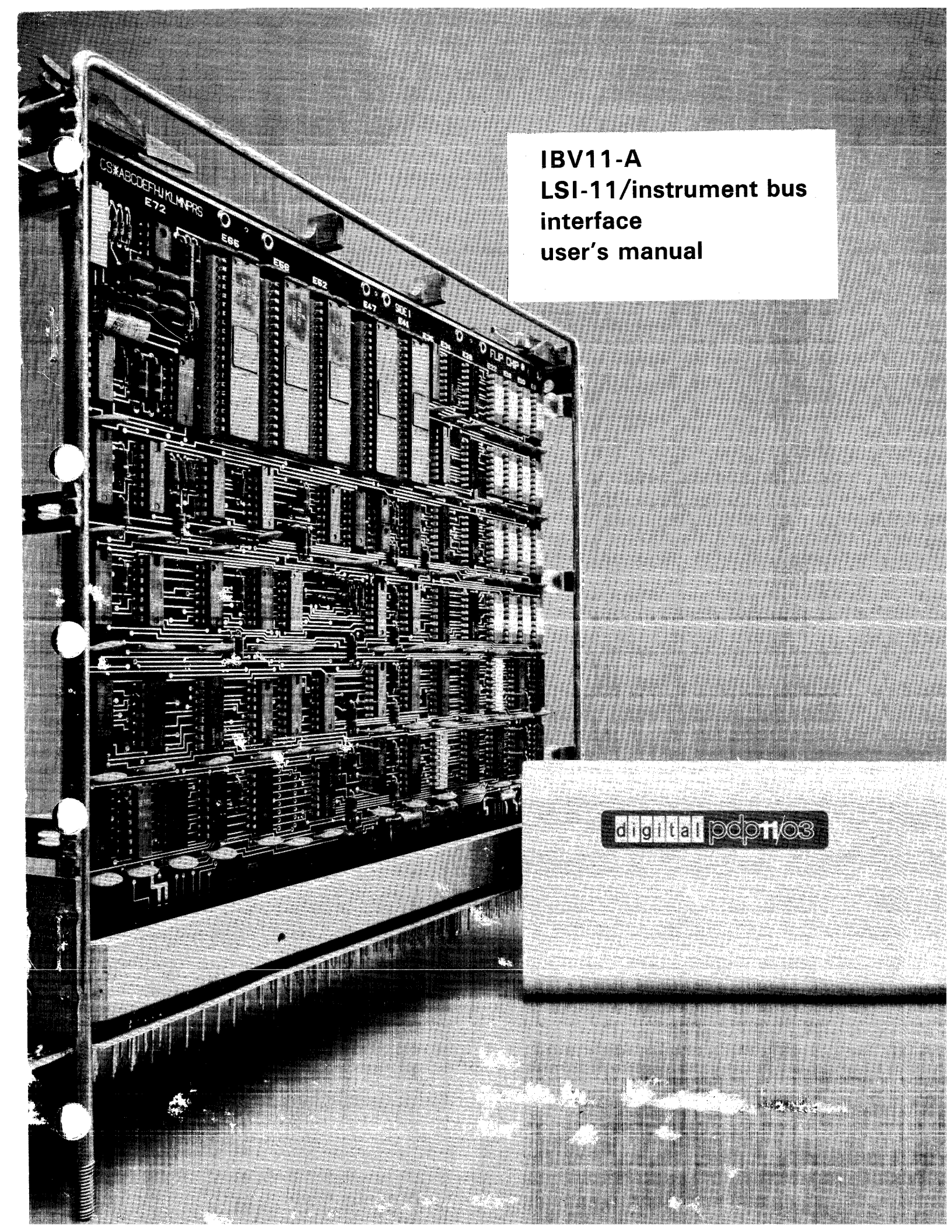

IBV11-A
LSI-11/instrument bus
interface
user's manual
digital pdp11/03

EK-IBV11-UG-001

IBV11-A
LSI-11/instrument bus interface
user's manual

digital equipment corporation • maynard, massachusetts

Preliminary Edition, January 1977
1st Edition, June 1977

Printed in U.S.A.

This document was set on DIGITAL's DECset-8000 computerized typesetting system.

CONTENTS

CONTENTS (CONT)

FIGURES

FIGURES (CONT)

TABLES

PREFACE

The rapid advancement in integrated circuit technology over the past ten years has produced a new generation of complex electronic instrumentation. This generation of instruments is less costly and more reliable and provides measurement accuracy previously available only in standards laboratory environments.

Most of these modern instruments are based on digital logic circuit designs, especially for control and display functions. Recognizing the fact that logic circuits can be readily interconnected and remotely controlled or monitored, the IEEE Standards Board approved IEEE Standard 488-1975, *Digital Interface for Programmable Instrumentation.* That document defines an instrument bus that has become the industry standard.

Basically, an instrument designed with an interface connector and signals that conform with the IEEE standard can communicate over a 16-line bus with other instruments and/or controllers designed to the same standard. Thus, a new family of instruments is emerging that can be interconnected to form "systems" that perform complex functions. Previously, systems were produced only by spending considerable design time and funding for each system; the instruments now conforming to the IEEE standard allow "off the shelf" purchase of instruments and system integration at minimal cost.

High-speed, automatic, programmable instrumentation can now easily be implemented by interfacing a computer to the "instrument bus" that conforms to the IEEE standard. DIGITAL's low-cost, high-performance LSI-11 microcomputer is the ideal solution for instrumentation designs. It provides minicomputer performance based on microprocessor technology. Its instruction set is compatible with most software presently available for the larger PDP-11/40 minicomputer. An instrumentation system designed in this manner is an easy to use, inexpensive, powerful tool for instrument system designers.

The IBV11-A option described in this manual is the instrument bus to LSI-11 bus interface. The information contained in this manual will enable a user to determine which LSI-11 options are required for a specific instrumentation application, to install the IBV11-A in the LSI-11 system, to connect the IBV11-A to instruments, and to program the LSI-11 to communicate with the instrument bus.

CHAPTER 1
INTRODUCTION

1.1 GENERAL

The IBV11-A (Figure 1-1) is an LSI-11 option that interfaces the LSI-11 bus with the instrument bus as described in IEEE Standard 488-1975, *Digital Interface for Programmable Instrumentation.* An IBV11-A can be installed in any basic LSI-11 system configuration.

1.1.1 LSI-11 Systems

Three basic LSI-11 system configurations comprise the LSI-11 family: LSI-11 system components, the PDP-11/03, and the PDP-11V03.

LSI-11 component systems include individual modules, backplane, etc., ordered as separate items. The user purchases only those items required for a specific application.

The PDP-11/03, a boxed version of the LSI-11, is designed for users that need a packaged microcomputer system. It consists of an LSI-11 microcomputer and 4K memory, a modular power supply, and a mounting box. It is easy to use an LSI-11-based microcomputer for system development or dedicated applications.

The PDP-11V03 is the latest addition to the LSI-11 family. It is a mass storage-based system, including the PDP-11/03, the RXV11 floppy disk system, a system cabinet and power distribution panel, either a VT52 DECscope or LA36 DECwriter terminal, RT-11 system software, and system diagnostics.

Refer to DIGITAL's *Microcomputer Handbook* for detailed information on theLSI-11 "family" of systems, including user-assembled component LSI-11 systems, the PDP-11/03 packaged LSI-11 microcomputer system, and the PDP-11V03 floppy disk-based LSI-11 system.

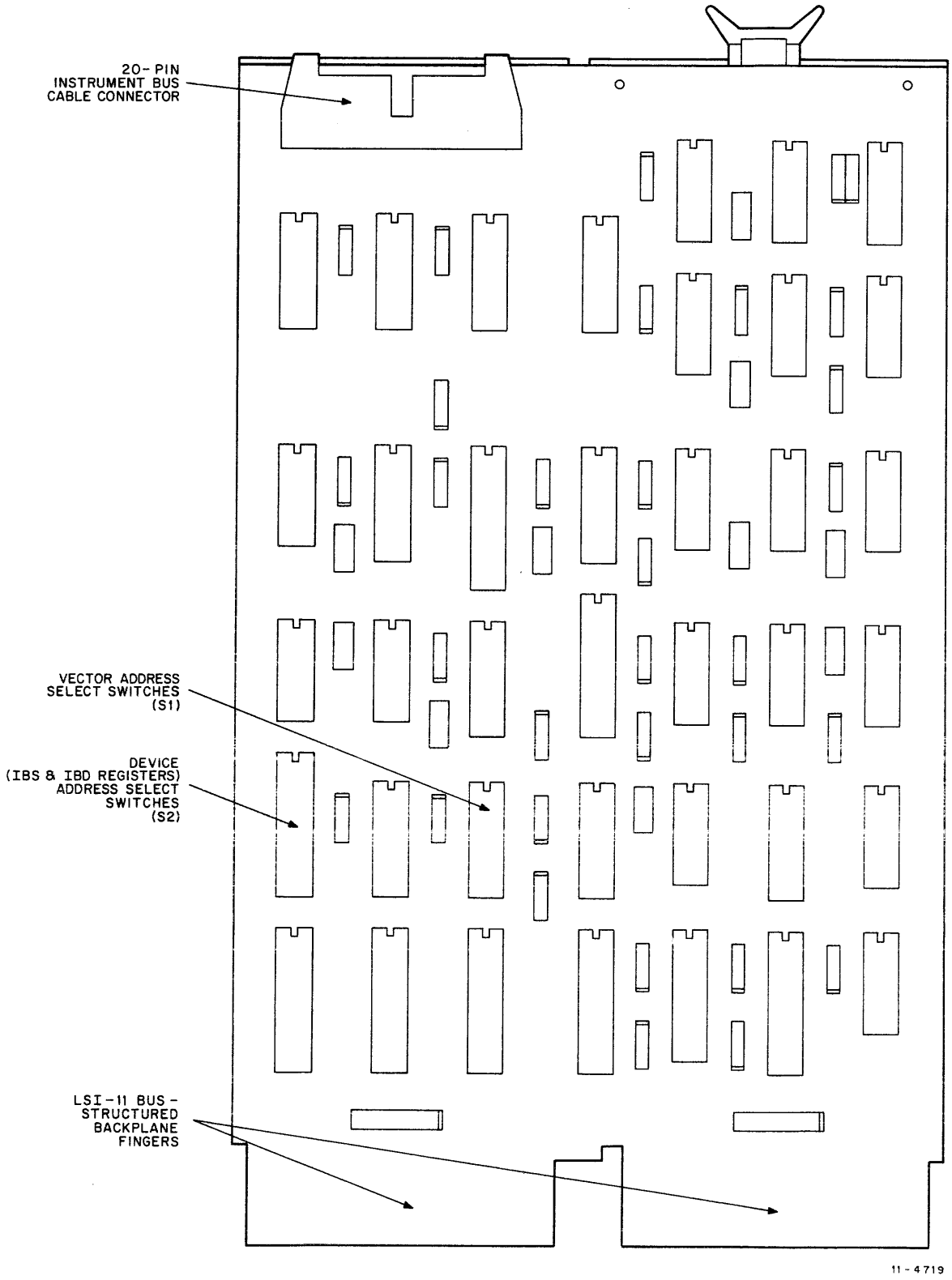


Figure 1-1 IBV11-A Instrument Bus Interface Module

1.1.2 IBV11-A Interface Option

The IBV11-A is contained on a 13.2 cm (5.2 in) by 22.8 cm (8.9 in) printed circuit board assembly; a printed circuit board of this size and shape is also referred to as a "double-height module." Fingers on one end of the module plug directly into any LSI-11 bus-structured system backplane. A connector on the module mates directly with the instrument bus via the BN11A instrument bus cable supplied with the option.

The IBV11-A is the LSI-11 bus/instrument bus interface that makes an LSI-11-based programmable instrument system possible. On the LSI-11 bus side of the interface, the IBV11-A features include:

- PDP-11 software-compatible
- Board-mounted user-configured switches that allow easy device (register address) and interrupt vector address selection
- RT-11/FORTRAN IV* and RT-11/BASIC* (easy to learn and easy to use programming languages that are user-oriented; no need for users to become experts in computer programming)
- System hardware-compatible with any LSI-11 component system, PDP-11/03, and PDP-11V03 systems.

On the instrument bus side of the interface, the IBV11-A features include:

- Instrument bus-compatible with IEEE Standard 488-1975
- Supports cable length up to 20 m (65.6 ft) total
- 15 devices (maximum) can connect to the bus
- Instrument bus-compatible with instrument manufacturers' devices.

1.1.3 Instrument Bus

Before the potential applications for the IBV11-A can be realized, an understanding of the instrument bus functions is required. (The LSI-11 microcomputer's I/O bus is described in the Microcomputer Handbook and is not repeated here.)

The instrument bus is capable of supporting up to 15 devices, including the IBV11-A. The physical structure of the instrument bus, the functional relationships of devices connected to the bus, and the 16 signals that comprise the bus are discussed in the following paragraphs.

Physical Structure

Physically, the instrument bus is composed of the cables and instrument connectors that interconnect a system. The cables are terminated with a standard connector that will mate with any instrument or device conforming to the IEEE standard. Each connector contains a male plug and a female receptacle in one housing (Figure 1-2). This allows the instrument bus user to stack connectors for interconnecting instruments in a "star" or linear arrangement. Both ends of the cable are terminated with identical connectors.

*Planned software options.

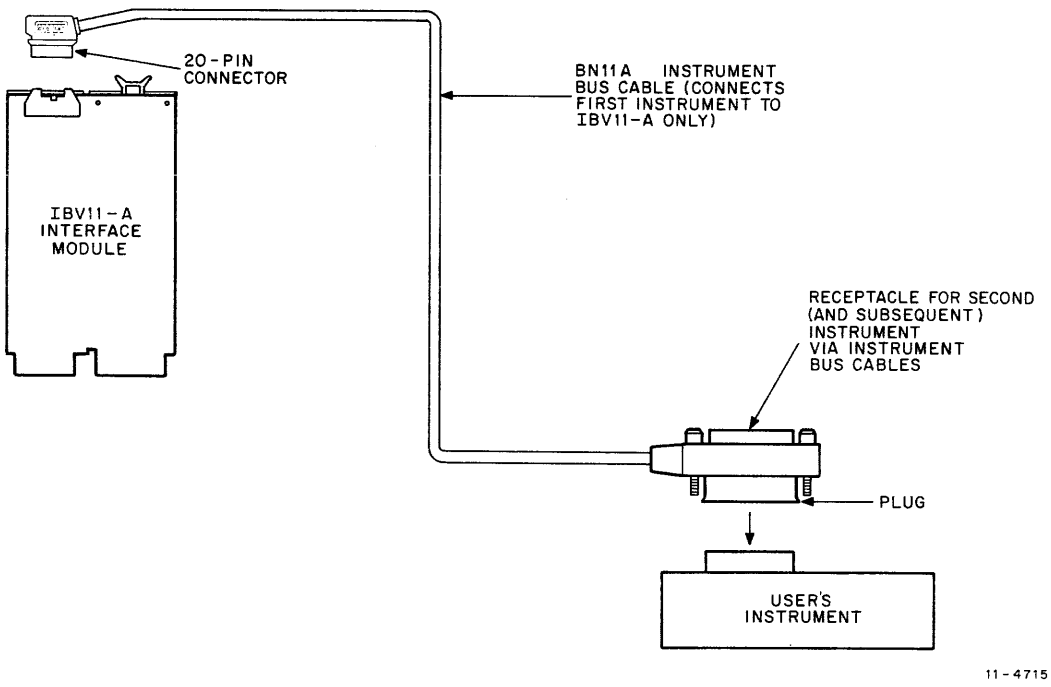


Figure 1-2 IBV11-A Instrument Bus Cable

The instrument bus cable (type BN11A) that connects the IBV11-A to the first instrument on the bus is different, however. One end (the instrument end) has a connector as previously described. The other end (at the IBV11-A) is terminated with a molded, 20-cavity housing that mates with the 20-pin connector on the IBV11-A module.

System Device Functions

System devices that connect to the instrument bus function as either "talkers," "listeners," or "controllers," or a combination of the three functions (Figure 1-3). The function of each in the instrumentation system must be understood before the individual bus signals can be described.

System Controller - The instrument bus always contains one device designated the system controller. The IBV11 usually performs this function. It can control all devices connected to the instrument bus comprising that system.

NOTE

When the IBV11-A is the only system controller, the ER1 switch (S1-8) must be in the OFF position. When the IBV11-A is used in a system that contains another system controller, S1-8 must be in the ON position.

Controller - A controller is capable of controlling talkers and listeners connected to the bus. Only one controller may be active at a time and it is designated the "controller-in-charge." The IBV11-A usually performs this function. All device addressing, device polling, commands, and data byte transfers are controlled by the controller-in-charge.

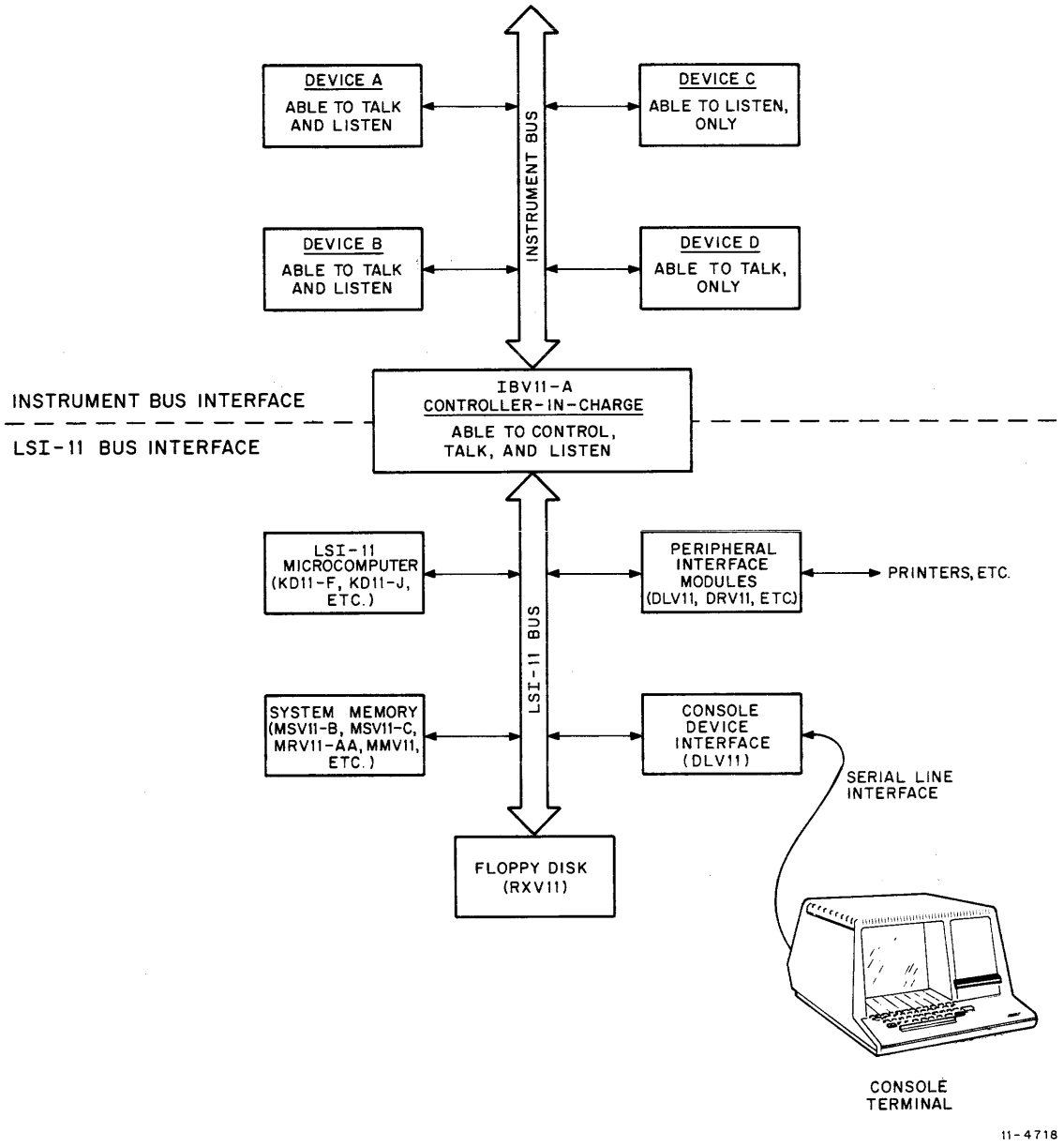


Figure 1-3 Typical System – Functional Relationship

Listener – A listener is a device that is capable of receiving commands and data from the instrument bus. More than one listener may be active at a time.

Talker – A talker is a device that is capable of receiving commands and transmitting data via the instrument bus. Only one talker may be active at a time.

Bus Signal Functions
Sixteen signal lines comprise the instrument bus. Eight lines comprise an 8-bit asynchronous bidirectional data bus. Logical 1s are produced by asserting data lines low (ground). The remaining eight lines provide control functions. Three of these lines are "message handshaking" signals that control data byte transfers over the instrument bus. The remaining five control signal lines provide general interface management functions. A detailed description for each signal is included in Chapter 2.

Controller-In-Charge Commands
Controller-in-charge commands are the IBV11-A to instrument bus byte transfers that control instrument system operation. The IBV11-A becomes active as controller-in-charge by asserting TCS. If any talker is active when the controller-in-charge is to become active, the IBV11-A first inhibits additional byte transfers by asserting NRFD and waits for DAV to become not asserted, indicating that the operation has been completed; it then asserts ATN. Approximately 0.5 μs later, NRFD becomes not asserted, and it can then transmit commands to all devices on the instrument bus.

Commands are transmitted via the DIO<7:1> lines. Commands are coded as 7-bit ASCII characters; the DIO8 line is not used for command transfers. The actual command set conforms to the instrumentation bus standard and is described in the IBV11-A specifications. The actual commands and the sequence in which they are issued are completely under the control of the LSI-11 system software being executed.

1.2 SCOPE
The remaining chapters of this manual contain all of the information normally required for the LSI-11 system user to install, program, and use the IBV11-A option. The manual is organized as follows:

Chapter 1 Introduction – General information and references.

Chapter 2 Specifications – Hardware and software specifications for IBV11-A users. Software specifications are included for programming the LSI-11 processor to communicate with the IBV11. The procedure for programming specific instruments varies depending on the particular instrument; refer to the instrument manufacturer's documentation for programming instructions.

Chapter 3 Installation – Minimum LSI-11 system requirements to support the IBV11-A, procedures for configuring IBV11-A module switches, installation in the LSI-11 backplane, and instrument bus cabling.

Chapter 4 Programming Examples – Sample instrument system application programs are discussed in this chapter.

Chapter 5 Technical Description – Includes a block diagram of the IBV11-A and functional theory of operation.

Chapter 6 Maintenance – Lists available diagnostics.

1.3 REFERENCES
Digital Equipment Corporation publications:

Basic Hardware Manuals
Microcomputer Handbook
PDP-11V03 System Manual

Hardware Option Manuals

RXV11 User's Manual (Floppy Disk System)
DRV11-B General Purpose DMA Interface User's Manual
DRV11-P Foundation Module User's Manual
H780 Power Supply User's Manual
LAV11 User's Manual (LSI-11 bus interface controller for the LA180 DECprinter I)
MSV11-C MOS Read/Write Memory User's Manual
MRV11-BA LSI-11 UV PROM-RAM User's Manual

IEEE Publications:

Digital Interface for Programmable Instrumentation (IEEE Std. 488-1975)

CHAPTER 2
SPECIFICATIONS

2.1 GENERAL

This chapter contains detailed specifications for IBV11-A users, including hardware and programming specifications for the module, and instrument bus specifications. Note that only the instrument bus specifications necessary for using the IBV11-A are included; detailed instrument bus specifications for designers are included in IEEE Standard 488-1975, *Digital Interface for Programmable Instrumentation.* Refer to instrument manufacturer's documentation for the correct procedure for instrument bus addressing, commands, etc. for that particular instrument.

Each IBV11-A option includes one IBV11-A (M7954) module, one BN11A-04 4 m (157.5 in) cable, and user documentation.

2.2 IBV11-A MODULE SPECIFICATIONS

2.2.1 General Specifications

Power Requirements	+5 V ± 5% 0.8 A typical (1.5 A maximum)
Mechanical	
Height	13.2 cm (5.2 in)
Length*	22.8 cm (8.9 in)
Width	1.27 cm (0.5 in)
Environmental	
Temperature	
Storage	–40° to 60° C (–40° to 140° F)
Operating	5° to 50° C (41° to 122° F)
Relative Humidity	10 to 95% (no condensation)

2.2.2 LSI-11 Bus Interface Signals

The IBV11-A conforms to LSI-11 bus specifications stated in the Microcomputer Handbook, Section 1, Chapter 3. Refer to that document for complete LSI-11 bus specifications. Backplane pin utilization is shown in Table 2-1.

*Length as stated is approximate and includes module handle. Actual module length is 21.6 cm (8.5 in).

Electrical

Input Logic Levels

TTL Logical Low	0.8 Vdc max
TTL Logical High	2.0 Vdc min

Output Logic Levels

TTL Logical Low	0.4 Vdc max
TTL Logical High	2.4 Vdc min

Bus Receivers

Logical Low	1.3 Vdc max, -10 μA max at 0 V
Logical High	1.7 Vdc min, 80 μA max at 2.5 V

Bus Drivers

Logical Low	0.8 Vdc max at 70 mA
Logical High	25 μA max at 3.5 V

Table 2-1 IBV11-A Backplane Pin Utilization

Row A		Row B	
Pin	**Signal Mnemonic**	**Pin**	**Signal Mnemonic**
Module Side 1 (Component Side)			
AA1	BSPARE1	BA1	BDCOK H
AB1	BSPARE 2	BB1	BPOK H
AC1	BAD16 L	BC1	SSPARE
AD1	BAD17 L	BD1	SSPARE5
AE1	SSPARE1	BE1	SSPARE6
AF1	SSPARE2	BF1	SSPARE7
AH1	SSPARE3	BH1	SSPARE8
AJ1	GND	BJ1	GND
AK1	MSPAREA	BK1	MSPAREB
AL1	MSPAREA	BL1	MSPAREB
AM1	GND	BM1	GND
AN1	BDMR L	BN1	BSACK L
AP1	BHALT L	BP1	BSPARE6
AR1	BREF L	BR1	BEVNT L
AS1	PSPARE3	BS1	PSPARE4
AT1	GND	BT1	GND
AU1	PSPARE1	BU1	PSPARE2
AV1	+5B	BV1	+5B

Table 2-1 IBV11-A Backplane Pin Utilization (Cont)

Row A		Row B	
Pin	Signal Mnemonic	Pin	Signal Mnemonic
Module Side 2 (Solder Side)			
AA2	+5	BA2	+5
AB2	−12	BB2	−12
AC2	GND	BC2	GND
AD2	+12	BD2	+12
AE2	BDOUT L	BE2	BDAL2 L
AF2	BRPLY L	BF2	BDAL3 L
AH2	BDIN L	BH2	BDAL4 L
AJ2	BSYNC L	BJ2	BDAL5 L
AK2	BWTBT L	BK2	BDAL6 L
AL2	BIRQ L	BL2	BDAL7 L
AM2	BIAKI L	BM2	BDAL8 L
AN2	BIAKO L	BN2	BDAL9 L
AP2	BBS7 L	BP2	BDAL10 L
AR2	BDMGI L	BR2	BDAL 11 L
AS2	BDMGO L	BS2	BDAL12 L
AT2	BINIT L	BT2	BDAL13 L
AU2	BDAL0 L	BU2	BDAL14 L
AV2	BDAL1 L	BV2	BDAL15 L

2.2.3 Instrument Bus Interface

Nominal Logic Levels

0 = +2.0 V minimum
1 = +0.8 V maximum

Bus Drivers and Receivers

Driver low state output voltage	+0.4 V (maximum) at +48 mA sink current
Driver high state output voltage	+2.6 V (minimum)
Receiver low state output voltage	+0.4 V (maximum)
Receiver high state output voltage	+2.4 V (minimum)
Receiver input hysteresis	400 mV (minimum) (typically, 900 mV)

Signal Termination

Each instrument bus signal line is terminated within the device by a resistive load (3.0K pull-up to V_{cc}, 6.2K to ground) to establish a steady-state voltage when all drivers on a line are in the high impedance state.

Electrical Length of Instrument Bus Transmission
The length of the electrical transmission path over the bus must be within the following specifications:

Number of Bus Segments	Bus Segments (m)	Maximum Number of Devices	Transmission Path (m)
5	4	6	20
10	2	11	20
14	1	15	14

Instrument Bus Data Rate
The maximum data transfer rate over the instrument bus is 250K bytes per second (4 μs/byte) provided that all electrical specifications stated herein are met.

Software Data Rate
The maximum estimated data rate for a tight, dedicated routine to move data from the instrument bus to the LSI-11 memory is 40 kilobytes/second. The maximum estimated data rate for general-purpose and high-level language routines is 5 kilobytes/second.

Instrument Bus Cable Connector Pinning
Instrument bus cable connector pins are identified in Figure 2-1. Pin utilization and signal names for the IBV11-A end and user's instrument end of the cable are identified in Table 2-2.

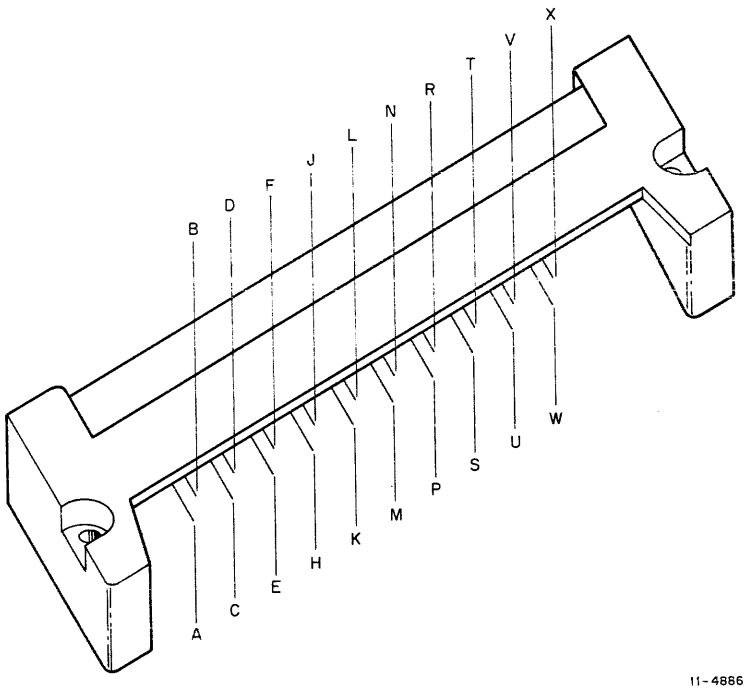


Figure 2-1 IBV11-A Instrument Bus Connector

Table 2-2 BN11A IBV11 to Instrument Bus Cable - Connector Pinning

Connector Pins on IBV11-A Module	Instrument Bus Signal Name	Instrument Bus Cable Connector Pins (Instrument End)
U	DIO1	1
S	DIO2	2
P	DIO3	3
M	DIO4	4
R	EOI	5
T	DAV	6
V	RFD	7
X	DAC	8
B	IFC	9
J	SRQ	10
F	ATN	11
W	SHIELD	12
K	DIO5	13
H	DIO6	14
E	DIO7	15
C	DIO8	16
D	REN	17
N	GND (6)	18
N	GND (7)	19
N	GND (8)	20
A	GND (9)	21
L	GND (10)	22
L	GND (11)	23
W	GND (logic)	24

NOTE

GND(n) (where n = 6, 7, 8, 9, 10, 11) is the ground return for the signal on pin "N" of the instrument bus connector. For example, GND (6), pin 18, is the ground return for the signal DAV on pin 6. These ground returns are returned to the logic ground of their respective drivers and receivers on the IBV11-A module.

Signal Functions

Sixteen signal lines comprise the instrument bus as shown in Figure 2-2. Each signal line is partially terminated by each device connected to the bus. Eight lines (DIO<1:8> L) comprise an 8-bit asynchronous bidirectional data bus. Logical 1s are produced by asserting data lines low (ground). The remaining eight lines provide control functions, as described in the following paragraphs.

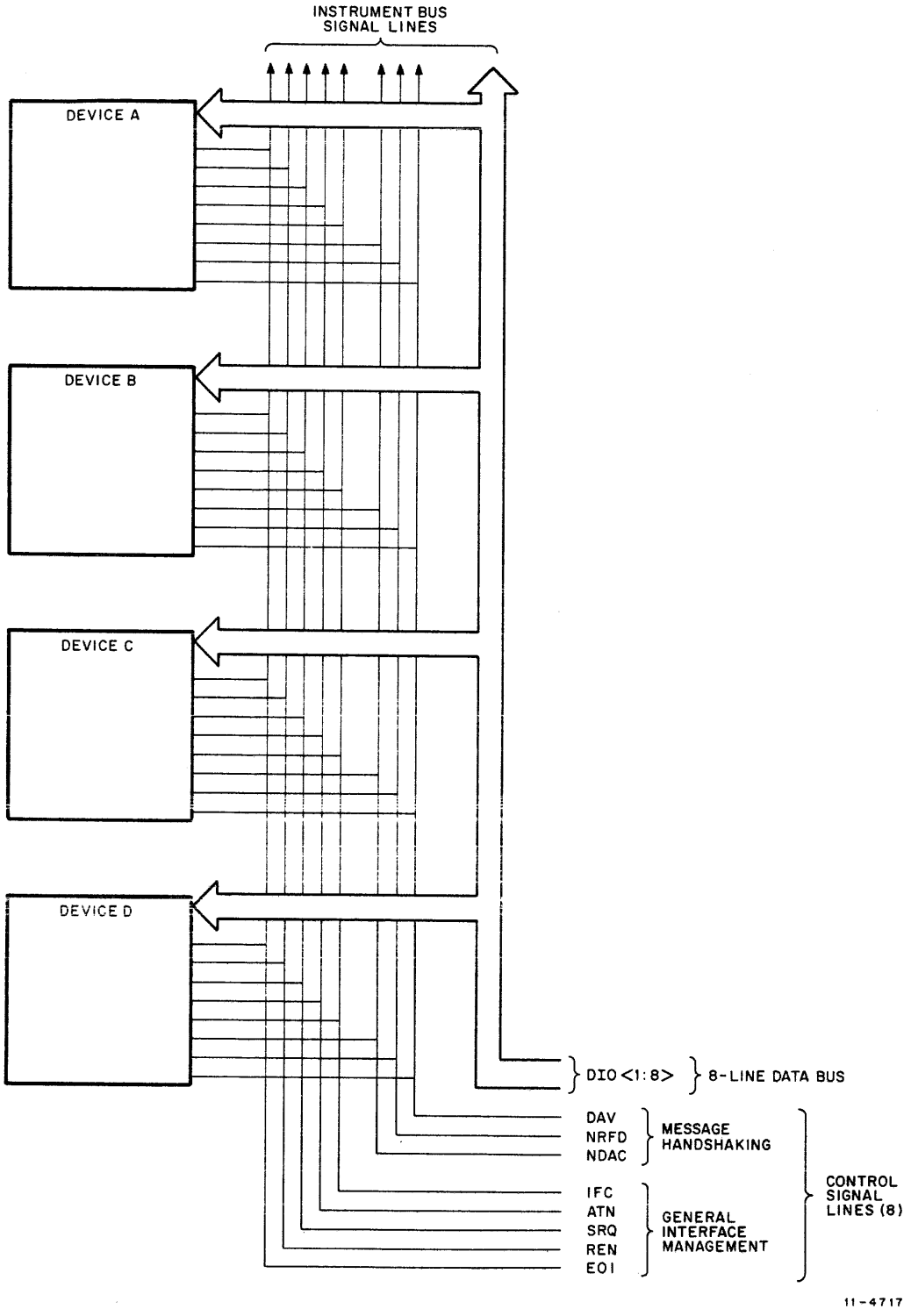


Figure 2-2 Instrument Bus Signal Lines

Three "message handshaking" control signals control data byte transfers over the 8-line data bus. They provide the necessary handshaking to complete interlocked, asynchronous data transfers. These control signals are described below.

Data Valid (DAV) – DAV is asserted by the active talker or active controller to indicate that it has placed valid data on the DIO signal lines. DAV is asserted only after all listener devices are ready for data and valid data has been placed on the DIO lines for 2 μs (minimum).

Not Ready for Data (NRFD) – NRFD enables an active listener to indicate that it is "busy," or not ready for data. All active listeners must be "ready" to not assert NRFD, enabling a talker to transmit a data byte.

Not Data Accepted (NDAC) – All *active* listeners assert NDAC. All inactive listeners do not assert NDAC. The active listener indicates that it has accepted data during a data transfer by not asserting NDAC. When more than one active listener is on the instrument bus, the last active listener to not assert NDAC indicates that all devices have accepted the data.

The remaining five control signals are for general interface management. Each signal is described below.

Interface Clear (IFC) – IFC is a "master clear" for all devices that connect to the interface bus. IFC is asserted by the IBV11-A only when it is an active system controller. All devices respond to the active IFC signal by returning to the idle state within 100 μs

NOTE

When the IBV11-A is the only system controller, the ER1 switch (S1-8) must be in the OFF position. When the IBV11-A is used in a system that contains another system controller, S1-8 must be in the ON position.

Remote Enable (REN) – Typical devices (instruments) connected to the instrument bus are capable of local (device control panel) or remote control operation via the instrument bus. The IBV11-A, as system controller, asserts REN to enable *remote* operation, or negates REN to allow local operation of all devices. When REN is asserted, the IBV11-A may return selected devices to the local mode by addressing the devices and issuing a "go to local" command. When in the local mode, devices may respond to control and data transmissions over the instrument bus that do not conflict with local control functions.

Attention (ATN) – The IBV11-A, as controller-in-charge, asserts the ATN signal line when it is transmitting commands over the DIO lines. An active ATN signal causes any previously active talker to become inactive and all devices may receive commands from the IBV11-A.

End or Identify (EOI) – EOI can be asserted either by the IBV11-A, as controller-in-charge, or by an active talker. A talker asserts EOI to indicate the last byte of a message is being transmitted. The IBV11-A can assert EOI while also asserting ATN to conduct a parallel poll.

Service Request (SRQ) – The IBV11-A monitors the SRQ line for service requests from devices connected to the instrument bus. The IBV11-A's control/status register (CSR) can be programmed to cause IBV11-A-generated interrupts to occur whenever the SRQ line is asserted. Thus, interrupt-driven routines can be included in system software that will automatically service devices on the instrument bus.

2.2.4 Programming

The IBV11-A communicates with devices connected to the instrument bus under the control of the LSI-11 program being executed. All communication between the LSI-11 and the IBV11-A is via the instrument bus status (IBS) and instrument bus data (IBD) registers. The programmer must be aware of the functional significance of each bit in both registers before any programs can be written that will control specific devices on the instrument bus. In addition, the programmer must establish instrument (device) addresses and conform to programming rules specified for each instrument connected to the instrument bus.

IBS and IBD Registers

The IBS register provides the means for controlling the instrument bus signals and IBV11-A functions relative to the LSI-11 bus. The low byte of the IBD register, on the other hand, is used for passing commands to devices connected to the bus and for transmitting and receiving data between the LSI-11 processor and talker and listener devices. In addition, the high byte of the IBD register allows for LSI-11 processor monitoring of all instrument bus signal (data and control) lines. IBS and IBD registers are shown in Figure 2-3 and described in Tables 2-3 and 2-4.

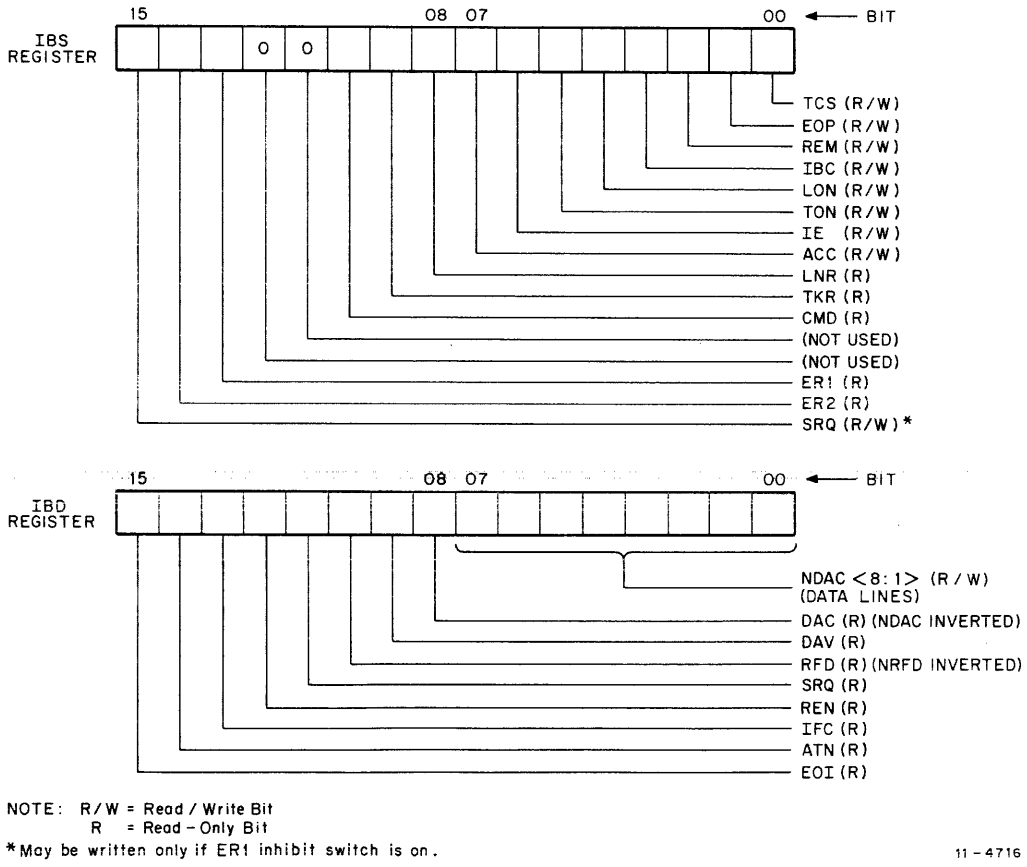


Figure 2-3 IBV11-A Register and Bit Assignments

Table 2-3 IBS Register Bits Descriptions

Bit	Function
00	Take Control Synchronously (TCS). Set and cleared under program control to enable or disable the IBV11-A controller-in-charge function by taking control synchronously or by negating ATN. Setting TCS will cause NRFD to be asserted for at least 500 ns before DAV is checked. ATN is then asserted when DAV is not asserted. NRFD must be unasserted and CMD is set 500 ns (minimum) after ATN is asserted. TCS is cleared by BINIT L and IFC.
01	End or Poll (EOP). Set and cleared under program control to assert or unassert the EOI line. EOP is cleared by BINIT L and IFC.
02	Remote On (REM). Set and cleared under program control to assert or unassert the REN line. REM is cleared by BINIT L and IFC.
03	Interface Bus Clear (IBC). When set, the leading edge of IBC produces IFC for 125 μs (approximately). At the end of IFC, TCS is automatically asserted and IBC is automatically cleared. IBC is cleared by BINIT L.
04	Listener On (LON). Set or cleared by the program to enable or disable the IBV11-A listener function. When LON is set and the DAV line is asserted, the IBS LNR bit (bit 08) becomes set. When LON is cleared, the IBV11-A ignores DAV LON is cleared by BINIT L and IFC.
05	Talker On (TON). Set or cleared by the program to enable or disable the IBV11-A talker function. TON is cleared by BINIT L and IFC.
06	Interrupt Enable (IE). Set and cleared by the program to enable or disable IBV11-A interrupts. IE is cleared by BINIT L.
07	Accept Data (ACC). Set and cleared by the program. When ACC is cleared, reading a data byte from the DIO lines will automatically assert the DAC line and clear the LNR bit (bit 08). When ACC is set, the program must clear the low byte of the IBD register in order to clear the LNR status bit and assert the DAC line. ACC is cleared by BINIT L and IFC when LON, TON, and TCS are all off; ACC may be set to assert NRFD.
08	Listener Ready (LNR). When set, LNR indicates that the IBV11-A has a data or command byte that is ready for reading from the low byte of the IBD. LNR is set when LON is set and the DAV line becomes asserted. LNR is cleared by reading the IBD low byte if ACC is cleared, or by clearing the IBD low byte if ACC is set. LNR is also cleared when LON is cleared by the program and by BINIT L and IFC.
09	Talker Ready (TKR). When set, TKR indicates to the LSI-11 processor that the IBV11-A is ready for the next data byte to be transmitted to the DIO lines via the low byte of the IBD register. TKR is set when TON is set, TCS is cleared, and when listeners are ready for data. TKR is cleared by BINIT L or IFC, by writing a command or data byte into the low byte of the IBD register, or by the program clearing TON or setting TCS.

Table 2-3 IBS Register Bit Descriptions (Cont)

Bit	Function
10	Command Done (CMD). When set, CMD indicates to the LSI-11 processor that the IBV11-A is ready for the next command byte to be transmitted to the DIO lines via the low byte of the IBD register. CMD is set by a successful TCS to indicate that the ATN line was asserted and that the next command byte may be transmitted over the instrument bus. CMD is also set when the DAC line is asserted after the command has been accepted by the addressed device on the instrument bus. CMD is cleared by BINIT L or IFC, by writing a byte (command) into the low byte of the IBD register, or by the program clearing the TCS bit.
11	Not used – read as 0.
12	Not used – read as 0.
13	Error 1 (ER1). Set whenever a conflict occurs between the instrument bus ATN, IFC, or REN lines and their IBV11-A control hardware. When set, ATN H is cleared and cannot be set. This condition can only be cleared by clearing the cause of the error. ER1 can occur when another system controller is connected to the instrument bus. The error can then be suppressed by setting the ER1 inhibit switch (S1-8) on the IBV11-A module to the ON position. If the IBV11-A is the only system controller, set S1-8 to the OFF position.
14	Error 2 (ER2). Set when the IBV11-A tries to send a data or command byte while there is no active listener or command acceptor on the instrument bus. ER2 is cleared by clearing both the TON and TCS bits.
15	Service Request (SRQ). This bit always indicates the status of the instrument bus SRQ line. It may be written (set and cleared) if the ER1 inhibit switch is set.

Table 2-4 IBD Register Bit Descriptions

Bit	Function
<15:8>	Instrument bus control line status. The program can monitor the signal status of all eight control signals by reading this byte. Note that DAC (bit 08) and RFD (bit 10) are inverted with respect to the actual instrument bus signal lines.
<7:0>	Instrument bus data input/output. The program can read or write via this register byte to receive or transmit command or data bytes over the instrument bus. Bits <7:0> correspond to DIO lines <8:1>.

Interrupts

The IBV11-A is capable of generating four separate interrupt requests; each has separate interrupt vectors and normally would have separate service routines. Interrupts can be requested only when the IBS interrupt enable (IE) bit is set. Interrupt requests are priority structured in the IBV11-A. A summary of the four interrupt types is provided in Table 2-5.

Table 2-5 IBV11-A Interrupt Vectors

Priority	Vector	Associated IBS Bit	Cause of Interrupt
Highest	000XNN00	ER2, ER1	Error condition.
Second highest	000XNN04	SRQ	A device connected to the instrument bus is requesting service.
Third highest	000XNN10	TKR, CMD	The IBV11-A is an active talker and it is ready for the LSI-11 processor to output a byte to the low byte of the IBD register. [The IBV11-A will normally then transmit the byte over the instrument bus to the active listener(s).]
Lowest	000XNN14	LNR	The IBV11-A is an active listener and has a data byte to be read by the LSI-11 processor.

Notes

1. **x = User-configured vector address octal digit.**
2. **N = User-configured vector address binary bits.**
3. **Associated IBS bits shown, when set, produce interrupt requests if the IE bit is set.**

2.3 INSTRUMENT BUS CONTROLLER-IN-CHARGE COMMANDS

2.3.1 General

Controller-in-charge commands may only be issued by an active controller-in-charge. When the system controller asserts the IFC control line, it designates a specific controller as the controller-in-charge. A controller may also become the controller-in-charge when the current controller-in-charge specifically transfers control to it with the "take control" command. The IBV11-A will detect a successful transfer of control as an ER1 condition (if S1-8 is in the OFF position) because it normally is the controller-in-charge as well as the system controller; ER1 can be avoided (when another device becomes controller-in-charge) by setting S1-8 on the IBV11-A module to the ON position.

A controller-in-charge becomes active 2 μs after asserting the ATN control line. First, the controller-in-charge stops further data byte transfers between the active talker and active listeners by asserting the NRFD line and then waits for the talker to unassert the DAV line. When DAV is unasserted, the controller-in-charge asserts ATN to declare its control of the DIO message lines. The controller-in-charge waits for 0.5 μs after asserting ATN before unasserting the NRFD line so that the talker and listeners will have enough time to recognize and respond to the asserted ATN line.

While active, the controller-in-charge uses the DIO lines to issue command bytes to all devices on the instrument bus. All controller-in-charge commands are coded into a 7-bit command byte on DIO<7:1>. The DIO8 line remains unasserted, and is not decoded by devices when a command byte is received. The command byte MSB is on DIO7; the LSB is on DIO1. Command bytes are restricted to 7 bits so that their coding may be correlated to the ISO 7-bit code (or the equivalent code in the American National Standard Code for Information Interchange, ANSI X3.4-1968) because it is convenient to both generate and interpret this code.

Command decoding in the IBV11-A, if desired, must be handled by software.

2.3.2 Addressed Command Group (ACG)

Commands in this group (Table 2-6) affect only the currently addressed talker or the currently addressed list of listeners.

Table 2-6 Addressed Command Group

Command Mnemonic	ASCII Character	ASCII Code	Keyboard Function	Devices Affected	Command Function
GTL	SOH	001	CTRL A	Listeners	Go to Local
SDC	EOT	004	CTRL D	Listeners	Selected Device Clear
PPC	ENQ	005	CTRL E	Listeners	Parallel Poll Configure
GET	BS	010	CTRL H	Listeners	Group Execute Trigger
TCT	TAB	011	CTRL I	Talker	Take Control

Go to Local (GTL) - The GTL command causes addressed listeners to go from the remote mode to the local mode. When in the local mode, a device is controlled by its front and rear panel controls.

Selected Device Clear (SDC) - The SDC command causes addressed listeners to be cleared (initialized).

Parallel Poll Configure (PPC) - The PPC command causes addressed listeners to enter the parallel poll configure mode. The next command must be from the secondary command group; otherwise, the listeners will exit the parallel poll configure mode. While in parallel poll configure mode, the listeners will interpret all secondary command group commands as 1 of the 16 possible Parallel Poll Enable (PPE) commands or as the PPD (Parallel Poll Disable) command.

Group Execute Trigger (GET) - The GET command causes the addressed listeners to start the basic operation of the device that the listener is a part of (only if that operation is at rest when this command is received).

Take Control (TCT) - The TCT command causes the addressed talker of a controller to enable the controller to become the controller-in-charge as soon as the current controller-in-charge unasserts ATN. Controller-in-charge status can be transferred in an orderly manner from one controller to another by the current controller-in-charge addressing the talker of the next controller-in-charge, and then issuing the TCT command followed by unasserting ATN.

2.3.3 Universal Command Group (UCG)

Commands in this group (Table 2-7) affect all devices able to respond without having to be previously addressed. The IEEE standard permits the first three of these commands to be issued while the system controller is asserting IFC. However, the ability to issue commands during IFC is not permitted in the IBV11-A.

Local Lockout (LLO) – The LLO command causes all instruments to ignore their local "return to local" signal. This command can be countermanded only by powering the instrument off and then on, or by the system controller unasserting the REN control line.

Device Clear (DCL) – The DCL command causes all devices on the instrument bus to be cleared (initialized).

Parallel Poll Unconfigure (PPU) – The PPU command causes all parallel poll configurations to be cleared (unconfigured). Only devices that have been parallel poll configured will respond to a "parallel poll" request. After the PPU command, no device will respond to a "parallel poll" request.

Serial Poll Enable (SPE) – The SPE command causes all talkers to enter the serial poll mode. When in serial poll mode, an addressed talker will respond to the unassertion of the ATN control line by sending its device's status byte to the controller-in-charge. When the controller-in-charge asserts ATN after a talker has responded with a status byte, bit 7 of the status byte will be cleared if it was set. This will in turn cause the talker to unassert its SRQ line driver if it was asserted.

Serial Poll Disable (SPD) – The SPD command causes all talkers to exit the serial poll mode and return to the normal data mode where an active talker sends data bytes rather than a single status byte.

Table 2-7 Universal Command Group

Command Mnemonic	ASCII Character	ASCII Code	Keyboard Function	Command Function
LLO	DC1	021	CTRL Q	Local Lockout
DCL	DC4	024	CTRL T	Device Clear
PPU	NAK	025	CTRL U	Parallel Poll Unconfigure
SPE	CAN	030	CTRL X	Serial Poll Enable
SPD	EM	031	CTRL Y	Serial Poll Disable

2.3.4 Listener Address Group (LAG)

Commands in this group (Table 2-8) are used to address one or more listeners at a time or to unaddress all listeners at once. Listener addresses specified by these commands are called primary listener addresses because they can be followed by a secondary address from the secondary command group to address an extended listener. Addressed listeners become active listeners when the controller-in-charge unasserts ATN.

Primary and secondary listener address decoding in the IBV11-A, if desired, must be handled by software.

My Listen Address (MLAn) – The MLA command covers 31 primary listener addresses. Any listener that recognizes its own address becomes an addressed listener. Only addressed listeners are able to receive data bytes from a talker.

Unlisten (UNL) – The UNL command causes all listeners to become unaddressed.

Table 2-8 Listener Address Group Commands

Command Mnemonic	ASCII Character	ASCII Code	Keyboard Function	Command Function
MLA00	SP	040	Space	My Listen Address 0
MLA01	!	041	!	1
MLA02	"	042	"	2
MLA03	#	043	#	3
MLA04	$	044	$	4
MLA05	%	045	%	5
MLA06	&	046	&	6
MLA07	'	047	'	7
MLA08	(	050	(	8
MLA09	)	051	)	9
MLA10	*	052	*	10
MLA11	+	053	+	11
MLA12	,	054	,	12
MLA13	-	055	-	13
MLA14	.	056	.	14
MLA15	/	057	/	15
MLA16	0	060	0	16
MLA17	1	061	1	17
MLA18	2	062	2	18
MLA19	3	063	3	19
MLA20	4	064	4	20
MLA21	5	065	5	21
MLA22	6	066	6	22
MLA23	7	067	7	23
MLA24	8	070	8	24
MLA25	9	071	9	25
MLA26	:	072	:	26
MLA27	;	073	;	27
MLA28	<	074	<	28
MLA29	=	075	=	29
MLA30	>	076	>	My Listen Address 30
UNL	?	077	?	Unlisten

2.3.5 Talker Address Group (TAG)

Commands in this group (Table 2-9) are used to address or unaddress one talker at a time. Talker addresses in this group are called primary talker addresses because they can be followed by a secondary address from the secondary command group to address an extended talker. An addressed talker becomes active when the controller-in-charge unasserts ATN.

Primary and secondary talker address decoding in the IBV11-A, if desired, must be handled by software.

My Talk Address (MTAn) – The MTA command covers 31 primary talker addresses. Any talker that recognizes its own address becomes addressed while all other talkers become unaddressed. Only an addressed talker is able to send data bytes.

Untalk (UNT) – The UNT command causes the addressed talker to become unaddressed without addressing another talker.

Table 2-9 Talker Address Group Commands

Command Mnemonic	ASCII Character	ASCII Code	Keyboard Function	Command Function
MTA00	@	100	@	My Talk Address 0
MTA01	A	101	A	1
MTA02	B	102	B	2
MTA03	C	103	C	3
MTA04	D	104	D	4
MTA05	E	105	E	5
MTA06	F	106	F	6
MTA07	G	107	G	7
MTA08	H	110	H	8
MTA09	I	111	I	9
MTA10	J	112	J	10
MTA11	K	113	K	11
MTA12	L	114	L	12
MTA13	M	115	M	13
MTA14	N	116	N	14
MTA15	O	117	O	15
MTA16	P	120	P	16
MTA17	Q	121	Q	17
MTA18	R	122	R	18
MTA19	S	123	S	19
MTA20	T	124	T	20
MTA21	U	125	U	21
MTA22	V	126	V	22
MTA23	W	127	W	23
MTA24	X	130	X	24
MTA25	Y	131	Y	25
MTA26	Z	132	Z	26
MTA27	[	133	[	27
MTA28	\	134	\	28
MTA29	] or ↑	135	] or ↑	29
MTA30	^	136	^	My Talk Address 30
UNT	– or ←	137	– or ←	Untalk

2.3.6 Secondary Command Group (SCG)

The commands in this group are used to modify the parallel poll configure (PPC) command or to specify an extended talker or listener address. This group consists of 32 codes starting with the ASCII code for "grave." However, the last code, equivalent to the ASCII code for delete, is not used.

Parallel Poll Enable/Disable Commands (PPE/PPD) – Each parallel poll enable (PPE) command (Table 2-10) must follow a parallel poll configure (PPC) command, which puts currently addressed listeners into parallel poll configure mode. Usually, only one listener is addressed before a PPC command is issued so that the PPE command will affect only one device at a time. However, more than one device may be affected by a single PPE command when this is desirable. The PPE command is used to dynamically instruct (program) some devices as to how to respond to a parallel poll request. The response to a parallel poll request in some devices is determined by hardware that is field-settable during system configuration. Some devices, including the IBV11-A, are unable to respond to a parallel poll request.

A device responds to a parallel poll request by returning one bit of status to the controller-in-charge on one of the eight DIO lines.

The second digit of the PPE command mnemonic specifies the DIO line to use. The first digit of the PPE command mnemonic specifies which logical state of the status bit should cause the DIO line to be asserted low. Thus, the PPE15 form of the PPE command instructs the device to respond to a parallel poll request by asserting DIO line 5 if the status bit is a logical 1.

Table 2-10 Parallel Poll Enable/Disable Commands

Command Mnemonic	ASCII Character	ASCII Code	Keyboard Function	Command Function
PPE01	`	140	`	Parallel Poll Enable 1
PPE02	a	141	a	2
PPE03	b	142	b	3
PPE04	c	143	c	4
PPE05	d	144	d	5
PPE06	e	145	e	6
PPE07	f	146	f	7
PPE08	g	147	g	8
PPE11	h	150	h	11
PPE12	i	151	i	12
PPE13	j	152	j	13
PPE14	k	153	k	14
PPE15	l	154	l	15
PPE16	m	155	m	16
PPE17	n	156	n	17
PPE18	o	157	o	Parallel Poll Enable 18
PPD	p	160	p	Parallel Poll Disable
(Not used)	q	161	q	
↓	↓	↓	↓	
(Not used)	DEL	177	DEL	

The process of selecting a device by addressing a listener, commanding it to enter the parallel poll configure mode, and instructing it how to respond to a parallel poll request is called configuring a parallel poll response. Once configured, a device remembers how to respond to the parallel poll request until it is reconfigured by another PPE or unconfigured by a parallel poll disable (PPD) or a parallel poll unconfigure (PPU) command.

A parallel poll request is issued by the controller-in-charge by asserting the EOI control line along with the ATN control line.

The PPD command, like all secondary commands, must follow its associated primary command. The PPD command selectively disables (unconfigures) devices from responding to the parallel poll request. The particular devices affected by the PPD command are those devices that have active listeners that respond to the preceding PPC command.

My Secondary Address Commands (MSAn) – An MSA command, when used, must follow an MLA or MTA primary address command. The 31 MSA commands are denoted by the mnemonics MSA00 through MSA30 with codes equivalent to the ASCII codes from "grave" to "tilde." A secondary address is used to specify an extended talker or listener address. Devices that use extended addressing do not become addressed until the proper secondary address follows the proper primary address. "My secondary address" commands are listed in Table 2-11.

Table 2-11 My Secondary Address Commands

Command Mnemonic	ASCII Character	ASCII Code	Keyboard Function	Command Function
MSA00	`	140	`	My Secondary Address 0
MSA01	a	141	a	1
MSA02	b	142	b	2
MSA03	c	143	c	3
MSA04	d	144	d	4
MSA05	e	145	e	5
MSA06	f	146	f	6
MSA07	g	147	g	7
MSA08	h	150	h	8
MSA09	i	151	i	9
MSA10	j	152	j	10
MSA11	k	153	k	11
MSA12	l	154	l	12
MSA13	m	155	m	13
MSA14	n	156	n	14
MSA15	o	157	o	15
MSA16	p	160	p	16
MSA17	q	161	q	17
MSA18	r	162	r	18
MSA19	s	163	s	19
MSA20	t	164	t	20
MSA21	u	165	u	21
MSA22	v	166	v	22
MSA23	w	167	w	23
MSA24	x	170	x	24
MSA25	y	171	y	25
MSA26	z	172	z	26
MSA27	{	173	{	27
MSA28	|	174	|	28
MSA29	}	175	}	29
MSA30	~	176	~	My Secondary Address 30
Not used	DEL	177	DEL	

CHAPTER 3
INSTALLATION

3.1 MINIMUM SYSTEM REQUIREMENTS

3.1.1 System Hardware

System hardware can range from very simple LSI-11 execute-only systems through dual floppy disk systems running under the RT-11 operating system software. LSI-11 systems fall into three general categories: LSI-11 component systems, PDP-11/03 systems, and PDP-11V03 systems. LSI-11 component systems and PDP-11/03 (packaged LSI-11) systems are configured by the user, ranging from a basic 4K memory and LSI-11 processor upward to a more complex configuration. Expansion beyond the basic system requires the addition of various standard LSI-11 options, including system memory, serial and parallel interfaces for peripherals, floppy disk, etc. All PDP-11V03 systems, however, include 8K memory, a console terminal (VT52 or LA36), and RT-11 operating system software as part of the basic system. The basic PDP-11V03 is factory-configured and shipped as a ready-to-use system capable of developing user-software. The user can add hardware options, such as the IBV11-A, as desired.

For a simple execute-only system, the minimum system hardware requirements include a console terminal, a means for loading object programs, and the IBV11-A interface. In addition, LSI-11 component systems require an LSI-11 processor and 4K memory, one or more backplanes, and +5 V and +12 V power supplies. A simple execute-only system (that is, one that will execute but not develop user application software) can consist of the basic components listed below:

Option	System Function
KD11-F* or KD11-J	LSI-11 processor and 4K memory
DLV11	Serial line interface for console terminal (an optional BC05M or BC05C cable is required for connection to the terminal)
H9270* or DDV11-B	System backplane
VT50, VT52, LA36, LT33, etc.	Console terminal
H780* (or equivalent)	Power supply
IBV11-A	Instrument bus interface

*Included in the basic PDP-11/03 system.

Note that the KD11-F processor and 4K memory uses semiconductor memory which is volatile (programs and data are lost when system power is turned off) and programs must be loaded each time power is turned on. The KD11-J uses non-volatile 4K core memory and normally does not require reloading programs. Non-volatile storage can also be obtained by the use of the MRV11-AA 4K programmable read-only memory (PROM) option or the MRV11-BA 4K UV PROM/256 RAM option; these options are supplied without PROM integrated circuits (ICs). PROM options are available from DIGITAL.

The execute-only system, as described, is not capable of developing user application software. Minimum requirements for a paper tape system capable of program development include 8K (total) read/write memory and a paper tape reader/punch. The paper tape reader must be capable of being turned on and off under program control. Either a high-speed or low-speed (110 baud) reader/punch can be used. A low-speed reader/punch function can be implemented by connecting an LT33 Teletype® (DEC-modified ASR33) to a DLV11 serial line interface; the Teletype can also function as the console device. DIGITAL can supply a modification kit for the ASR33 as option model LT33-MB. A modified ASR33 for 115 V, 60 Hz operation is available from DIGITAL as option model LT33-DC. If desired, a high-speed reader/punch (user-supplied) can be interfaced to the LSI-11 system as directed in the Digital Components Group Application Note, *An LSI-11 Paper Tape Reader/Punch Interface.*

Floppy-disk-based systems are capable of maximum flexibility for user program development and applications. Any LSI-11 system capable of executing RT-11 operating system software is capable of supporting IBV11-A applications, including program development. Any LSI-11 system containing 8K read/write memory and a console device can be expanded for RT-11 operation by adding the RXV11 floppy disk option and appropriate software. The PDP-11V03 system includes the required hardware and software to support RT-11.

3.1.2 System Software

3.1.2.1 General – The only software required for IBV11-A operation in an LSI-11 system is the binary program for a specific application. However, in order to generate a binary program on the system, certain software options will aid the program development process. A brief list of software options is provided below.

For assembly language program development:

1. *Paper tape system software*
   QJV10-CB – This software option includes ED-11 Text Editor, PAS11S Assembler, LINK11 Linker, DUMPAB Memory Dump Utility, ODT-11 On-Line Debugging Technique, IOX Input/Output Executive, and the Absolute Loader.

   QJV11-CB, PROM Formatter – This software option reads binary object tapes and produces punched paper tapes (for automatic PROM programmers) and listings that are compatible with MRV11-AA and MRV11-BA applications.

2. *Floppy disk software*
   QJ003-AY or QJ003-CY, RT-11 Operating System – This software option includes: EDIT text editor, MACRO assembler (required 12K read/write memory), EXPAND macro expander, ASEMBL assembler, and various system and utility programs. EXPAND and ASEMBL allow program assembly on the basic 8K floppy disk LSI-11 system. RT-11 is a very comprehensive software option and is described in much greater detail in the Microcomputer Handbook, Section 4.

®Teletype is a registered trademark of Teletype Corporation.

For high-level language program development:

1. *FORTRAN*
   QJ925-AY or QJ925-CY - RT-11/FORTRAN

2. *BASIC*
   QJ920-AY or QJ920-CY - RT-11/BASIC

NOTE

The above software options run under the RT-11 operating system. Routines for IBV11-A support for FORTRAN and BASIC are planned options; however, they are not presently available.

3.2 CONFIGURING THE IBV11-A MODULE

Each IBV11-A module is factory-configured for standard device register and interrupt vector addresses. Switches S1 (vector address) and S2 (device register) configure the addresses. A summary of register and vector addressing is provided in Figures 3-1 and 3-2. Observe that only the IBS register address is configured. The IBD register address is always the IBS address plus 2. Similarly, only the error interrupt vector address is configured. The remaining three vector addresses are permanently assigned sequential addresses in address increments of four, as follows:

Vector	**Address**
Error	"n" (configured address)
Service	n + 4
Command and Talker	$n + 10_8$
Listener	$n + 14_8$

Switches S1 and S2 are located on the IBV11-A module as shown in Figure 3-3. S1 and S2 are switch assemblies, each containing several individual switches. The individual switches indicated in Figures 3-1 and 3-2 are clearly marked on the S1 and S2 assemblies. The ON and OFF positions are also clearly marked.

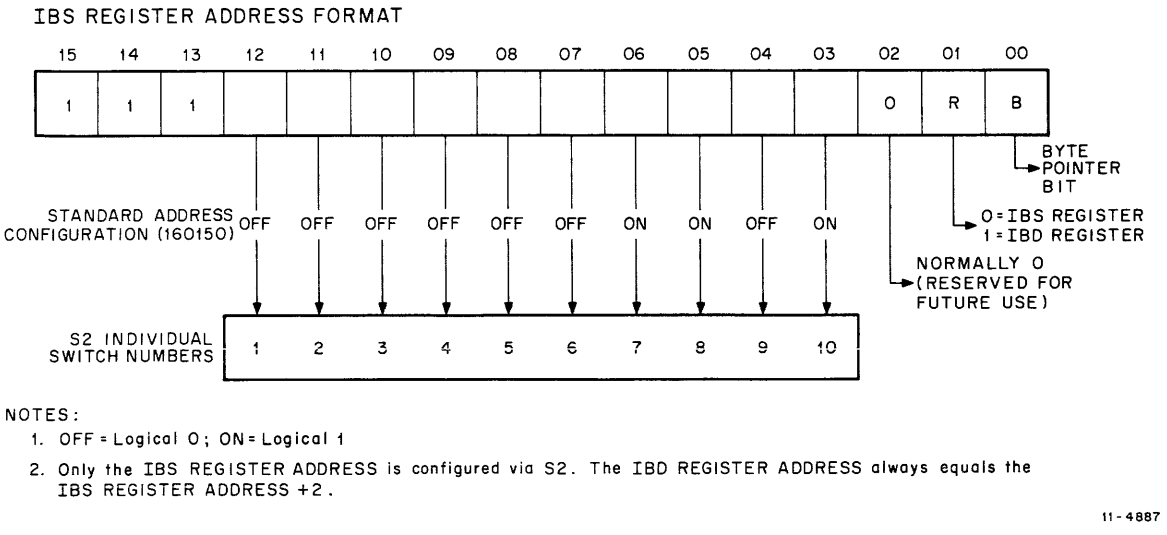
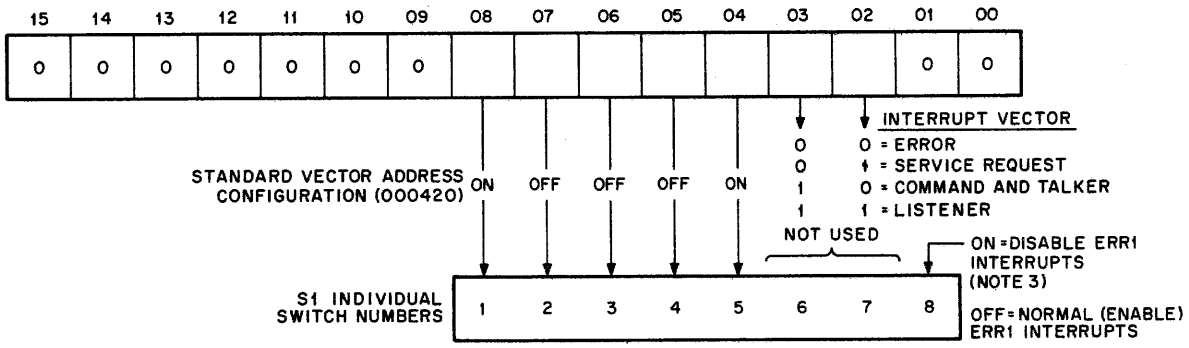


Figure 3-1 Configuring the IBV11-A Register Addresses



NOTES:

1. OFF = Logical 0; ON = Logical 1
2. Only the VECTOR ADDRESS bits (8:4) are configured via S1. Bits 3 and 2 are IBV11-A hardware-selected for the functions shown
3. S1-8 OFF = IBV11-A is the only system controller connected to the instrument bus.
   S1-8 ON = Another system controller is connected to the instrument bus.

11-4888

Figure 3-2 Configuring the IBV11-A Interrupt Vector Addresses

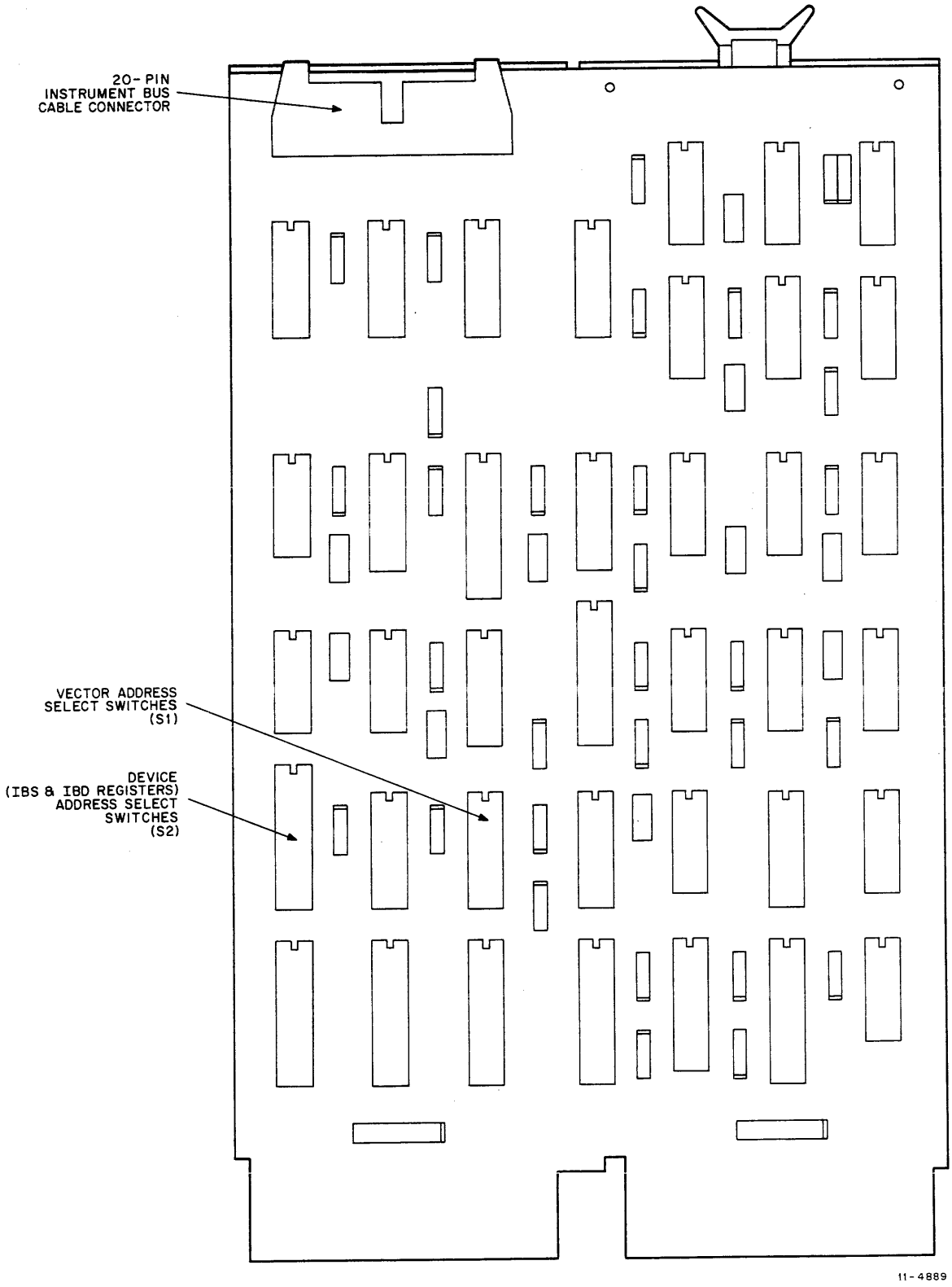


Figure 3-3 IBV11 Module Switch Locations

3.3 INSTALLING IN THE LSI-11 BACKPLANE

The general procedure for installing the IBV11-A module in the LSI-11 backplane is the same as for any other peripheral interface. Prior to installing the module, determine the desired peripheral device priority as described in the Microcomputer Handbook, Section 1, Chapter 3, and Chapter 6, Paragraph 6.3. Module insertion and removal is described in Section 1, Chapter 6, Paragraph 6.4.

3.4 CONNECTING TO EXTERNAL EQUIPMENT

Connection from the IBV11-A to the first device on the instrument bus is via a type BN11A cable as shown in Figure 3-4. One end is terminated with a 20-pin connector that mates with the 20-pin connector on the IBV11-A module. The other end is terminated with a 24-pin "double-ended" connector that conforms with the IEEE 488-1975 standard; the cable can be connected to any device conforming to that standard. The double-ended connector contains a male 24-pin and a female 24-pin connector in the same connector housing. This allows for "linear" and "star" connections to instruments connected to the instrument bus, as shown in Figure 3-5.

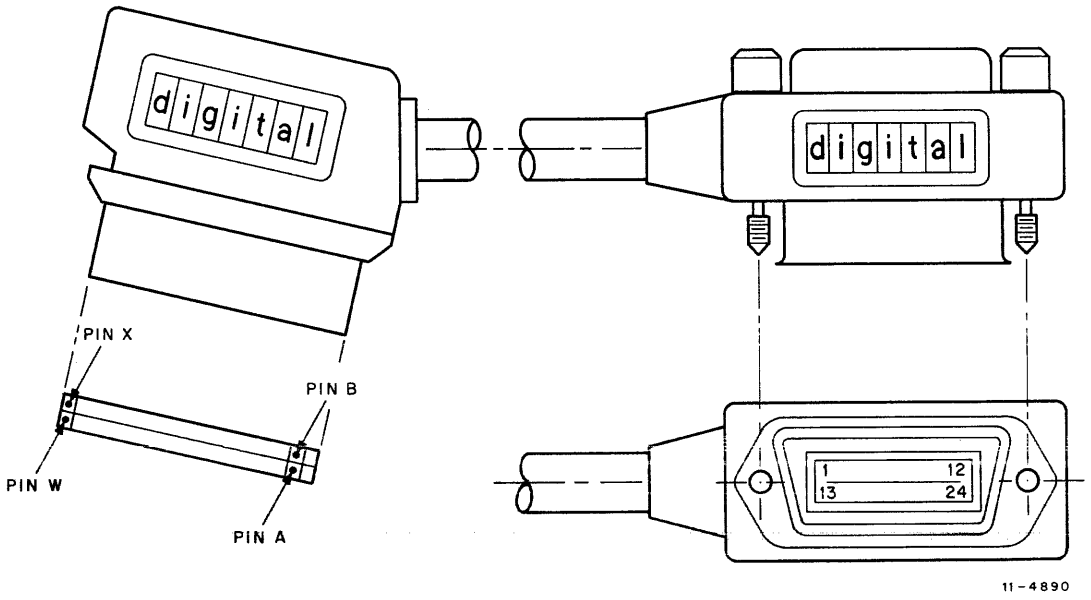


Figure 3-4 BN11A Instrument Bus Cable

The linear arrangement shown in Figure 3-5 includes five devices (or instruments), A through E. There is no particular significance to the sequence shown or electrical position along the instrument bus. Unlike the LSI-11 bus, position along the bus does not structure device priority in the system.

The star arrangement shown in the figure allows five devices to be connected by stacking BN01A instrument cable connectors on the BN11A's double-ended connector. Double-ended connectors on instrument bus cables will normally include captive locking screws on each connector assembly (two each), allowing stacked connectors to be secured together in a single assembly.

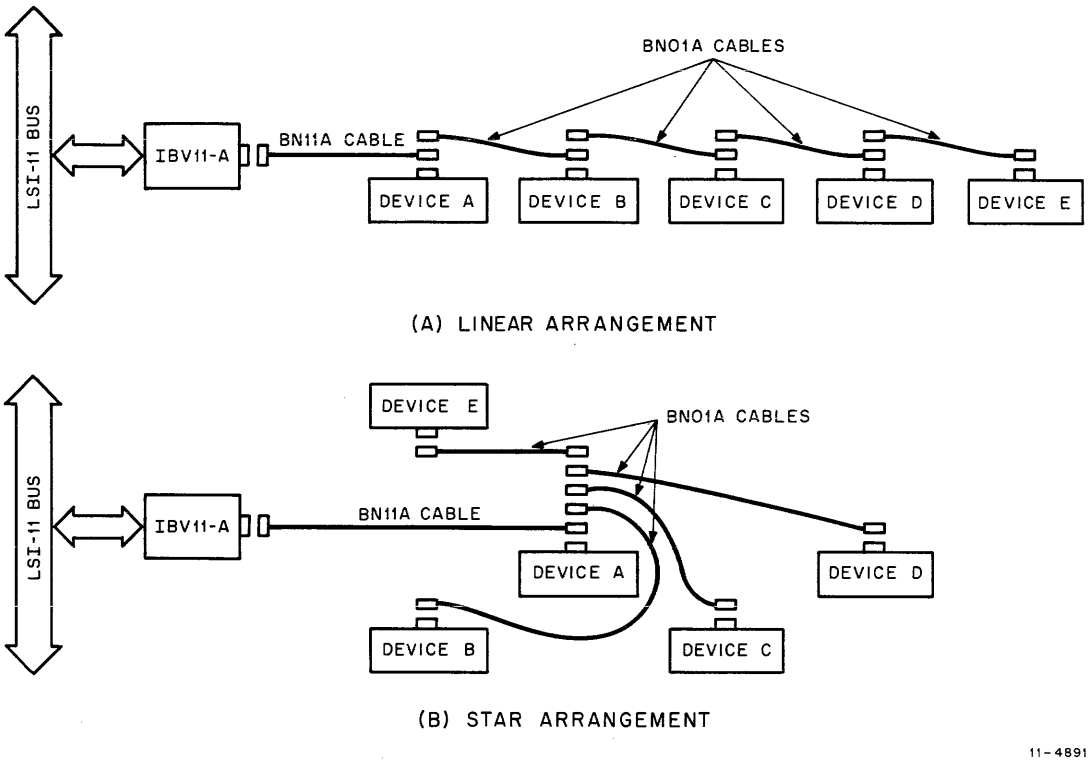


Figure 3-5 Linear and Star Configurations

BN11A and BN01A cable connector pin signal assignments are listed in Table 3-1 for each connector. One 4 m (157.5 in) BN11A-04 cable is supplied with each IBV11-A option. Additional BN11A cables are available in the following lengths:

Model	**Length**
BN11A-01	1 m (39.4 in)
BN11A-02	2 m (78.7 in)
BN11A-04	4 m (157.5 in)

BN01A cables are available in the following lengths:

Model	**Length**
BN01A-01	1 m (39.4 in)
BN01A-02	2 m (78.7 in)
BN01A-04	4 m (157.5 in)

Table 3-1 BN11A and BN01A Connector Pin Assignments

BN11A (only) IBV11-A Connector Pin	Signal Name	BN11A and BN01A Instrument Bus Connector Pin
U	DI01	1
S	DI02	2
P	DI03	3
M	DI04	4
R	EOI	5
T	DAV	6
V	NRFD	7
X	NDAC	8
B	IFC	9
J	SRQ	10
F	ATN	11
	(SHIELD)	12
K	DI05	13
H	DI06	14
E	DI07	15
C	DI08	16
D	REN	17
	GND (DAV GND)	18
N	GND (NRFD GND)	19
	GND (NDAC GND)	20
	GND (IFC GND)	21
A, L	GND (SRQ GND)	22
	GND (ATN GND)	23
W	GND (LOGIC)	24

CHAPTER 4
PROGRAMMING EXAMPLES

4.1 GENERAL

This chapter contains two programming examples that illustrate how the LSI-11 system can communicate with instruments via the IBV11-A. No attempt is made to restrict these programming examples to specific devices. Refer to programming instructions included with the programmable instruments for specific procedures, including codes (ASCII characters) for device addressing, setting modes, ranges, etc. LSI-11 program listings, flowcharts, memory maps, and detailed descriptions are included for each example.

Both programming examples have been executed on an LSI-11 system. The programs were entered manually via the console terminal using console ODT commands. In an actual applications environment, however, routines, similar to these programming examples, would function as parts of a larger program. Additional features, such as the ability to enter instrument parameters directly from the console keyboard and display processed results would be incorporated. No attempt is made in these examples to illustrate those features since each application, available hardware, and programmer's skill will vary. However, the examples do illustrate how the IBV11-A can be programmed to perform useful tasks.

4.2 EXAMPLE 1 - IBV11-A TO LISTENER DEVICE

4.2.1 General

This programming example illustrates how the IBV11-A communicates with a listener device. Standard device and vector addresses are used, as shown in Figures 3-1 and 3-2. Once the program is started, and after pointers have been initialized and the IBV11-A has taken control synchronously, the program communicates with the IBV11-A via an interrupt-driven service routine. No "background" program is used; the program simply "waits" until another interrupt occurs.

Communication with the listener device includes the transmission of 2 command bytes (read as words from a message buffer), followed by 24 message bytes that program device functions. After all message bytes have been transmitted, the program halts (displayed HALT PC address = 1066).

A program flowchart for this example is shown in Figure 4-1, and a symbolic listing is shown in Figure 4-2. Figure 4-3 is a memory map for the program. Refer to those figures when reading the description of program operation in the following paragraphs.

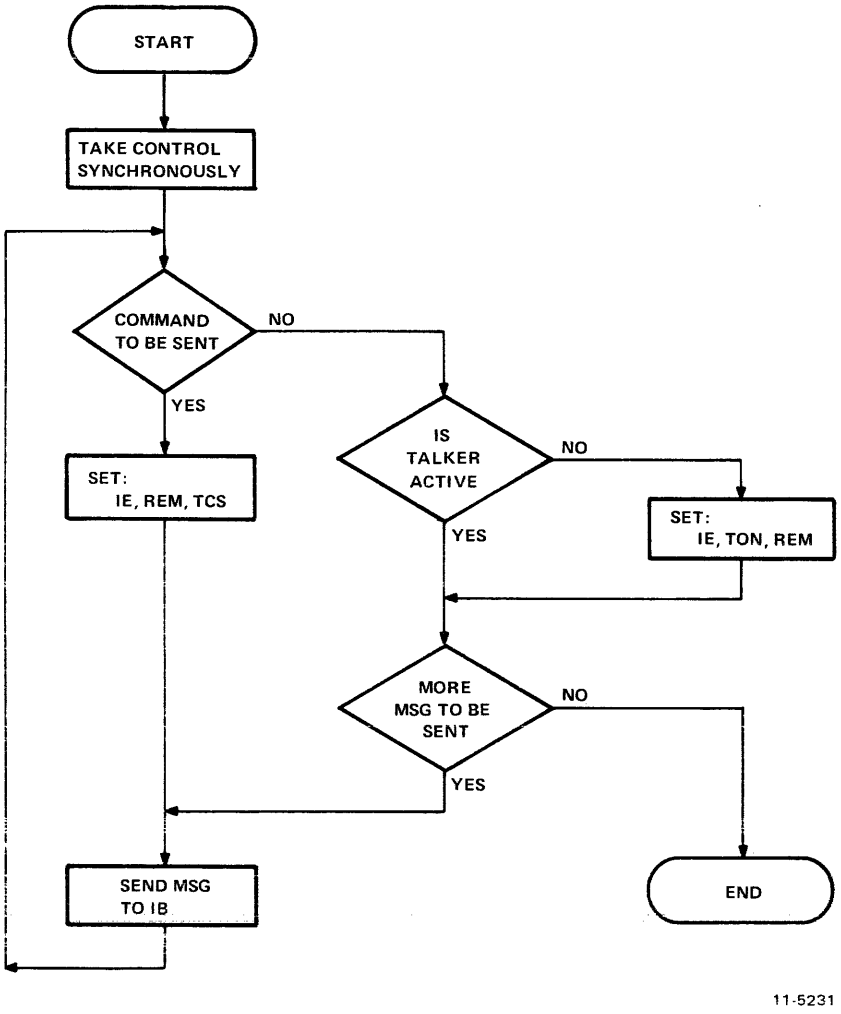


Figure 4-1 Example 1, Communicating with a Listener Device Program Flowchart

```
ADDRESS   OCTAL CODE      ASSEMBLER SYNTAX      COMMENTS

000430    001020                              ; INTR RETURN ADDRESS
000432    000200                              ; PSW

001000    012706   START: MOV #500,R6         ; SET UP STACK POINTER
001002    000500
001004    012700         MOV #2000,R0         ; R0 IS MSG BUFFER ADDRESS
001006    002000
001010    012737         MOV #110,160150      ; TAKE CONTROL
001012    000110                              ; SYNCHRONOUSLY TO BECOME
001014    160150                              ; CONTROLLER-IN-CHARGE

001016    000777   WAIT: BR .                 ; WAIT FOR INTERRUPT
001020    022700         CMP #2004,R0         ; MORE COMMANDS TO BE SENT?
001022    002004
001024    100006         BPL 20$              ; IF NO,GO TO 20$
001026    012737         MOV #105,160150      ; IF YES,SET IE,REM,AND
001030    000105                              ; TCS BITS OF IBS REG TO
001032    160150                              ; ACTIVATE CONTROLLER
001034    012037   SEND: MOV (R0)+,160152     ; SEND MSG TO IB
001036    160152
001040    000002         RTI                  ; RETURN TO WAIT--FOR
                                              ; MSG TO BE ACCEPTED
001042    022700   20$:  CMP #2004,R0         ; IS TALKER ACTIVE?
001044    002004
001046    003003         BGT 30$              ; IF YES,GO TO 30$
001050    012737         MOV #144,160150      ; OTHERWISE SET IE,TON
001052    000144                              ; AND REM BITS OF IBS REG
001054    160150                              ; TO ACTIVATE TALKER
001056    022700   30$:  CMP #2062,R0         ; HAD ALL MSG BEEN SENT?
001060    002062
001062    100364         BMI SEND             ; IF NO,GO SEND ANOTHER MSG
001064    000000         HALT                 ; OTHERWISE STOP
```

11-5232

Figure 4-2 Example 1, Communicating with a Listener Device Program Listing

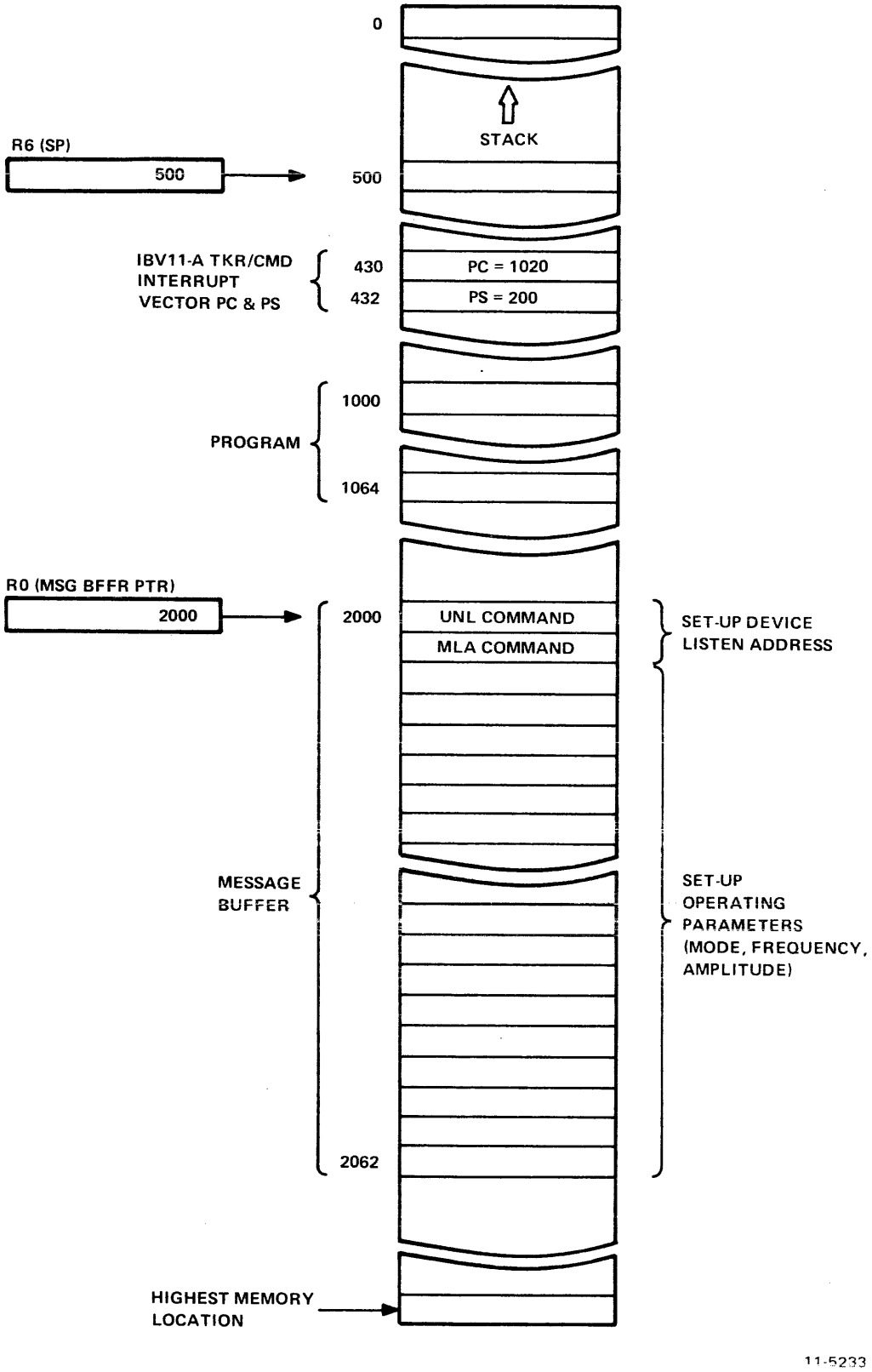


Figure 4-3 Memory Map for Example 1

4.2.2 Program Operation

Prior to starting the program, it is assumed that interrupts are enabled (PS bit 07 = 0). The program is started at address 1000. The first MOV instruction (at location 1000) initializes the stack pointer (R6) to 500. Note that this address is first decremented by two each time a new item is "pushed" onto the stack in the usual manner; the content of R6 always points to the address of the last word pushed onto the stack. Conversely, R6 is incremented after "popping" a word from the stack. In this programming example, the stack is used for servicing the IBV11-A's interrupt requests; the interrupts will always occur while executing the "wait" (BR.) instruction at location 1016. Thus, the PC pushed onto the stack for this example is 1016, followed by the PS. These two words are popped from the stack to restore the wait instruction by executing the RTI instruction at location 1040.

The next MOV instruction (at location 1004) initializes R0 to 2000 for use as a message buffer address pointer. The number of messages sent, and hence, the actual size of the message buffer, is dependent on the requirements of the instrument being programmed. In this example, 26_{10} locations are reserved for the message buffer. This means that the message buffer's last address is 2062. During program execution, the CMP instruction at location 1056 tests R0 to determine if all locations of the message buffer have been transmitted to the addressed listener device.

The MOV instruction at location 1010 sets the IBV11-A's IBS register IE and IBC bits (bits 06 and 03). The IBV11-A responds by generating a 125 μs IFC pulse, followed by "taking control synchronously." Once this operation is completed, the IBS CMD bit becomes set and the IBV11-A's TKR or CMD interrupt request is enabled. The vector address for this interrupt is 430 [the configured address (420_8) + 10_8].

The BR. ("branch to self") instruction at location 1016 causes the program to "hang" (or wait) until the interrupt occurs. The processor services the interrupt by pushing the previous PC and PS onto the stack and uses the new PC (contents of location 430 = 1020) and PS (contents of location 432 = 200) for the interrupt-driven routine; interrupts are disabled by the new PS.

In this example, the first two words in the message buffer are commands [UNL (unlisten) and MLA (my listen address "n")]. The CMP instruction at location 1020 tests if the two commands have been sent. Initially, R0 = 2000; executing the CMP and BPL instructions results in executing the MOV instruction at location 1026. The MOV instruction sets IBS IE, REM, and TCS bits to activate the IBV11-A as the controller-in-charge. The MOV instruction at location 1034 is then executed, transmitting the first command byte to the instrument bus via the low byte of the IBV11-A's IBD register (device address 160152). Note that R0 is auto-incremented after it is used, causing it to point to the next command in the message buffer. The RTI instruction at location 1040 is then executed, causing the program to return to the BR. instruction at location 1016, and the program waits for the next interrupt. When the interrupt occurs, it indicates that the command transfer over the instrument bus has been completed. The second command is then transmitted in exactly the same manner. However, note that the message buffer pointer content of R0 is now 2004, and no more commands are to be sent.

When the next interrupt occurs, the CMP and BPL instructions following WAIT cause the program to branch to 20$ (location 1042). Since the content of R0 = 2004 at this point of program execution, the branch (BGT) at location 1046 will not be executed; instead, the MOV instruction at location 1050 is executed, setting IBS register bits IE, TON, and REM, and the IBV11-A becomes an active talker.

Message bytes are now sequentially sent to the listener via the low byte of the IBV11-A's IBD register, functioning as a talker on the instrument bus. The CMP instruction at location 1062 first checks if all message bytes have been transmitted. If more bytes are to be transmitted, the BMI instruction causes the program to branch to SEND. A word is then transmitted to the IBD register and the program exits to WAIT until another interrupt occurs; the content of R0 is now 2006. When the interrupt occurs, CMP and BPL instructions cause the program to branch to 20$. Since R0 now contains a message buffer pointer greater than 2004, the CMP and BGT instructions cause the program to branch to 30$. The program then checks if the last message buffer word has been transmitted. If not, the program branches to SEND, another message byte is transmitted, etc., until all messages have been transmitted. When all messages have been transmitted, the content of R0 will be 2064. The CMP and BMI instructions (following 30$) will then result in the HALT instruction at location 1064 being executed, and the program execution is completed. The instrument will then operate according to the command/program data received over the instrument bus.

4.3 EXAMPLE 2 - IBV11-A TO TALKER DEVICE

4.3.1 General

This programming example illustrates how the IBV11-A communicates with a talker device. As in example 1, this programming example assumes standard IBV11-A device and interrupt vector addresses. Communication between the instrument and the LSI-11 system is via IBV11-A interrupt-driven service routines. No background program is used; the program simply "waits" until another interrupt occurs.

Communication with the instrument involves first transmitting the content of the command message buffer, in a manner similar to the program operation described for example 1, followed by accepting instrument output data and storing it in a received data buffer. The content of the command message buffer typically includes first activating the device via its listen address, followed by setting up range, mode, etc. operating parameters for the instrument, an execute command, and, finally, activating the device as an active talker via its talker address. Once the device has received the command message buffer data, it performs the programmed measurements (or the function, depending on the instrument) and returns data to the LSI-11 system via the IBV11-A; note that during this portion of program operation, the IBV11-A functions as an active listener on the instrument bus. Once all measurements have been stored by the program, the program halts with a displayed PC address = 1102.

A program flowchart for this example is shown in Figure 4-4 and a symbolic program listing is shown in Figure 4-5. Figure 4-6 is a memory map for the program. Refer to those figures when reading the description of program operation in the following paragraphs.

4.3.2 Program Operation

The program is started at location 1000. Prior to starting the program, it is assumed that interrupts are enabled (PS bit 07 = 0) and the program and interrupt vector code has been loaded into system memory as shown in Figure 4-5. The first four MOV instructions set up the stack pointer, initialize the command message and received data buffer pointers R0 and R1, respectively, and cause the IBV11-A to take control synchronously. The program portion starting at WAIT (location 1022) through the RTI instruction at location 1044 outputs the command message buffer in a manner similar to that described for example 1.

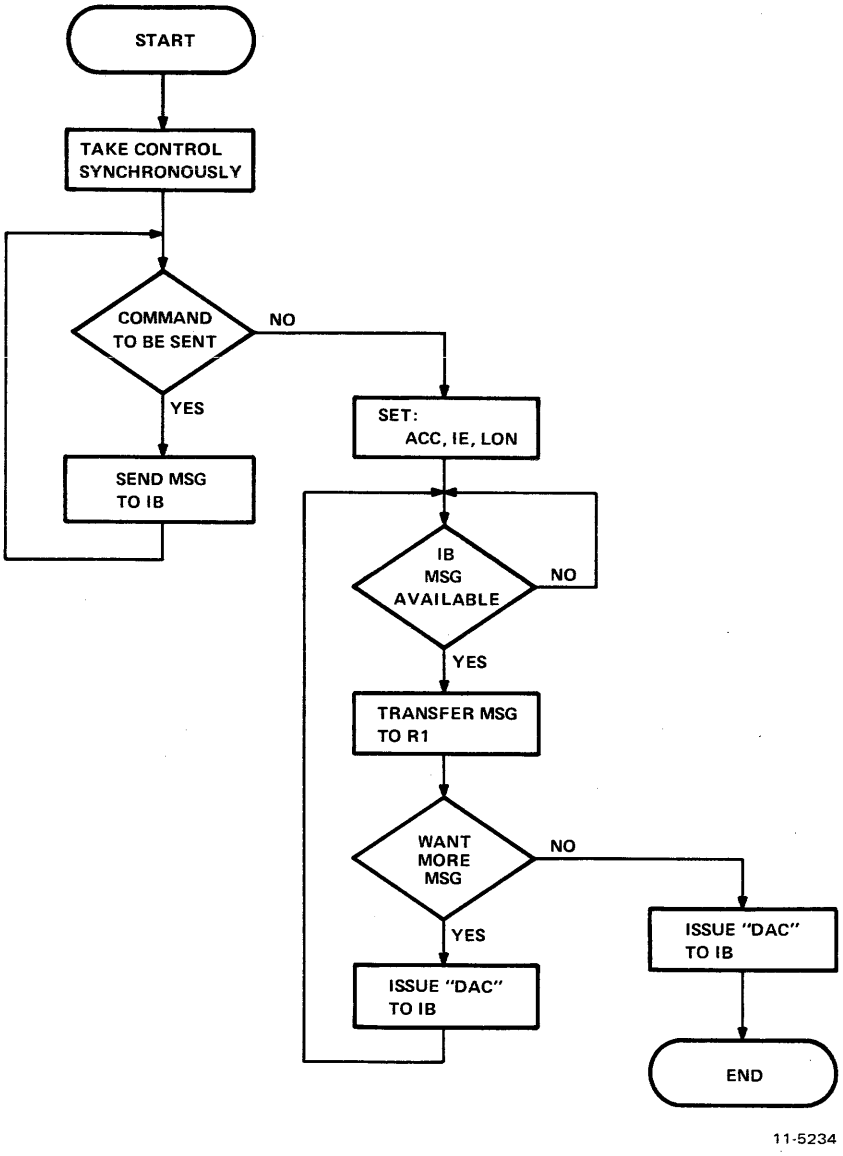


Figure 4-4 Example 2, Communicating with a Talker Device Program Flowchart

```
ADDRESS  CODE    ASSEMBLER SYNTAX          COMMENTS

000430   001024                           ; COMMAND/TALKER INTR
                                          ; RETURN ADDRESS
000432   000200                           ; PSW
000434   001056                           ; LISTENER RETURN ADDRESS
000436   000200                           ; PSW

001000   012706  START:  MOV #500,R6       ; SET UP STACK
001002   000500
001004   012700         MOV #2000,R0      ; IBV11-A MSG BUFFER
001006   002000
001010   012701         MOV #2500,R1      ; BUFF FOR RECEIVED MSG
001012   002500
001014   012737         MOV #110,160150   ; TAKE CONTROL SYNCHRONOUSLY
001016   000110                           ; TO BECOME CONTROLLER-
001020   160150                           ; IN-CHARGE,C-I-C
001022   000777  WAIT:   BR .              ; WAIT--FOR INTERRUPT
001024   012737         MOV #105,160150   ; PREPARE TO SEND
001026   000105                           ; COMMAND MESSAGES
001030   160150
001032   022700         CMP #2024,R0      ; HAD ALL COMMANDS
001034   002024                           ; BEEN SENT?
001036   001404         BEQ 20$           ; IF YES,GO TO 20$
001040   012037         MOV (R0)+,160152  ; OTHERWISE SEND MSG
001042   160152
001044   000002         RTI               ; RETURN TO WAIT--FOR
                                          ; MSG TO BE ACCEPTED
001046   012737  20$:    MOV #320,160150   ; IBV11-A SWITCHES FROM
001050   000320                           ; CONTROLLER TO LISTENER
001052   160150
001054   000002         RTI               ; RETURN TO WAIT--
                                          ; FOR DMM MSG
001056   013721         MOV 160152,(R1)+  ; SAVE THE RECEIVED
001060   160152                           ; MSG IN R1
001062   022701         CMP #2540,R1      ; HAD 20 (OCTAL) MSG
001064   002540                           ; BEEN ACCEPTED?
001066   001403         BEQ 30$           ; IF YES,GO TO 30$
001070   005037         CLR 160152        ; OTHERWISE ISSUE DAC
001072   160152
001074   000002         RTI               ; RETURN TO WAIT--FOR
                                          ; ANOTHER DMM MSG
001076   005037  30$:    CLR 160152        ; ISSUE DAC TO IB
001100   000000         HALT              ; STOP,20 MSG RECEIVED
```

11-5235

Figure 4-5 Example 2, Communicating with a Talker Device Program Listing

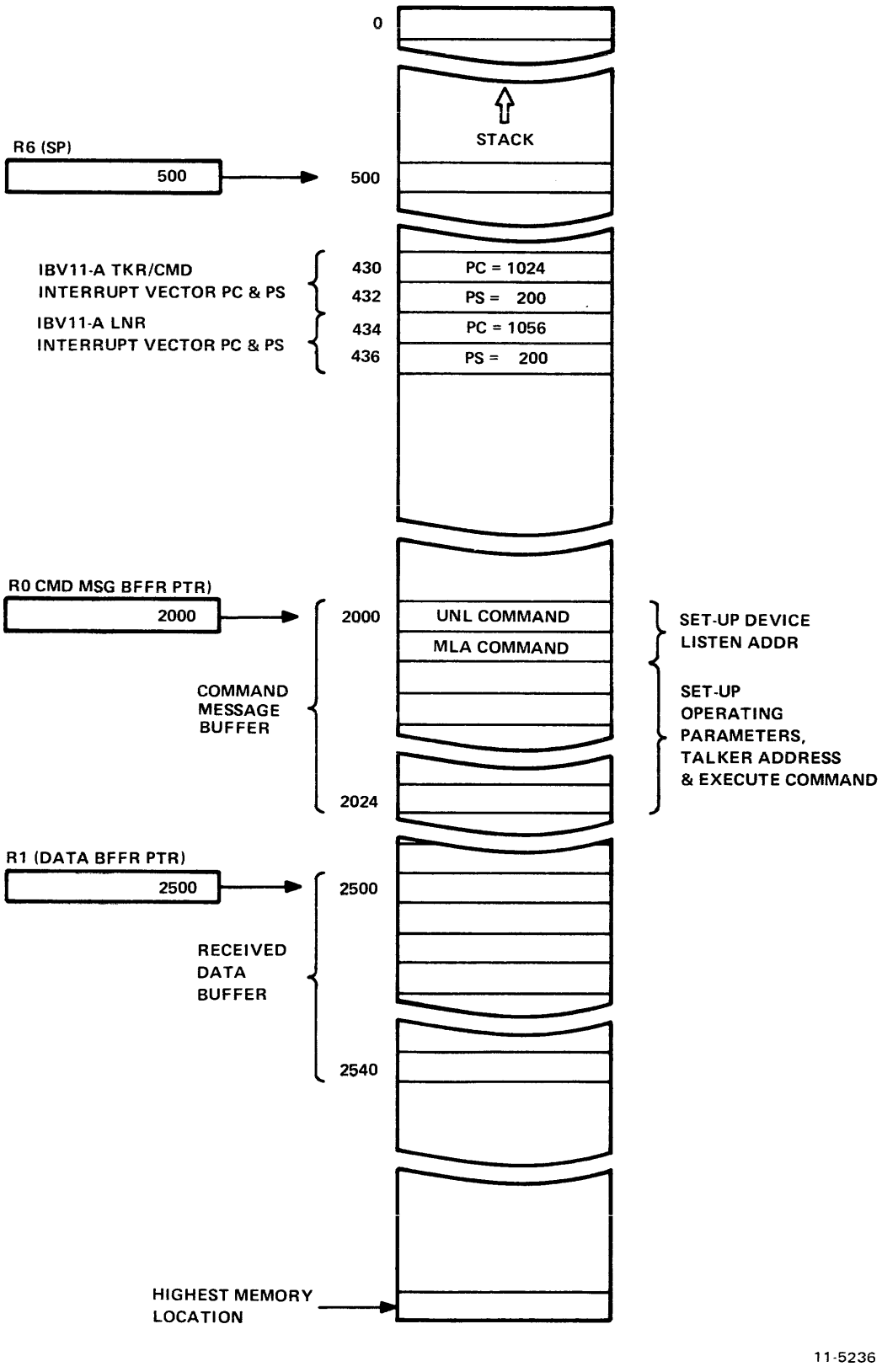


Figure 4-6 Memory Map for Example 2

When all command bytes have been transmitted to the addressed device, the program branches to 20$ (location 1046) and the MOV instruction at that address causes the IBV11-A to become an active listener. This is followed by an RTI instruction and the program waits for the IBV11-A's listener (LNR) interrupt to occur. When the interrupt occurs, the IBV11-A's IBD register, which contains the first measurement data word (byte), is read and stored in the first received data buffer location (2500); note that the received data buffer pointer (content of R1) is incremented by two (one word address) each time the IBD register is read. The CMP and BEQ instructions that follow test if all 16_{10} data words have been received. If not, the RTI instruction returns the program to WAIT until the next IBV11-A LNR interrupt occurs. Operation continues in this manner until all data transfers have been stored in the received data buffer. When the last buffer location has been filled, R1 contains 2540_8, the last address in the received data buffer. Executing the CMP instruction (location 1062) and BEQ instruction results in the program branching to 30$ (location 1076). Operation is then terminated by first clearing the IBD register; the IBD responds by asserting DAC to complete the data transfer from the instrument to the IBV11-A's IBD register. Finally, the program executes the HALT instruction and the operation is completed.

The data contained in the received data buffer is device-dependent for its actual significance. In a typical application, the HALT could be replaced by a branch (BR) or jump (JMP) instruction to transfer control to a program that processes the received data.

CHAPTER 5
TECHNICAL DESCRIPTION

5.1 GENERAL

The functional logic blocks that comprise the IBV11-A are shown in Figure 5-1. LSI-11 software controls and communicates with the IBV11-A via programmed I/O transfers and interrupts. Refer to the Microcomputer Handbook, Section 1, Chapter 3 for a complete description of LSI-11 bus cycles, including DATO, DATOB, DATI, DATIO, DATIOB, and interrupt transactions.

5.2 DEVICE REGISTERS

Programmed I/O transfers are made possible by assigning unique device addresses (also called "bus addresses") to the IBS and IBD registers. The IBS register is the instrument bus status register; it is generally similar in function to other PDP-11 device control/status registers (CSRs). The IBD register is the instrument bus data register. It is a 16-bit register that contains eight read/write data bits in the low byte and eight read-only bits in the high byte. The eight read-only bits allow the program to read the logical state of the instrument bus.

Functions of bits in the IBS and IBD registers are described in Paragraph 2.2.4.

5.3 LSI-11 BUS INTERFACE

LSI-11 bus address selection, interrupt vector address generation, and bus data driver/receiver (transceiver) functions are provided by transceiver integrated circuits (DC005). Each integrated circuit provides the interface for four BDAL bus lines; thus, four transceivers comprise the 16-line BDAL <00:15> L LSI-11 bus interface. Refer to Appendix A for a detailed description of the DC005 transceiver integrated circuit.

Device address switches provide a convenient means for the user to configure the IBV11-A's register addresses. Only switches corresponding to BDAL lines <03:12> are provided. By PDP-11 convention, the upper 4K address space (bank 7) is normally reserved for peripheral devices, such as the IBV11-A. The LSI-11 processor module asserts BBS7 L whenever a bank 7 address (BDAL <13:15> L are asserted) is placed on the bus. Thus, BBS7 L must be asserted to enable an "address match" output from the address selection function. Any address ranging from 16000X to 17777X can be configured, as long as it does not conflict with other device addresses within the system; the X in the address represents register and byte selection within the module.

The preferred addresses (for DEC software compatibility) are:

IBS = 160150
IBD = 160152

Bit 1 of the least significant octal digit selects the IBS or IBD register. The least significant bit (BDAL 0) is a byte pointer and it is significant for DATOB and DATIOB bus cycles only. Register address selection is actually performed in the LSI-11 bus protocol and register selection function; the transceiver integrated circuit simply routes the received low-order three address bits (DA <2:0>) to that function.

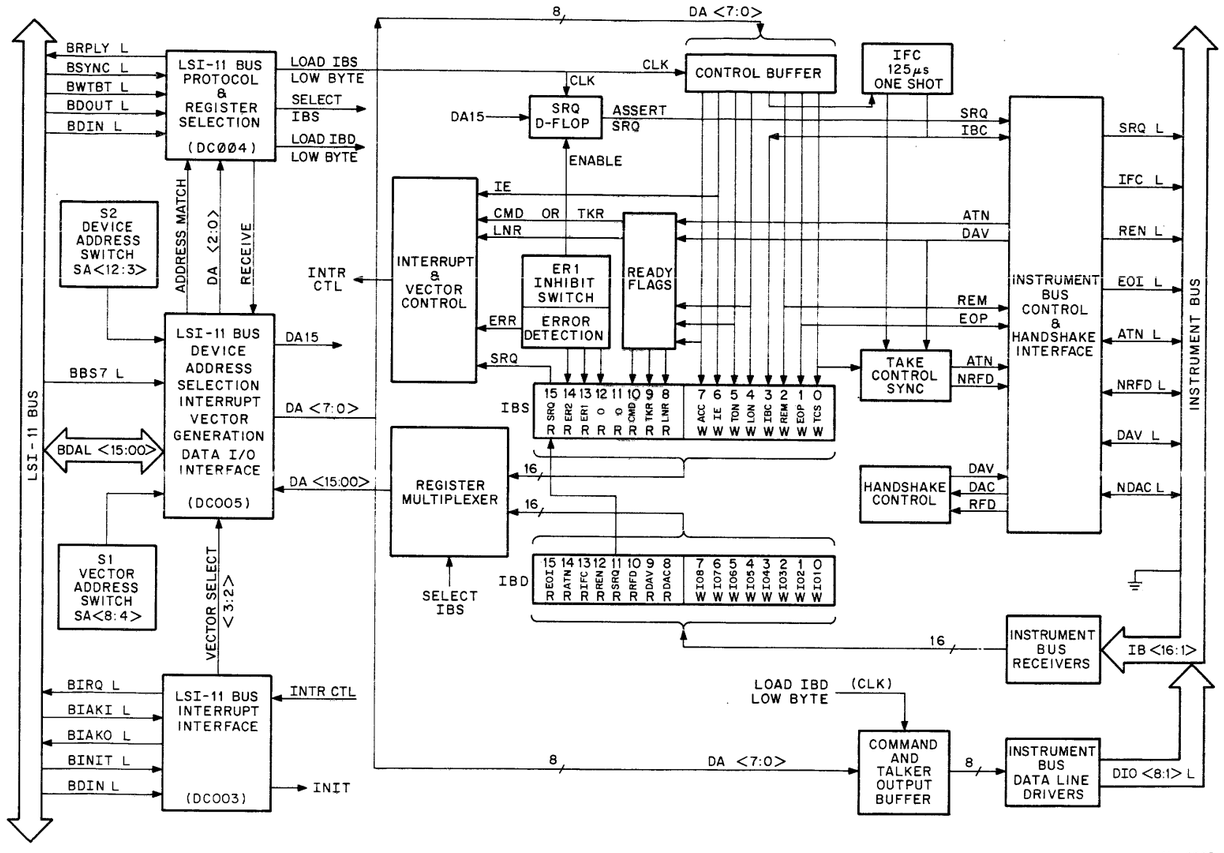


Figure 5-1 IBV11-A
Functional Block Diagram

All I/O transfers over the LSI-11 bus are done according to a strict protocol. One bus protocol integrated circuit (DC004) performs this function and the register address selection previously discussed. When an active ADDRESS MATCH signal is present on the leading edge of the BSYNC L signal, the bus protocol integrated circuit is enabled to complete its register selection function. BWTBT L, BDOUT L, and BDIN L bus signals are decoded in the integrated circuit, as appropriate, to produce the LOAD IBS LOW BYTE, SELECT IBS, LOAD IBD LOW BYTE, and RECEIVE internal control signals for the IBV11-A logic functions. The integrated circuit also asserts BRPLY L as required during the I/O sequence to complete the programmed transfer. Refer to Appendix A for a detailed description of the DC004 integrated circuit.

Interrupts are generated by one interrupt integrated circuit (DC003). Four vector addresses can be generated by this LSI-11 bus interrupt interface function. A 5-bit vector address switch allows the user to select the lowest vector address for the IBV11-A module. The preferred lowest IBV11-A vector address is 420. IBV11-A vector addresses can range from 0 to 774; however, they must not conflict with other LSI-11 bus device and reserved system vectors.

The error vector always has the lowest address; this is the actual address configured by the user. The remaining three vector addresses are relative to the error vector address. For example, the preferred vector address (420) selection produces the following vector addresses:

Address	**IBV11-A Interrupt Vector**
000420	Error
000424	Service request
000430	Command and talker
000434	Listener

These vector addresses allow the IBV11-A to generate interrupts that can most efficiently be serviced by four separate service routines. Each vector address points to a pair of locations in system memory containing the PC and PS words for the interrupt service routine. For example, an IBV11-A service request interrupt request, when acknowledged by the processor, will be serviced by the 000424 vector address as follows:

System Memory Location	**Contents**
000424	Service Routine PC word (starting address)
000426	Service Routine PS word

Interrupt and vector control logic on the IBV11-A module generates the INTR CTL signals that initiate the interrupts. Inputs for this logic function include the interrupt enable (IE) IBS bit (stored in the control buffer), command or talker (CMD or TKR) and listener (LNR) ready flags, error (ERR) status from the error detection logic, and the device service request (instrument bus control signal). Refer to Appendix A for a detailed description of the DC003 interrupt logic integrated circuit.

5.4 INSTRUMENT BUS CONTROL

The control buffer is an 8-bit register that functions as the low byte of the IBS register. Bits stored in this register control generation of interrupts, instrument bus clear, and instrument bus control and status logic. Setting the IBC bit actually triggers a one-shot producing a 125 μs pulse that clears the instrument bus. Take control sync and handshake control logic function together with instrument bus control and handshake interface logic to communicate with instruments on the bus according to instrument bus protocol. LSI-11 output transactions with the low byte of the IBD register result in data being stored in the 8-bit command and talker output buffer. Instrument bus line drivers gate this byte onto the instrument bus when the IBV11-A is an active talker or when it is an active controller.

5.5 INSTRUMENT BUS INTERFACE

The IBV11-A interfaces with the instrument bus via four MC3441 integrated circuits. These integrated circuits are bus transceivers, each containing four bus drivers, four bus receivers, and bus terminations that comply with instrument bus specifications.

CHAPTER 6
MAINTENANCE

6.1 GENERAL

Maintenance for the IBV11-A involves executing a diagnostic program and, if required, repairing the module. As a general rule, the user should first check that the IBV11-A is properly installed as described in Chapter 3. If the module appears to be properly installed (address and interrupt vector switches are properly set, and no unoccupied device locations are located in the backplane between the IBV11-A and the LSI-11 processor module), confirm the operational status of the module by executing the IBV11-A's diagnostic program.

6.2 IBV11-A DIAGNOSTIC SOFTWARE

IBV11-A diagnostic software (MAINDEC-11-DVIBA) includes a diagnostic program tape and a program listing. Detailed operating instructions are included in the listing. In order to completely test the IBV11-A option, a second "known good" IBV11-A module and BN11A instrument bus interface cable are required. If those items are not available, the program will allow tests to be run on those IBV11-A functions that do not require the second module.

Minimum hardware requirements for running diagnostics include the basic LSI-11 system, including 4K read/write memory (minimum), a console terminal, the IBV11-A to be tested, and a paper tape reader. A second IBV11-A (known functional) module and two BN11A cables will allow full use of the diagnostic program. Refer to the Microcomputer Handbook, Section 1, Paragraph 9.3 for general instructions for using paper tape diagnostics.

6.3 DIGITAL SERVICES

Maintenance services can be performed by the user or by DIGITAL, as desired. DIGITAL's services are described in the Microcomputer Handbook, Section 5, Chapter 3.

APPENDIX A
IC DESCRIPTIONS

A.1 DC003 INTERRUPT LOGIC

The interrupt chip is an 18-pin, 0.300-inch center, DIP device that provides the circuits to perform an interrupt transaction in a computer system that uses a "pass-the-pulse" type arbitration scheme. The device is used in peripheral interfaces and provides two interrupt channels labeled "A" and "B," with the A section at a higher priority than the B section. Bus signals use high-impedance input circuits or high-drive open collector outputs, which allow the device to directly attach to the computer systems bus. Maximum current required from the V_{cc} supply is 140 mA.

Figure A-1 is a simplified logic diagram of the DC003 IC. Figure A-2 shows the timing for the "A" interrupt section, while Figure A-3 shows the timing for both "A" and "B" interrupt sections. Table A-1 describes the signals and pins of the DC003 by pin and signal name.

A.2 DC004 PROTOCOL LOGIC

The protocol chip is a 20-pin, 0.300-inch center, DIP device that functions as a register selector, providing the signals necessary to control data flow into and out of up to four word registers (8 bytes). Bus signals can directly attach to the device because receivers and drivers are provided on the chip. An RC delay circuit is provided to slow the response of the peripheral interface to data transfer requests. The circuit is designed such that if tight tolerance is not required, then only an external 1K ±20 percent resistor is necessary. External RCs can be added to vary the delay. Maximum current required from the V_{cc} supply is 120 mA.

Figure A-4 is a simplified logic diagram of the DC004 IC. Signal timing with respect to different loads is shown in Table A-2 and in Figure A-5. Figure A-6 shows the loading for the test conditions in Table A-2. Signal and pin definitions for the DC004 are presented in Table A-3.

A.3 DC005 TRANSCEIVER LOGIC

The 4-bit transceiver is a 20-pin, 0.300-inch center, DIP, low-power Schottky device; its primary use is in peripheral device interfaces, functioning as a bidirectional buffer between a data bus and peripheral device logic. In addition to the isolation function, the device also provides a comparison circuit for address selection and a constant generator, useful for interrupt vector addresses. The bus I/O port provides high-impedance inputs and high-drive (70 mA) open collector outputs to allow direct connection to a computer's data bus structure. On the peripheral device side, a bidirectional port is also provided, with standard TTL inputs and 20 mA 3-state drivers. Data on this port is the logical inversion of the data on the bus side.

Three address jumper inputs are used to compare against three bus inputs and to generate the signal MATCH. The MATCH output is open collector, which allows the output of several transceivers to be wire-ANDded to form a composite address match signal. The address jumpers can also be put into a third logical state that disconnects that jumper from the address match, allowing for "don't care" address bits. In addition to the three address jumper inputs, a fourth high-impedance input line is used to enable/disable the MATCH output.

Three vector jumper inputs are used to generate a constant that can be passed to the computer bus. The three inputs directly drive three of the bus lines, overriding the action of the control lines.

Two control signals are decoded to give three operational states: receive data, transmit data, and disable.

Maximum current required from the V_{cc} supply is 100 mA.

Figure A-7 is a simplified logic diagram of the DC005 IC. Timing for the various functions is shown in Figure A-8. Signal and pin definitions for the DC005 are presented in Table A-4.

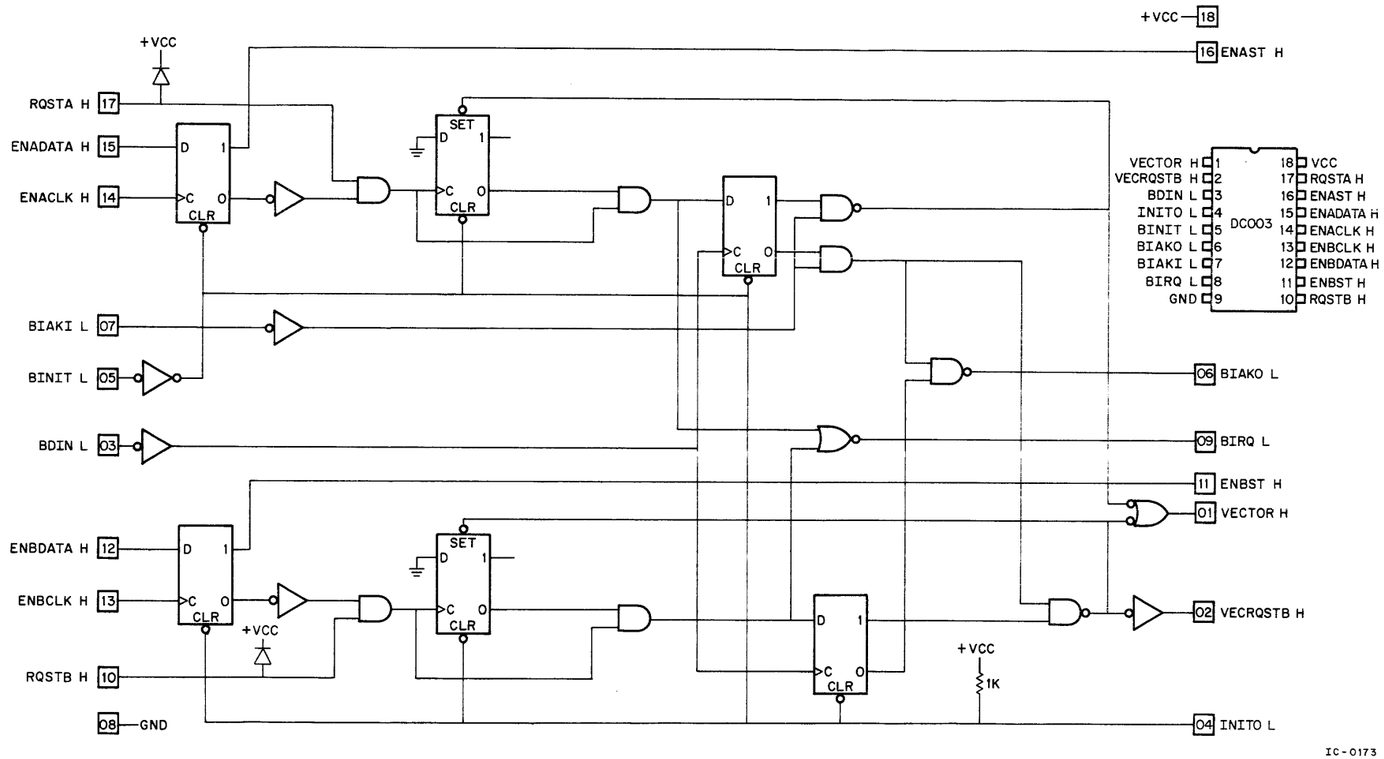


Figure A-1 DC003
Simplified Logic Diagram

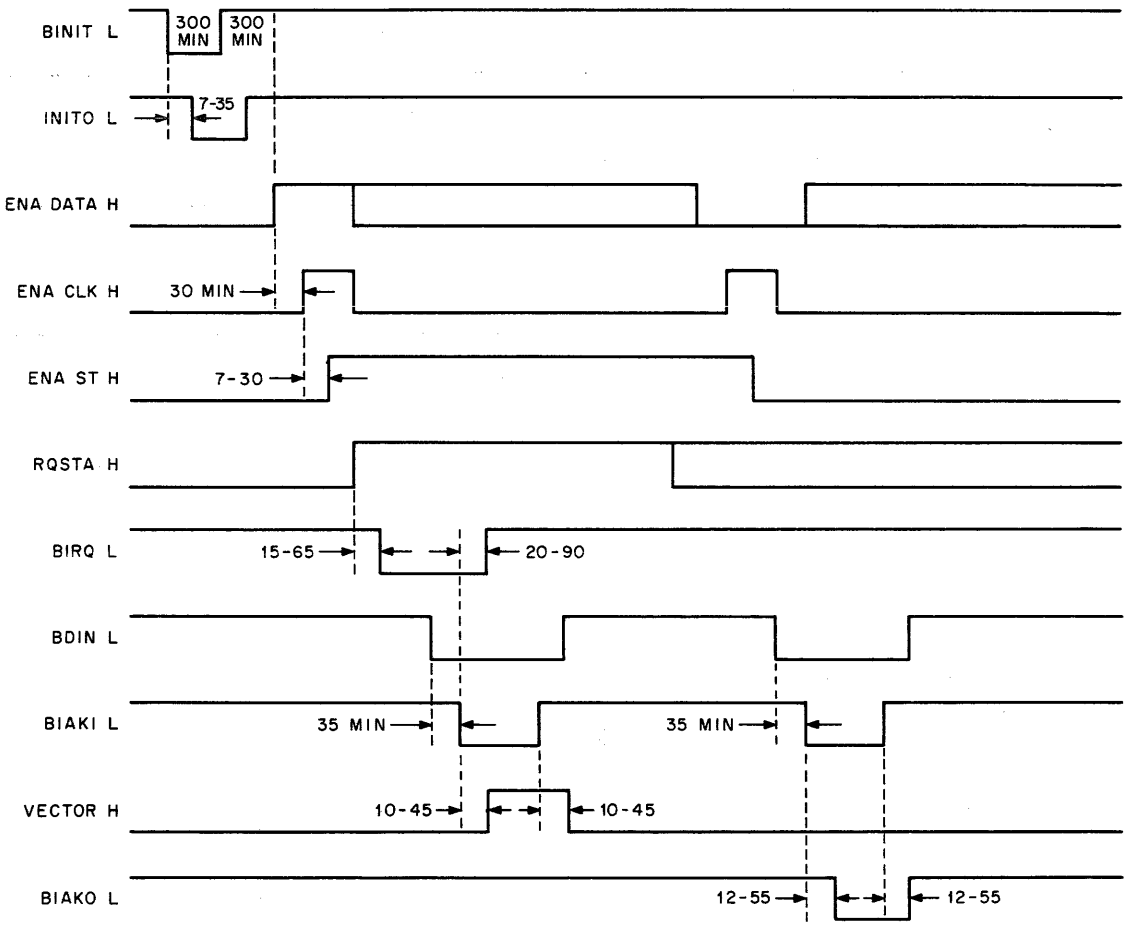


Figure A-2 DC003 "A" Interrupt Section Timing Diagram

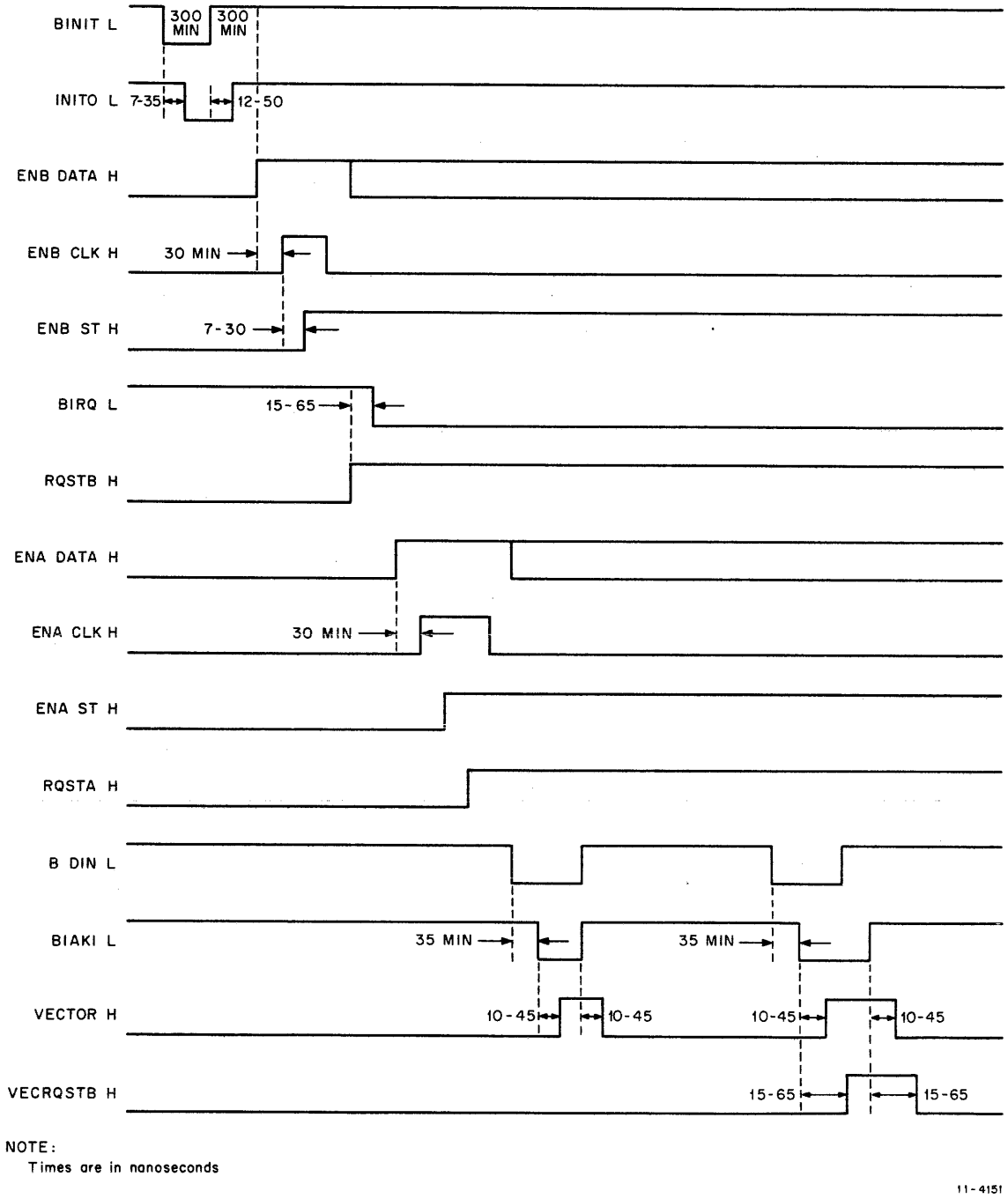


Figure A-3 DC003 "A" and "B" Interrupt Sections Timing Diagrams

Table A-1 DC003 Pin/Signal Descriptions

Pin	Signal	Description
1	VECTOR H	INTERRUPT VECTOR GATING SIGNAL. This signal should be used to gate the appropriate vector address onto the bus and to form the bus signal called BRPLY L.
2	VEC RQSTB H	VECTOR REQUEST "B" SIGNAL. When asserted indicates RQST "B" service vector address is required. When unasserted indicates RQST "A" service vector address is required. VECTOR H is the gating signal for the entire vector address: VEC RQST B H is normally bit 2 of the vector address.
3	BDIN L	BUS DATA IN. This signal generated by the processor BDIN always preceeds a BIAK signal.
4	INITO L	INITIALIZE OUT signal. This is the buffered BINIT L signal used in the device interface for general initialization.
5	BINIT L	BUS INITIALIZE signal. When asserted, this signal brings all driven lines to their unasserted state (except INITO L).
6	BIAKO L	BUS INTERRUPT ACKNOWLEDGE signal (OUT). This signal is the daisy-chained signal that is passed by all devices not requesting interrupt service (see BIAKI L). Once passed by a device, it must remain passed until a new BIAKI L is generated.
7	BIAKI L	BUS INTERRUPT ACKNOWLEDGE signal (IN). This signal is the processor's response to BIRQ L true. This signal is daisy-chained such that the first requesting device blocks the signal propagation while non-requesting devices pass the signal on as BIAKO L to the next device in the chain. The leading edge of BIAKI L causes BIRQ L to be unasserted by the requesting device.
8	BIRQ L	ASYNCHRONOUS BUS INTERRUPT REQUEST from a device needing interrupt service. The request is generated by a true RQST signal along with the associated true interrupt enable signal. The request is removed after the acceptance of the BDIN L signal and on the leading edge of the BIAKI L signal or the removal of the associated interrupt enable or due to the removal of the associated request signal.
10 17	REQSTB H REQSTA H	DEVICE INTERRUPT REQUEST SIGNAL. When asserted with the enable "A" flip-flop asserted will cause the assertion of BIRQ L on the bus. This signal line normally remains asserted until the request is serviced.

Table A-1 DC003 Pin/Signal Descriptions (Cont)

Pin	Signal	Description
11 16	ENB ST H ENA ST H	INTERRUPT ENABLE "A" STATUS signal. This signal indicates the state of the interrupt enable "A" internal flip-flop which is controlled by the signal line ENA DATA H and the ENA CLK H clock line.
12 15	ENB DATA H ENA DATA H	INTERRUPT ENABLE "A" DATA signal. The level on this line, in conjunction with the ENA CLK H signal, determines the state of the internal interrupt enable "A" flip-flop. The output of this flip-flop is monitored by the ENA ST H signal.
13 14	ENB CLK H ENA CLK H	INTERRUPT ENABLE "A" CLOCK. When asserted (on the positive edge), interrupt enable "A" flip-flop assumes the state of the ENA DATA H signal line.

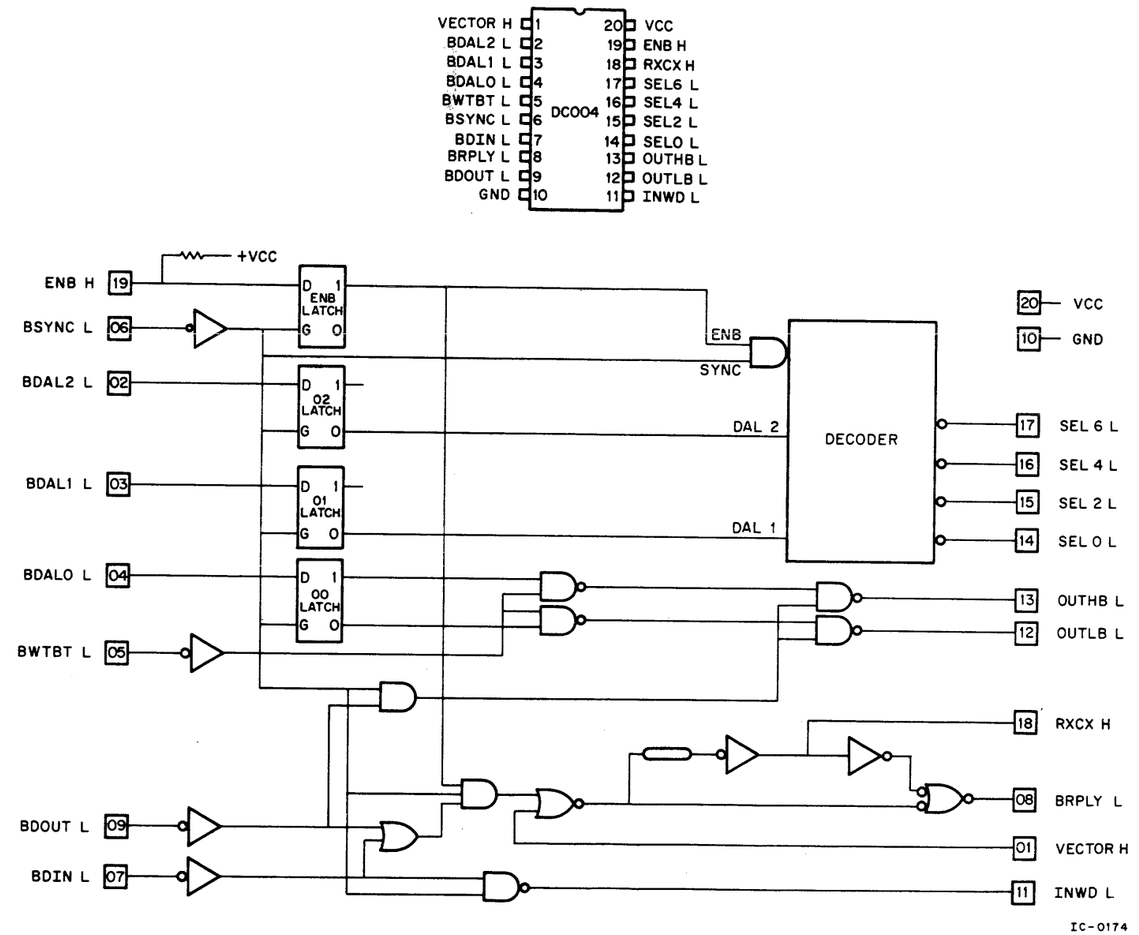


Figure A-4 DC004
Simplified Logic Diagram

Table A-2 DC004 Signal Timing vs Output Loading

	Signal	With Respect to	Signal	Test Cond.	Output Being Asserted Min (ns)	Output Being Asserted Max (ns)	Output Being Asserted Min (ns)	Output Being Asserted Max (ns)	Fig. A-5 Ref.
	SEL (0,2,4,6) L		BSYNC L	Load B	15	35	5	25	t_5, t_6
				Load C	15	40	5	30	
	OUTLB L		BDOUT L	Load B	5	25	5	25	t_9, t_{10}
				Load C	5	30	5	30	
	OUTHB L		DBOUT L	Load B	5	25	5	25	t_9, t_{10}
				Load C	5	30	5	30	
	INWD L		BDIN L	Load A	5	25	5	25	t_{11}, t_{12}
				Load B	5	30	5	30	
Pin 18 Connection RX = 1K ±5% 350Ω ±5% 15 pf ±5%	BRPLY L (Load A)		OUTLB L (Load B)		20	60	-10	45	t_{13}, t_{14}
	BRPLY L (Load A)		OUTHB L (Load B)		20	60	-10	45	t_{13}, t_{14}
	BRPLY L (Load A)		INWD L (Load B)		20	60	-10	45	t_{13}, t_{14}
	BRPLY L (Load A)		VECTOR H		30	70	0	45	t_{13}, t_{14}
Pin 18 Connection RX = 4.64K ±1%	BRPLY L (Load A)		OUTLB L (Load B)		300	400	-10	45	t_{13}, t_{14}
	BRPLY L (Load A)		OUTHB L (Load B)		300	400	-10	45	t_{13}, t_{14}

Table A-2 DC0004 Signal Timing vs Output Loading (Cont)

	Signal	With Respect to	Signal	Test Cond.	Output Being Asserted Min (ns)	Output Being Asserted Max (ns)	Output Being Asserted Min (ns)	Output Being Asserted Max (ns)	Fig. A-5 Ref.
CX = 220 pf ±1%	BRPLY L (Load A)		INWD L (Load B)		300	400	-10	45	t_{13}, t_{14}
	BRPLY L (Load A)		VECTOR H		330	430	0	45	t_{13}, t_{14}

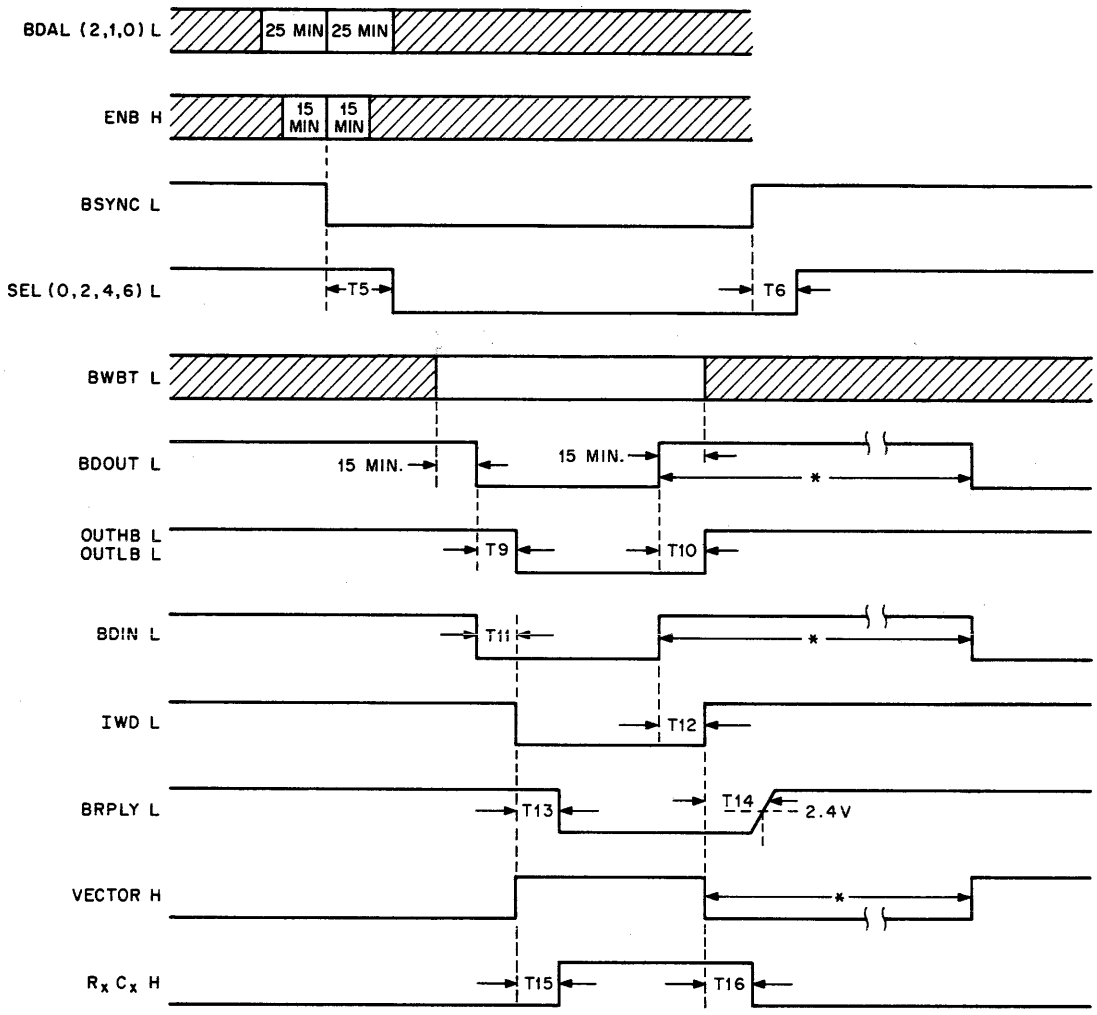


Figure A-5 DC004 Timing Diagram

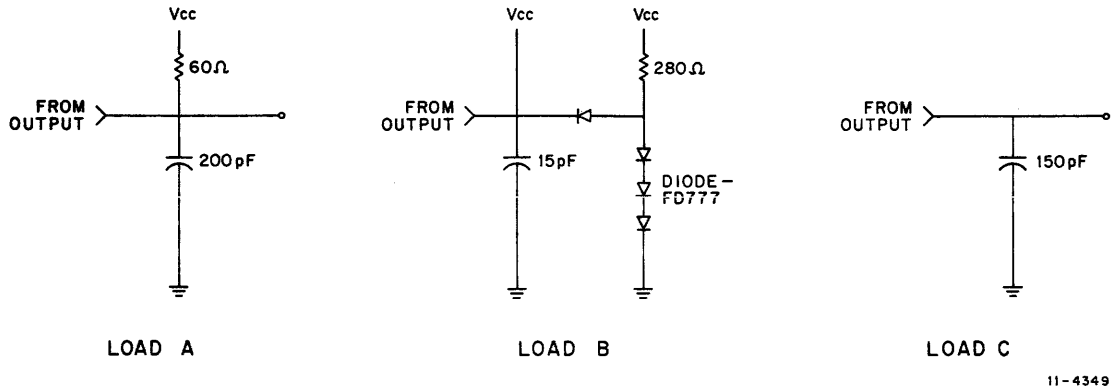


Figure A-6 DC004 Loading Configurations for Table A-2

Table A-3 DC004 Pin/Signal Descriptions

Pin	Signal	Description
1	VECTOR H	VECTOR. This input causes BRPLY L to be generated through the delay circuit. Independent of BSYNC L and ENB H.
2 3 4	BDAL2 L BDAL1 L BDAL0 L	BUS DATA ADDRESS LINES. These signals are latched at the assert edge of BSYNC L. Lines 2 and 1 are decoded for the select outputs; line 0 is used for byte selection.
5	BWTBT L	BUS WRITE/BYTE. While the BDOUT L input is asserted, this signal indicates a byte or word operation: Asserted = byte unasserted = word. Decoded with B OUT L and latched BDAL0 L to form OUTLB L and OUTHB L.
6	BSYNC L	BUS SYNCHRONIZE. At the assert edge of this signal, address information is trapped in four latches. While unasserted, disables all outputs except the vector term of BRPLY L.
7	BDIN L	BUS DATA IN. This is a strobing signal to effect a data input transaction. Generates BRPLY L through the delay circuit and INWD L.
8	BRPLY L	BUS REPLY. This signal is generated through an RC delay by VECTOR H, and strobed by BDIN L or BDOUT L, and BSYNC L and latched ENB H.
9	BDOUT L	BUS DATA OUT. This is a strobing signal to effect a data output transaction. Decoded with BWTBT L and BDAL0 to form OUTLB L and OUTHB L. Generates BRPLY L through the delay circuit.
11	INWD L	IN WORD. Used to gate (read) data from a selected register on to the data bus. Enabled by BSYNC L and strobed by BDIN L.
12 13	OUTHB L OUTLB L	OUT LOW BYTE, OUT HIGH BYTE. Used to load (write) data into the lower, higher, or both bytes of a selected register. Enabled by BSYNC L and decode of BWTBT L and latched BDAL0 L, and strobed by BDOUT L.
14 15 16 17	SEL0 L SEL2 L SEL4 L SEL6 L	SELECT LINES. One of these four signals is true as a function of BDAL2 L and BDAL1 L if ENB H is asserted at the assert edge of BSYNC L. They indicate that a word register has been selected for a data transaction. These signals never become asserted except at the assertion of BSYNC L (then only if ENB H is asserted at that time) and once asserted, are not unasserted until BSYNC L becomes unasserted.

Table A-3 DC004 Pin/Signal Descriptions (Cont)

Pin	Signal	Description
18	RXCX	EXTERNAL RESISTOR CAPACITOR NODE. This node is provided to vary the delay between the BDIN L, BDOUT L, and VECTOR H inputs and BRPLY L output. The external resistor should be tied to VCC and the capacitor to ground. As an output, it is the logical inversion of BRPLY L.
19	ENB H	ENABLE. This signal is latched at the asserted edge of BSYNC L and is used to enable the select outputs and the address term of BRPLY L.

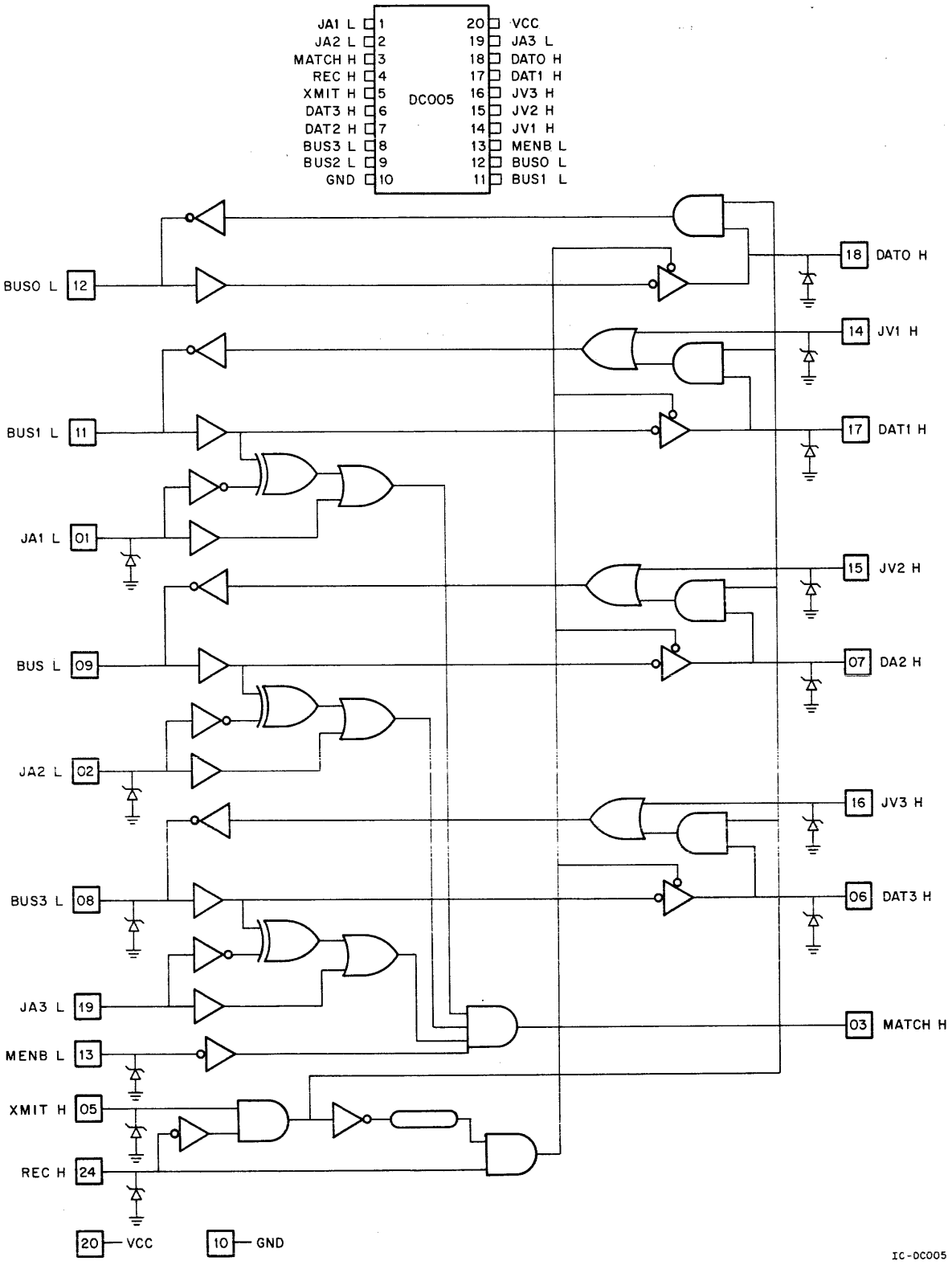


Figure A-7 DC005 Simplified Logic Diagram

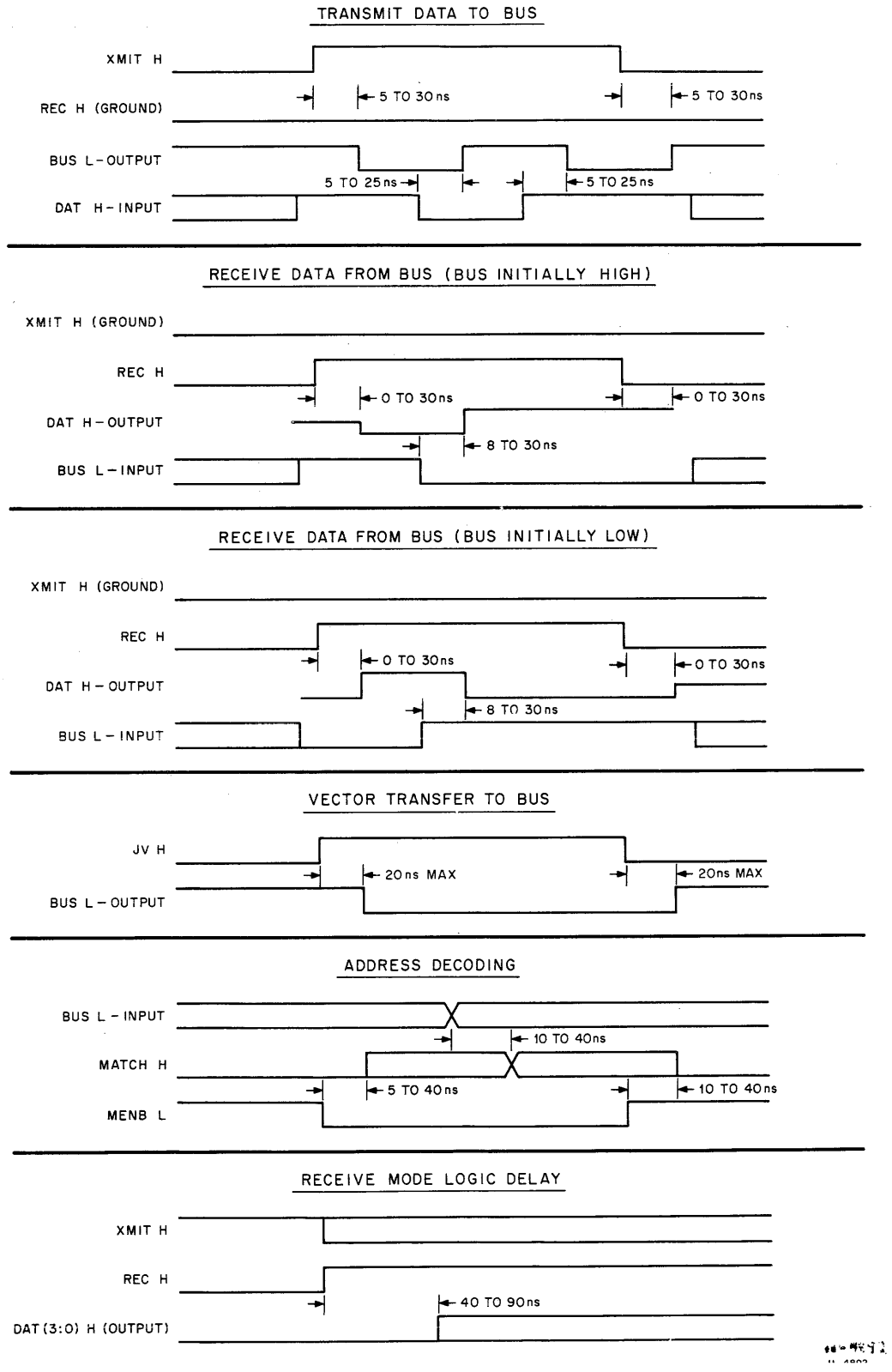


Figure A-8 DC005 Timing Diagram

Table A-4 DC005 Pin/Signal Descriptions

Pin	Name	Function
	BUS<3:0> L	BUS DATA. This set of four lines constitutes the bus side of the transceiver. Open collector outputs; high-impedance inputs. LOW = 1.
12	BUS0	
11	BUS1	
9	BUS2	
8	BUS3	
	DAT<3:0> H	PERIPHERAL DEVICE DATA. These four 3-state lines carry the inverted received data from BUS <3:0> when the transceiver is in the receive mode. When in transmit data mode, the data carried on these lines is passed inverted to BUS ↓3:0>. When in the disabled mode, these lines go open (hi-Z). HIGH = 1.
18	DAT0	
17	DAT1	
7	DAT2	
6	DAT3	
	JV<3:1> H	VECTOR JUMPERS. These inputs, with internal pull-down resistors, directly drive BUS <3:1>. A low or open on the jumper pin will cause an open condition on the corresponding bus pin if XMIT H is low. A high will cause a one (low) to be transmitted on the bus pin. Note that BUS0 L is not controlled by any jumper input.
14	JV1	
15	JV2	
16	JV3	
13	MENB L	MATCH ENABLE. A low on this line will enable the MATCH output. A high will force MATCH low, overriding the match circuit.
3	MATCH H	ADDRESS MATCH. When BUS <3:1> match with the state of JA <3:1> and MENB L is low, this output is open; otherwise it is low.
	JA <3:1> L	ADDRESS JUMPERS. A strap to ground on these inputs will allow a match to occur with a one (low) on the corresponding BUS line; an open will allow a match with a zero (high); a strap to V_{CC} will disconnect the corresponding address bit from the comparison.
1	JA1 L	
2	JA2 L	
19	JA3 L	
5	XMIT H	CONTROL INPUTS. These lines control the operation of the transceiver as follows:
4	REC H	**REC** **XMIT** 0 0 DISABLE: BUS DAT open 0 1 XMIT DATA: DAT BUS 1 0 RECEIVE: BUS DAT 1 1 RECEIVE: BUS DAT To avoid 3-state signal overlap conditions, an internal circuit delays the change of modes between XMIT DATA and RECEIVE mode and delays 3-state drivers on the DAT lines from enabling. This action is independent of the DISABLE mode.

Reader's Comments

IBV11-A LSI-11/INSTRUMENT BUS
INTERFACE USER'S MANUAL
EK-IBV11-UG-001

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgment is it complete, accurate, well organized, well written, etc.? Is it easy to use? _____

_____

_____

_____

What features are most useful? _____

_____

_____

_____

What faults do you find with the manual? _____

_____

_____

_____

Does this manual satisfy the need you think it was intended to satisfy? _____

Does it satisfy *your* needs? _____ Why? _____

_____

_____

_____

Would you please indicate any factual errors you have found. _____

_____

_____

_____

Please describe your position. _____

Name _____ Organization _____

Street _____ Department _____

City _____ State _____ Zip or Country _____

— — — — — — — — — — — — Fold Here — — — — — — — — — — — —

— — — — — — — — — — Do Not Tear - Fold Here and Staple — — — — — — — — —

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

Digital Equipment Corporation
Technical Documentation Department
Maynard, Massachusetts 01754