 3)—The maximum ac loading for any lumped load should not exceed 20 ac loads. The exception to this rule is a multiple bus segment containing no Unibus cables.

If a lumped load is greater than 20 ac loads, it may generate a reflection on the Unibus to create a false logic signal. This reflection, shown in Figure 1-41, is greater than the threshold to turn on the receiver. This false signal creates an error. To compensate for this reflection, install an M9202 (jumper with cable) splitting the load into two equal loads. The implementation of this rule is shown in Figure 1-42. The effect of the M9202 causes the peak reflections to occur at slightly different times as one half the load must travel through an extra 0.6 m (2 ft) of cable. The reduced peak reflections do not reach the threshold of the receiver, as shown in Figure 1-43.

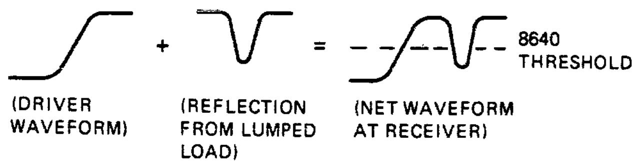
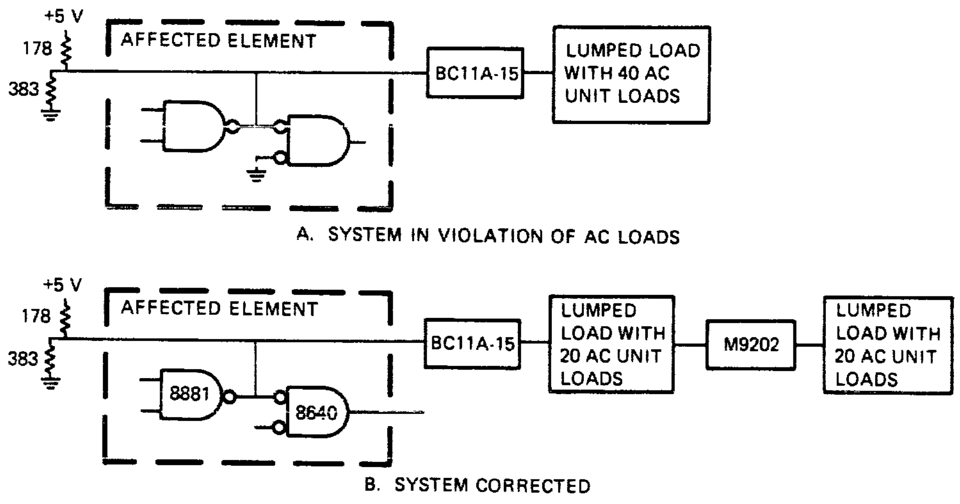


Figure 1-41 Reflection from Too Many AC Loads



+5 V
178
383
AFFECTED ELEMENT
BC11A-15
LUMPED LOAD WITH 40 AC UNIT LOADS
A. SYSTEM IN VIOLATION OF AC LOADS
+5 V
178
383
AFFECTED ELEMENT
8881
8640
BC11A-15
LUMPED LOAD WITH 20 AC UNIT LOADS
M9202
LUMPED LOAD WITH 20 AC UNIT LOADS
B. SYSTEM CORRECTED

Figure 1-42 Compensating for Too Many AC Loads (Implementing Rule 3)

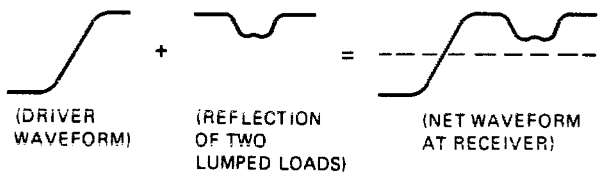


Figure 1-43 Effect of Separating Reflections from AC Loads

NOTE

An M9202 has 0.6 m (2 ft) of folded Unibus cable. Use the M9202, in place of an M920 jumper in the circuit, to reduce the effect of too great an ac load. If an M9202 is unavailable, use a BC11A-02.

Multiple Bus Segment—The exception to rule 3 is a multiple bus segment, as shown in Figure 1-44. This example consists of one lumped load, and it obeys all configuration rules.

1. The length of Unibus cable is zero.
2. There are no more than 20 dc loads.
3. The number of ac loads is irrelevant. There is no 120-ohm cable in the segment on which reflections can travel. None of the M920 jumpers have to be replaced by M9202s.

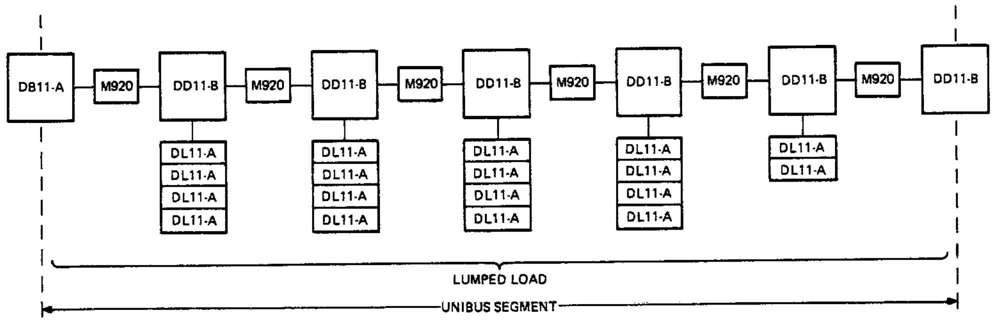


Figure 1-44 Multiple Bus Segment with No Unibus Cable

Different Cable Lengths (Rule 4)—Select cables with different lengths when one of the following conditions exist.

1. A lumped load has cables longer than 2.6 m (8.5 ft) at its bus-in and bus-out connections.
2. The sum of the ac loads for two lumped loads connected by cables exceeds 18.
3. The sum of the ac loads for two semi-lumped loads connected by cables exceeds 36.

Consider the example in Figure 1-45. The sum of the lumped loads connected to the opposite ends of the Unibus cable exceeds 18. (The total ac loads equals 27.) Signal reflections from the ends of the bus-in and bus-out cables arrive at the affected lumped load simultaneously and superimpose. The net reflection, shown in Figure 1-46, crosses the threshold of the receiver, turning it on erroneously.

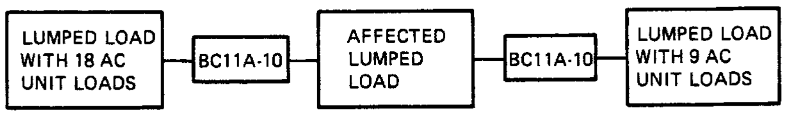
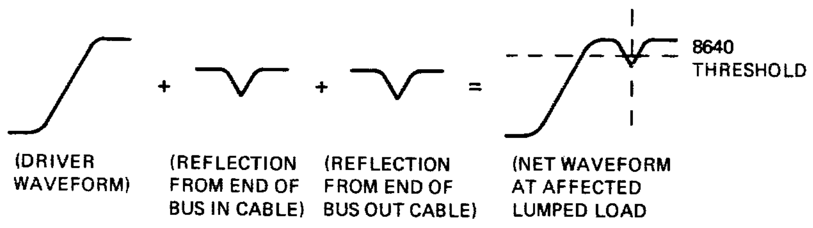

LUMPED LOAD WITH 18 AC UNIT LOADS
BC11A-10
AFFECTED LUMPED LOAD
BC11A-10
LUMPED LOAD WITH 9 AC UNIT LOADS

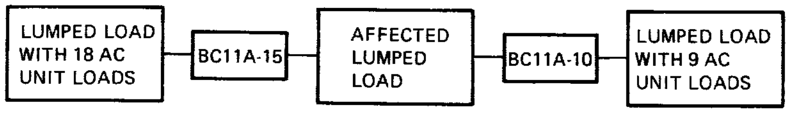
Figure 1-45 Example of Bus Segment Requiring Different Cable Lengths



+
+
=
8640
THRESHOLD
(DRIVER WAVEFORM)
(REFLECTION FROM END OF BUS IN CABLE)
(REFLECTION FROM END OF BUS OUT CABLE)
(NET WAVEFORM AT AFFECTED LUMPED LOAD

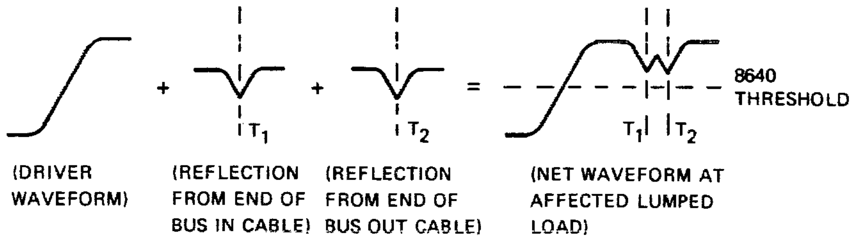
Figure 1-46 Net Reflections with Identical Cable Lengths

Rule 4 may be implemented to compensate for the net reflection. Select cables with lengths that differ by at least 1.52 m (5 ft). Connect the cable with the longer length to the greater ac load, as shown in Figure 1-47. The reflections arrive at different times and do not cross the threshold of the receiver, as shown in Figure 1-48.



LUMPED LOAD WITH 18 AC UNIT LOADS
BC11A-15
AFFECTED LUMPED LOAD
BC11A-10
LUMPED LOAD WITH 9 AC UNIT LOADS

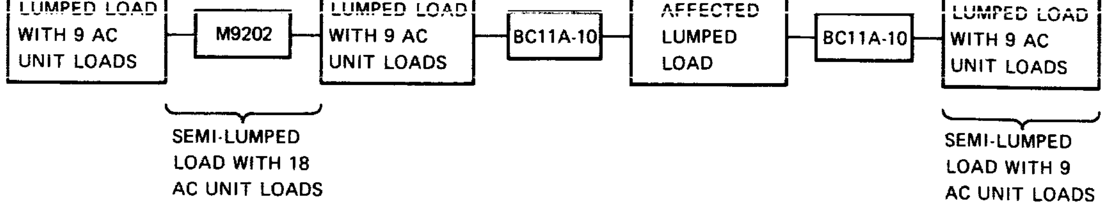
Figure 1-47 Selecting Different Cable Lengths



+
+
=
T_1
T_2
T_1 T_2
8640
THRESHOLD
(DRIVER WAVEFORM)
(REFLECTION FROM END OF BUS IN CABLE)
(REFLECTION FROM END OF BUS OUT CABLE)
(NET WAVEFORM AT AFFECTED LUMPED LOAD)

Figure 1-48 Net Reflections with Different Cable Lengths

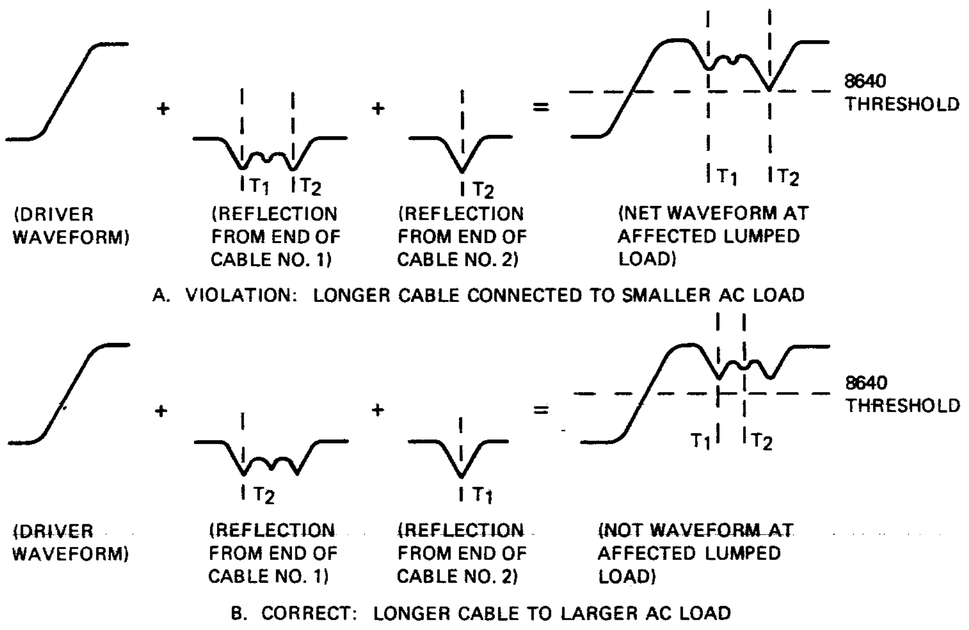
Another method to eliminate the problem is to reduce the ac loads connected by the cables. Split the greater ac load with an M9202 making two smaller loads, as shown in Figure 1-49. The M9202 has a 0.6 m cable to separate the larger load. The total ac loads seen by the affected lumped load does not exceed 18.



LUMPED LOAD WITH 9 AC UNIT LOADS
M9202
LUMPED LOAD WITH 9 AC UNIT LOADS
BC11A-10
AFFECTED LUMPED LOAD
BC11A-10
LUMPED LOAD WITH 9 AC UNIT LOADS
SEMI-LUMPED LOAD WITH 18 AC UNIT LOADS
SEMI-LUMPED LOAD WITH 9 AC UNIT LOADS

Figure 1-49 Alternate Solution: Reducing AC Load

Supplement to Rule 4—When selecting cables of different lengths, attach the cable with the greater length to the larger ac load. The reflections superimpose on each other, but not as much. This is shown in Figure 1-50. When the longer cable is connected to the smaller ac load, the reflection from this cable is greater and the net reflection crosses the threshold of the receiver. Connect the longer cable to the larger ac load to reduce the net reflection.



8640 THRESHOLD
T1 T2
T2
T1 T2
(DRIVER WAVEFORM)
(REFLECTION FROM END OF CABLE NO. 1)
(REFLECTION FROM END OF CABLE NO. 2)
(NET WAVEFORM AT AFFECTED LUMPED LOAD)
A. VIOLATION: LONGER CABLE CONNECTED TO SMALLER AC LOAD
8640 THRESHOLD
T1 T2
T2
T1
(DRIVER WAVEFORM)
(REFLECTION FROM END OF CABLE NO. 1)
(REFLECTION FROM END OF CABLE NO. 2)
(NOT WAVEFORM AT AFFECTED LUMPED LOAD)
B. CORRECT: LONGER CABLE TO LARGER AC LOAD

Figure 1-50 Effect of AC Load on Longer Cable

NOTE

Common sense should be exercised when implementing rules 1 through 4. On rare occasions, it may not be practical to implement all the rules. For example: Suppose the Unibus on a system exceeds the maximum 15.24 m (50 ft) by 1.5 m (5 ft). Implementing rule 1 requires another DB11-A repeater. This may require another BA11 expander box, which may require another cabinet. Common sense must be exercised if any rule is violated. In all cases perform the voltage margin tests.

Voltage Margin Tests (Rule 5)—Voltage margin tests should be performed when the system is originally configured and when any Unibus element is added or changed in the system. On rare occasions, implementing rules 1-4 may not eliminate all reflection problems. On these occasions call the local DIGITAL Field Service office for assistance in performing margin tests.

UNIBUS HARDWARE

Unibus Cable (BC11A)

The Unibus cable, shown in Figure 1-51, is constructed of two parallel 60-conductor flat mylar ribbon cables separated by foam. An M929 bus connector is connected to one end, and an M919 external bus connector is connected to the other end. The cable contains 56 signal lines and 64 ground lines. The connector cards have 56 gold plated fingers devoted to signals and 14 devoted to ground. The BC11A cables are used to connect system units that are not adjacent.

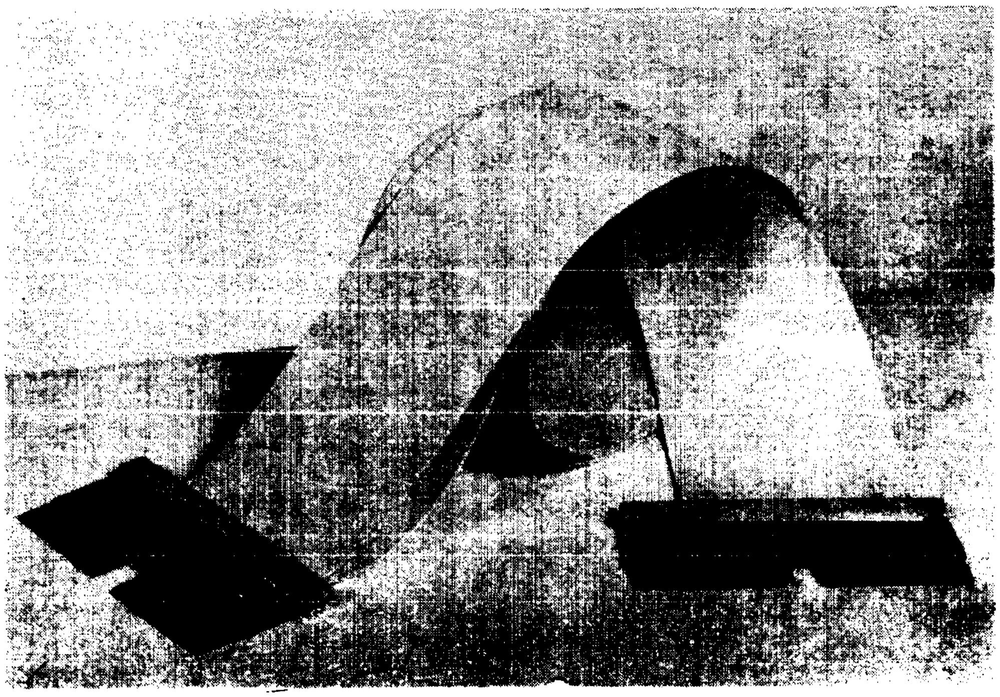


Figure 1-51 Unibus Cable (BC11A)

The BC11A cable comes in the following standard lengths.

Cable Part Number	Meters	Feet
BC11A-0F	0.15	0.5
BC11A-02	0.61	2.0
BC11A-03	0.92	3.0
BC11A-05	1.52	5.0
BC11A-06	1.82	6.0
BC11A-8F	2.59	8.5
BC11A-10	3.04	10.0
BC11A-15	4.57	15.0
BC11A-20	6.07	20.0
BC11A-25	8.60	25.0

NOTE

The M9202 Unibus jumper also has 0.61 m (2 ft) of Unibus cable.

M920 Unibus Jumper Module

The M920, shown in Figure 1-52, is a double module connected by a short 60-conductor cable. The module connects one Unibus system unit to the next. The printed circuit cards are held rigidly 2.54 cm (1 in) apart by a handle. A single M920 carries all 56 Unibus signals and 14 grounds.

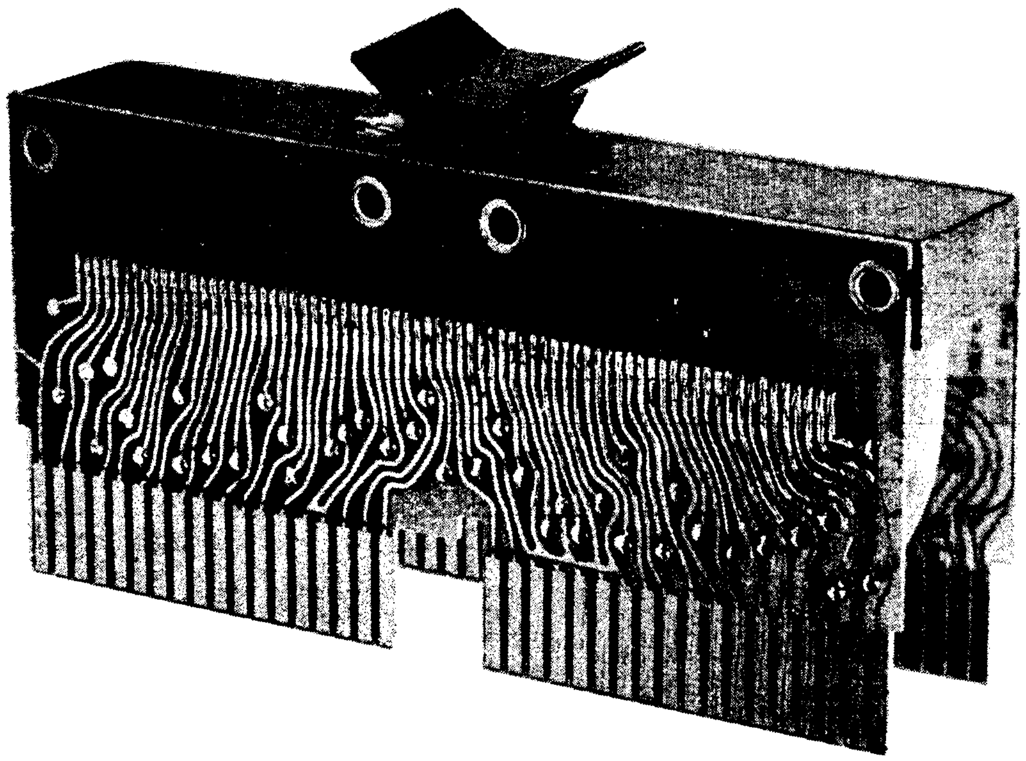


Figure 1-52 M920 Unibus Jumper Module

M9202 Unibus Jumper with Cable

The M9202, shown in Figure 1-53, is a double module connected by 0.61 m (2 ft) of 60-conductor cable. The modules are held rigidly 2.54 cm (1 in) apart by spacers. The flexible ribbon cable is folded between the modules. The M9202 is used in place of M920 to distribute ac loads and separate Unibus segments. The M9202 has the same effect as a BC11A-02.

M930 Unibus Terminator Module

The M930, shown in Figure 1-54, is a short, double-size module that terminates all signal lines on the Unibus. This module provides the signals with termination to prevent reflections. The module plugs into a bus-out slot, the slot normally used for adding another Unibus segment.

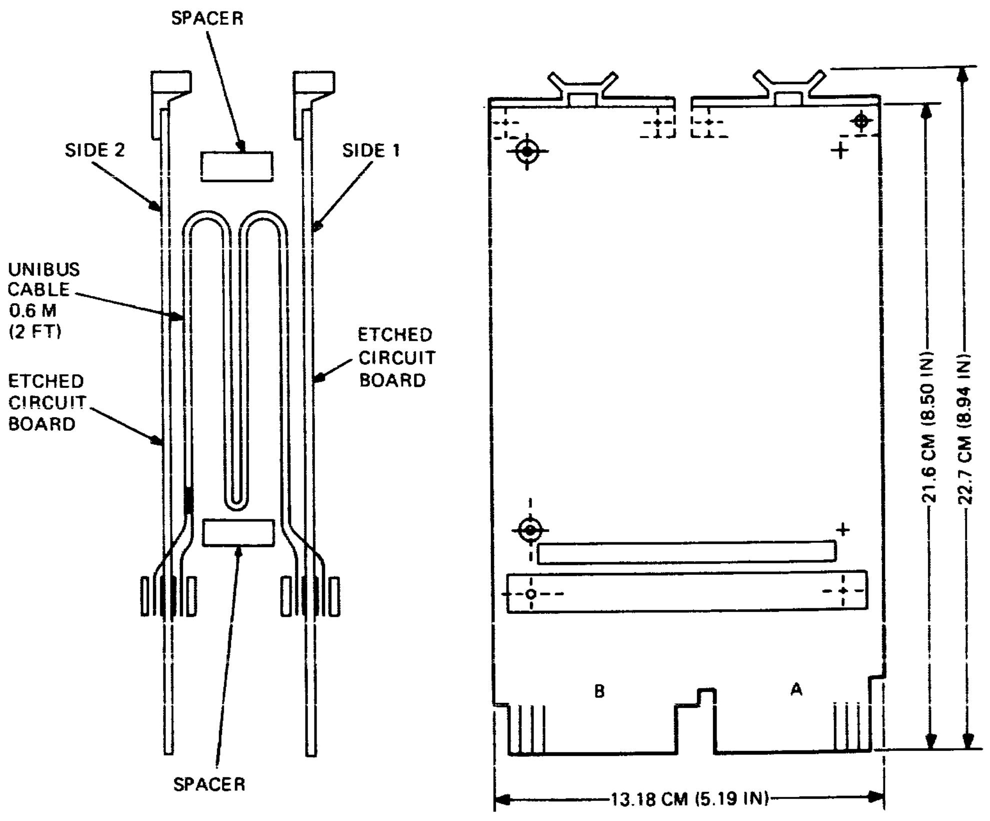


Figure 1-53 M9202 Unibus Jumper with Cable

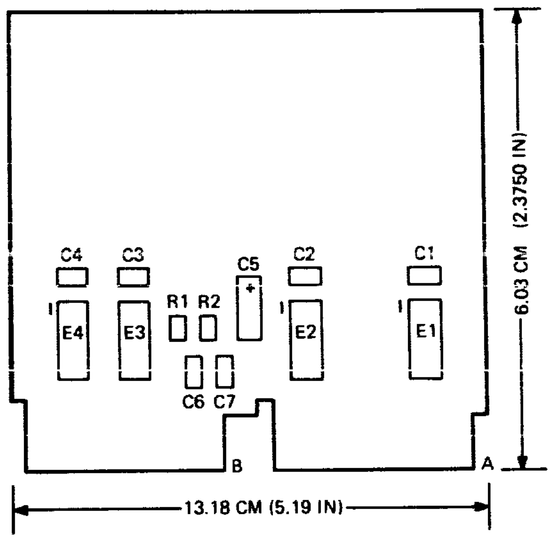


Figure 1-54 M930 Unibus Terminator Module

The module requires 1.25 A at 5 V ± 5%. All pins have a resistive divider termination of 178 ohms to +5 V and 383 ohms to ground, except those listed in Table 1-5.

Table 1-5
M930 Unibus Terminator Pin Assignment

383 ohms in parallel with 0.001 μf to +5 V (for AC LO, DC LO)	178 ohms to +5 V (for grant lines)	Ground Pins		+5 V Input Pins
BF1	AV1	AB2	BB2	AA2
BF2	AU1	AC2	BC2	BA2
	BA1	AN1	BD1	
	BB1	AP1	BE1	
	BE2	AR1	BT1	
		AS1	BV2	
		AT1		
		AV2		

NOTE

All other pins have a resistor divider termination of 178 ohms to +5 V and 383 ohms to GND.

M981 Internal Unibus Terminator Assembly

The M891, shown in Figure 1-55, is a double module connected by a short 60-conductor cable. The M981 is effectively an M930 configured on a M920 jumper module for internal termination of the Unibus. The printed circuit cards are held rigidly 2.54 cm (1 in) apart by a handle. Voltage requirements and pin connections are the same as an M930.

NOTE

Other terminator modules are available with bootstrap programs, diagnostic programs, and ROM options. Contact the local DIGITAL sales office for details.

Drivers, Receivers, and Transceivers

The recommended drivers, receivers, and transceivers used to interface with the Unibus cable are listed below.

Function	Type IC	Description	DEC Part No.	DEC Option No.
Receiver	8640	Quad NOR gate	19-11469	DEC 00956
Transceiver	8641	Quad transceiver (transmitter/ receiver)	19-11579	DEC 00964
Driver (Transmitter)	8881	Quad NAND gate	19-09705	DEC 00957

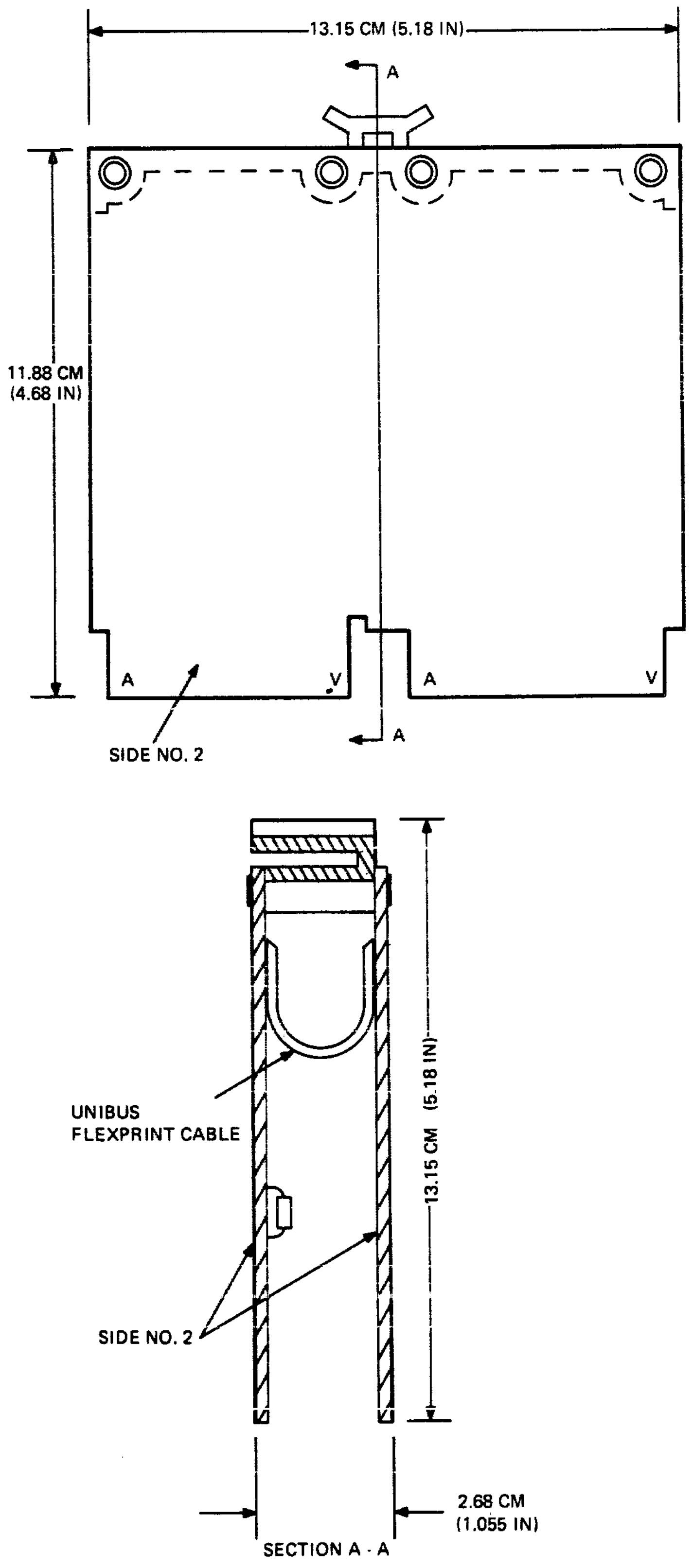


Figure 1-55 M981 Internal Unibus Terminator Module

Customers may purchase these IC chips from DIGITAL using the DEC option number. Each option number defines a package containing ten (10) IC chips of the type described.

Unibus Connector Block Pin Assignments

The Unibus cable normally plugs into rows A and B of a slot in a backplane. The Unibus signals for the backplane are listed in Tables 1-6 and 1-7. Table 1-6 lists the pin assignments by pin number; Table 1-7 lists the pin assignments by signal name.

Table 1-6
Unibus Pin Assignments (By Pin Numbers)

Pin	Signal	Pin	Signal
AA1	INIT L	BA1	BG6 H
AA2	POWER (+5 V)	BA2	POWER (+5 V)
AB1	INTR L	BB1	BG5 H
AB2	GROUND	BB2	GROUND
AC1	D00 L	BC1	BR5 L
AC2	GROUND	BC2	GROUND
AD1	D02 L	BD1	GROUND
AD2	D01 L	BD2	BR4 L
AE1	D04 L	BE1	GROUND
AE2	D03L	BE2	BG4 H
AF1	D06 L	BF1	ACLO L
AF2	D05 L	BF2	DCLO L
AH1	D08 L	BH1	A01 L
AH2	D07 L	BH2	A00 L
AJ1	D10 L	BJ1	A03 L
AJ2	D09 L	BJ2	A02 L
AK1	D12 L	BK1	A05 L
AK2	D11 L	BK2	A04 L
AL1	D14 L	BL1	A07 L
AL2	D13 L	BL2	A06 L
AM1	PA L	BM1	A09 L
AM2	D15 L	BM2	A08 L
AN1	GROUND	BN1	A11 L
AN2	PB L	BN2	A10 L
AP1	GROUND	BP1	A13 L
AP2	BBSY L	BP2	A12 L
AR1	GROUND	BR1	A15 L
AR2	SACK L	BR2	A14 L
AS1	GROUND	BS1	A17 L
AS2	NPR L	BS2	A16 L
AT1	GROUND	BT1	GROUND
AT2	BR7 L	BT2	C1 L
AU1	NPG H	BU1	SSYN L
AU2	BR6 L	BU2	CO L
AV1	BG7 H	BV1	MSYN L
AV2	GROUND	BV2	GROUND

Table 1-7
Unibus Pin Assignments (By Signal Name)

Signal	Pin	Signal	Pin
A00 L	BH2	D06 L	AF1
A01 L	BH1	D07 L	AH2
A02 L	BJ2	D08 L	AH1
A03 L	BJ1	D09 L	AJ2
A04 L	BK2	D10 L	AJ1
A05 L	BK1	D11 L	AK2
A06 L	BL2	D12 L	AK1
A07 L	BL1	D13 L	AL2
A08 L	BM2	D14 L	AL1
A09 L	BM1	D15 L	AM2
A10 L	BN2	GROUND	AB2
A11 L	BN1	GROUND	AC2
A12 L	BP2	GROUND	AN1
A13 L	BP1	GROUND	AP1
A14 L	BR2	GROUND	AR1
A15 L	BR1	GROUND	AS1
A16 L	BS2	GROUND	AT1
A17 L	BS1	GROUND	AV2
ACLO L	BF1	GROUND	BB2
BBSY L	AP2	GROUND	BC2
BG4 H	BE2	GROUND	BD1
BG5 H	BB1	GROUND	BE1
BG6 H	BA1	GROUND	BT1
BG7 H	AV1	GROUND	BV2
BR4 L	BD2	INIT L	AA1
BR5 L	BC1	INTR L	AB1
BR6 L	AU2	MSYN L	BV1
BR7 L	AT2	NPG H	AU1
CO L	BU2	NPR L	AS2
C1 L	BT2	PA L	AM1
DOO L	AC1	PB L	AN2
DO1 L	AD2	+5 V*	AA2
D02 L	AD1	+5 V*	BA2
D03 L	AE2	SACK L	AR2
D04 L	AE1	DCLO L	BF2
D05 L	AF2	SSYN L	BU1

* +5 V is wired to these pins to supply power to the bus terminator only. +5 V should never be connected via the Unibus between system units.

part 2

lsi11 bus specification

PART 2

LSI-11 BUS SPECIFICATION

INTRODUCTION

NOTE

This is not the *complete* LSI-11 bus specification. The next printing of this document will contain greater detail in a number of areas. The goal is to permit users to design and implement *typical* interfaces to the LSI-11 bus.

The processor, memory and I/O devices communicate via 38 bidirectional signal lines that constitute the LSI-11 bus. Addresses, data, and control information are sent along these signal lines, some of which contain time-multiplexed information. The lines are functionally divided as follows:

18 Data/address lines—BDAL<17:00>
6 Data transfer control lines—BBS7, BDIN, BDOUT, BRPLY, BSYNC, BWTBT
3 Direct memory access control lines—BDMG, BDMR, BSACK
6 Interrupt control lines—BEVNT, BIAK, BIRQ4, BIRQ5, BIRQ6, BIRQ7
5 System control lines—BDCOK, BHALT, BINIT, BPOK, BREF

Most LSI-11 bus signals are bidirectional and use terminations for a negated (high) signal level. Devices connect to these lines via high-impedance bus receivers and open collector drivers. The asserted state is produced when a bus driver asserts the line low. Although bidirectional lines are electrically bidirectional (any point along the line can be driven or received), certain lines are functionally unidirectional. These lines communicate to or from a bus master (or signal source), but not both. Interrupt Acknowledge (BIACK) and Direct Memory Access Grant (BDMG) signals are physically unidirectional in a daisy-chain fashion. These signals originate at the processor output signal pins. Each is received on device input pins (BIAKI or BDMGI) and conditionally retransmitted via device output pins (BIAKO or BDMGO). These signals are received from higher priority devices and are retransmitted to lower priority devices along the bus.

Master/Slave Relationship

Communication between devices on the bus is asynchronous. A master/slave relationship exists throughout each bus transaction.

At any time, there is one device that has control of the bus. This controlling device is termed the "bus master." The master device controls the bus when communicating with another device on the bus, termed the *slave*. The *bus master* (typically the processor or a DMA device) initiates a bus transaction. The *slave device* responds by acknowledging the transaction in progress and by receiving data from, or transmitting data to, the bus master. LSI-11 bus control signals transmitted or received by the bus master or bus slave device must complete the sequence according to bus protocol.

The processor controls bus arbitration (i.e., who becomes bus master at any given time). A typical example of this relationship is the processor, as master, fetching an instruction from memory (which is always a slave). Another example is a disk, as master, transferring data to memory as slave. Theoretically, any device can be master or slave depending on the circumstances. Communication on the LSI-11 bus is interlocked so that for each control signal issued by the master device, there must be a response from the slave in order to complete the transfer. It is the master/slave signal protocol that makes the LSI-11 bus asynchronous. The asynchronous operation precludes the need for synchronizing with clock pulses.

Since bus cycle completion by the bus master requires response from the slave device, each bus master must include a timeout error circuit that will abort the bus cycle if the slave device does not respond to the bus transaction within 10 microseconds.

The actual time before a timeout error occurs must be longer than the reply time of the slowest peripheral or memory device on the bus.

The signals and pin assignments are tabulated in Table 2-1. The pin nomenclature is for reference and is only required when examining DIGITAL modules and circuit schematics. Table 2-2 contains a functional description of the LSI-11 bus pins and signals.

Table 2-1

Categories of LSI-11 Bus Signal Lines

Number of Pins	**Functional Category**	**DIGITAL's Nomenclature** (Name) (Pin)					
18	Data/Address	BDAL0, AU2	BDAL1, AV2	BDAL2 → BE2 →	BDAL15, BV2	BDAL16, AC1	BDAL17 AD1
6	Data Control	BDOUT, AE2	BRPLY, AF2	BDIN, AH2	BSYNC, AJ2	BWTBT, AK2	BBS7 AP2
6	Interrupt Control	BIRQ7, BP1	BIRQ6, AB1	BIRQ5, AA1	BIRQ4, AL2	BIAKO, AN2	BIAKI AM2
4	DMA Control	BDMR, AN1	BDMGO, AS2	BDMGI, AR2	BSACK BN1		
6	System Control	BHALT, AP1	BREF, AR1	BDCOK, BA1	BPOK, BB1	BEVNT, BR1	BINIT AT2
3	+5 Vdc	AA2, BA2, BV1					
2	+12 Vdc	AD2, BD2					
2	−12 Vdc	AB2, BB2					
2	+12B (battery)	AS1, BS1					
1	+5B (battery)	AV1, (AE1, AS1 alternates)					
8	GND	AC2, AJ1, AM1, AT1, BC2, BJ1, BM1, BT1					
8	S SPARES	AE1, AF1, AH1, BC1, BD1, BE1, BF1, BH1					
4	M SPARES	AK1 — AL1, BK1 — BL1 (pairs connected)					
2	P SPARES	AU1, BU1					
72							

Table 2-2

Functional Descriptions

Bus Pin	Signal Mnemonic	Signal Function
AA1	BIRQ5 L	Interrupt Request priority level 5
AB1	BIRQ6 L	Interrupt Request priority level 6
AC1	BDAL16 L	Address line 16 during addressing protocol; memory error line during data transfer protocol.
AD1	BDAL17 L	Address line 17 during addressing protocol; memory error enable during data transfer protocol.
AE1	SSPARE1 (alternate +5B)	Special spare—not assigned or bused in DIGITAL cable or backplane assemblies; available for user connection. Optionally, this pin may be used for +5 V battery (+5B) backup power to keep critical circuits alive during power failures. A jumper is required on LSI-11 bus options to open (disconnect) the +5B circuit in systems that use this line as SSPARE1.
AF1	SSPARE2	Special spare—not assigned or bused in DIGITAL cable or backplane assemblies; available for user interconnection. In the highest priority device slot, the processor may use this pin for a signal to indicate its RUN state.
AH1	SSPARE3 SRUN	Special spare—not assigned nor bused in DIGITAL cable or backplane assemblies; available for user interconnection.
AJ1	GND	Ground—System signal ground and dc return.
AK1 AL1	MSPAREA MSPAREB	Maintenance Spare—Normally connected together on the backplane at each option location (not bused connection).
AM1	GND	Ground—System signal ground and dc return.
AN1	BDMRL	Direct Memory Access (DMA) Request—device asserts this signal to request bus mastership. The processor arbitrates bus mastership between itself and all DMA devices on the bus. If the processor is

Table 2-2 Functional Descriptions (Cont.)

Bus Pin	Signal Mnemonic	Signal Function
		not bus master (it has completed a bus cycle and BSYNC L is not being asserted by the processor), it grants bus mastership to the requesting device by asserting BDMGO L. The devices responds by negating BDMR L and asserting BSACK L.
AP1	BHALT L	Processor Halt—When BHALT L is asserted, the processor responds by going into console ODT mode.
AR1	BREF L	Memory refresh—used to refresh dynamic memory devices. The LSI-11 processor microcode features automatic refresh control. BREF L is asserted during this time to override memory bank selection decoding. Interrupt requests and BBS7 are blocked out during this time.
AS1	+5B or +12B (battery)	+12 or +5 Vdc battery backup power to keep critical circuits alive during power failures. This signal is not bused to BS1 in all DIGITAL backplanes. A jumper is required on all LSI-11 bus options to open (disconnect) the backup circuit from the bus in systems that use this line at the alternate voltage.
AT1	GND	Ground—System signal ground and dc return.
AU1	PSPARE1	Power spare 1 (not assigned a function; not recommended for use). If a module is using −12 V (on pin BB2) and if the module is accidentally inserted backward in all the backplane, −12 Vdc appears on pin AU1.
AV1	+5B	+5 V Battery Power—Secondary +5 V power connection. Battery power can be used with certain devices.
BA1	BDCOK H	DC Power OK—Power supply-generated signal that is asserted when there is sufficient dc voltage available to sustain reliable system operation.

Table 2-2 Functional Descriptions (Cont.)

Bus Pin	Signal Mnemonic	Signal Function
BB1	BPOK H	AC Power OK—Asserted by the power supply when primary power is normal. When negated during processor operation, a power fail trap sequence is initiated.
BC1 BD1 BE1 BF1 BH1	SSPARE 4 SSPARE 5 SSPARE 6 SSPARE 7 SSPARE 8	Special spares—not assigned or bused in DIGITAL cable and backplane assemblies; available for user interconnections. Caution. These pins may be used as test points by Digital in some options.
BJ1	GND	Ground—System signal ground and dc return.
BK1 BL1	MSPAREB MSPAREB	Maintenance Spares—Normally connected together on the backplane at each option location (not a bused connection).
BM1	GND	Ground—System signal ground and dc return.
BN1	BSACK L	This signal is asserted by a DMA device in response to the processor's BDMGO L signal, indicating that the DMA device is bus master.
BP1	BIRQ7 L	Interrupt request priority level 7
BR1	BEVNT L	External Event Interrupt Request—When the processor latches the leading edge and arbitrates as a level 6 interrupt. A typical use of this signal is a line time clock interrupt.
BS1	+12B	+12 Vdc battery backup power (not bussed to AS1 in all DIGITAL backplanes)
BT1	GND	Ground—System signal ground and dc return.
BU1	PSPARE2	Power spare 2 (not assigned a function, not recommended for use). If a module is using −12 V (on pin AB2) and if the module is accidentally inserted backwards in the backplane, −12 Vdc appears on pin BU1.
BV1	+5	+5 V power—Normal +5 Vdc system power

Table 2-2 Functional Descriptions (Cont.)

Bus Pin	Signal Mnemonic	Signal Function
AA2	+5	+5 V power—Normal +5 Vdc system power
AB2	−12	−12 V Power—−12 Vdc (optional) power for devices requiring this voltage.
AC2	GND	Ground—System signal ground and dc return
AD2	+12	+12 V Power—+12 Vdc system power
AE2	BDOUT L	Data Output—BDOUT, when asserted, implies that valid data is available on BDAL <0:15> L and that an output transfer, with respect to the bus master device, is taking place. BDOUT L is deskewed with respect to data on the bus. The slave device responding to the BDOUT L signal must assert BRPLY L to complete the transfer.
AF2	BRPLY L	Reply—BRPLY L is asserted in response to BDIN L or BDOUT L and during IAK transaction. It is generated by a slave device to indicate that it has placed its data on the BDAL bus or that it has accepted output data from the bus.
AH2	BDIN L	Data Input—BDIN L is used for two types of bus operation: 1. When asserted during BSYNC L time, BDIN L implies an input transfer with respect to the current bus master, and requires a response (BRPLY L). BDIN L is asserted when the master device is ready to accept data from a slave device. 2. When asserted without BSYNC L, it indicates that an interrupt operation is occurring. The master device must deskew input data from BRPLY L.
AJ2	BSYNC L	Synchronize—BSYNC L is asserted by the bus master device to indicate that it has placed an address on the bus. The

Table 2-2 Functional Descriptions (Cont.)

Bus Pin	Signal Mnemonic	Signal Function
		transfer is in process until BSYNC L is negated.
AK2	BWTBT L	Write/Byte—BWTBT L is used in two ways to control a bus cycle: 1. It is asserted during the leading edge of BSYNC L to indicate that an output sequence is to follow (DATO or DATOB), rather than an input sequence. 2. It is asserted during BDOUT L, in a DATOB bus cycle, for byte addressing.
AL2	BIRQ4 L	Interrupt request priority level 4
AM2 AN2	BIAKI L BIAKO L	Interrupt acknowledge—In accordance with interrupt protocol, the processor asserts BIAKO L to acknowledge the honoring of an interrupt. The bus transmits this to BIAKI L of the next priority device (electrically closest to the processor). This device accepts the Interrupt Acknowledge under two conditions: 1. The device requested the bus by asserting an interrupt, and 2. The device has the highest priority interrupt request on the bus at that time. If these conditions are not met, the device asserts BIAKO L to the next device on the bus. This process continues in a daisy-chain fashion until the device with the highest interrupt priority receives the Interrupt Acknowledge (IAK) signal.
AP2	BBS7 L	Bank 7 select—The bus master asserts this signal to reference the I/O page (including that portion of the I/O page reserved for nonexistent memory). The address in BDAL<0:12> L when BBS7 L is asserted is the address within the I/O page.

Table 2-2 Functional Descriptions (Cont.)

Bus Pin	Signal Mnemonic	Signal Function
AR2 AS2	BDMGI L BDMGO L	Direct memory access grant—The bus arbitrator asserts this signal to grant bus mastership to a requesting device, according to bus mastership protocol. The signal is passed in a daisy-chain from the arbitrator (as BDMGO L) through the bus to BDMGI L of the next priority device (electrically closest device on the bus). This device accepts the grant only if it requested to be bus master (by a BDMR L). If not, the device passes the grant (asserts BDMGO L) to the next device on the bus. This process continues until the requesting device acknowledges the grant.
AT2	BINIT L	Initialize—This signal is used for system reset. All devices on the bus are to return to a known, initial state; i.e., registers are reset to zero, and logic is reset to state 0. Exceptions should be completely documented in programming and engineering specifications for the device.
AU2	BDAL0 L	Data/address line 00.
AV2	BDAL1 L	Data/address line 01.
BA2	+5	+5 Vdc power.
BB2	−12	−12 Vdc power (optional, not required for DIGITAL LSI-11 hardware options).
BC2	GND	Power supply return.
BD2	+12	+12 Vdc power.
BE2	BDAL2 L	Data/address line 02.
BF2	BDAL3 L	Data/address line 03.
BH2	BDAL4 L	Data/address line 04.
BJ2	BDAL5 L	Data/address line 05.
BK2	BDAL6 L	Data/address line 06.
BL2	BDAL7 L	Data/address line 07.
BM2	BDAL8 L	Data/address line 08.
BN2	BDAL9 L	Data/address line 09.
BP2	BDAL10 L	Data/address line 10.

Table 2-2 Functional Descriptions (Cont.)

Bus Pin	Signal Mnemonic	Signal Function
BR2	BDAL11 L	Data/address line 11.
BS2	BDAL12 L	Data/address line 12.
BT2	BDAL13 L	Data/address line 13.
BU2	BDAL14 L	Data/address line 14.
BV2	BDAL15 L	Data/address line 15.

DATA TRANSFER BUS CYCLES

Data transfer bus cycles are as follows:

Bus Cycle Mnemonic	Description	Function (with respect to the bus master)
DATI	Data word input	Read
DATO	Data word output	Write
DATOB	Data byte output	Write byte
DATIO	Data word input/output	Read-modify-write
DATIOB	Data byte input/byte output	Read-modify-write byte

These bus cycles, executed by bus master devices, transfer 16-bit words or 8-bit bytes to or from slave devices. The following bus signals are used in a data transfer operation:

Mnemonic	Description	Function
BDAL<17:00> L	18 Data/address lines	BDAL<15:00> L are used for word and byte transfers. BDAL<17:16> L are used for extended addressing, memory parity error, and memory parity error enable functions.
BSYNC L	Synchronize	Strobe signals
BDIN L	Data input strobe	
BDOUT L	Data output strobe	
BRPLY L	Reply	
BWTBT L	Write/byte control	Control signals
BBS7 L	Bank select 7	

Data transfer bus cycles can be reduced to three basic types; DATI, DATO(B) and DATIO(B). These transactions occur between the bus master and one slave device selected during the addressing portion fo the bus cycle.

Bus Cycle Protocol
Before initiating a bus cycle, the previous bus transaction must have been completed (BSYNC L negated) and the device must become bus master. The bus cycle can be divided into two parts, an addressing portion and a data transfer portion. During the addressing portion, the bus master outputs the address for the desired slave device (memory or device register). The selected slave device responds by latching the address bits and holding this condition for the duration of the bus cycle (until BSYNC L becomes negated). During the data transfer portion, the actual data transfer occurs.

Device Addressing—The device addressing portion of a data transfer bus cycle comprises an address setup and deskew time and an address hold/deskew time. During the address setup and deskew time the bus master does the following:

1. Asserts BDAL<17:00> L with the desired slave device address bits,
2. Asserts BBS7 L if a device in the I/O page is being addressed, and
3. Asserts BWTBT L if the cycle is a DATO(B) bus cycle.
4. Asserts BSYNC at least 150 ns after BDAL<17:00> L, BBS7 L, and BWTBT L are valid.

During this time the address, BBS7 L, and BWTBT L signals are asserted at the slave bus receiver for at least 75 ns before BSYNC goes active. Devices in the I/O page ignore the five high-order address bits BDAL<17:13> and instead decode BBS7 along with the thirteen low-order address bits. An active BWTBT L signal indicates that a DATO(B) operation follows, while an inactive BWTBT L indicates a DATI or DATIO(B) operation.

The address hold/deskew time begins after BSYNC L is asserted. The master must hold the address at BDAL at least 100 ns after the assertion of BSYNC.

The slave device uses the active BSYNC L bus receiver output to clock BDAL address bits, BBS7 L and BWTBT L, into its internal logic. BDAL<17:00>L, BBS7 L, and BWTBT L will remain active for 25 ns (minimum) after the BSYNC L bus receiver goes active. BSYNC L remains active for the duration of the bus cycle.

Device selected logic must be reset at the end of the current bus cycle. The device should not wait until the next BSYNC L signal to reset the device selected logic.

Memory and peripheral devices are addressed similarly except for the way the slave device responds to BBS7. Addressed peripheral devices must not decode address bits on BDAL<17:13> L. Addressed peripheral devices may respond to a bus cycle only when BBS7 L is asserted (low) during the addressing portion of the cycle. When asserted, BBS7 L indicates that the device address resides in the I/O page (the upper 4K address space). Memory devices generally do not respond to addresses in the I/O page; however, some system applications may permit memory to reside in the I/O page for use as DMA buffers, read-only-memory bootstraps or diagnostics, etc.

DATI (Figures 2-1 and 2-2)—The DATI bus cycle is a read operation. During DATI data is input to the bus master. Data consists of 16-bit word transfers over the bus. During the data transfer portion of the DATI bus cycle the bus master asserts BDIN L 100 ns minimum, 8 ns maximum, after BSYNC L is asserted. The slave device responds to BDIN L active in the following ways:

1. Asserts BRPLY L after receiving BDIN L and 125 ns (maximum) before BDAL bus driver data bits are valid
2. Asserts BDAL<17:00> L with the addressed data and error information

When the bus master receives BRPLY L, it does the following:

1. Waits at least 200 ns deskew time and then accepts input data at BDAL<17:00> L bus receivers. BDAL<17:16> L are used for transmitting parity errors to the master.
2. Negates BDIN L 150 ns (minimum) to 2 microseconds (maximum) after BRPLY L goes active.

The slave device responds to BDIN L negation by negating BRPLY L and removing read data from BDAL bus drivers. BRPLY L must be negated 100 ns (maximum) prior to removal of read data. The bus master responds to the negated BRPLY L by negating BSYNC L.

Conditions for the next BSYNC L assertion are as follows:

1. BSYNC L must remain negated for 200 ns (minimum)
2. BSYNC L must not become asserted within 300 ns of previous BRPLY L negation.

NOTE

Continuous assertion of BSYNC L retains control of the bus by the bus master, and the previously addressed slave device remains selected. This is done for DATIO(B) bus cycles where DATO or DATOB follows a DATI without BSYNC L negation and a second device addressing operation. Also, a slow slave device can hold off data transfers to itself by keeping BRPLY L asserted, which will cause the master to keep BSYNC L asserted. Exceeding 15 μsec hold time will cause loss of memory if bus refresh is being used.

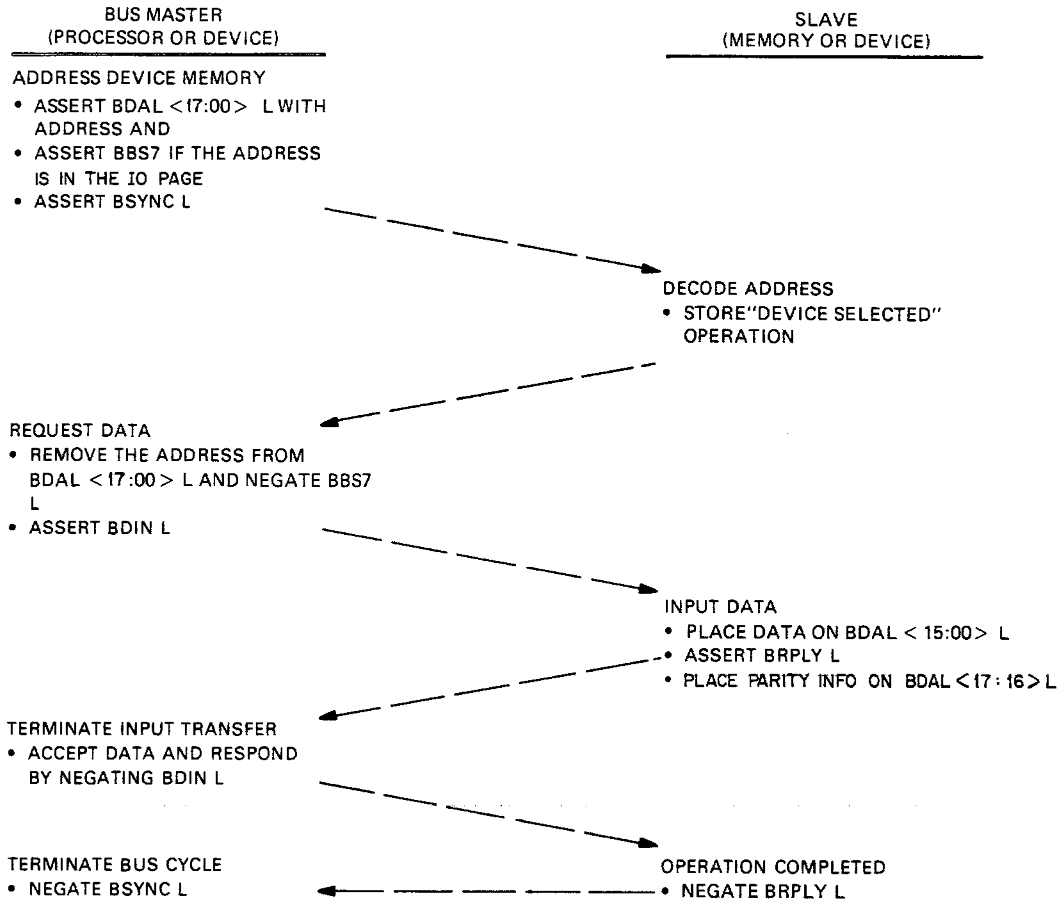
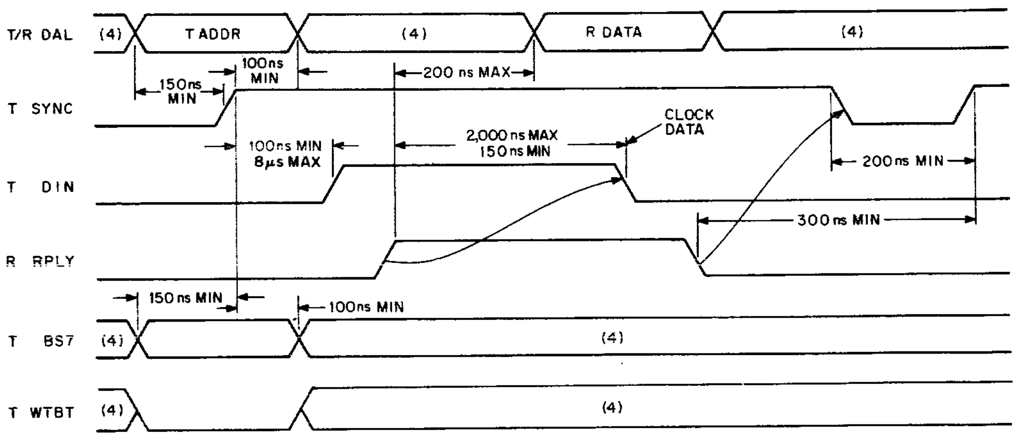

BUS MASTER (PROCESSOR OR DEVICE)
SLAVE (MEMORY OR DEVICE)
ADDRESS DEVICE MEMORY
• ASSERT BDAL <17:00> L WITH ADDRESS AND
• ASSERT BBS7 IF THE ADDRESS IS IN THE IO PAGE
• ASSERT BSYNC L
DECODE ADDRESS
• STORE "DEVICE SELECTED" OPERATION
REQUEST DATA
• REMOVE THE ADDRESS FROM BDAL <17:00> L AND NEGATE BBS7 L
• ASSERT BDIN L
INPUT DATA
• PLACE DATA ON BDAL < 15:00> L
• ASSERT BRPLY L
• PLACE PARITY INFO ON BDAL <17:16> L
TERMINATE INPUT TRANSFER
• ACCEPT DATA AND RESPOND BY NEGATING BDIN L
TERMINATE BUS CYCLE
• NEGATE BSYNC L
OPERATION COMPLETED
• NEGATE BRPLY L

Figure 2-1 DATI Bus Cycle

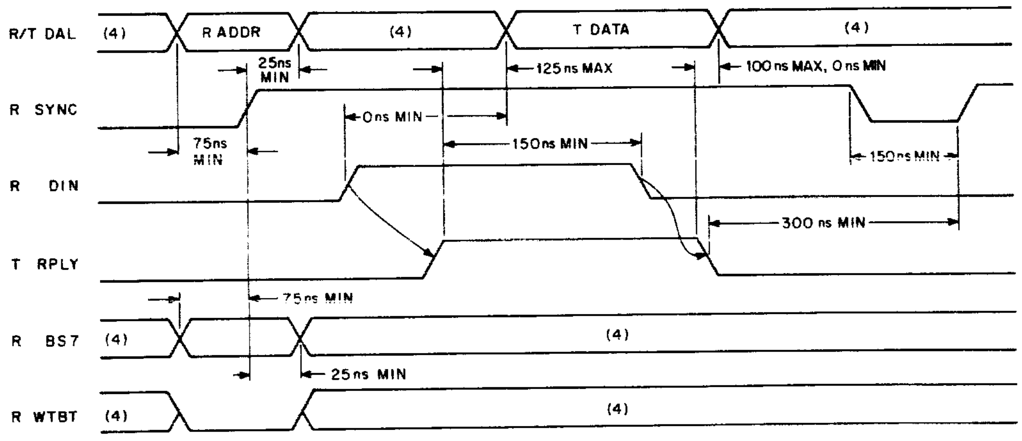
DATO(B) (Figures 2-3 and 2-4)—DATO(B) is a write operation. Data is transferred in 16-bit words (DATO) or 8-bit bytes (DATOB) from the bus master to the slave device. The data transfer output can occur after the addressing portion of a bus cycle when BWTBT L had been asserted by the bus master, or immediately following an input transfer part of a DATIO(B) bus cycle.

The data transfer portion of a DATO(B) bus cycle comprises a data setup and deskew time and a data hold and deskew time.

During the data setup and deskew time, the bus master outputs the data on BDAL<16:00> L at least 100 ns after BSYNC L is asserted. If it is a word transfer, the bus master negates BWTBT L at least 100 ns after BSYNC L assertion and BWTBT L remains negated for the length of the bus cycle. If the transfer is a byte transfer, BWTBT L remains asserted. If it is the output of a DATIOB, BWTB L becomes asserted and lasts the duration of the bus cycle. During a byte transfer, BDAL 00 L selects the high or low byte. This occurs while in the addressing portion of the cycle. If asserted, the high byte (BDAL<15:08> L) is selected; other-



TIMING AT MASTER DEVICE



TIMING AT SLAVE DEVICE

NOTES.

1. Timing shown at Master and Slave Device Bus Driver inputs and Bus Receiver Outputs.
2. Signal name prefixes are defined below:
   T = Bus Driver Input
   R = Bus Receiver Output
3. Bus Driver Output and Bus Receiver Input signal names include a "B" prefix
4. Don't care condition

Figure 2-2 DATI Bus Cycle Timing

wise, the low byte (BDAL<07:00> L) is selected. An asserted BDAL 16 L at this time will force a parity error to be written into memory if the memory is a parity-type memory. BDAL 17 L is not used for write operations. The bus master asserts BDOUT L at least 100 ns, but not more than 8 µsec, after BDAL and BWTBT L bus drivers are stable. The slave device responds by asserting BRPLY L within 10 microseconds to avoid bus timeout. This completes the data setup and deskew time.

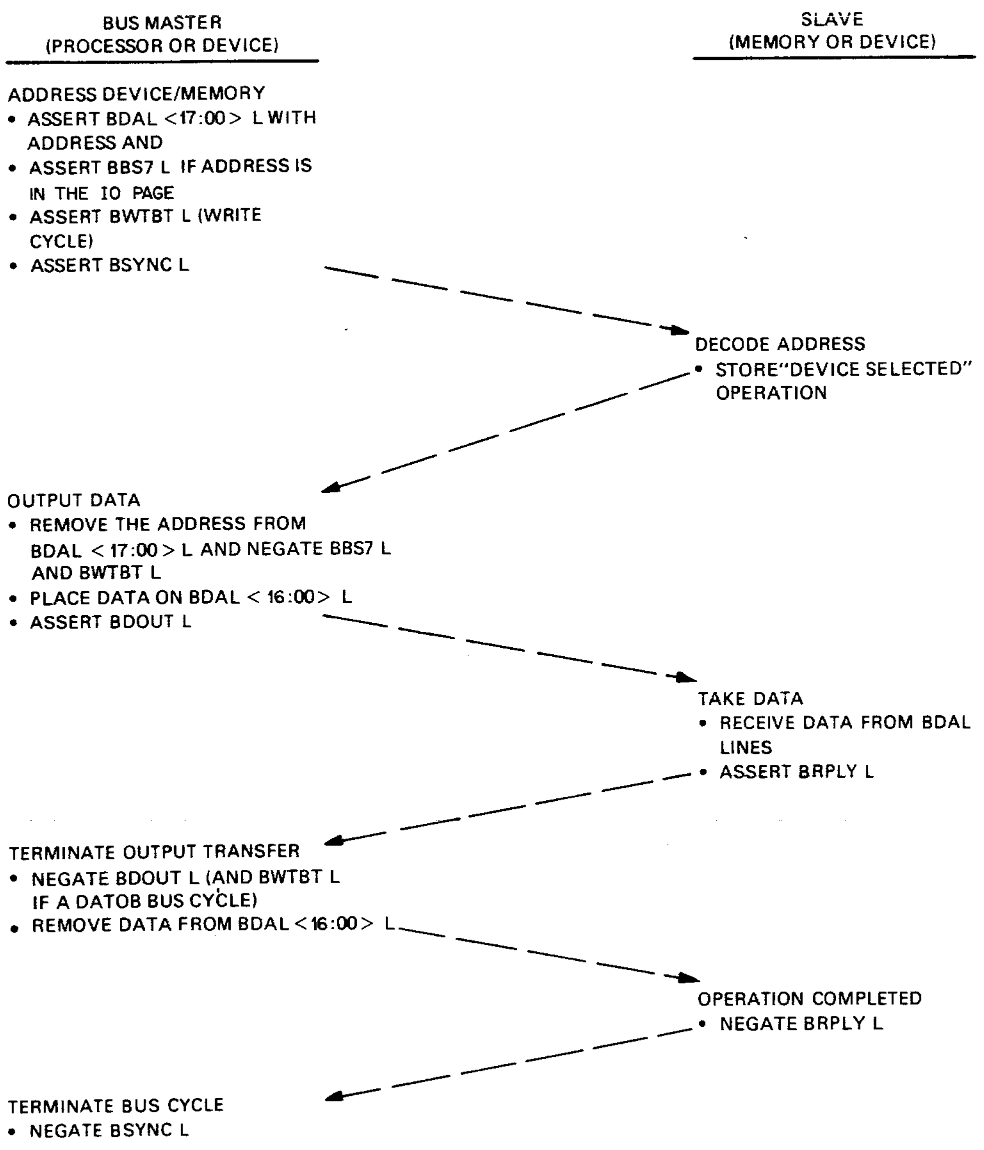
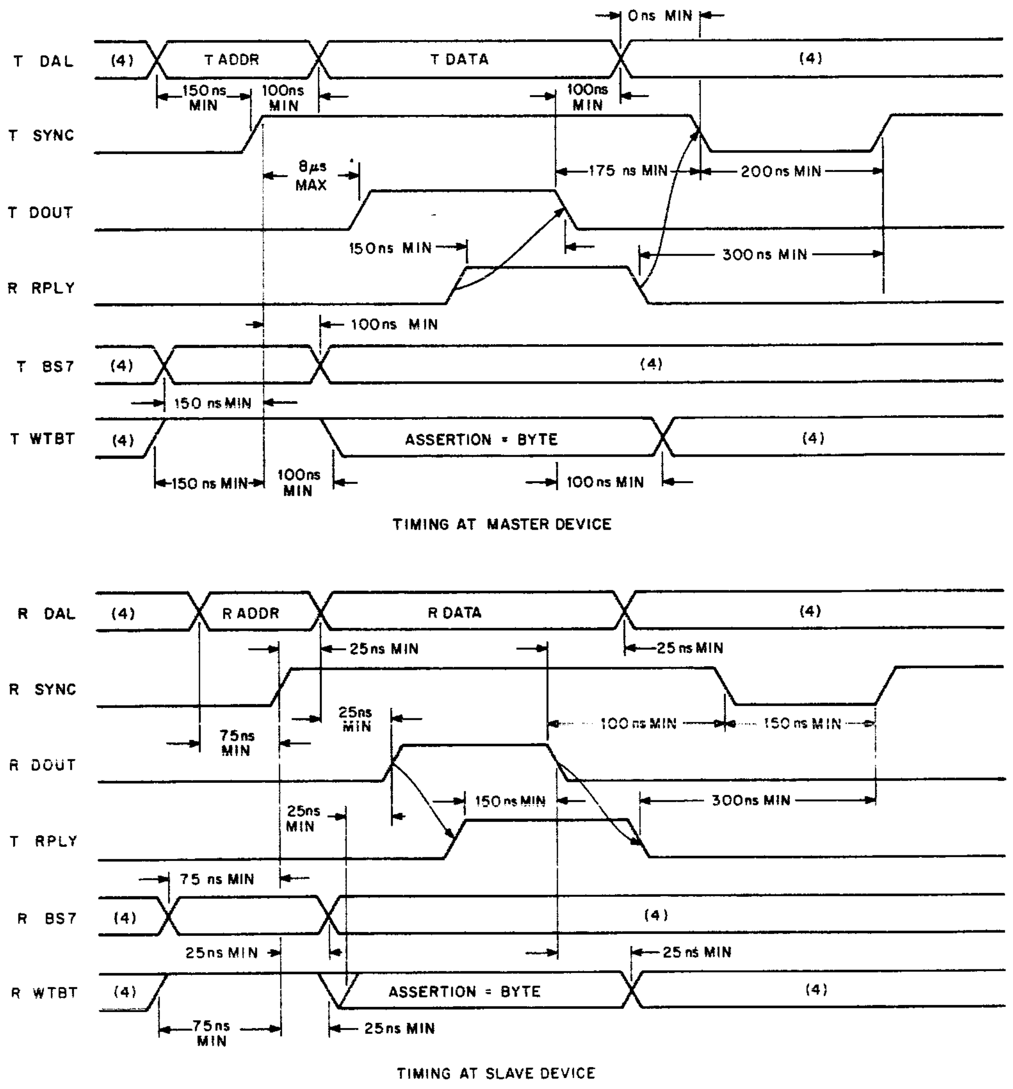

BUS MASTER (PROCESSOR OR DEVICE)
SLAVE (MEMORY OR DEVICE)
ADDRESS DEVICE/MEMORY
• ASSERT BDAL <17:00> L WITH ADDRESS AND
• ASSERT BBS7 L IF ADDRESS IS IN THE IO PAGE
• ASSERT BWTBT L (WRITE CYCLE)
• ASSERT BSYNC L
DECODE ADDRESS
• STORE"DEVICE SELECTED" OPERATION
OUTPUT DATA
• REMOVE THE ADDRESS FROM BDAL <17:00> L AND NEGATE BBS7 L AND BWTBT L
• PLACE DATA ON BDAL <16:00> L
• ASSERT BDOUT L
TAKE DATA
• RECEIVE DATA FROM BDAL LINES
• ASSERT BRPLY L
TERMINATE OUTPUT TRANSFER
• NEGATE BDOUT L (AND BWTBT L IF A DATOB BUS CYCLE)
• REMOVE DATA FROM BDAL <16:00> L
OPERATION COMPLETED
• NEGATE BRPLY L
TERMINATE BUS CYCLE
• NEGATE BSYNC L

Figure 2-3 DATO or DATOB Bus Cycle

During the data hold and deskew time the bus master receives BRPLY L and negates BDOUT L. BDOUT L must remain asserted for at least 150 ns from the receipt of BRPLY L before being negated by the bus master. BDAL<16:00> L bus drivers remain asserted for at least 100 ns after BDOUT L negation. The bus master then negates BDAL inputs.

During this time, the slave device senses BDOUT L negation. The data is accepted and the slave device negates BRPLY L. The bus master responds by negating BSYNC L. However, the processor will not negate BSYNC L for at least 175 ns after negating BDOUT L. This completes the DATO(B) bus cycle. Before the next cycle BSYNC L must remain unasserted for at least 200 ns.



NOTES

1 Timing shown at Master and Slave Device Bus Driver Inputs and Bus Receiver Outputs.

2. Signal name prefixes are defined below:

T = Bus Driver Input
R = Bus Receiver Output

3 Bus Driver Output and Bus Receiver input signal names include a "B" prefix.

4 Don't care condition

Figure 2-4 DATO or DATOB Bus Cycle Timing

DATIO(B) (Figures 2-5 and 2-6)—The protocol for a DATIO(B) bus cycle is identical to the addressing and data transfer portions of the DATI and DATO(B) bus cycles. After addressing the device, a DATI cycle is performed as explained above; however, BSYNC L is not negated. BSYNC L remains active for an output word or byte transfer [DATO(B)]. The bus master maintains at least 200 ns between BRPLY L negation during the DATI cycle and BDOUT L assertion. The cycle is terminated when the bus master negates BSYNC L, which is the same as described for DATO(B).

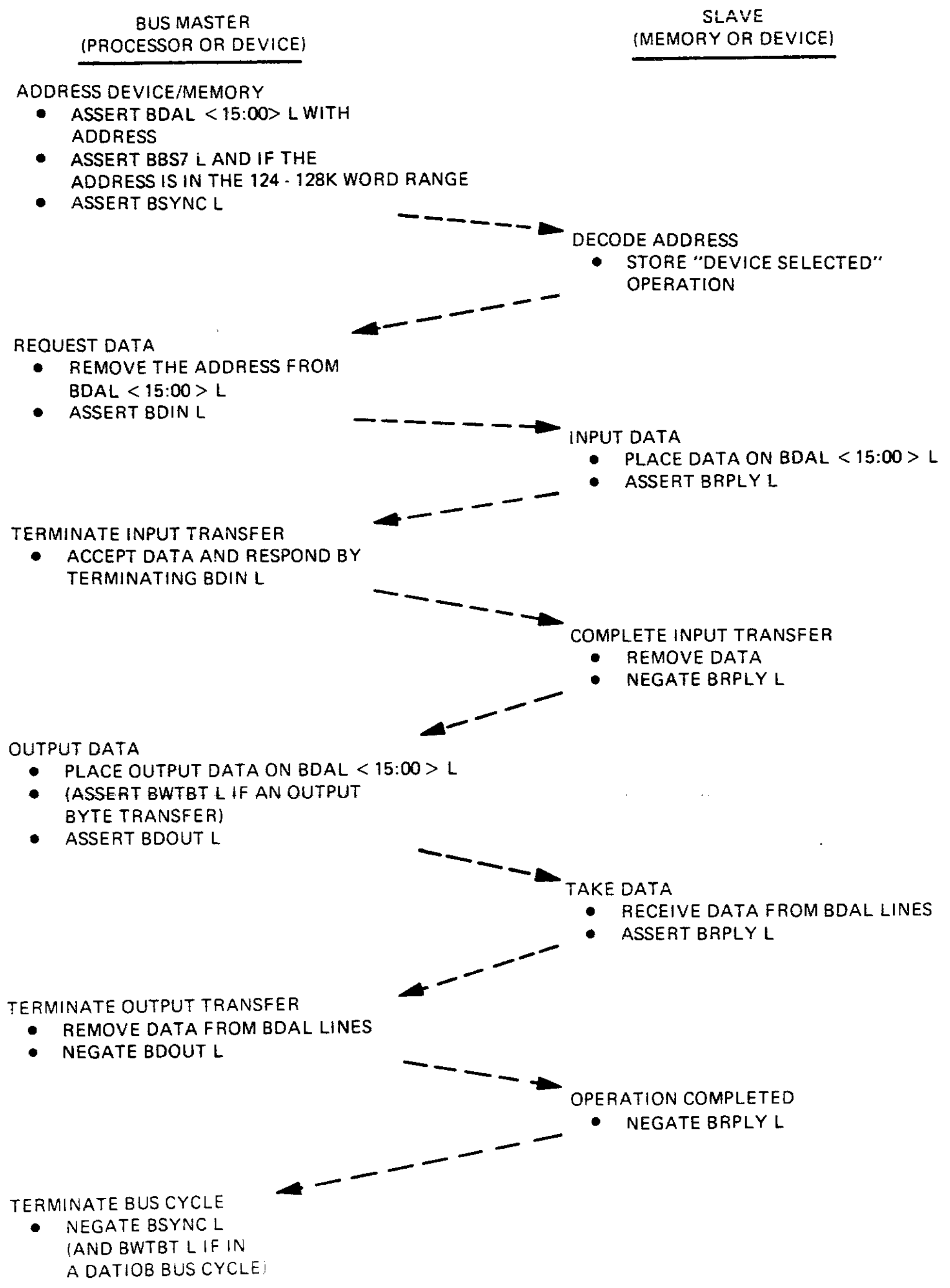
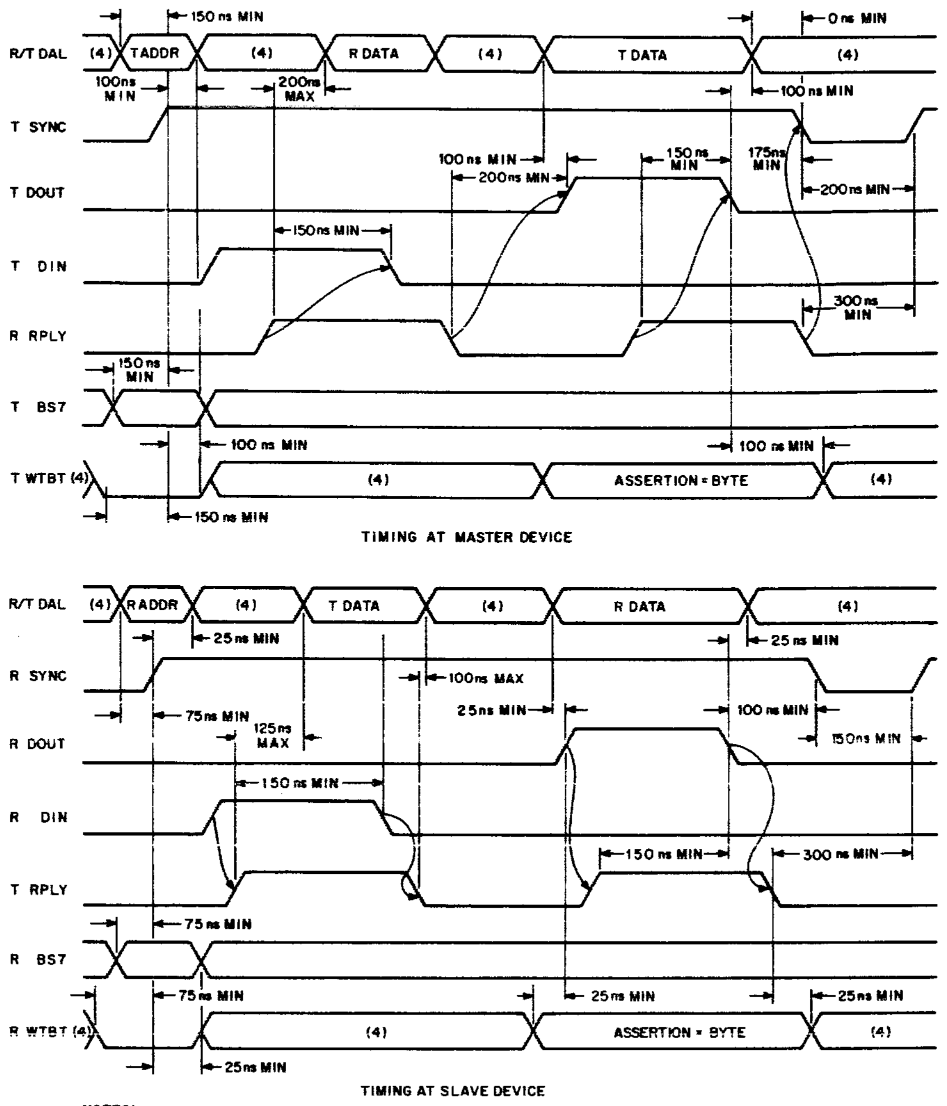

BUS MASTER (PROCESSOR OR DEVICE)
SLAVE (MEMORY OR DEVICE)
ADDRESS DEVICE/MEMORY
• ASSERT BDAL <15:00> L WITH ADDRESS
• ASSERT BBS7 L AND IF THE ADDRESS IS IN THE 124 - 128K WORD RANGE
• ASSERT BSYNC L
DECODE ADDRESS
• STORE "DEVICE SELECTED" OPERATION
REQUEST DATA
• REMOVE THE ADDRESS FROM BDAL <15:00> L
• ASSERT BDIN L
INPUT DATA
• PLACE DATA ON BDAL <15:00> L
• ASSERT BRPLY L
TERMINATE INPUT TRANSFER
• ACCEPT DATA AND RESPOND BY TERMINATING BDIN L
COMPLETE INPUT TRANSFER
• REMOVE DATA
• NEGATE BRPLY L
OUTPUT DATA
• PLACE OUTPUT DATA ON BDAL <15:00> L
• (ASSERT BWTBT L IF AN OUTPUT BYTE TRANSFER)
• ASSERT BDOUT L
TAKE DATA
• RECEIVE DATA FROM BDAL LINES
• ASSERT BRPLY L
TERMINATE OUTPUT TRANSFER
• REMOVE DATA FROM BDAL LINES
• NEGATE BDOUT L
OPERATION COMPLETED
• NEGATE BRPLY L
TERMINATE BUS CYCLE
• NEGATE BSYNC L (AND BWTBT L IF IN A DATIOB BUS CYCLE)

Figure 2-5 DATIO or DATIOB Bus Cycle

Parity Protocol

The KDF11-AA recognizes memory parity errors and traps to location 114_8 if one occurs. A parity error detection occurs during every DATI or DATI portion of a DATIO(B) cycle. The processor samples BDAL 16 L and BDAL 17 L after the 200 ns REPLY deskew time similar to BDAL <15:00> L. BDAL 16 L is interpreted as a parity error signal from memory and BDAL 17 L is interpreted as a parity error enable signal from an external parity controller module. BDAL 17 L is used by software to enable parity detection



NOTES:

1. Timing shown at Requesting Device Bus Driver Inputs and Bus Receiver Outputs.
2. Signal name prefixes are defined below:
   T = Bus Driver Input
   R = Bus Receiver Output
3. Bus Driver Output and Bus Receiver Input signal names include a "B" prefix.
4. Don't care condition.

Figure 2-6 DATIO or DATIOB Bus Cycle Timing

which is done by addressing a parity status register on the LSI-11 bus. Parity status register hardware then asserts BDAL 17 L during the BDIN L portion of DATI cycles to inform the processor or bus master that detection is enabled. BDAL 16 L is used to indicate a parity error and is asserted by the selected memory at REPLY time. Upon system power-up, memory may contain random data and erroneous parity error signals may be issued (BDAL 16 L asserted).

Until known data is written into memory, software keeps BDAL 17 L negated, to avoid false traps. After known data and correct parity have been written into memory, software can enable parity detection in the parity status register. If both BDAL 16 L and BDAL 17 L are asserted at REPLY time, an abort and trap to location 114_8 will occur. The assertion of BDAL 16 L during BDOUT L will cause memory to write wrong parity as a diagnostic tool for maintenance purposes.

Direct Memory Access

The direct memory access (DMA) capability allows direct data transfers between I/O devices and memory. This is useful when using mass storage devices (e.g., disks) that move large blocks of data to and from memory. A DMA device only needs to know the starting address in memory, the starting address in mass storage, the length of the transfer and whether the operation is read or write. When this information is available the DMA device can transfer data directly to (or from) memory. Since most DMA devices must perform data transfers in rapid succession or lose data, DMA devices are provided the highest priority.

DMA is accomplished after the processor (normally bus master) has passed bus mastership to the highest priority DMA device that is requesting the bus. The processor arbitrates all requests and grants the bus to the DMA device located electrically closest to the processor. A DMA device remains bus master indefinitely until it relinquishes its mastership. The following control signals are used during bus arbitration:

BDMGI L	DMA Grant Input
BDMGO L	DMA Grant Output
BDMR L	DMA Request Line
BSACK L	Bus Grant Acknowledge

DMA Protocol (Figures 2-7 and 2-8)—A DMA transaction can be divided into three phases: the bus mastership acquisition phase, the data transfer phase, and the bus mastership relinquish phase.

During the bus mastership acquisition phase, a DMA device requests the bus by asserting BDMR L. The processor arbitrates the request and initiates the transfer of bus mastership by asserting BDMGO L. The maximum time between BDMR L assertion and BDMGO L assertion is DMA latency. This time is processor-dependent. BDMGO L/BDMGI L is one signal that is daisy-chained through each module in the backplane. It is driven out of the processor on the BDMGO L pin, enters each module on the BDMGI L pin and exits on the BDMGO L pin. Propagation delay from BDMGI L to BDMGO L must be less than 500 ns per LSI-11 bus slot. Since this delay directly affects system performance, it should be kept as short as possible. This signal passes through the modules in descending order of priority until it is stopped by the requesting device. The requesting device blocks the output of BMDGO L and asserts BSACK L if BRPLY L and BSYNC L are negated. Propagation delay from BDMGI L to BSACK L must be less than 500 ns.

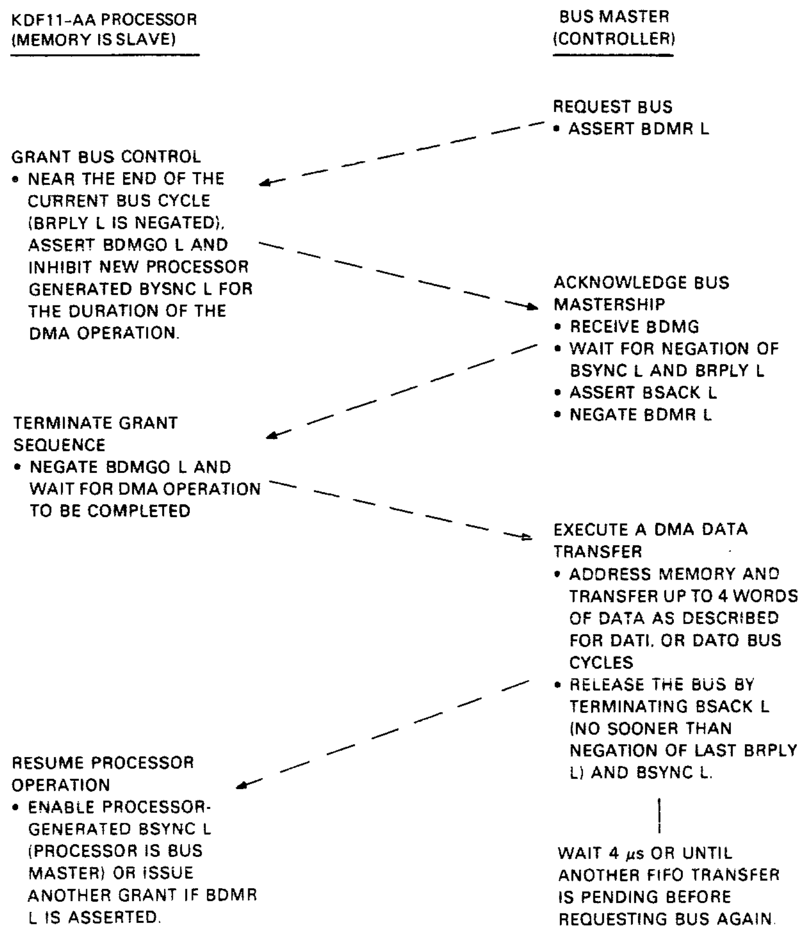

KDF11-AA PROCESSOR (MEMORY IS SLAVE)
BUS MASTER (CONTROLLER)
REQUEST BUS
• ASSERT BDMR L
GRANT BUS CONTROL
• NEAR THE END OF THE CURRENT BUS CYCLE (BRPLY L IS NEGATED), ASSERT BDMGO L AND INHIBIT NEW PROCESSOR GENERATED BYSNC L FOR THE DURATION OF THE DMA OPERATION.
ACKNOWLEDGE BUS MASTERSHIP
• RECEIVE BDMG
• WAIT FOR NEGATION OF BSYNC L AND BRPLY L
• ASSERT BSACK L
• NEGATE BDMR L
TERMINATE GRANT SEQUENCE
• NEGATE BDMGO L AND WAIT FOR DMA OPERATION TO BE COMPLETED
EXECUTE A DMA DATA TRANSFER
• ADDRESS MEMORY AND TRANSFER UP TO 4 WORDS OF DATA AS DESCRIBED FOR DATI, OR DATO BUS CYCLES
• RELEASE THE BUS BY TERMINATING BSACK L (NO SOONER THAN NEGATION OF LAST BRPLY L) AND BSYNC L.
RESUME PROCESSOR OPERATION
• ENABLE PROCESSOR-GENERATED BSYNC L (PROCESSOR IS BUS MASTER) OR ISSUE ANOTHER GRANT IF BDMR L IS ASSERTED.
WAIT 4 μs OR UNTIL ANOTHER FIFO TRANSFER IS PENDING BEFORE REQUESTING BUS AGAIN.

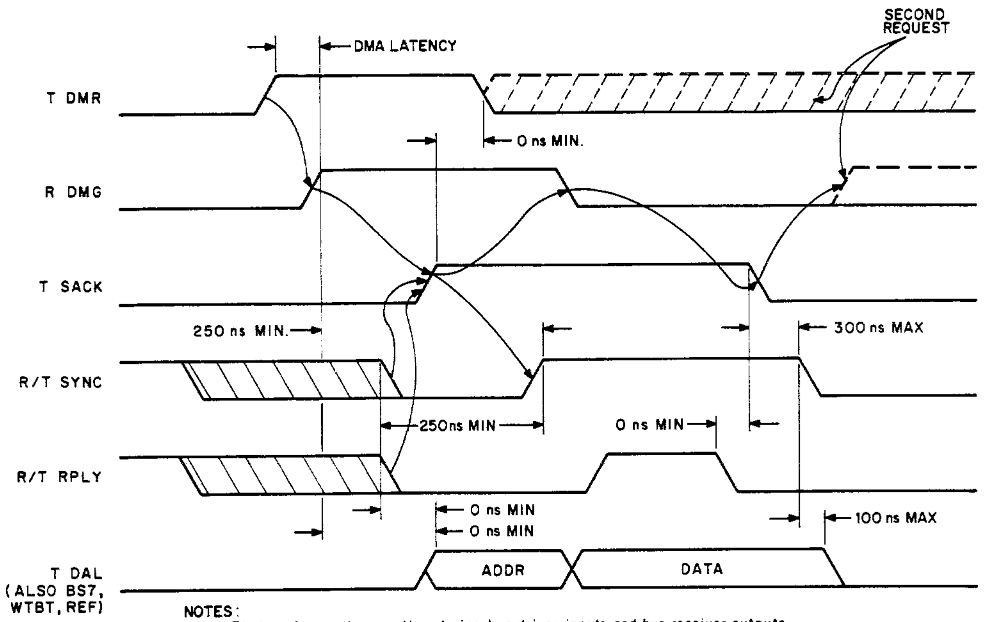
Figure 2-7 DMA Request/Grant Sequence

During the data transfer phase, the DMA device continues asserting BSACK L. The actual data transfer is performed as described in the sections on DATI, DATO, and DATIO.

NOTE

If multiple-data transfers are performed during this phase, consideration must be given to the use of the bus for other system functions, such as memory refresh (if required).

The DMA device can assert BSYNC L for a data transfer 250 ns (minimum) after it receives BDMGI L and its BSYNC L and BRPLY L become negated.



NOTES:
1. Timing shown at requesting device bus driver inputs and bus receiver outputs.
2. Signal name prefixes are defined below:
T = Bus Driver Input
R = Bus Receiver Output
3. Bus Driver Output and Bus Receiver Input signal names include a "B" prefix.

Figure 2-8 DMA Request/Grant Timing

During the bus mastership relinquish phase the DMA device relinquishes the bus by negating BSACK L. This occurs after completing (or aborting) the last data transfer cycle (BRPLY L negated). BSACK L may be negated up to 300 ns (maximum) before negating BSYNC L.

INTERRUPTS

The interrupt capability of the LSI-11 bus allows any I/O device to temporarily suspend (interrupt) current program execution and divert processor operation to service the requesting device. The processor inputs a vector from the device to start the service routine (handler). Like the device register address, hardware fixes the device vector at locations within a designated range (below location 001000). The vector is used as the first of a pair of contiguous addresses. The content of the first address is read by the processor and is the starting address of the interrupt handler. The content of the second address is a new processor status word (PS). The new PS can raise the interrupt priority level, thereby preventing lower level interrupts from breaking into the current interrupt service routine. Control is returned to the interrupt program when the interrupt handler is ended. The original (interrupted) program's address (PC) and its associated PS are stored on a *stack*. The original PC and PS are restored by a return from interrupt (RTI or RTT) instruction at the end of the handler. The use of the stack and the LSI-11 bus interrupt scheme can allow interrupts to occur within interrupts (nested interrupts), depending on the PS.

Interrupts can be caused by LSI-11 bus options. Interrupt operations can also originate from within the processor. These interrupts are called *traps.* Traps are caused by programming errors, hardware errors, special instructions and maintenance features.

The LSI-11 bus signals that are used in interrupt transactions are the following:

BIRQ4 L	Interrupt request priority level 4
BIRQ5 L	Interrupt request priority level 5
BIRQ6 L	Interrupt request priority level 6
BIRQ7 L	Interrupt request priority level 7
BIAKI L	Interrupt acknowledge input
BIAKO L	Interrupt acknowledge output
BDAL<15:00> L	Data/address lines
BDIN L	Data input strobe
BRPLY L	Reply

There are two classes of LSI-11 CPU's. One, the 11/03 CPU class, treats all interrupts as level 4. The other, the 11/23 CPU class, can distinguish between the four interrupt levels.

Device Priority
The LSI-11 bus supports the following two methods of device priority:

1. Distribution arbitration—Priority arbitration is implemented in logic on the interrupting device based on request priority information on the bus. When devices of equal priority level request an interrupt, priority is given to the device electrically closest to the processor.
2. Position-defined arbitration—Priority is determined solely by electrical position on the bus. The closer a device is to the processor, the higher its priority is.

Interrupt Protocol (Figures 2-9 and 2-10)
Interrupt protocol has three phases: interrupt request phase, interrupt acknowledge and priority arbitration phase, and interrupt vector transfer phase.

The interrupt request phase begins when a device meets its specific conditions for interrupt requests. For example, the device is *ready, done,* or an error has occurred. The interrupt enable bit in a device status register must be set. The device then initiates the interrupt by asserting the interrupt request line(s). BIRQ4 L is the lowest hardware priority level and is asserted for all interrupt requests for compatibility with previous LSI-11 processors. The level a device is configured at must also be asserted. A special case exists for level 7 devices which must also assert level 6. See item 2 of the arbitration discussion involving the 4-level scheme (below) for an explanation.

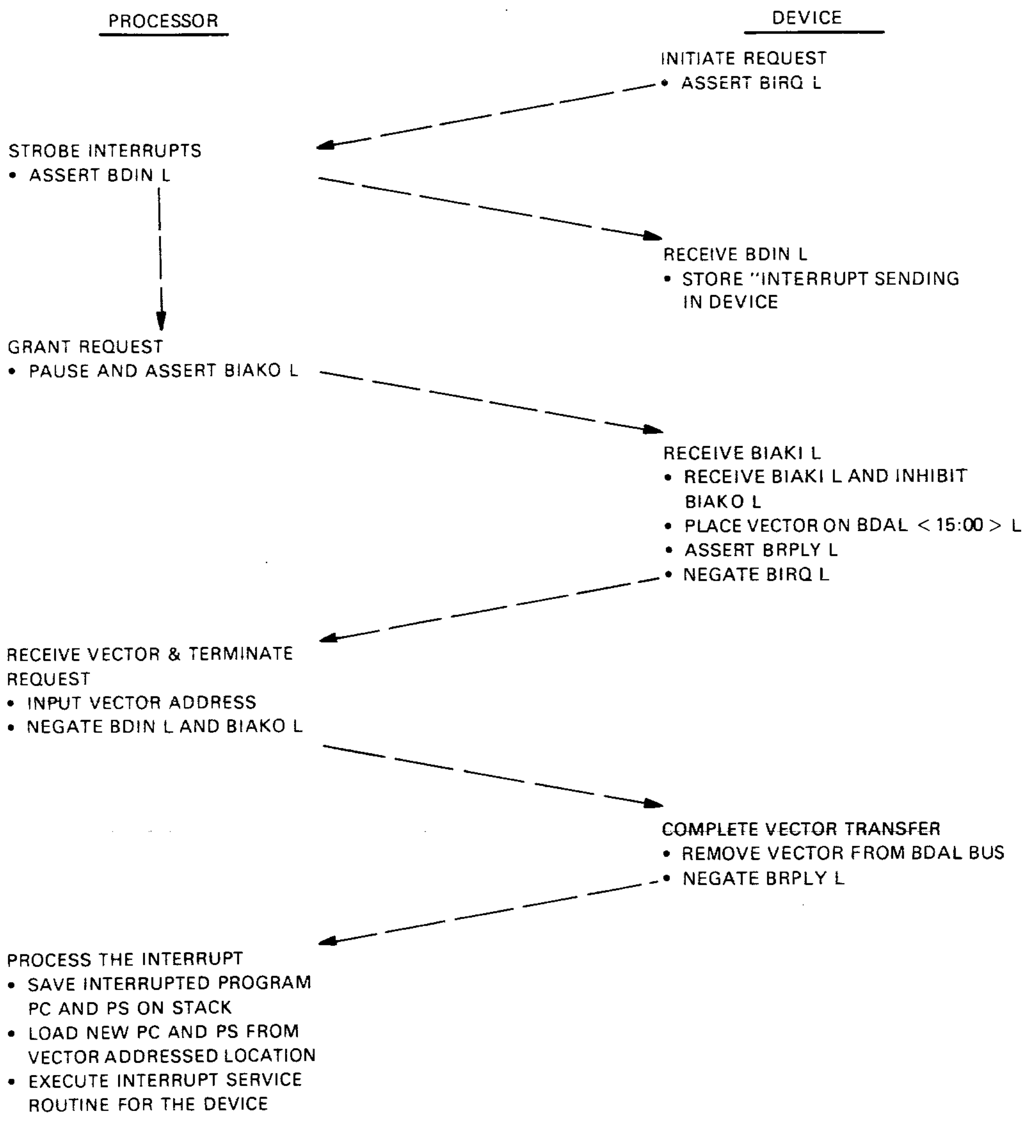

PROCESSOR
DEVICE
INITIATE REQUEST
• ASSERT BIRQ L
STROBE INTERRUPTS
• ASSERT BDIN L
RECEIVE BDIN L
• STORE "INTERRUPT SENDING IN DEVICE
GRANT REQUEST
• PAUSE AND ASSERT BIAKO L
RECEIVE BIAKI L
• RECEIVE BIAKI L AND INHIBIT BIAKO L
• PLACE VECTOR ON BDAL < 15:00 > L
• ASSERT BRPLY L
• NEGATE BIRQ L
RECEIVE VECTOR & TERMINATE REQUEST
• INPUT VECTOR ADDRESS
• NEGATE BDIN L AND BIAKO L
COMPLETE VECTOR TRANSFER
• REMOVE VECTOR FROM BDAL BUS
• NEGATE BRPLY L
PROCESS THE INTERRUPT
• SAVE INTERRUPTED PROGRAM PC AND PS ON STACK
• LOAD NEW PC AND PS FROM VECTOR ADDRESSED LOCATION
• EXECUTE INTERRUPT SERVICE ROUTINE FOR THE DEVICE

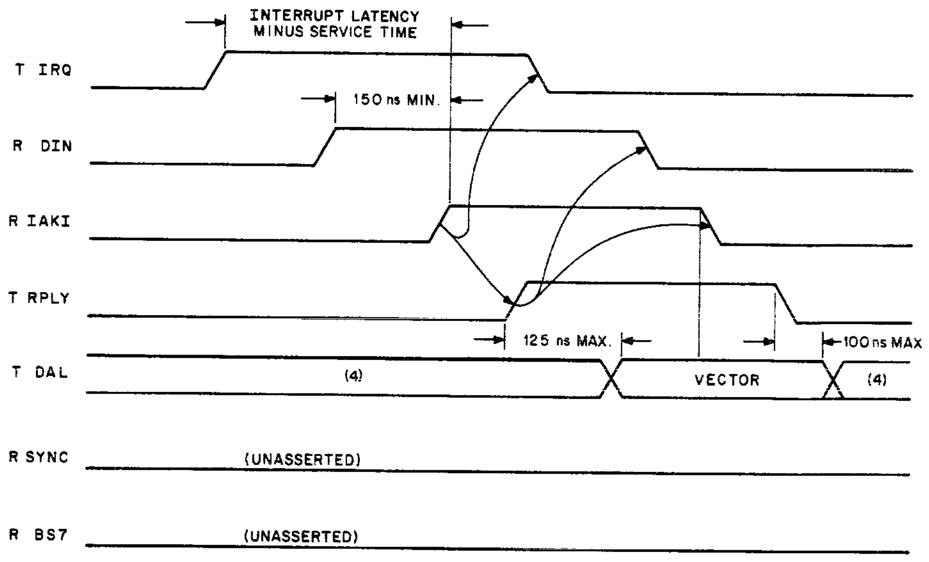
Figure 2-9 Interrupt Request/Acknowledge Sequence

Interrupt Level	**Lines Asserted by Device**
4	BIRQ4 L
5	BIRQ4 L, BIRQ5 L
6	BIRQ4 L, BIRQ6 L
7	BIRQ4 L, BIRQ6 L, BIRQ7 L

The interrupt request line remains asserted until the request is acknowledged.

During the interrupt acknowledge and priority arbitration phase the processor will acknowledge interrupts under the following conditions:

1. On the 11/03 class processors the PS bit 7 is cleared. On 11/23 class processors the device interrupt priority is higher than the PS<07:05>



NOTES:

1. Timing shown at Requesting Device Bus Driver Inputs and Bus Receiver Outputs.
2. Signal Name Prefixes are defined below:
   T = Bus Driver Input
   R = Bus Receiver Output
3. Bus Driver Output and Bus Receiver Input signal names include a "B" prefix.
4. Don't care condition

Figure 2-10 Interrupt Protocol Timing

2. The processor has completed instruction execution and no additional bus cycles are pending.

The processor acknowledges the interrupt request by asserting BDIN L, and 225 ns (minimum) later asserting BIAKO L. The device electrically closest to the processor receives the acknowledge on its BIAKI L bus receiver.

At this point the two types of arbitration must be discussed separately. If the device that receives the acknowledge uses the 4-level (distributed) interrupt scheme, it reacts as described below:

1. If not requesting an interrupt, the device asserts BIAKO L and the acknowledge propagates to the next device on the bus.
2. If the device is requesting an interrupt it must check to see that no higher level device is currently requesting an interrupt. This is done by monitoring higher level request lines. The table below lists the lines that need to be monitored by devices at each priority level.

   In addition to asserting levels 7 and 4, level 7 devices must drive level 6. This is done to simplify the monitoring and arbitration by level 4 and 5 devices. In this protocol, level 4 and 5 devices need not monitor level 7 since level 7 devices assert level 6. Level 4 and 5 devices will become aware of a level 7 request since they monitor the level 6 request.

Device Priority Level	Line(s) Monitored
4	BIRQ5, BIRQ6
5	BIRQ6
6	BIRQ7
7	—

3. If no higher level device is requesting an interrupt, the acknowledge is blocked by the device. (BIAKO L is not asserted). Arbitration logic within the device uses the leading edge of BDIN L to clock a flip-flop that blocks BIAKO L. Arbitration is won and the interrupt vector transfer phase begins.
4. If a higher level request line is active, the device disqualifies itself and asserts BIAKO L to propagate the acknowledge to the next device along the bus.

Signal timing must be carefully considered when implementing 4-level interrupts. Refer to Figure 2-10 for interrupt protocol timing.

If a single-level interrupt (position defined) device receives the acknowledge, it reacts as follows:

1. If not requesting an interrupt, the device asserts BIAKO L and the acknowledge propagates to the next device on the bus.
2. If the device was requesting an interrupt, the acknowledge is blocked using the leading edge of BDIN L and arbitration is won. The interrupt vector transfer phase begins.

The interrupt vector transfer phase is enabled by BDIN L and BIAKI L. The device responds by asserting BRPLY L and its BDAL<15:00> L bus driver inputs with the vector address bits. The BDAL bus driver inputs must be stable within 125 ns (maximum) after BRPLY L is asserted. The processor then inputs the vector address and negates BDIN L and BIAKO L. The device then negates BRPLY L and 100 ns (maximum) later removes the vector address bits. The processor then enters the device's service routine.

NOTE

Propagation delay from BIAKI L to BIAKO L must be no greater than 500 ns per LSI-11 bus slot.

The device must assert BRPLY L within 10 microseconds (maximum) after BIAKI L is asserted at the input to the module. Since the magnitude of both these times directly affects system performance, they should be kept as short as possible. Typical DIGITAL designs have less than 55 ns propagation delay from BIAKI L to BIAKO L.

4-Level Interrupt Configurations (LSI-11/23)

Users who have high-speed peripherals and desire better software performance can use the 4-level interrrupt scheme. Both position-independent and position-dependent configurations can be used with the 4-level interrupt scheme.

The position-independent configuration is shown in Figure 2-11. This allows peripheral devices that use the 4-level interrupt scheme to be placed in the backplane in any order. These devices must send out interrupt requests and monitor higher level request lines as described above. The level 4 request is always asserted by a requesting device regardless of priority, to allow compatibility if an LSI-11 or LSI-11/2 processor is in the same system. If two or more devices of equally high priority request an interrupt, the device physically closest to the processor will win arbitration.

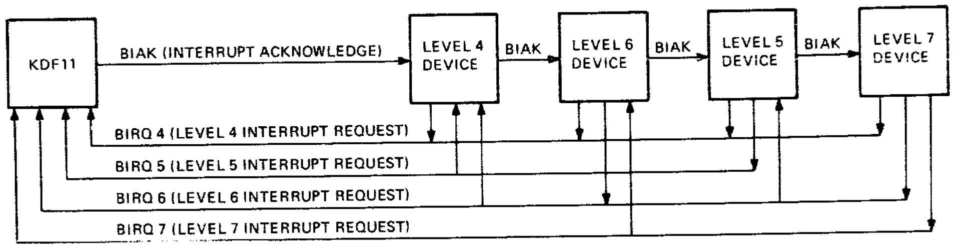

KDF11
BIAK (INTERRUPT ACKNOWLEDGE)
LEVEL 4 DEVICE
BIAK
LEVEL 6 DEVICE
BIAK
LEVEL 5 DEVICE
BIAK
LEVEL 7 DEVICE
BIRQ 4 (LEVEL 4 INTERRUPT REQUEST)
BIRQ 5 (LEVEL 5 INTERRUPT REQUEST)
BIRQ 6 (LEVEL 6 INTERRUPT REQUEST)
BIRQ 7 (LEVEL 7 INTERRUPT REQUEST)

Figure 2-11 Position-Independent Configuration (LSI-11/23)

The position-dependent configuration is shown in Figure 2-12. This configuration is simpler to implement. A constraint is that peripheral devices must be inserted with the highest priority device located closest to the processor and the remaining devices placed in the backplane in decreasing order of priority, with the lowest priority devices farthest from the processor. With this configuration each device only has to assert its own level and level 4 (for compatibility with an LSI-11 or LSI-11/2). Monitoring higher level request lines is unnecessary. Arbitration is achieved through the physical positioning of each device on the bus. Devices which use the position dependent scheme must be placed on the bus behind all position independent devices and in order of decreasing priority.

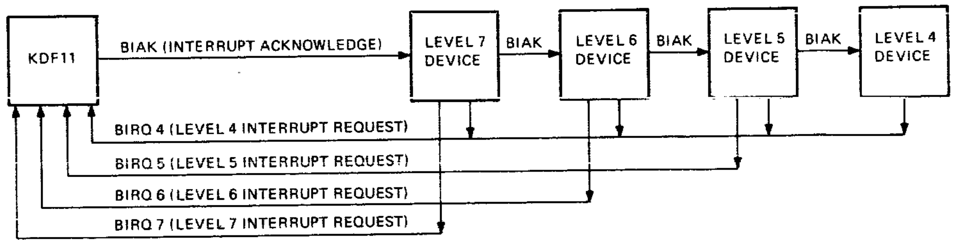

KDF11
BIAK (INTERRUPT ACKNOWLEDGE)
LEVEL 7 DEVICE
BIAK
LEVEL 6 DEVICE
BIAK
LEVEL 5 DEVICE
BIAK
LEVEL 4 DEVICE
BIRQ 4 (LEVEL 4 INTERRUPT REQUEST)
BIRQ 5 (LEVEL 5 INTERRUPT REQUEST)
BIRQ 6 (LEVEL 6 INTERRUPT REQUEST)
BIRQ 7 (LEVEL 7 INTERRUPT REQUEST)

Figure 2-12 Position-Dependent Configuration (LSI-11/23)

CONTROL FUNCTIONS

The following LSI-11 bus signals provide control functions.

BREF L	Memory refresh
BHALT L	Processor halt
BINIT L	Initialize
BPOK H	Power OK
BDCOK H	dc power OK

Memory Refresh

If BREF is asserted during the address portion of a bus data transfer cycle, it causes all dynamic MOS memories to be simultaneously addressed. The sequence of addresses required for refreshing the memories is determined by the specific requirements for each memory. The complete memory refresh cycle consists of a series of refresh bus transactions. A new address is used for each transaction. The effect of multiple data transfers by DMA devices must be carefully considered since they could delay memory refresh cycles.

Halt

Assertion of BHALT L stops program execution and forces the processor unconditionally into console ODT mode.

Initialization

Devices along the bus are initialized when BINIT L is asserted. The processor can assert BINIT L as a result of executing a RESET instruction or as part of a power-up sequence. BINIT L is asserted for approximately 10 microseconds when RESET is executed.

Power Status

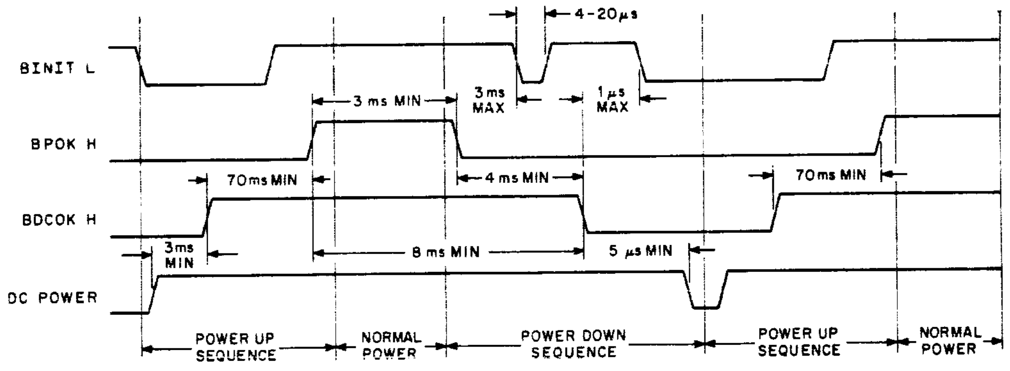
Power status protocol is controlled by two signals, BPOK H and BDCOK H. These signals are driven by some external device (usually the power supply) and are defined as follows.

BDCOK H—The assertion of this line indicates that dc power has been stable for at least 3 ms. The negation of this line indicates that only 5 microseconds of dc power reserve remains. Once BDCOK H is negated it must remain in this state for at least 1 microsecond before being asserted again. BDCOK H may be pulsed low for a minimum of 1 μs to cause the CPU to restart.

BPOK H—The assertion of this line indicates that there is at least an 8 ms reserve of dc power and that BDCOK H has been asserted for at least 70 ms. Once BPOK H has been asserted, it must remain asserted for at least 3 ms. The negation of this line indicates that power is failing and that only 4 ms of dc power reserve remains.

Power-Up/Down Protocol (Figure 2-13)
Power-up protocol begins when the power supply applies power with BDCOK H negated. This forces the processor to assert BINIT L. When the dc voltages are stable, the power supply or other external device asserts BDCOK H. The power supply asserts BPOK H 70 ms (minimum) after BDCOK H is asserted. The processor then performs its power-up sequence. Normal power must be maintained at least 3 ms before a power-down sequence can begin.

A power-down sequence begins when the power supply negates BPOK H. The current bus master, if not the processor, should relinquish the bus as soon as possible (maximum 1 ms). When the current instruction is completed, the processor traps to a power-down routine. The processor traps to location 24_8 which contains the PC that points to the power-down routine. The end of the routine is terminated with a HALT instruction to avoid any possible memory corruption as the dc voltages decay. The power fail routine has 4 ms to execute and HALT from the time BPOK L is negated.



NOTE:
Once a power down sequence is started, it must be completed before a power-up sequence is started

Figure 2-13 Power-Up/Power-Down Timing

BUS ELECTRICAL CHARACTERISTICS
This section contains information about the electrical characteristics of the LSI-11 bus.

AC Load Definition
AC load is a unit of measure of capacitance between a signal line and ground, as specified below. A unit load is defined as 9.35 pF of capacitance.

DC Load Definition
DC load is a unit of measure of the DC current flowing in a signal line. A unit load is defined as 105 μA flowing into a device when the signal line is in the high state.

120 Ohm LSI-11 Bus

The electrical conductors interconnecting the bus device slots are treated as transmission lines. A uniform transmission line, terminated in its characteristic impedance, will propagate an electrical signal without reflections. Insofar as bus drivers, receivers and wiring connected to the bus have finite resistance and nonzero reactance, the transmission line impedance becomes nonuniform, and thus introduces distortions into pulses propagated along it. Passive components of the LSI-11 bus (such as wiring, cabling and etched signal conductors) are designed to have a nominal characteristic impedance of 120 ohms.

The maximum length of interconnecting cable excluding wiring within the backplane is limited to 4.88 m (16 ft).

Bus Drivers

Devices driving the 120 ohm LSI-11 bus must have open collector outputs and meet the following specifications.

DC Specifications

Output low voltage when sinking 70 mA of current: 0.7 V maximum

Output high leakage current when connected to 3.8 Vdc: 25 μA (even if no power is applied to them, except for BDCOK H and BPOK H)

These conditions must be met at worst-case supply voltage, temperature, and input signal levels.

AC Specifications

Bus driver output pin capacitive load: Not to exceed 10 pF
Propagation delay: Not to exceed 35 ns

Skew (difference in propagation time between slowest and fastest gate): Not to exceed 25 ns

Rise/Fall Times: Transition time from 10% to 90% for positive transition, and from 90% to 10% for negative transition, must be no faster than 10 ns and no slower than 1 μs.

Bus Receivers

Devices that receive signals from the 120 ohm LSI-11 bus must meet the following requirements.

DC Specifications

Input low voltage (maximum): 1.3 V

Input high voltage (minimum): 1.7 V

Maximum input current when connected to 3.8 Vdc: 80 μA even if no power is applied to them.

These specifications must be met at worst-case supply voltage, temperature, and output signal conditions.

AC Specifications

Bus receiver input pin capacitance load: Not to exceed 10 pF

Propagation delay: Not to exceed 35 ns

Skew (difference in propagation time between slowest and fastest gate): Not to exceed 25 ns

Bus Termination (Figure 2-14)
The 120 ohm LSI-11 bus must be terminated at each end by an appropriate terminator. This is to be done as a voltage divider with its Thevenin equivalent equal to 120 ohms and 3.4 V nominal. This type of terminataion is provided by an REV11-A refresh/boot/ terminator, or the BDV11-AA.

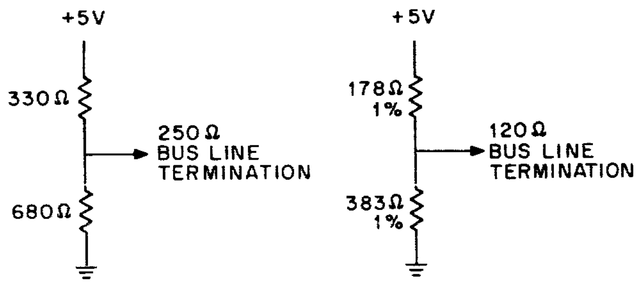


Figure 2-14 Bus Line Terminations

Each of the several LSI-11 bus lines (all signals whose mnemonics start with the letter B) must see an equivalent network with the following characteristics at each end of the bus.

Input impedance (with respect to ground): $Z = 120$ ohm $\pm 10\%$.

Open circuit voltage: 3.4 Vdc $+5\%$

Capacitance Load: Not to exceed 30 pF

NOTE

The resistive termination may be provided by the combination of two modules (i.e., the processor module supplies 220 ohms to ground). Both of these two terminators must be physically resident within the same backplane.

Bus Interconnecting Wiring
This section contains the electrical characteristics of the bus transmission lines.

Backplane Wiring—The wiring that interconnects all device interface slots on the LSI-11 bus must meet the following specifications:

1. The conductors must be arranged such that each line exhibits a characteristic impedance of 120 ohms (measured with respect to the bus common return).
2. Crosstalk between any two lines must be no greater than 5%. Note that worst-case crosstalk is manifested by simultaneously driving all but one signal line and measuring the effect on the undriven line.
3. DC resistance of signal path, as measured between near-end terminator and far-end terminator module (including all intervening connectors, cables, backplane wiring, connector-module etch, etc.) must not exceed 2 ohms.
4. DC resistance of common return path, as measured between near-end terminator and far-end terminator module (including all intervening connectors, cables, backplane wiring, connector-module etch, etc.) must not exceed an equivalent of 2 ohms per signal path. Thus, the composite signal return path dc resistance must not exceed 2 ohms divided by 40 bus lines, or 50 milliohms. Note that although this common return path is nominally at ground potential, the conductance must be part of the bus wiring; the specified low impedance return path must be provided by the bus wiring as distinguished from common system or power ground path.

Intra-Backplane Bus Wiring—The wiring that interconnects the bus connector slots within one contiguous backplane is part of the overall bus transmission line. Due to implementation constraints, the nominal characteristic impedance of 120 ohms may not be achievable. Distributed wiring capacitance in excess of the amount required to achieve the nominal 120 ohm impedance may not exceed 60 pF per signal line per backplane.

Power and Ground—Each bus interface slot has connector pins assigned for the following dc voltages.*

+5 Vdc—Three pins (4.5 A maximum per bus device slot)

+12 Vdc—Two pins (3.0 A maximum per bus device slot)

Ground—Eight pins (shared by power return and signal return).

NOTE

Power is not bused between backplanes on any interconnecting bus cables.

*The maximum allowable current per pin is 1.5 A. +5 Vdc must be regulated to ±5%; maximum ripple: 100 mV pp. +12 Vdc must be regulated to ±3%; maximum ripple: 200 mV pp.

SYSTEM CONFIGURATIONS

LSI-11 bus systems can be divided into two types:

1. Systems containing one backplane
2. Systems containing multiple backplanes

Before configuring any system, three characteristics for each module in the system must be known. These characteristics include:

1. Power consumption—+5 Vdc and +12 Vdc current requirements.
2. AC bus loading—the amount of capacitance a module presents to a bus signal line. AC loading is expressed in terms of ac loads where one ac load equals 9.35 pF of capacitance.
3. DC bus loading—the amount of dc leakage current a module presents to a bus signal when the line is high (undriven). DC loading is expressed in terms of dc loads where one dc load equals 105 microamperes (nominal).

Power consumption, ac loading, and dc loading specifications for each module are included in the Microcomputer Handbook Series.

NOTE

The ac and dc loads and the power consumption of the processor module, terminator module, and backplane must be included in determining the total loading of a backplane.

Rules for Configuring Single Backplane Systems (Figure 2-15)

1. The bus can accommodate modules that have up to 20 ac loads (total) before an additional termination is required. The processor has on-board termination for one end of the bus. If more than 20 ac loads are included, the other end of the bus must be terminated with 120 ohms.
2. A single backplane, terminated bus can accommodate modules comprising up to 35 ac loads (total).
3. The bus can accommodate modules up to 20 dc loads (total).
4. The bus signal lines on the backplane can be up to 35.6 cm (14 in) long.

Rules for Configuring Multiple Backplane Systems (Figure 2-16)

1. a. Up to three backplanes may compose the system.
   b. The signal lines on each backplane can be up to 25.4 cm (10 in) long.
2. Each backplane can accommodate modules that have up to 20 ac loads (total). Unused ac loads from one backplane may not

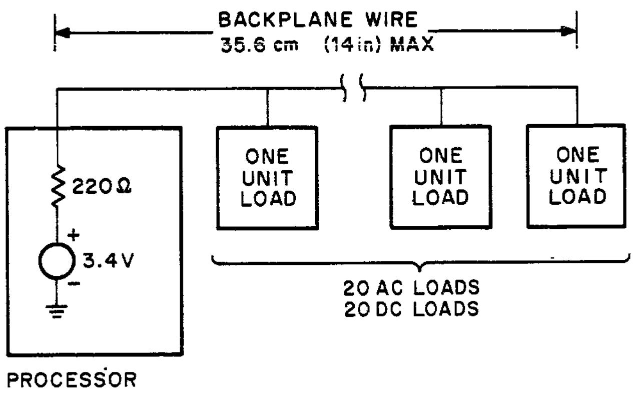


Figure 2-15 Single Backplane Configuration

be added to another backplane. It is desirable to load backplanes equally, or with the highest ac loads in the first and second backplanes.

3. DC loading of all modules in all backplanes cannot exceed 20 loads (total).
4. Both ends of the bus must be terminated with 120 ohms. This means that the first backplane must have an impedance of 120 ohms (obtained via the processor 220 ohm terminations and a separate 220 ohm terminator), and the last backplane must have a termination of 120 ohms.
5. a. The cable(s) connecting the first two backplanes is 61 cm (2 ft) or greater in length.
   b. The cable(s) connecting the second backplane to the third backplane is 22 cm (4 ft) longer or shorter than the cable(s) connecting the first and second backplanes.
   c. The combined length of both cables cannot exceed 4.88 m (16 ft).
   d. The cables used must have a characteristic impedance of 120 ohms.

Power Supply Loading

Total power requirements for each backplane can be determined by obtaining the total power requirements for each module in the backplane. Obtain separate totals for +5 V and +12 V power. Power requirements for each module are specified in the Microcomputer Handbook Series.

When distributing power in multiple backplane systems, do not attempt to distribute power via the LSI-11 bus cables. Provide separate, appropriate power wiring from each power supply to each backplane. Each power supply should be capable of asserting BPOK H and BDCOK H signals according to bus protocol; this is required if automatic power fail/restart programs are imple-

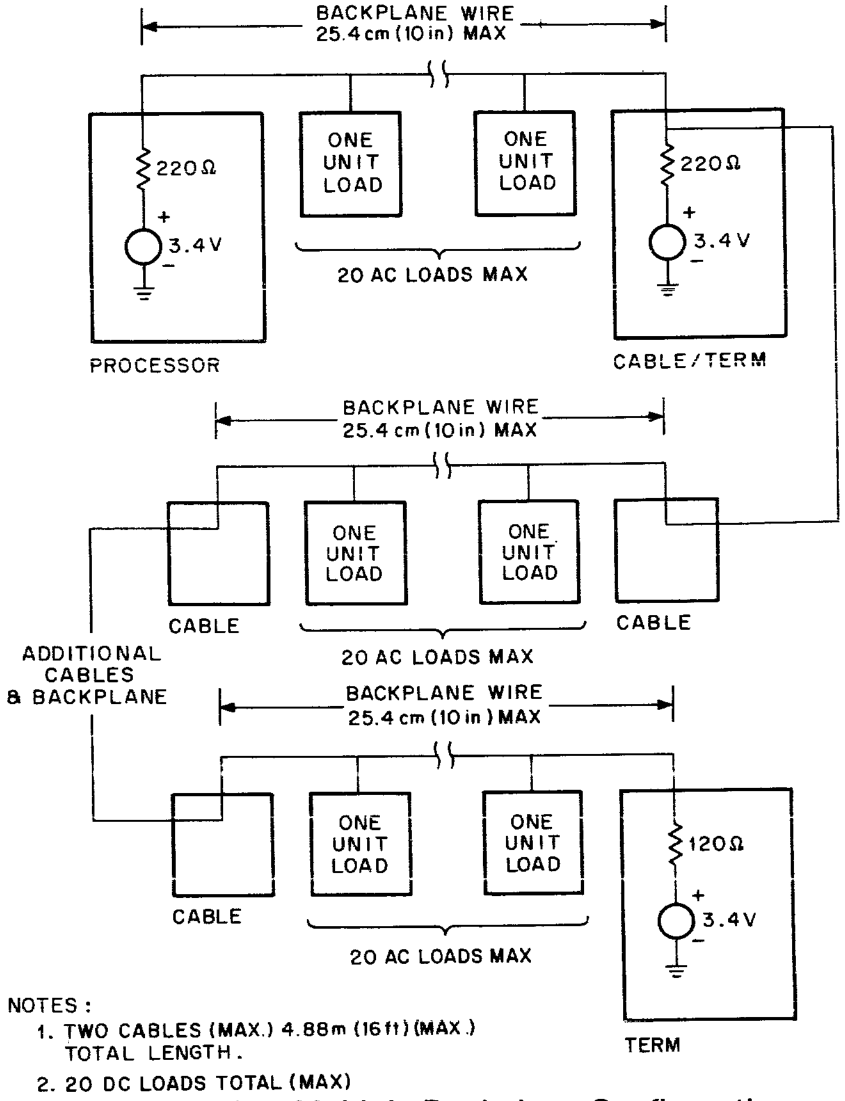


Figure 2-16 Multiple Backplane Configuration

mented, or if specific peripherals require an orderly power-down halt sequence. The proper use of BPOK H and BDOK H signals is strongly recommended.

NOTE

Timing diagrams reference signals at driver inputs (eg. TSYNC) and receiver outputs (eg. RSYNC). However, the accompanying text refers to the signals names in their bus specific form (eg. BSYNC).Figure 2-17 shows the relationship between the three signal names. Most timing numbers indicated in the text are given with respect to the R and T versions of the signals shown in the timing diagrams. In all cases the timing diagrams are the overriding authority.

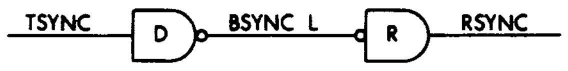


Figure 2-17 Signal Naming Conventions

appendices

MASSBUS

The MASSBUS is a peripheral-to-controller bus that has no arbitration. A single controller at one end of the bus receives or sends on each data transfer. Control information is transferred as on the Unibus, but the "master" is always the controller. Data blocks are transferred using a peripheral-generated clock, and the transfers are always initiated by writing a control word into a register in the in the peripheral.

Interruptions to the CPU are generated by the controller on demand from any peripheral. For this purpose an Attention signal exists in the control section of the Massbus. Each peripheral is capable of asserting this signal.

The Massbus was developed specifically to interface to high density and high data transfer rate disk storage devices of the IBM 3330 class. The RH11 controller connects the Massbus to the Unibus of PDP-11/34 system as shown in Figure A-1.

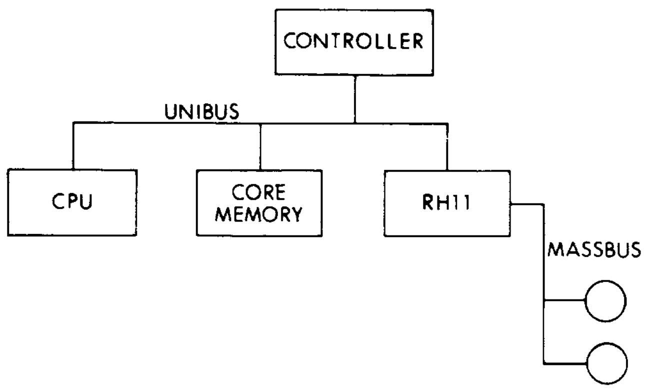


Figure A-1 The PDP-11/34 Massbus Configuration

In order to gain memory bandwidth for increases in both CPU and DMA performance, a new "back store" memory bus with a 32-bit wide data path was added to the PDP-11/70. Closely related to the memory bus was a backplane interconnection to the RH70 Massbus controllers (up to four of them) which can carry 32 bits at a time. In Figure A-2 the RH70-to-memory path is shown going through the cache because of a look-aside feature of the cache memory.

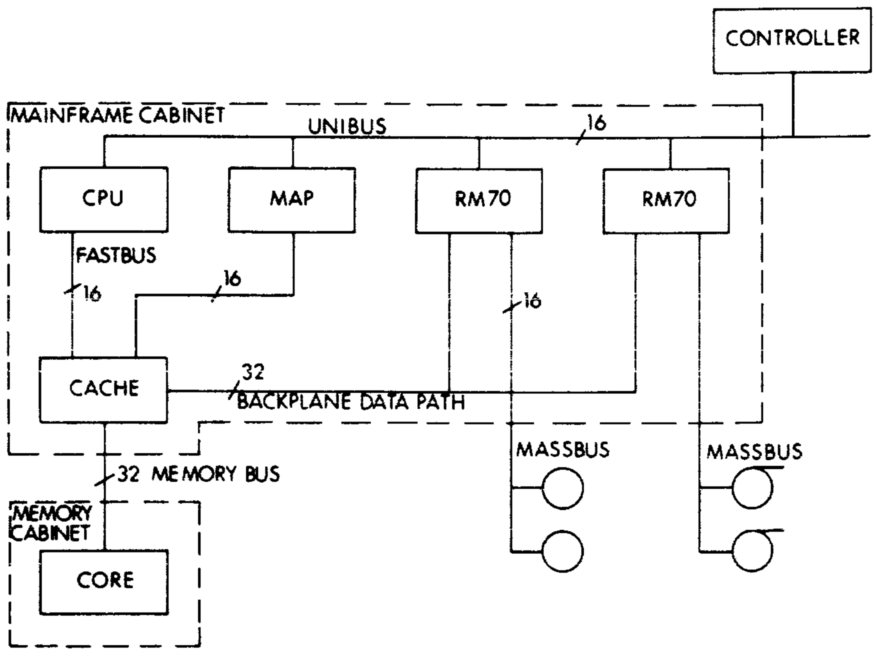


Figure A-2 The PDP-11/70 Massbus Configuration

The Massbus was designed to provide very high block transfer bandwidth, while keeping the control registers accessible to the central processor via the Unibus at all times. The splitting of the Unibus and the backplane data path in the PDP-11/70 matched well with the Massbus design.

The Massbus consists of two sections: a Control Section for reading and writing the contents of registers in the peripherals, and a Data Section for moving blocks of data. All transfers are between the controller and one of the (up to eight) peripherals. The two sections operate independently, except that a Control Section write into a control register of a peripheral is required to initiate a block transfer on the Data Section.

MASSBUS CONTROL SECTION

The Control Section of the Massbus is a miniature Unibus. However, the controller is always the master, and one of the peripherals is always the slave in the transfer. Figure A-3 shows the Control Section signals involved in data (i.e., control and status register) transfers. The Demand (DEM) signal takes the place of MSYN, and Transfer (TRA) takes the place of SSYN. Instead of Address and Control lines, there is an eight-bit address on the Massbus Control Section: three bits of Drive Select (DS), and five bits of Register Select (RS). Thus, each of eight peripherals (drives) may contain up to 32 two-byte registers. The Controller to Drive (CTOD) signal, when asserted, indicates that the transfer is a write into a peripheral register.

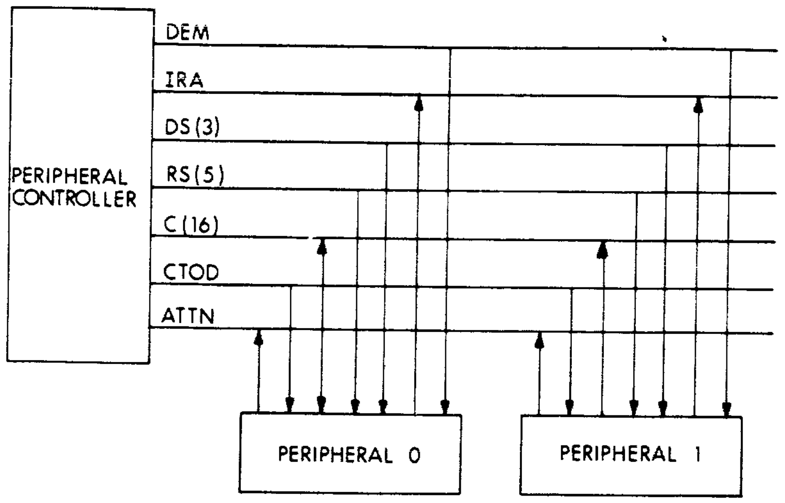


Figure A-3 The Control Signals of the Massbus

Control information is transferred 16 bits at a time on the C lines. Timing of these transfers is equivalent to that of the Unibus.

There is also a shared Attention (ATTN) signal in the Control Section that may be asserted at any time by a peripheral which requires CPU intervention. The controller normally creates an interruption to the CPU soon after ATTN is asserted.

The Massbus Control Section closely resembles the Unibus in timing, but it does carry one data parity check signal. If an error occurs on reading a control register, the controller passes the "bad parity" indication on to the CPU, with consequences the same as a memory parity error.

If an error occurs on writing a control register, the peripheral ignores the data word and asserts the Attention signal. "Control Bus Parity Error" is displayed in the Peripheral Error Status Register.

The Massbus Control Section also has the same acknowledgment and time-out properties as the Unibus, with the exception of reading the Attention Summary pseudo-register, which always uses the time-out to terminate the read cycle.

MASSBUS DATA SECTION

The Massbus Data Section is shown in Figure A-4. It contains 18 Data (D) lines, which carry data in both directions. Two clock signal lines, Synchronizing Clock (SCLK) and Write Clock (WCLK) carry a clock from and back to the peripheral, respectively. The RUN and End-of-Block (EBL) signals control the termination of a block data transfer. The Exception (EXC) signal is used to indicate error conditions.

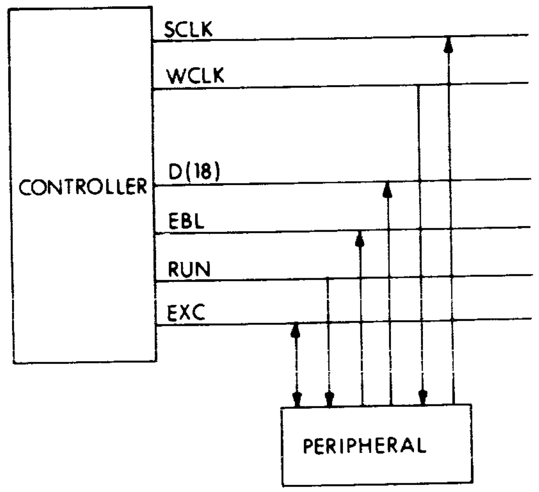


Figure A-4 Massbus Data Section

Data in the Massbus Data Section is always transferred in multiple-word blocks. The data read from or written to a mass storage device, such as a disk drive, must be synchronized with the mechanical motion of the recording medium. Therefore, the clock (SCLK) originates in the peripheral.

A Massbus Data Read begins when a control register in the selected peripheral is written with a Read command code. The controller asserts the RUN signal as soon as it is ready to receive data.

When the peripheral has received the RUN assertion, it begins reading data from its storage medium. The peripheral asserts SCLK when a new data word is present on the D lines. The peripheral continues to assert and negate the SCLK signal at the characteristic data rate.

Each time the controller receives the negation of SCLK, the controller captures and stores the data word from the D lines. Note that the peripheral does not receive any positive indication that the data word was received by the controller: the data transfer is "open loop."

At the end of the block of data words, the peripheral asserts EBL to indicate that it has reached the end of a data block.

When the controller receives the EBL assertion, it decides whether to continue (usually by inspecting a word count register). The controller must then negate RUN or else accept another block of data.

As the peripheral negates EBL, it senses the RUN signal. If it is negated the peripheral disconnects itself from the Massbus Data Section. Otherwise, the peripheral would transmit the next block of data.

If the number of words desired by the controller is less than an integral number of data blocks, the controller may negate RUN before EBL is asserted. The controller then simply ignores the remaining data words being transmitted.

During a Massbus Data Write, as for a data read, the peripheral controls the rate at which data is transmitted. However, this time the data is coming from the controller, which asserts the WCLK signal whenever it puts data onto the D lines.

The controller must have a data word ready each time it receives the negation of SCLK. Otherwise a "data overrun" condition occurs, which causes abnormal termination of the transfer.

The Massbus Data Section carries a parity check bit with each 18-bit word of data. A signal called Exception (EXC) can be asserted from either end to indicate a bad data transfer or other exceptional conditions.

NOTE

APPENDIX B

DECdataway

The DECdataway is a multidrop communication bus using an inexpensive twisted-pair cable to link terminals, hosts, and process I/O subsystems into integrated, intelligent information and control systems.

Up to 63 devices can be addressed by each DECdataway controller in a system host. Small hosts have two controllers; larger hosts, four. Each DECdataway can be up to 15,000 feet (4500 meters) long. Any device along the run can be removed without disturbing the system. Data transmission uses sophisticated error-checking codes.

The DECdataway greatly reduces plant wiring costs, initially and in modification and expansion. And it greatly improves the manager's access to the system's unified data base, providing what is known as Distributed Plant Management capability.

Distributed Plant Management is a concept that integrates host computers, interactive, intelligent operator-input terminals, sensor-based subsystems, communication links, high-level programming languages and efficient operating systems. It offers the managerial decision-makers real-time access to real-time data.

The piece-work employee can use the system to record output and receive acknowledgement. The hourly employee can use it to record attendance. The first-line supervisor can query the system on-line for specific answers about attendance, production rates, and machine status. The foundry foreman can use the system to manage core inventory. The production manager can use it to manage raw material inventory more productively and profitably. The plant controller can use it daily to update in-process inventory value.

Distributed Plant Management can tie together, in one reliable, low-cost system, all the information available about the many operations in a plant that must be managed in a timely way.

The DECdataway consists of a controller interfaced to a PDP-11 computer and to a number of terminals and peripherals (slaves), each containing a port. The port connector has address jumpers installed so that the controller can select any one device with which to communicate. The number of addresses per controller may not exceed 63. The DECdataway cable is linear and free of any branches, stubs, splitters, or repeaters. Positioning of the controller is optional, Figure B-1 is a block diagram of two typical DECdataway configurations.

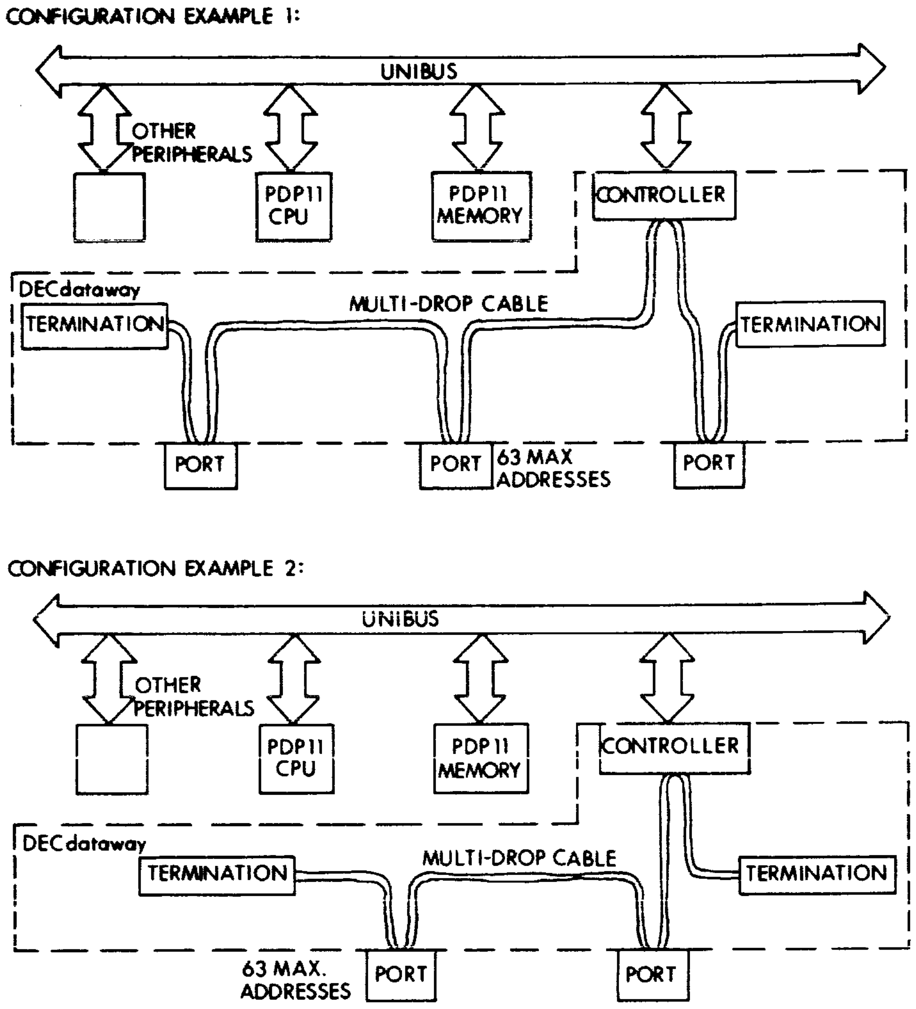


Figure B-1 Typical DECdataway Configurations

DECdataway provides the capability to transfer data between a PDP-11 and a maximum of 63 slaves in a block or message-oriented fashion via controller and multidrop cable. Data transfer between the PDP-11 and the controller is accomplished by the Unibus; data transfer between the controller and slaves is via the multidrop cable.

The Unibus program controlled interface appears as eight bytes of read/write registers in the PDP-11 external page. The registers are used primarily as a one-way port by the CPU to pass information to the controller. Communication of data from the controller to the CPU is done via an area in main memory set aside by the operating system.

A shielded, twisted pair cable provides a path for information transfer between the controller and the ports. Transfers are executed sequentially on the DECdataway (multidrop) cable.

DECdataway CONTROLLER

The controller is always master of the DECdataway and all transactions on the bus consist of two messages: (1) a command from the controller to the port and (2) a response from the port to the controller. There is one exception in that the controller has the capability of broadcasting a message to all ports simultaneously with no response allowed from any port.

Each message is formatted in DSBC (DEC Serial Bus Control) protocol to ensure error-free communication between the controller and ports by assuring accurate recovery from communication errors.

The controller controls access to the bus and thereby establishes bus priorities via the "round robin" polling scheme. The controller recovers from bus error conditions and reports errors to the host. The same controller monitors bus connections and notifies the host of devices that connect or disconnect to/from the bus.

The DECdataway controller maintains a logically separate nonprocessor request (NPR) in and NPR out channel for each device on the bus (up to 64 into the host CPU, including one unsolicited data channel, and up to 64 out of the host CPU, including one broadcast channel). The controller formats information transfers into messages with DSBC Protocol, and adds sequencing and Cyclic Redundancy Checking (CRC) information to each data transfer. Finally, the controller receives messages in DSBC protocol, passes data to the host via an NPR, and notifies the host of successful completion of message transfers.

DECdataway PORTS

DECdataway ports receive messages in DSBC protocol and check for port address and error-free message reception. If these conditions are met, a port acts in response to the received command. Ports transfer responses back through the bus to the host in DSBC protocol. Ports also perform DSBC protocol functions for bus initialization (and reinitialization) and error recovery.

DECdataway PERFORMANCE

If a communication error is detected during a transaction between the controller and a port, the controller generates a recovery procedure so that the transaction is successfully completed. A transaction may involve information transfer in one or two directions on the bus (only one transaction in a broadcast). The controller supports messages with up to 255 bytes (maximum) of information. A single bus transaction between the controller and a port can result in up to 255 bytes of information being moved in each direction on the bus in the correct order and without error. It is this ability to move error-free blocks of information of variable length in a single bus transaction that gives the DECdataway the throughput potential to be an effective multidevice bus for both terminals and process control peripherals.

The performance of the DECdataway is a function of:

- The communication rate
- The number and type of devices on the bus
- The protocol used on the bus

DECdataway data efficiency is the percent of data bytes versus total message transactions for (1) data transferred in one direction only and (2) in both directions for a complete transaction. Single byte transfers make poor utilization of the bus; transfers of 8 to 16 bytes can achieve 50% utiliaztion.

PHYSICAL DESCRIPTION

The DECdataway consists of a controller that includes a microprocessor module, a line unit module, and associated interface cables.

DECdataway MICROPROCESSOR SPECIFICATIONS

- Cycle time—300 ns
- Power requirements—5.0 A @ +5 V
- Environment must be in accordance with applicable portions of DEC Std. 102 for Class C (factory) environment. However, in most cases, the computer system is restricted to Class A (office) environments.

DECdataway LINE UNIT SPECIFICATIONS

The DECdataway line unit must be used with the DECdataway microprocessor. It is not a stand-alone device.

Power requirements	+4.75 V to +5.25 V @ 3.02 A (max.) +14 V to +16 V @ 0.1 A (max.) −14 V to −16 V @ 0.1 A (max.)
Operating mode	Half Duplex
Data format	Synchronous serial by bit, LSB first
Character size	8 bits plus 0 to 2 stuffing bits
Data rate	55,500 bits/second
Modulation	Biphase (space)
Transmitter timing	Adjustable RC clock
Receiver timing	From received signal
Line interface	Transformer coupled

Transmitter signal	6 V peak (min.) into terminated 200 ohm cable
Receiver signal threshold	150 mV peak (min.)
Error-free signal level	250 mV peak (min.)
Common mode isolation	350 Vac peak (min.) 500 Vdc
Receiver bandpass	6 kHz—130 kHz 3 db points

CABLE SPECIFICATIONS

The integral modem is designed to operate with shielded, balanced, or unbalanced 75 to 200 ohm cables. Digital Equipment Corporation recommends Belden 9182 cable. The cable consists of a shielded, twisted pair of 22 AWG conductors. The maximum length of this cable is 4,572 m (15,000 ft.). This distance must include the lengths required at the host and terminals for servicing. Belden 9182 specifications are given below.

Conductors	22 AWG stranded tinned copper (19 x 34)
Dielectric material	Cellular polypropylene
Nominal insulation	1.65 mm (0.065 in.)
Number of pairs thickness	One
Shield	Duofoil (polyester film laminated on both sides with aluminum foil)
Drain wire	22 AWG (19 x 34) tinned copper
Jacket	PVC
Jacket thickness	0.89 mm (0.035 in.)
Cable nominal diameter	10.26 mm (0.404 in.)
Nominal dc resistance of conductor at 20°C (68°F)	52.5 ohms/km (16 ohms/1000 ft.)
Nominal capacitance between conductors of a pair	29.5 pF/m (9 pF/ft.)
Nominal velocity (V) of propogation (% of velocity of light)	67 @ 56 kHz
Nominal impedance (Zo) of a pair (ohms)	150 @ 1 mHz; 200 @ 56 kHz

Other cables than Belden 9182 may be used, provided the following criteria are met. However, if cable impedance is less than 200 ohms then the maximum distance is reduced.

- Characteristic impedance $75 \leq Zo \leq 200$ ohms measured at 56 kHz.
- Cable loss <15 db measured at 56 kHz with termination at both end points.
- Cable run is homogeneous with shield continuity maintained.
- Step response $\leq$ 5 microseconds, 10%—90% risetime measured differentially at end point and driven from a differential source into terminated line.
- Conductor size—22 AWG.

The ends of the cable must be terminated in the characteristic impedance of 200 ohms measured at 56 kHz. The ports and controller do not provide termination.

CABLE CONNECTIONS

Each DECdataway consists of a continuous, homogeneous length of shielded, twisted pair cable with a 200 ohm termination resistor fixed across the signal pair at each end. Each DECdataway has one "DECdataway Controller Connector" and from one to 63 "DECdataway Port Connectors" as required by the application. The controller connector may be placed anywhere along the length of the cable. The port connectors may also be placed anywhere along the length of the cable. The only restriction on how close together port connectors may be placed is the need for sufficient slack to enable servicing terminals conveniently. Digital recommends a minimum of 3.05 m (10 ft.) of cable between port connectors. The maximum total length of cable permitted is 4572 m (15,000 ft.), if Belden 9182 is used. The maximum length of cable is not specified for any other cable type.

There is only one controller for each DECdataway; therefore no address selection is necessary. The addresses of the DECdataway port connectors are set by jumpers inserted in each port connector at the time of installation. This arrangement, along with the fact that the DECdataway is a true multidrop communication line, will allow terminals to be removed from and replaced on the line for servicing without disturbing the DECdataway operation.

APPENDIX C

PCL11

The Parallel Communications Link (PCL11) is a high performance computer link used for interconnecting multiple PDP-11 or VAX-11/780 computers in a local distributed processing network. Up to 16 processors may be connected to the PCL11 network. See Figure C-1. Each computer may exchange messages or data blocks with any other computer in the network. Communications occur in a DMA block transfer mode over a time division multiplexed (TDM) 16-bit parallel bus. Because of the TDM nature of the PCL11 bus, to 16 conversations may be conducted simultaneously. The PCL11 is used to build local networks in a variety of applications. These include distributed processing, distributed data base management, industrial data collection and control systems, simulation systems, transaction processing, laboratory data collection and control networks.

The power and features of the PCL11 system make it the ideal multiprocessor communications link for applications of several processors in local networks. The PCL11 allows for communications between any computer pair in the network, a flexibility which would otherwise require a very large number of two processor links. The PCL11 offers the ability to add more processors to the network by simply adding additional PCL11 nodes.

SOFTWARE SUPPORTED

Ease of use is provided by optionally available RSX-11M and VMS device driver software which allows user programs to interface with the PCL11 by means of standard format I/O calls.

The PCL11 will be supported by DECnet for RSX-11M and S operating systems beginning in Summer 1979.

TRANSPARENCY OF OPERATION

The user experiences near perfect transparency of operation. The PCL11 hardware manages the protocol and error checking for communication with a recipient computer and for successful transfer of the data or message. The user simply tells the PCL11 what data is to be sent and to which recipient computer. The user is then notified upon successful completion of the transfer. If, after multiple attempts, the PCL11 is unsuccessful in transferring the data, it will inform the user together with an indication of the reason (receiver busy, no acknowledgment, unresolved errors, etc.).

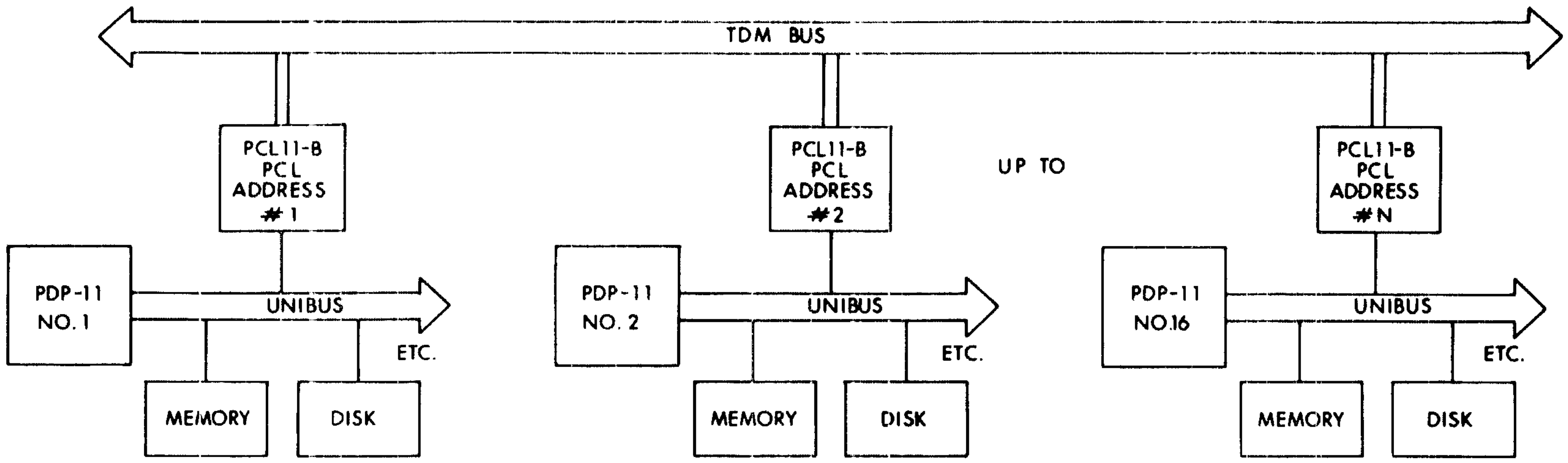


Figure C-1 PCL11 Communications System Block Diagram

ERROR CHECKING

The PCL11 hardware provides error checking in the hardware by use of word parity and block CRC-16. If an error is detected, the message is automatically retransmitted by the software driver.

HIGH AVAILABILITY SYSTEMS

The PCL11 is designed for use in high availability systems. The connections of the PCL11 interface on each computer to the PCL11 bus is such that a computer and interface may be powered on or off without disabling the bus or the rest of the network. The bus is not "daisy chained" but is connected to the computer and interface by a "T connection." A computer may even be physically unplugged from the PCL11 bus, moved away, and replaced without stopping the network. If a data error should occur, the error will be detected by the hardware and the data automatically retransmitted by the software driver.

Provision is even made for backup of the PCL11 unit which is clocking the PCL11 bus. If the PCL11 interface which has been designated to clock and control the PCL11 bus fails, a designated secondary or backup PCL11 unit automatically assumes control. The new controlling computer is notified of the action and the user may, via software, designate a new secondary or backup PCL11 control unit. All PCL11 units are identical and any unit may be designated as master or secondary.

HIGH BANDWIDTH AND FLEXIBILITY

The PCL11 provides high bandwidth data transfer rates plus flexibility in the allocation of that bandwidth among the various nodes in the network.

The maximum PCL11 bus bandwidth is 1 million bytes per second. There are two mechanisms for dividing the bandwidth among nodes. The default allocation simply divides the bus bandwidth equally among the PCL11 nodes on the bus. That is, the TDM time slices simply go "round robin" among the nodes.

The allocation of the TDM time slices can be set and varied under software control. The user may load a register in the PCL11 with an explicit list of the time slice allocations by transmitter number. This may be done to give specific bandwidth allocations to different PCL11 transmitters in the network. Different transmitters may have to handle differing data rate requirements up to a maximum of half of the system bandwidth to one node and a minimum of none to another. For example, a large data base management processor in a network might be given a greater proportion of the bandwidth if it needs to send large amounts of data to other processors in the network.

For total PCL11 bus lengths in excess of 50 feet, the bandwidth is reduced as a function of the bus length.

MULTIPLE UNITS AND MULTIPLE BUSES

Additional flexibility and power are provided by the ability to put multiple PCL11 units on one processor and the ability to implement multiple bus systems.

The use of two or three PCL11 buses to interconnect processors can give increased reliability through redundancy as well as increased throughput. In the unlikely event of a failure disabling one PCL11 bus, the system would continue operation using the other bus. The use of dual bus systems is recommended for medium to large networks. See Figure C-2.

A dual or triple bus system provides still greater flexibility. For example, one bus could be heavily loaded with long data transfers while another bus is kept lightly loaded and used for command and control messages where quick response is important. A third bus could be kept in reserve for backup.

A processor may also have more than one PCL11 node on the same PCL11 bus. The use of two nodes on the same bus might be valuable for processors having large amounts of traffic. The two nodes operate independently and concurrently, thus allowing greater throughput on that processor.

BUILDING THE SYSTEM

A PCL11 network is a system constructed from several components which are combined to meet the particular application needs. The PCL11-B is the Unibus interface which contains the PCL11 electronics and connects to the PCL11 bus cable. One PCL11-B forms one node on the network. The PCL11-B interfaces are interconnected by cables of appropriate length to suit the site geometry.

The PCL11-B can be used with any of the family of Unibus computers ranging from the small PDP-11/04 to the new VAX-11/780 computer system. This interfacing to the common Unibus also means that the user can achieve graceful growth of the network to incorporate more CPUs, larger CPUs, and future newer technology computers.

If the network is small, say three computers, interconnection can effectively be made using point-to-point communications links such as the DMC11 network link. This is illustrated in Figure C-3 for a three computer network where three linkages are needed for the complete interconnection of the computers.

However, as the number of computers in the network increases to four and beyond, the number of linkages required to interconnect the computers increases geometrically. For a four computer network, six linkages are required. Fifteen linkages are required to fully interconnect a six computer network system. The PCL11-B is the device of choice for interconnecting local comptuer net-

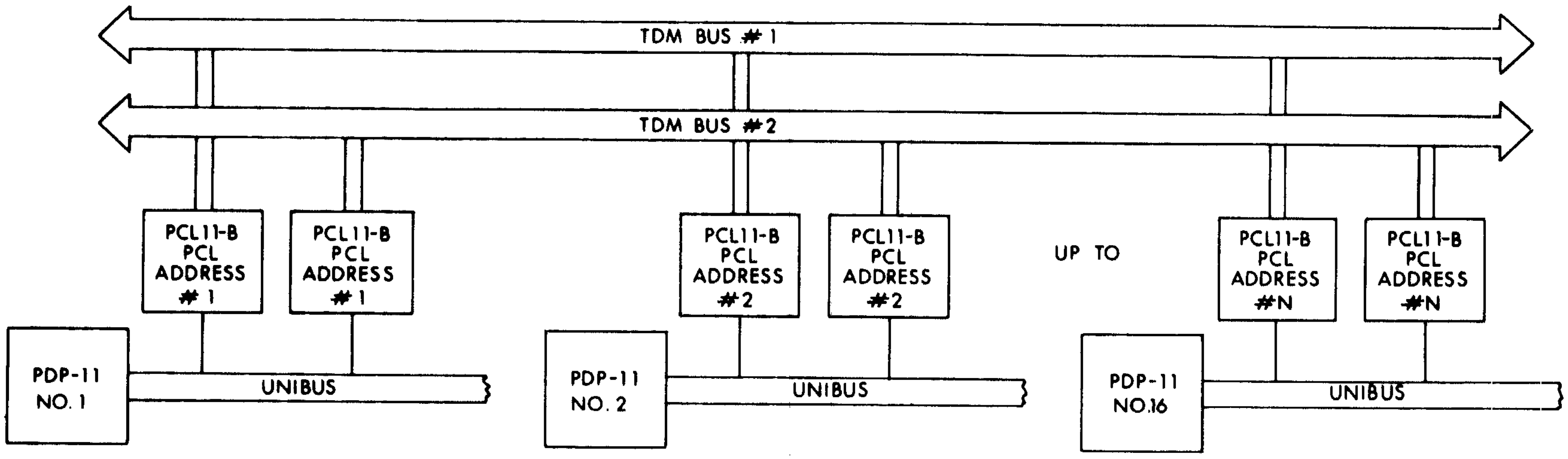


Figure C-2 Dual Bus Communications System Block Diagram

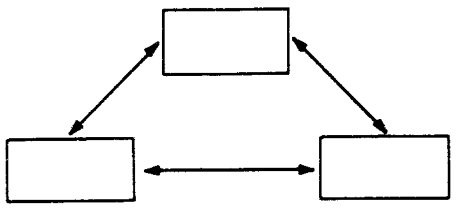


Figure C-3 Point-to-Point Links for Three Processors

works of more than three computers or where the network is starting at a lesser number but future growth is anticipated.

Trained personnel from Digital's Computer Special Systems group work with you to design the PCL11 network and then integrate the system.

LOCAL FUNCTIONALLY DISTRIBUTED NETWORKS

Local functionally distributed networks consist of several computers located at one site, typically in one computer room, and interconnected so as to form one large computer system.

The application tasks in such a network can be split into functional modules and distributed among the various computers in the network; hence the name "Local Functionally Distributed Networks." The computer nodes in the network can be treated as computer modules in the larger system which is applied to the overall application.

A typical example of such a system might be the one shown in Figure C-4. In that example system there are five computers of differing sizes with the general tasks being: two computers as communications front ends, two computers for data base management, and one computer for special processing.

Such distributed networks are not for all applications. Some applications can not be split up among functional modules and can run only as one very large program on a single large computer.

One of the immediate benefits of functionally distributed systems is potential major increases in power and throughput because different portions of the application (different functional modules) are running in different computers simultaneously. Thus, the power of the overall system approaches the power of the individual computers times the number of computers. In fact, the overall power and throughput of a large functionally distributed work would exceed that of even the very large mainframe computers available today.

Flexibility of design is achieved in such systems by the ability to configure the individual CPU nodes to best fit the modular task for that node. For example, in the sample system of Figure C-4,

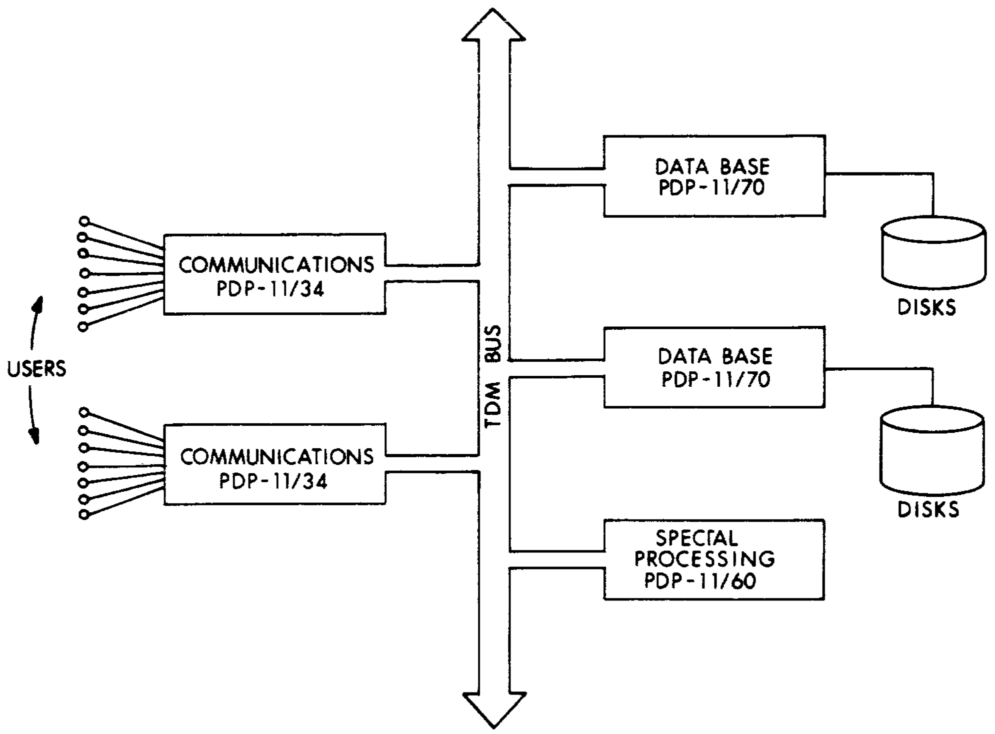


Figure C-4 Example Functionally Distributed System

the communications processors are PDP-11/34s because of their excellent communications handling characteristics and relatively low prices, the data base computers are PDP-11/70s because of the large memory size available and the high speed Massbus I/O structure connected to large disks, and the PDP-11/60 is used for special processing tasks because of the ability to specially program its high speed CPU.

Once the initial system is designed and built, or even while it is being implemented, there is flexibility for growth and expansion as the application requires. As the application grows, the number of CPU nodes in the network or the power of individual nodes can be increased. The end result is that the potential maximum power of the system is not limited by the initial design of the system. The system may be expanded and obsolescence is avoided. A final, and for many applications the major, benefit of such a system design is its high availability.

GENERAL SPECIFICATIONS

Mechanical:

Mounting	PCL11-B interface: 9-slot double system unit. Space required in standard PDP-11 mounting box or in PDP-11 processor box.

Electrical:	Powered from PDP-11 mounting box. 14 amp at +5 V 0.5 amp at −15 V
Operational:	
PCL11 Bus Length	300 foot (91 meter) maximum.
Units	Any number up to 16 maximum may be accommodated on one PCL11 bus.
Transfer Rate	Per Unit: 500K bytes/sec maximum PCL11 bus aggregate bandwidth:

PCL11 bus cable length up to	Bandwidth 8 bit bytes/sec
50 ft. (15m)	1000K
100 ft. (30m)	800K
140 ft. (42m)	666K
240 ft. (73m)	500K
300 ft. (91m)	400K

Message Length	64K bytes maximum
Error Checking	Word parity on each transfer CRC character after each 400_8 bytes
Unibus:	
Loads	1.5 Unibus loads
Mode	NPR DMA Transfers, full duplex, fully silo buffered, 16 bit word mode