DX11-B system 360/370 channel to PDP-11 unibus interface maintenance manual

DEC-11-HDXBA-B-D

DX11-B system 360/370 channel to PDP-11 unibus interface maintenance manual

digital equipment corporation • maynard, massachusetts

1st Edition, August 1972
2nd Printing (Rev) March 1973
3rd Printing, July 1973

CONTENTS

ILLUSTRATIONS

ILLUSTRATIONS (Cont)

TABLES

FOREWORD

This manual provides the information necessary to maintain the DX11-B, System 360/370 Channel to PDP-11 Unibus Interface. Chapter 1 contains introductory information with brief specifications. For more information concerning performance characteristics, refer to the Engineering Specification, Spec No. KFS-DX11-B-31. Chapter 2 of this manual is concerned with the basic information needed to unpack and install the equipment, although specific reference should be made to the individual Customer Site Plan for the particular installation. Included are general programming procedures required in testing, following installation. Although Chapters 3 and 4 provide considerable information as to formats, programming considerations, etc., further reference should be made to the programming documentation.

This manual is primarily concerned with the operation of the DX11-B hardware and the ways in which it interacts with its specially designed software. Chapter 5, Theory of Operation, provides information about the 360/370 Channel operation as it bears on the interface with the DX11-B. Included are detailed discussions of the functional parts and the various sequences of operation. Some of the more obscure aspects of the hardware are also discussed.

This is not a basic manual, some prior knowledge of the IBM 360/370 is required, together with familiarity with the PDP-11 System, particularly with regards to the Unibus Interface. The following table lists related PDP-11 and IBM documents applicable to the DX11-B System.

Title	Number	Coverage
PDP-11 Unibus Interface Manual, 2nd Edition	DEC-11-HIAB-D	How to interface to the PDP-11 Unibus
IBM System/360 and System/370 I/O Interface Channel to Control Unit OEMI.	GA22-6974	How to interface to an IBM 360/370 Channel
IBM System/360 Principles of Operation	GA22-6821	How the 360 works

CHAPTER 1
GENERAL DESCRIPTION

1.1 SCOPE

This chapter describes a typical DX11-B System in the most general of terms. Included is a listing of its mechanical, electrical, environmental, and performance specifications.

1.2 INTRODUCTION

The DX11-B, System 360/370 Channel to PDP-11 Unibus Interface, shown in Figure 1-1, is a data and status transfer controller that provides communication between an I/O channel of a System 360 or System 370 IBM Computer and software within a Digital Equipment Corporation PDP-11 Computer. The system is "table-driven" for convenience and efficiency. When installed, the DX11-B responds automatically to any 360/370 I/O command or status interrogation that is directed to its assigned addresses. It will conduct data transfers (upon PDP-11 program initiation) between the 360/370 Channel and the Unibus, automatically transferring data between the DX11-B buffers and the Unibus locations; participating in data transfers on the 360/370 Channel, and stopping the transfer when the prescribed number of bytes have been transferred. Additionally, the DX11-B communicates with the PDP-11 processor which, in turn, controls the status indications provided to the 360/370 Channel, sets up the DX11-B to execute the commands transmitted by the channel, and provides communication between other devices on the Unibus and the address space used by the DX11-B.

1.3 SYSTEM DESCRIPTION

As shown in Figure 1-2, the typical system that is attached to an IBM processor consists of a channel (CH), the control units (CU) attached to the channel, and the devices attached to each controller. Several channels can operate with one processor. The DX11-B operates as a control unit, routing data from devices that store data in different formats to a channel that provides selection and timing control.

The DX11-B is used to perform some of the functions of a controller. However, much of the functional operation of the controller, such as determining the status of the devices, initiating transfers, and detecting errors, is performed by a PDP-11 programmed processor that is interposed between the DX11-B and the devices with which it operates. In addition, the DX11-B, under control of the PDP-11 processor, can appear to the channel to operate with several different types of devices.

As shown in Figure 1-3, from a functional standpoint, the DX11-B can be considered to be two interfaces: a standard hardware interface to the IBM 360/370 and a standardized software interface (within the PDP-11) to device emulator programs and application programs. The basic package can be programmed to emulate most standard IBM control units with up to 128 devices attached. The emulation only requires the addition of software to the basic package. As a result, the PDP-11 based equipment can be substituted directly for a 360/370 device or devices, with no reprogramming of the 360/370 required. Although hardware and software are described

separately in this manual, because of the interactive characteristics between hardware and software in the DX11-B, they are, in reality, inseparable and *must* be considered as one device in order to gain a complete understanding of the operation of this equipment.

The outstanding features of the DX11-B are:

- Operates on either multiplexer, block multiplexer, or selector channels.
- Emulates up to 128 devices with as many as eight separate sets of contiguous addresses.
- Handles concurrent I/O on all emulating devices.
- Operates with standard 360/370 operating software.
- Is plug-compatible with the standard IBM channel.
- Contains powerful built-in maintenance capability.
- Contains an interface simulator, allowing complete off-line diagnosis.
- Provides a fully buffered I/O (4-byte buffer).
- Handles sequence checking.
- Operates on multiple subchannels.
- Operates on shared or unshared subchannels.
- Operates on selector subchannels.
- Has table-driven status response and burst mode selection.
- Makes available all control unit options.
- Provides hardware fail-soft on program or system latency errors.

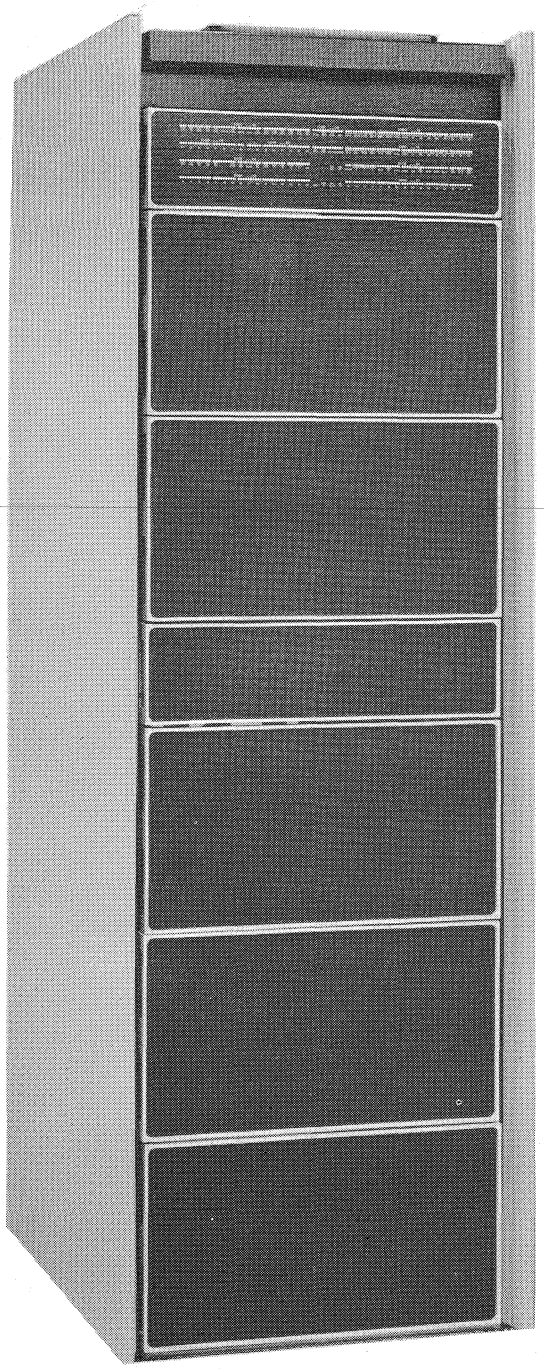


Figure 1-1 DX11-B, System 360/370 Channel to PDP-11 Unibus Interface

1.4 MECHANICAL DESCRIPTION

The DX11-B is housed in a standard 19-in. rack (CAB-11 H960 or H957), containing four H911 Mounting Panels for M-series logic, and IBM 360 connectors for cable connection from the channel and to the next channel unit. Power is provided by an H720e Logic Power Supply, a 716 Power Supply for the indicator panel, and a unit source power panel (EPO) with its power sequence control to the IBM power interface logic. Indicators are provided for logic operation on one panel located on the top front of the unit. Controls are located on the EPO panel available behind a removable front cover, and on the back of the H720e Power Supply, which contains its own power control circuitry. Caravelle fan cooling is provided as part of the basic cabinet. Connections are made between this cabinet and other cabinets containing the PDP-11 Computer and associated equipment. In some installations the PDP-11 can be mounted in the same cabinet with the DX11-B.

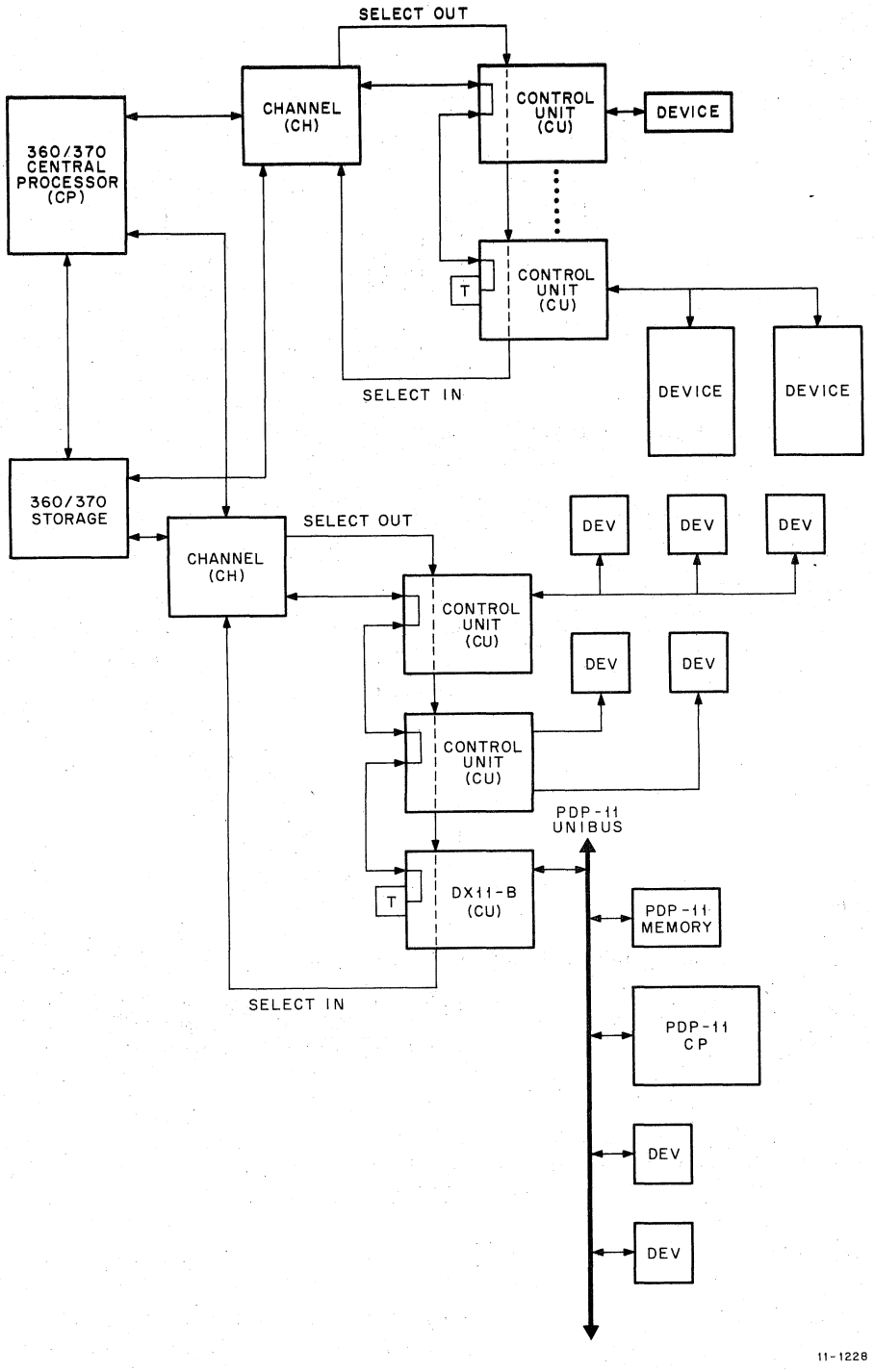


Figure 1-2 Typical DX11-B System Block Diagram

1.5 SPECIFICATIONS

Physical, environmental, and electrical specifications for the DX11-B are contained in the following paragraphs.

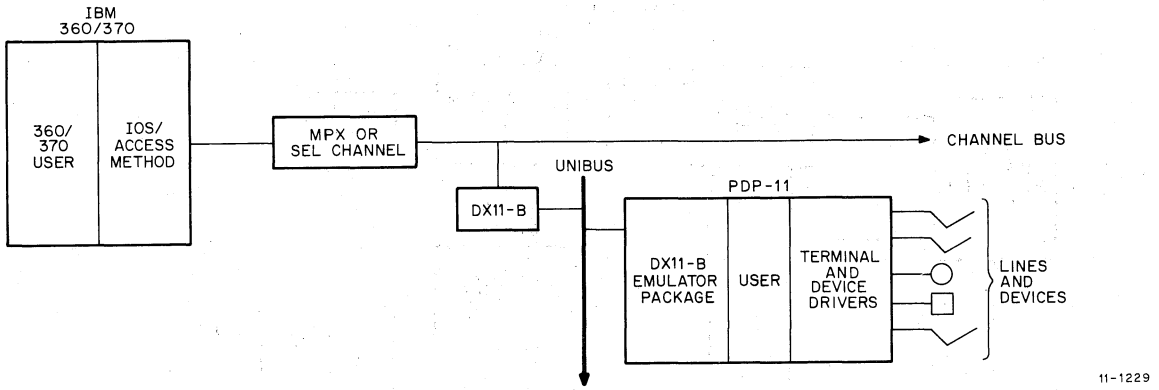


Figure 1-3 DX11-B Functional Block Diagram

1.5.1 Physical

Dimensions

	(CAB H960)	(CAB H957)
Width:	20-3/16 in.	20-3/16 in.
Height:	71-7/16 in.	47-1/2 in.
Depth:	25 in.	25 in.

Service Access Dimensions

Front:	42 in.	42 in.
Rear:	42 in.	42 in.

1.5.2 Environmental

The DX11-B operates under normal conditions of humidity, shock, and vibration.

Ambient Temperature:	50°F to 122°F (10°C to 50°C)
Relative Humidity:	20% to 95% (noncondensing)

1.5.3 Electrical

Estimated Power Consumption (H720e) Logic:	20A @ +5 Vdc
Overall AC:	220 VA @ 120 Vac

1.5.4 Performance

Operating Modes:

Program transfer and non-processor transfer depending upon control unit state and channel function being performed.

Transfer Width:

DX11-B to IBM Channel

Eight bits plus parity for information and six individual lines for selection and control.

IBM Channel to DX11-B

Eight bits plus parity for information and eight bits for selection and control.

Transfer Width (Cont):	**DX11-B to/from PDP-11** All standard Unibus interface signals. Sixteen-bit data path only for data transfers with IBM channel (two bytes/ PDP word). **Parity used between DX11-B and IBM Channel** Odd in both directions.
DX11-B Clock Frequency:	Effective: 2.5 MHz Internal: 5.0 MHz Two discrete time pulses, each standard 50 ns width occurring sequentially 200 ns apart.
DX11-B to CH/CH to DX11-B Transfer Rate:	Nominal 250K bytes/sec depending on system (PDP-11 and 360/370) configuration. The theoretical limiting speed caused by the DX11-B clock is 1M bytes/sec.

1.6 ENGINEERING DRAWINGS

A complete set of engineering drawings is supplied with each DX11-B System. The engineering drawings are bound in a separate volume entitled *DX11-B System Engineering Drawings.* A list of applicable engineering drawings is included in Chapter 7 of this manual. The general logic symbols used on these drawings are described in the *DEC Logic Handbook,* 1970. Specific symbols and conventions are also included in the *PDP-11 Conventions Manual,* DEC-11-HR6B-D. Specialized symbols and terminology, particularly those used in the flow diagrams for the DX11-B, are explained on the drawings as notes and are further defined in Chapter 5.

1.7 TERMINOLOGY

The *PDP-11 Conventions Manual* contains a list of terminology and abbreviations used with the PDP-11 family of systems. A glossary of PDP-11 terms, as well as general computer and programming terms, is also included. A generic glossary of mnemonic terms used in the DX11-B is given in Appendix A of this manual.

NOTE

The documents described in Paragraphs 1.4 and 1.5 are available from the nearest DEC Field Office or from:

Digital Equipment Corporation
Communications Services (Direct Mail)
146 Main Street
Maynard, Massachusetts 01754

CHAPTER 2
INSTALLATION

2.1 SCOPE

This chapter contains information on the installation of the DX11-B System. This chapter is not intended to supersede the installation requirements stipulated in the Customer's Site Plan; however, this chapter supplements that information by providing an installation and checkout procedure to confirm operation of the equipment after it has been installed.

CAUTION
OEMs should not attempt installation until DEC has been notified and a Field Service representative is present. Failure to do so can void equipment warranty.

2.2 CONFIGURATION DIFFERENCES

Installation procedures can vary dependent on the particular system configuration. For example, the DX11-B can be housed in a separate cabinet from the PDP-11, or they can be mounted in the same cabinet. In some installations, a paper-tape reader will be supplied in the second cabinet with the PDP-11; in still other installations, an RC11 and its companion disk unit can be supplied in a third cabinet. The configurations will vary according to customer's requirements and the 360/370 configuration with which the DX11-B is to operate.

A listing of the basic option designations is given in Table 2-1.

Table 2-1
DX11-B Option Designations

Nomenclature	Cab Type	Voltage
DX11-BA	H950	115 Vac
DX11-BB	H950	230 Vac
DX11-BC	H957	115 Vac
DX11-BD	H957	230 Vac

2.3 UNPACKING

The equipment should be unpacked as follows:

1. Place the equipment package within the installation site near its final location. Cut the shipping straps and carefully remove all packing material.

(continued on next page)

2. Remove the machine screws that hold the cabinet to the shipping pallet. Slide the cabinet off of the pallet and move to its final location.

3. Remove the tape holding the power and interconnecting cables to the cabinet floor.

2.4 INSPECTION

Inspect all of the equipment before installing it, checking each piece against the parts list. Any damage must be reported immediately to the shipper and to the DEC representative. Check the power supplies and power wiring for damage in shipment. Seat modules and perform general visual inspection of the equipment.

2.5 SPACE REQUIREMENTS

No special site preparation is required to install the DX11-B; however, when installing the system, make certain that sufficient access is provided at the front and rear of the cabinets for maintenance personnel. If the cabinets are separated by long distances, consider installing overhead trenching ducts or floor ducts for the cabling. Equipment dimensions and access dimensions are given in Paragraph 1.5 and illustrated in Figure 2-1.

2.6 POWER AND CABLE REQUIREMENTS

The DX11-B operates from a line voltage of either 115 Vac at 60 Hz or 240 Vac at 50 Hz, depending on the model ordered. The maximum current required is 600 VA @ 120 Vac. Interconnecting cables to the 360/370 System are not supplied as part of the DX11-B but must be supplied by the customer. BC11-A cables connecting the DX11-B and the PDP-11 Unibus are supplied as part of the DX11-B. See Table 2-2 for a list of interconnecting cables and Figure 2-2 for the connections required. If the DX11-B is located on the end of the 360/370 Channel bus, special IBM terminators should be installed in the IBM connectors located inside the back door of the DX11-B cabinet. Install IBM part TAG 2282676 in CONTROL (TAG) TO THE NEXT UNIT receptacle, and IBM part BUS 2282675 in TO THE NEXT UNIT receptacle (Figure 2-3). Figure 2-4 illustrates connections if *not* on end-of-bus.

CAUTION

Do not connect 360 cables until the off-line checkout is complete.

Table 2-2

DX11-B List of Interconnecting Cables

Qty Per EQ	Function	MFG. Part Number
1	Power Control Cable	DEC 70-08288-8
1	Unibus Cable	DEC BC11-A
1	EPO Control Cable	IBM 5351178
2	I/O Cables (360 only)	IBM 5353920
3	I/O Cables (370 only)	IBM 5466456

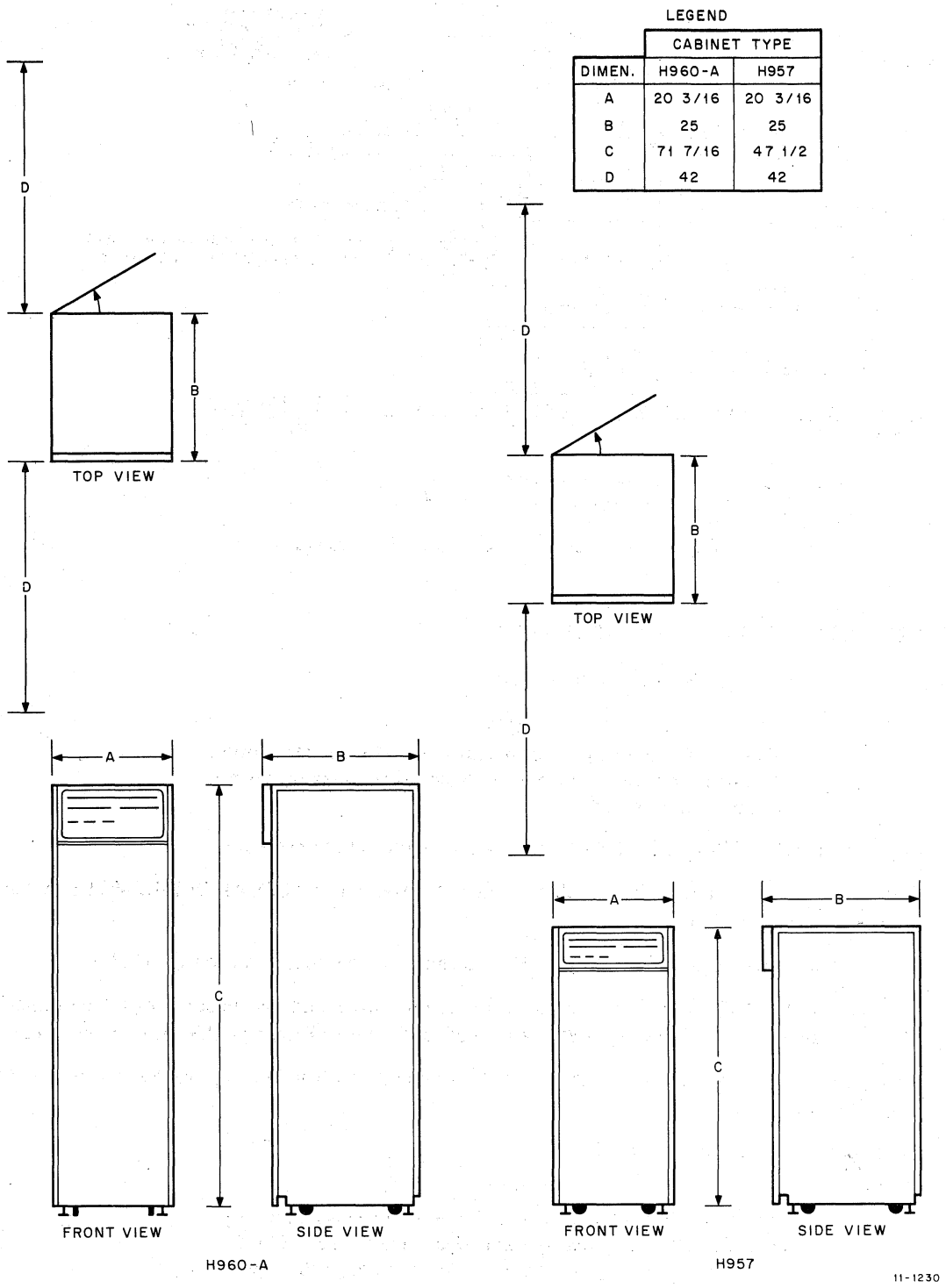


Figure 2-1 Overall Dimensions

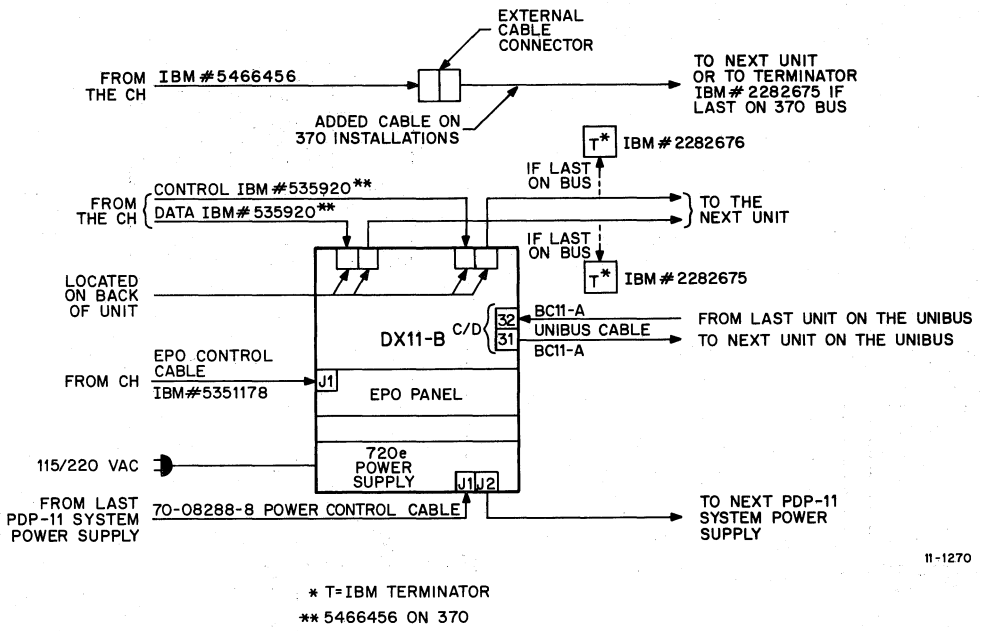


Figure 2-2 System Power and Cabling Diagram

2.7 INSTALLATION PROCEDURE AND CHECKOUT

To install and checkout the DX11-B, proceed as follows:

NOTE

The specifics in this procedure apply to 2848 emulation only. As other emulators are made available, addendums to this procedure will be issued.

1. Ensure that the modules, their hold-down bars, and module clips are secure.
2. Ensure, on the H720e Power Supply, that the main breaker and the LOCAL-OFF-REMOTE switch are both in the OFF position.
3. Connect the Unibus cable (if necessary). DX11-B should be the first device on the Unibus.
4. Wire the M908-YB-A20 360 address jumper card for the correct address selection for the customer's 360/370 installation (see Drawing D-BS-DX11-B-26, and refer to Paragraph 2.9 for more information).
5. Check (and rejumper if necessary) the interrupt vector address. The interrupt vector is assigned to the floating vector space.
6. Apply power to the PDP-11 and DX11-B.

NOTE

When testing the DX11-B with the 360/370, it is mandatory that the IBM computer by solely dedicated to the DX11-B and not occupied in any way with its own operating schedule.

7. Load the DX11 diagnostic (MAINDEC-11-DZDXA-[REV]-PB). See the software abstract (MAINDEC-11-DZDXA-[REV]-PB-D) for instructions. Run diagnostic off-line tests.
8. Calibrate M306 NPR timeout mono (refer to Paragraph 2.10 for procedure). Perform Cable Check per Paragraph 2.15.

NOTE

BLLM and DXTO will be calibrated later.

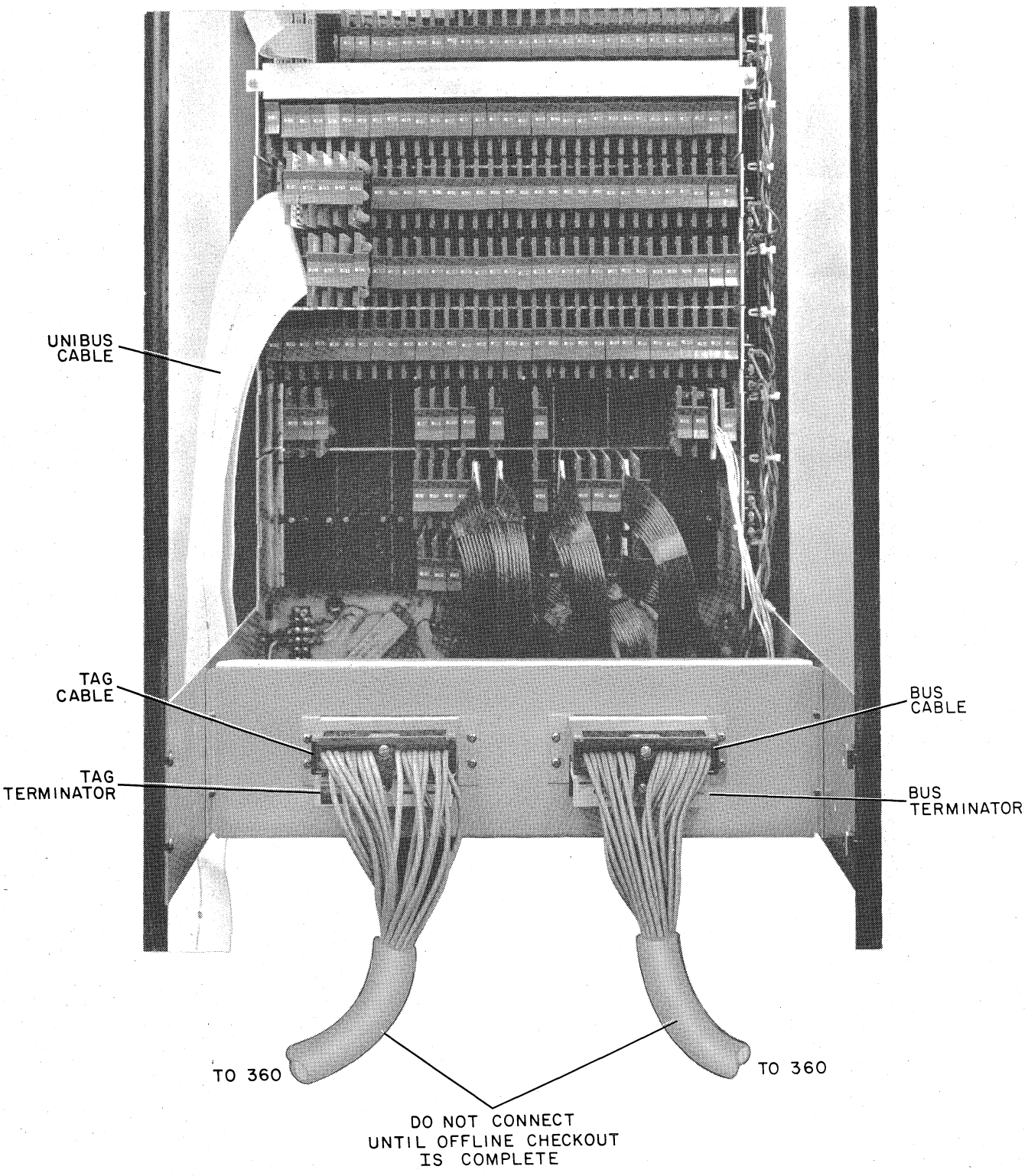


Figure 2-3 Normal Cable Connections to Channel Bus if on End of Bus

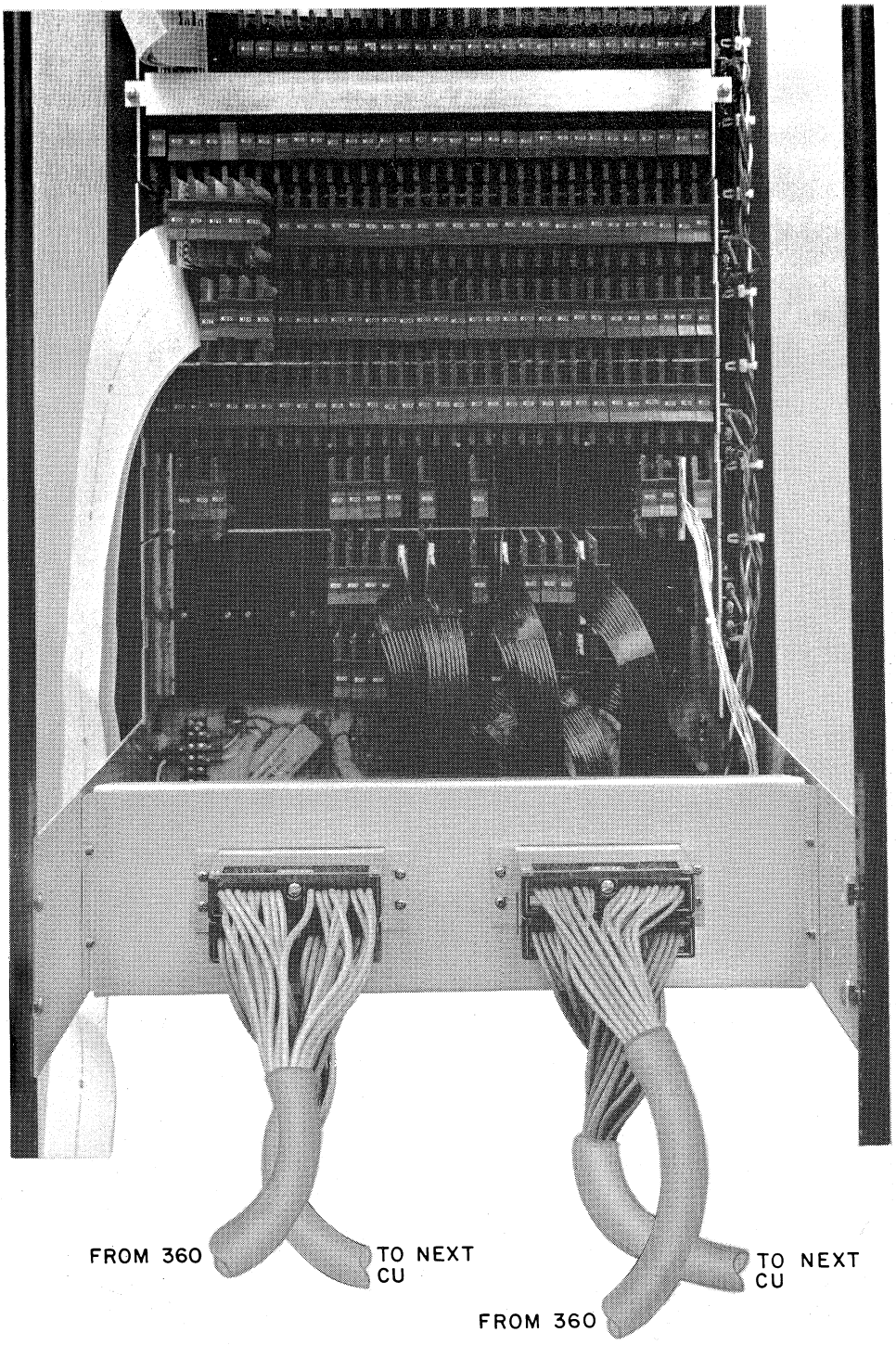


Figure 2-4 Normal Cable Connections to Channel Bus if *Not* on End of Bus

9. Temporarily label cables as "TAG" and "BUS" and attach DX11-B cables for on-line loop-around tests (Figure 2-5). To do this, remove H/J 07 and plug into H/J 17. Remove H/J 14 and plug into H/J 18. Install M984 terminator cards in H 07, J 07, and H 14. Install terminator card M984-YA in J 14. Include customer-supplied cables in this loop by plugging one side of "TAG" cable into the CONTROL (TAG) FROM THE CHANNEL receptacle and its opposite end into the CONTROL (TAG) TO THE NEXT UNIT receptacle. Loop the "BUS" cable in the same manner using the DATA (BUS) receptacles. The purpose of this test is to assure continuity of the customer-supplied cables. If this test fails, use loop test without cables by removing the receptacles from their panel and mating them to form the loop (Figure 2-6). Once mated, secure connectors with their retaining screws.

NOTE

The IBM cables should always be mated dark-to-light and light-to-dark. Interchanging TAG and BUS cables will cause incorrect operation, but will not harm the equipment. Use care in mating these plugs to prevent damage to pins.

10. Run diagnostic on-line tests.

11. Calibrate DXTO by running diagnostic DXTO calibration test (adjust to 5–10 sec). See Paragraph 2.11 for more information.

12. Calibrate BLLM by running diagnostic Fast NPR test. Put oscilloscope on SRVI flip-flop (E32,F2) and adjust BLLM until rising edges of SRVI are 4 μs apart: 250K byte/sec data rate.

13. Run CTP with off-line DX11-B tests and all other CTP-supported devices to checkout system operation.

14. Run GTP with the DX11 overlay (off-line) in 12K systems. Recheck cables per Paragraph 2.15.

15. Reset DX11-B cables to normal operational positions. Move cable connector at H/J 17 to H/J 07. Move cable connector at H/J 18 to H/J 14. Remove terminator cards M984 from H07, J07, and H14 and replace in J29, J30, and J31, respectively. Remove terminator card M984-YA from J14 and replace in J32.

16. Checkout EPO panel operation as follows (Figure 3-1):

    a. Put ENABLE switch in ON LINE position.

    b. Load and start 2848 RESPONDER.

    c. Check that the DX11-B is on-line (ONLINB=1).

    d. Turn the PDP-11 Processor power off and then on. Check that the DX11-B power first goes off, then that the processor is in the Run state, and that the DX11-B is back on-line (ONLINB=1).

    e. Turn the DX11-B LOCAL POWER switch to OFF position. Check that the DX11-B system power goes off.

    f. Turn the LOCAL POWER switch back to ON position. Check that the processor is in the Run state and that the DX11-B is back on-line (ONLINB=1).

    g. Pull the plug from its outlet and see that the same sequence occurs when it is replaced.

17. Connect IBM EPO cable to the DX11-B and put EPO control in SYSTEM position; LOCAL PWR in OFF position. Request that IBM CE power 360/370 up and down. Check for the same indications as described in Step 16. Check PDP-11 power control in REMOTE mode.

(continued on Page 2-9)

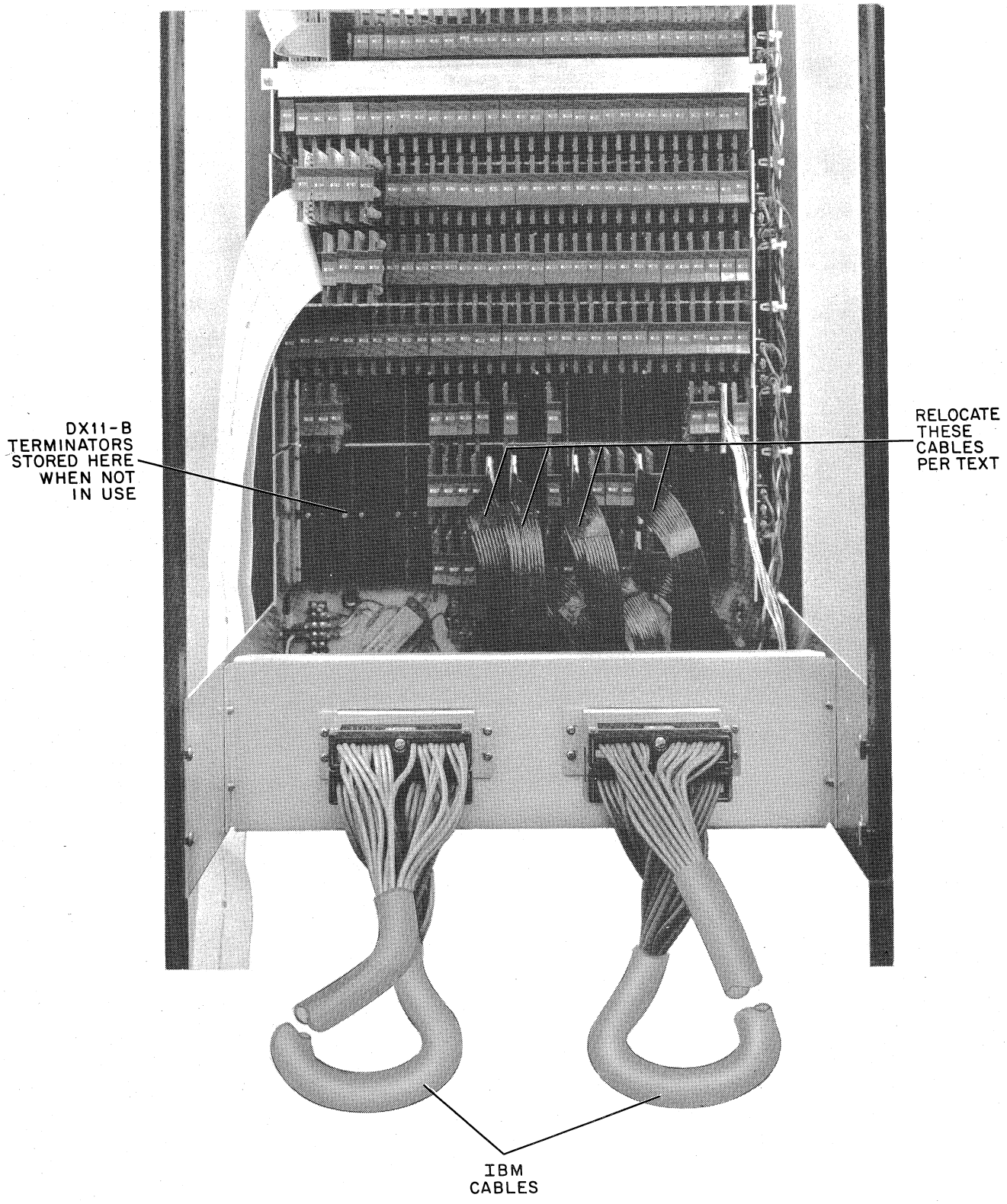


Figure 2-5 Test Bus Cable Connections with IBM Cables in the Loop

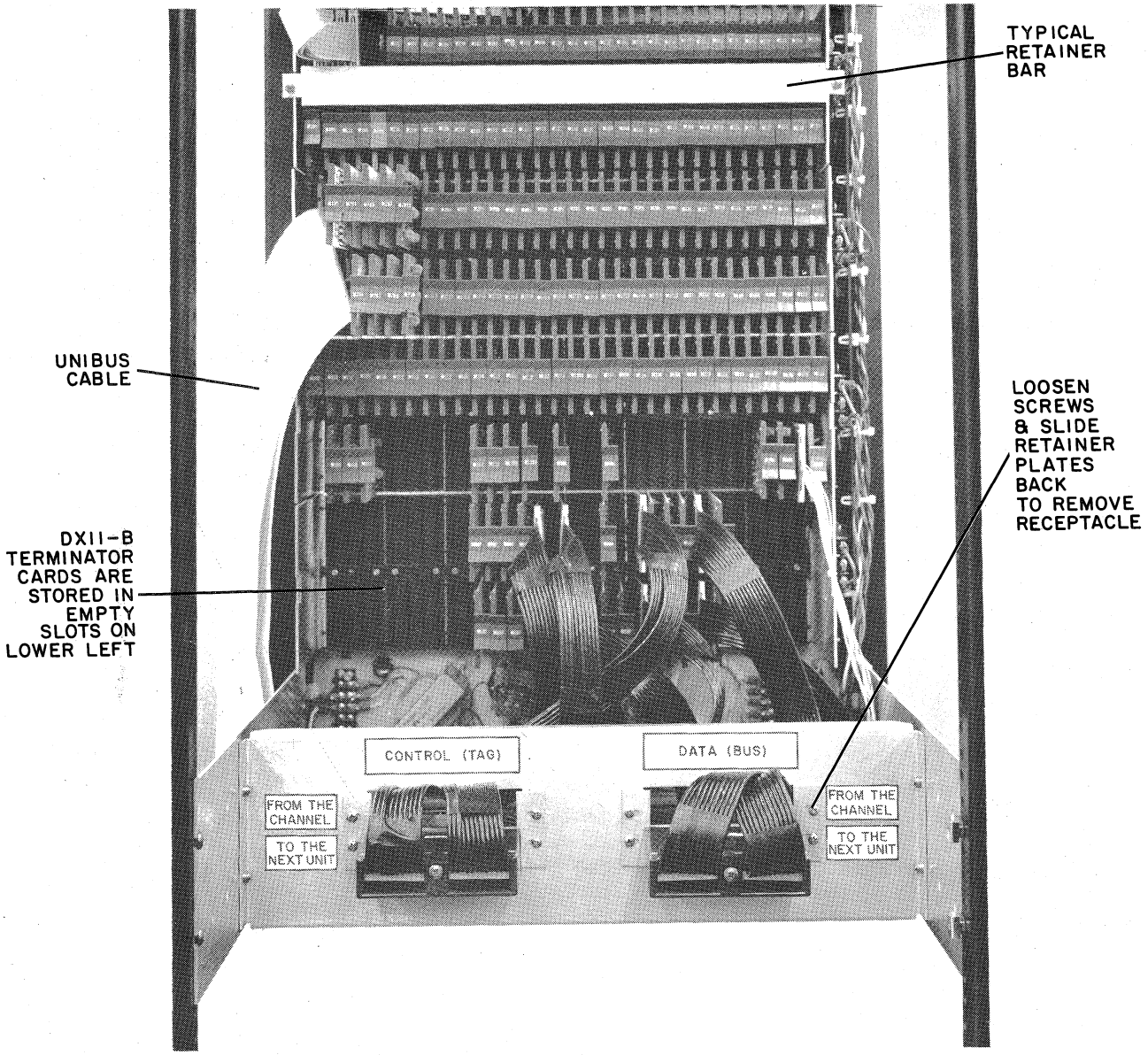


Figure 2-6 Test Bus Connections Without IBM Cables

18. Have IBM CE attach 360 cables to the channel and run channel diagnostics and device diagnostics with cables butted together or terminated (not connected to DX11-B) until he is satisfied that his equipment is operating satisfactorily.

19. Disconnect 360 cables, butted together in Step 15, and connect to the DX11-B. Cable connectors are color coded so that unlike shades mate.

20. With the DX11-B off-line (ENABLE switch in OFF LINE position), but powered up, have IBM CE run diagnostics again to ensure that DX11-B is not interfering with other devices on the channel.

21. Put ENABLE switch in ON LINE position.

(continued on next page)

22. Restart the 2848 RESPONDER in the PDP-11. The DX11-B should be on-line (ONLINB=1) (Paragraph 3.3).
23. Have the IBM CE run diagnostics a final time to ensure that the DX11-B in the on-line mode is not interfering with other 360 devices.
24. With the RESPONDER still running and with switch 14 down, load and execute the IBM 2848 diagnostic in the 360 for the first two DX11-B addresses (Paragraph 3.3).
25. Load FRIEND into the 360 and run the 2848 test procedure (Paragraphs 3.3 and 2.12).
26. On the 360 Selectric Keyboard, press REQUEST to stop FRIEND, load CTP, and then select the DX11-B on-line test.
27. Run all CTP devices on the PDP-11; with FRIEND in the 360, run the CTP test procedure to check total system operation (Paragraphs 3.3 and 2.13).

2.8 CUSTOMER ACCEPTANCE PROCEDURE

To run a customer acceptance procedure, proceed as follows:

1. Run off-line diagnostic.
2. Cable for "on-line loop-around" and run on-line diagnostic.
3. Recable to 360.
4. Run GTP (in systems of 12K or more) with DX11 overlay (off-line).
5. Run CTP with off-line DX11 test.
6. Run CTP with on-line DX11 test driven by FRIEND in the 360.
7. Run 2848 RESPONDER with 2848 diagnostic in 360.
8. With RESPONDER still running, the customer can optionally run 2848 Online Test Program (OLTEP).

2.9 IBM CONTROL UNIT AND DEVICE ADDRESS JUMPER CARD

The M908B Jumper Card located at A19 is shown in Figure 2-7 as it appears at installation, before the control unit and device addresses have been set by jumpering. This card is equipped with split lugs for easy wiring. In wiring this card, any standard wire can be used. The no. 30 wirewrap is carried by all DEC Field Service representatives and is adequate for this procedure.

The I/O address format is shown in Figure 2-8. The DX11-B never sees the 8-bit channel address portion of the I/O instruction. There are four of these instructions by which the 360/370 CP controls I/O operations. These are: 1) Start I/O, 2) Test I/O, 3) Halt I/O, and 4) Test Channel. A Test Channel instruction does not issue an I/O address, but for the other three, the address is generated and decoded as described in Paragraph 5.5.21. Figures 2-9 through 2-11 give these wiring examples.

In Figure 2-9, jumpering pin B to pin V2 and then running the jumper to pins U through M, the first hex digit (1) is decoded by an M155 module at location A19. When pins K, H, E, and B are jumpered to pin C (gnd), the DX11-B ignores the second hex digit causing it to automatically respond to device addresses 10 through 1F.

Figure 2-10 is an example of wiring for the 32 device addresses 50 through 6F. Note that pin F (5x) is tied to pin V2, while pin H (6x) is tied to pin U before being jumpered through pin M. The second digit is responded to as it is in Figure 2-7.

Figure 2-11 shows the wiring for four device addresses in the range 3x. Pins K and H are grounded by jumpering to pin C. Pin L is jumpered to pin B and pin E is tied to pin D.

From these examples, the wiring for any combination can be derived.

NOTE
All addressing, including systems with less than 16 devices, must start on zero boundaries.

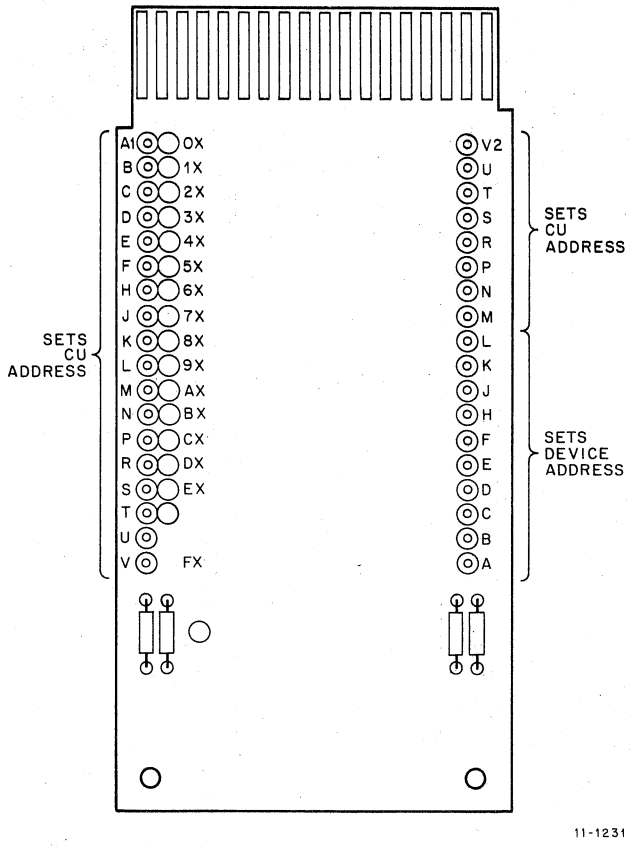


Figure 2-7 M908B Module Before Wiring

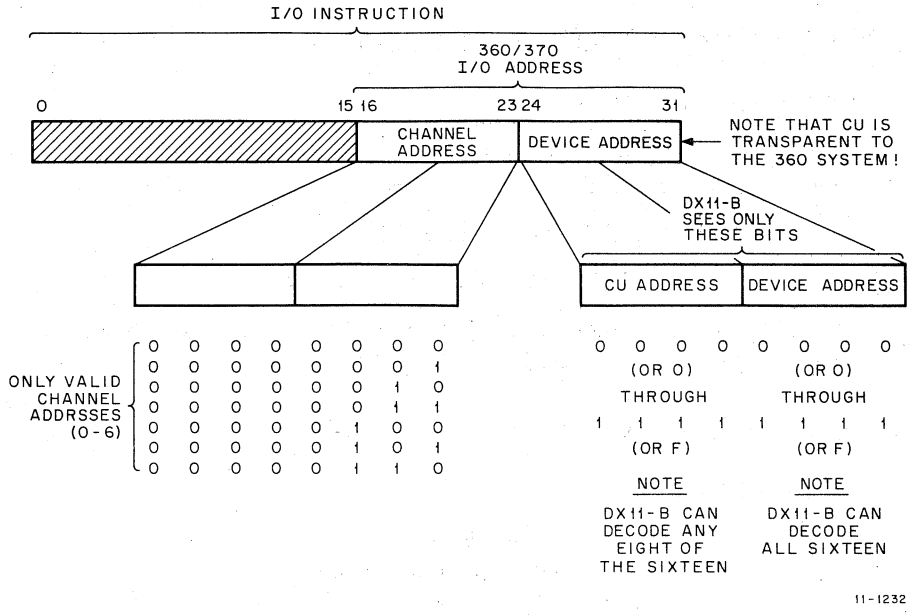


Figure 2-8 I/O Address Format

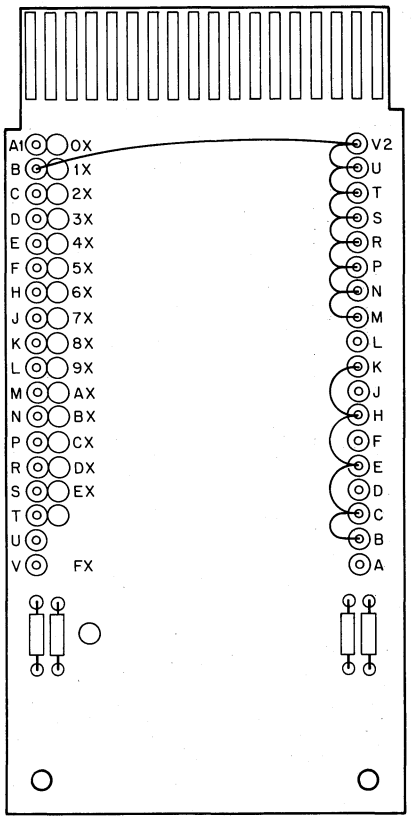


Figure 2-9 Example of Wiring for 16 Devices 10 through 1F (1x)

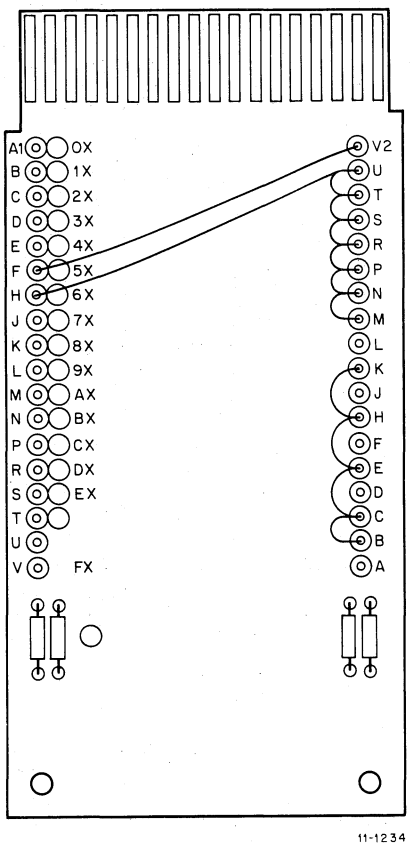


Figure 2-10 Example of Wiring for 32 Devices 50 through 6F (5x and 6x)

2.10 NPR TIMEOUT MONO (M306) CALIBRATION PROCEDURE

To calibrate the NPR Timeout Mono, proceed as follows:

1. Place channel 1 (sync) probe on the output of the NPRDLY Mono (Pin F04S2). Put sync controls to: Ac Low Frequency Rej., pos. sync. Time base to 100 μs/div.
2. Place channel 2 probe on the NPRX flip-flop (pin F04H2).
3. Load the diagnostic and run the NPRTO calibration procedure.
4. Adjust the mono potentiometer in slot F04 to 12 μs. (NPRX(1) L neg. edge to NPRDLY). Put X10 mag. on to make measurement (Figure 2-12).

2.11 DXTO CALIBRATION PROCEDURE

To calibrate the DX timeout, proceed as follows:

NOTE
This is a visual checkout procedure.

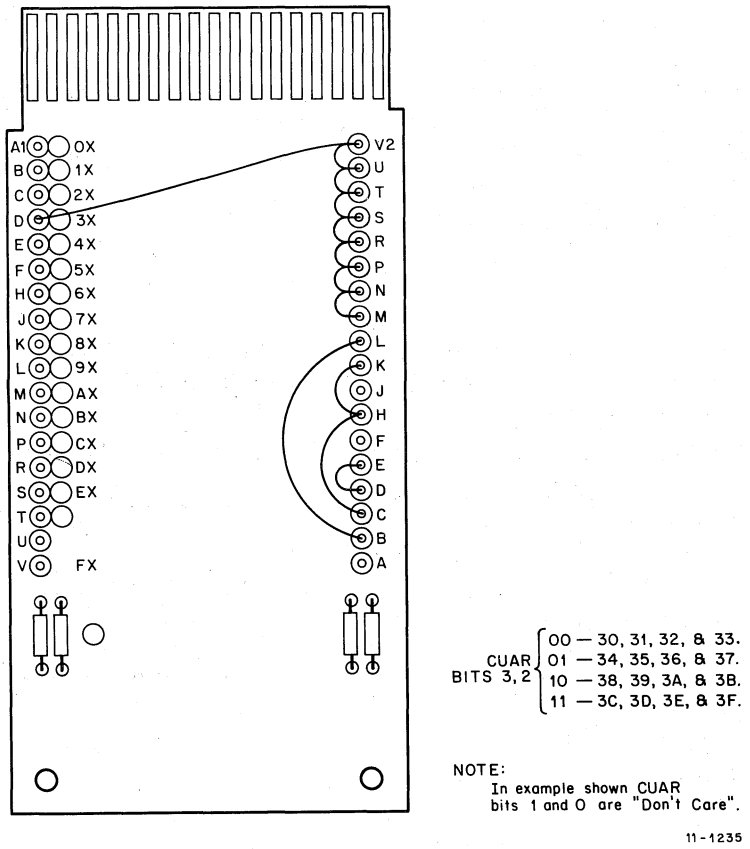


Figure 2-11 Example of Wiring for Four Devices

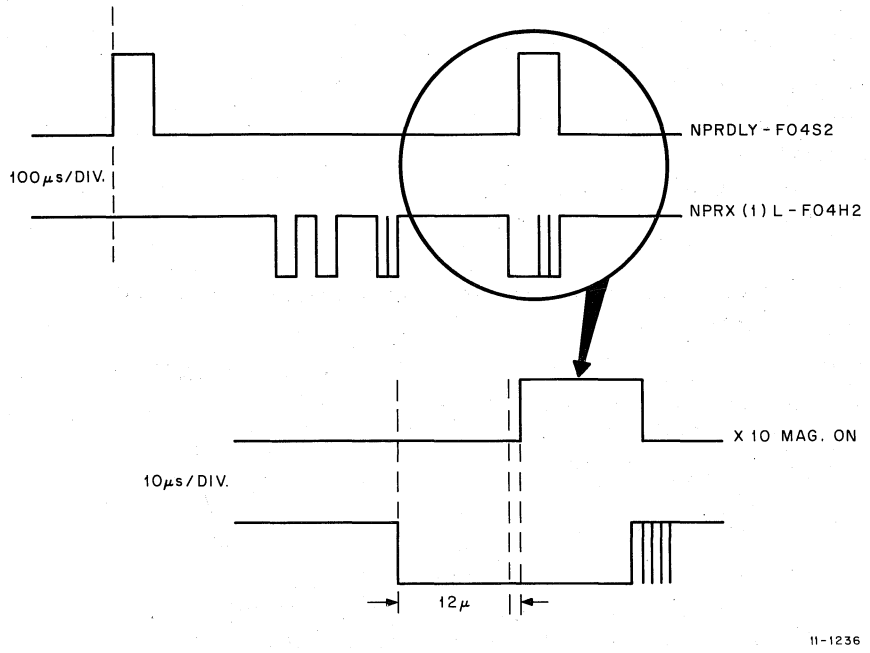


Figure 2-12 NPR Timeout Mono Calibration Waveform

1. Place processor in HALT state.
2. Hit START to cause DX11 initialization.
3. Set switch register to 176216 and hit LOAD ADDRESS.
4. Set switch register (SR) to 100000 and hit DEPOSIT to cause Operational-In to set.
5. Time approximately 5 sec. At this time, DXTO should set.
6. Repeat Steps 2–5 to calibrate DXTO mono M306-F03 to approximately 5 sec.

2.12 2848/FRIEND TEST PROCEDURE

To conduct a 2848/FRIEND Test Procedure, proceed as follows:

NOTE

On the 360 all console input is terminated by [EOB], obtained by pressing both ALT mode and 5. On the 370, hit END. This refers to every line entered.

1. Load FRIEND into 360 (see Paragraph 3.3.2 for 360 load procedure) and reply to CPU model.
2. Run 2848 RESPONDER in PDP-11 (see Paragraph 3.3 for operating RESPONDER).
3. Type following to FRIEND (underline is program query):

```
DEV= XXX [EOB]          (XXX = DX11 low-order address)
CMD 01 from $A [EOB]
DATA= 480C0 [EOB]
CMD 06 into $B [EOB]          includes CH number
DL= 480 [EOB]
COMPARE $A, $B [EOB]
GO [EOB]
```

NOTE

This performs a "write" chained to a "read", then compares the data read back to see that it equals the data written.

4. Hit REQUEST on 360 console.
5. Type following to FRIEND (if DX11 has more than one contiguous device address):

```
DEV= YYY [EOB]          (YYY = XXX + 1)
CMD 01 from $C [EOB]
DATA= 480C1 [EOB]
CMD 06 into $D [EOB]
DL= 480 [EOB]
COMPARE $C, $D [EOB]
GO [EOB]
```

NOTE

This performs the same operation as that above sequentially on the next device.

6. Hit REQUEST, type RESET to terminate test. FRIEND should loop continuously until terminated by this step. Errors will be reported on IBM console typewriter.

2.13 CTP/FRIEND SYSTEM TEST

To perform a CTP/FRIEND System Test, proceed as follows:

NOTE

On the 360 all console input is terminated by [EOB]. On the 370, hit END. This refers to every line entered.

1. Load FRIEND and reply to CPU model.
2. Load and start CTP for on-line DX11 test and all other CTP-supported devices.
3. Type following to FRIEND:

```
DEV= XXX [EOB]          (XXX is DX11 low-order address)
CMD 01 from $A [EOB]
DATA= 256Xff [EOB]
CMD 02 into $B [EOB]
DL= 256 [EOB]
COMPARE $A, $B [EOB]
GO [EOB]
```

Includes the CH number

4. Hit REQUEST and type RESET to end test.

2.14 CAUTIONS AND WARNINGS

Various cautions and warnings concerning the installation and operation of the DX11-B have been gathered here for easy reference. As field experience is gained in the use of this equipment, revisions and additions to this list will be made. In this paragraph, a CAUTION is intended to prevent equipment damage or malfunction; a WARNING is intended to prevent bodily harm.

WARNING

The ac voltage to the power supply is also provided to the EPO panel terminals 1, 2, 3. This power is NOT switched at any place in the entire system. It can be removed from the cabinet only by pulling the power plug.

CAUTION

Be extremely careful when working around the backplane pins in the H and J rows. These pins are connected to the IBM cables and carry signals at all times while the 360/370 system is operating (whether or not the DX11-B has power).

CAUTION

Do not remove the G890 Select Bypass Module from the logic rack while the system is cabled to the 360/370 channel. If the module is removed, the SELECT signal path is broken and the 360/370 channel cannot use any control unit attached. If it becomes necessary to remove the G890 while cabled to the channel, follow the directions noted on Drawing DX11-B-29. Do not vibrate the G890.

CAUTION

Do not remove any of the 360/370 cable connectors from the logic rack while cabled to the IBM system. All of the IBM signals (control and data) go through the DX11-B. Pulling a cable would cause the 360/370 System to stop. Also, do not disturb the connectors while connected. If it becomes necessary to remove the cables, proceed as follows:

1. Arrange for the 360/370 System operating personnel to *stop* the system for about 5 minutes.
2. Remove both cables from the control (TAG) side of the connector assembly (power need not be dropped on the 360/370 for this operation).

(continued on next page)

3. Mate the light and dark cables together. Repeat the operation for the data (bus) pair of cables.

4. Notify the IBM operating personnel of completion.

NOTE

Use care to not bend the IBM cable pins during this operation. They bend easily.

CAUTION

Do not leave the EPO connector plug unplugged after system installation is completed. If the EPO plug is left out and the 360 System is powered down and then up, the power sequencing will hang trying to turn on the DX11-B.

2.15 CABLE CHECK

In addition to a confusion factor in polarizing the IBM connector end of TAG and BUS cables, there is a great chance that the module connector ends of these cables can be incorrectly plugged in to the DX11-B if a cable check is not made prior to making these connections.

There are two possible cable connectors supplied in the field for the IBM double-connector end of these cables. Although they are wired identically, they are marked differently and must be inspected before deciding which connector to plug into which slot.

The dark gray cable which utilizes the IBM-manufactured cable is illustrated in Drawing D-IA-7008777-0-0, titled *IBM 360 Cable Assembly.* The manner in which the double connector is marked is illustrated to the right of the photographic portion of this print. The letters B, D, G, and J appear twice, embossed on the outside lips of the connector. The pins are labeled once between the two rows of pins, also embossed with the numbers 2 through 13. There are two rows of 13 pins each for the B/D side of the connector and for the G/J side of the connector.

The other cable supplied in the field is dark brown and is labeled differently. Although the labels are the same, the B, D, G, and J designations appear embossed within circles either side of the center securing hardware. The pin numbers are embossed below the lower set of pins within the connector lip.

The IBM cable comprises two sets of coaxial leads arranged in a flat ribbon, each set terminating in an individual module connector at the other end. The other cable assembly (brown connector) also consists of two sets of cables, but each set is made up of twisted pairs, formed into a round cable that is wrapped with a polyethylene cover.

Note that although the parts of the two different cables are not interchangeable, they are compatible in that they can be mated if necessary.

Polarization of the module connector ends of these cables is performed identically. Note that one set of cables corresponds to the B and D side of the double connector while the other set of cables correspond to the G and J side.

Referring now to Drawing D-BS-DX11-B-27, Sheet 3, the logic slots into which the module connectors are to be inserted can be derived. On that drawing, the rectangle to the right of each connector represents the pin numbers on the double connector to which these signals go. Depending on whether the large connector is labeled TAG or DATA/1st BUS or NEXT BUS, the proper slot can be determined from this diagram.

For example, if the cable in question is 1st BUS/TAG line, the connector for the cable associated with pins G and J should plug into slot H12. The connector associated with pins B and J plugs into slot H14.

Once this check has been made, it is recommended that these module connectors be labeled for future reference since they are moved around for various tests.

CHAPTER 3
OPERATION

3.1 SCOPE

This chapter provides the information necessary to operate the DX11-B. The description is divided into two major parts: Controls and Indicators, and Special Operating Procedures.

3.2 CONTROLS AND INDICATORS

Figure 3-1 shows the DX11-B control and indicator panels. The indicator panel is located at the top front of the unit. The control panel (EPO) is located in the center front of the unit behind a removable panel. The figure is keyed, by index number, to the functional descriptions given in Table 3-1.

The power controls, located inside the cabinet back door on the Type 720e Power Supply, are not included.

Table 3-1
DX11-B Controls and Indicators

Index No.	Name	Function
1	DXDS__00	*Device-Status-Register* – Sixteen lamps which indicate the contents of the DX11-B Device Status Register as follows:
	PARER	*Parity-Error* – Indicates an even parity condition on Bus-Out, when either a command or data information is sent from CH to CU (DX11-B). (Error indicator.)
	NXM (PDPTO)	*Non-existent-Memory* – Indicates more than 20 μs were required to complete a single NPR. (Error indicator.)
	SELRST	*Selective-Reset* – Indicates a malfunctioning DX11-B device. (IBM Reset indicator.)
	SYSRST	*System-Reset* – Indicates the execution of a System Reset sequence. (IBM Reset indicator.)
	INFDSC	*Interface-Disconnect* – Indicates that the channel has performed a disconnect operation with the DX11-B. (IBM Reset indicator.)
	UCHKS	*Unit-Check-Sent* – Indicates that Unit Check has been presented in the status byte. (Channel Status flag.)

(continued on Page 3-3)

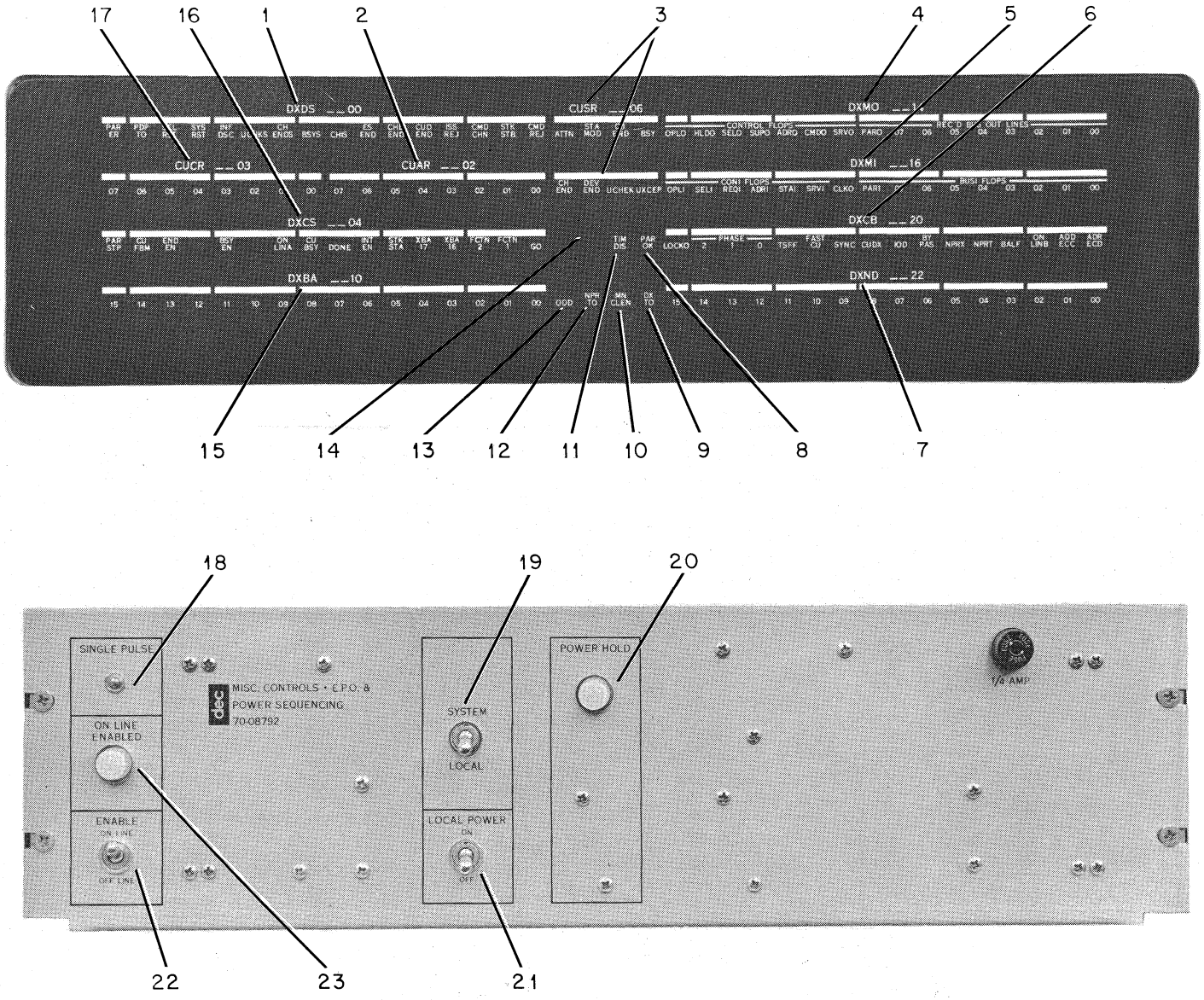


Figure 3-1 DX11-B Controls and Indicators

Table 3-1 (Cont)
DX11-B Controls and Indicators

Index No.	Name	Function
1 (cont)	CHENDS	*Channel-End-Sent* – Indicates to the emulator that Channel End status was sent in a status response. (Channel Status flag.)
	BSYS	*Busy-Sent* – Indicates that Busy status was sent to the channel. (Channel Status flag.)
	CHIS	*Channel-Initiated-Selection-Sequence* – Indicates a channel-initiated sequence is in progress with the DX11-B.
	ESEND	*Ending-Sequence-End* – Indicates a status byte was presented to the channel as a result of either a program-initiated sequence or a stacked status having been accepted. Can indicate a CUI-ISS contention. (CUI End indicator.)
	CHDEND	*Channel-Data-End* – Indicates a channel byte count overflow during a Data Transfer sequence. (CUI End indicator.)
	CUDEND	*Control-Unit-Data-End* – Indicates an all zero state in the DXBC Register during a Data Transfer sequence. (CUI End indicator.)
	ISSREJ	*ISS-Rejected* – Indicates a channel-initiated selection sequence addressed to the DX11-B was answered by a CU Busy status indication and a short CU Busy sequence. (Special indicator.)
	CMDCHN	*Command-Chaining* – Indicates that another CCW operation will probably follow for the CU/Device currently connected when Device End is presented. (Special indicator.)
	STKSTB	*Stack-Status-Byte* – Indicates that the status byte being presented on the Bus-In cannot currently be accepted by the channel. (Special indicator.)
	CMDREJ	*Command Reject* – Indicates the command sent to the DX11-B was illegal. Unit check is presented in the status byte. The rejected command is indicated in the CUCR. (Special indicator.)
2	CUAR__02	*Control Unit Address Register* – Eight lamps which indicate the contents of the CUAR constituting the right-hand byte of the DXCA (Command and Address Register). Contains the device address from Bus-Out during an ISS.
3	CUSR__06	*Control Unit Status Register* – Eight lamps which indicate the contents of the CUSR constituting the right-hand byte of the DXOS (Offset and Status Register). These status bits, transmitted to the channel are as follows:
	ATTN	Attention
	CUEND	Control Unit End
	BSY	Busy
	CHEND	Channel End
	DEVEND	Device End
	UCHECK	Unit Check
	UXCEP	Unit Exception

(continued on next page)

Table 3-1 (Cont)
DX11-B Controls and Indicators

Index No.	Name	Function
4	DXMO__14	*Maintenance Out Register* – Sixteen lamps which indicate the contents of the Maintenance Out Register. This register comprises two subregisters, the left-hand byte constituting the CONO or Control Lines Out Register, and the right-hand byte, the BUSO or Data Lines or IBM Bus-Out Register. These bits are as follows:
	OPLO	Operational-Out – Indicates that the channel is in operation (Selection Control Line)
	HLDO	Hold-Out (Selection Control Line)
	SELO	Select-Out (Selection Control Line)
	SUPO	Suppress-Out (Selection Control Line)
	ADRO	Address-Out (Tag Line)
	CMDO	Command-Out (Tag Line)
	SRVO	Service-Out (Tag Line)
	PARO	Parity-Out (Parity Line)
	REC'D BUS OUT LINES (BUSO)	Indicate the Bus-Out data bits as seen either directly from the Bus-Out cables or from BUSOB if off-line.
5	DXMI__16	*Maintenance In Register* – Sixteen lamps which indicate the contents of the Maintenance In Register. This register comprises two subregisters, the left-hand byte constituting the CONI or Control Lines In Register, and the right-hand byte the BUSI or Data Lines or IBM Bus-In Register. These bits are as follows:
	OPLI	Operational-In (Selection Control Line)
	SELI	Select-In (Selection Control Line)
	REQI	Request-In (Selection Control Line)
	ADRI	Address-In (Tag Line)
	STAI	Status-In (Tag Line)
	SRVI	Service-In (Tag Line)
	CLKO	Clock-Out (from Bus-Out)
	PARI	Bus Parity In
	BUSI	Indicates the data bits enabled to the Bus-In lines for transmission back to the IBM 360 Channel.
6	DXCB__20	*Control Bits Register* – Sixteen lamps which indicate the contents of the Control Bits Register as follows:
	LOCKO	*Lockout* – Indicates that the first five programmable registers are made read-only to the PDP-11 program.
	PHASE 2,1,0	Indicate the existing major phase of the DX11-B.
	TSFF	*Time State Flip-Flop* – Indicates the state of the clock control (time state) flip-flop. 1 = TS1 0 = TS2

(continued on next page)

Table 3-1 (Cont)
DX11-B Controls and Indicators

Index No.	Name	Function
6 (cont)	FASTCU	*Fast Control Unit Response* – This signal is made available for diagnostic test purposes.
	SYNC	*Synchronization Flip-Flop* – This is a phase synchronizing flip-flop used to disable certain portions of logic within any particular phase.
	CUDX	*Control Unit Data Control* – Indicates that the Control Unit data (CUCR and CUSR) is available to either the bus-out or bus-in lines.
	IOD	*I/O-Done* – Indicates an Input-Output Done condition.
	BYPAS	*Bypass* – Indicates one of the following: a. This is the first byte of data that is being transmitted or received. b. The device status table will not be accessed by this particular channel-initiated sequence. c. No address-out has been received. d. A copy of the parity okay signal (between Phases 2 and 3). e. A copy of status pending (between TS1 and TS2 of Phase 0 during a CHI sequence). f. In TS2, indicates that a fast CU Busy was made in TS1.
	NPRX	*NPR Control Switch* – Indicates that the DX11-B is requesting a non-processor or memory-to-device transfer from the PDP-11. When extinguished, indicates that the transfer has been completed.
	NPRT	*NPR Transfer Direction* – Indicates that data is being sent into PDP-11 core. If extinguished, indicates that data is being taken from PDP-11 core.
	BALF	*Byte Alternator Load Flop* – Indicates that status is pending and that the device for which status is pending has been addressed by the channel. During Phases 5 and 6, it keeps track of which half of the data word is currently being used.
	ONLINB	*On line B* – Indicates whether or not the DX11-B is connected to the I/O interface.
	ADRECC	*Control Unit Address Compare* – Indicates that the control unit portion of the address matches Bus-Out.
	ADRECD	*Device Address Compare* – Indicates that the device portion of address matches Bus-Out.

(continued on next page)

Table 3-1 (Cont)
DX11-B Controls and Indicators

Index No.	Name	Function
7	DXND__22	*NPR Data Register* – Sixteen lamps that indicate each data word that is transmitted to or from PDP-11 core via NPRs.
8	PAROK	*Parity OK* – Indicates odd parity on the channel and device address from the channel to the DX11-B.
9	DXTO	*DX11-B Timeout* – Indicates that while Operational-In was up, the program did not interact for 5 sec. (Program Response Latency error.)
10	MNCLEN	*Maintenance Clock Enable* – (MCLKEN) Indicates that the DX11-B is enabled for maintenance clock mode. When set, the DX11-B will not change time state until Maintenance Clock pulse is set. This flip-flop is pulsed by the SINGLE PULSE switch on the EPO panel. An on and an off condition of the MNCLEN indicator will result in the DX11-B switching from one time state to the other.
11	TIMDIS	*Timeout Disable* – Indicates disabling of DXTO during program debugging.
12	NPRTO	*NPR Timeout* – Indicates that Bus Grant was not received during timeout interval. (NPR Latency error.)
13	ODD	*ODD Flip-Flop* – Copy of DXBA (00). Always zero when presented to Unibus (or read by program). This data (odd buffered address) is saved here to initialize BALF.
14		Two undesignated lamps that can be jumpered to any signal for observation.
15	DXBA__10	*NPR Bus Address Register* – Fifteen lamps that indicate the following: **NOTE** **00 is always (0). See Item 13.** a. During data transfers, points to PDP-11 core location to or from which data will be transferred. b. During CHIS, points to the SPW and device status byte in PDP-11 core. c. When information is to be stored in the Tumble Table (TT), indicates that address.
16	DXCS__04	*Control Unit Status Register* – Fifteen lamps that indicate the contents of the Control Unit Status Register as follows:
	PARSTP	*Parity Error Stop* – Indicates that a data transfer sequence will be terminated upon a Bus-Out Parity error. If out, will not prevent a parity error from being raised, but the sequence will end normally.

(continued on next page)

Table 3-1 (Cont)
DX11-B Controls and Indicators

Index No.	Name	Function
16 (cont)	CUFBM	*Control-Unit-Forced-Burst-Mode* – Indicates that Forced Burst mode has been enabled. (Program set.)
	ENDEN	*Control Unit End Enable* – Indicates that the CUEND will be asserted in the device status presented to the channel during a CU Busy sequence. (Program set.)
	BSYEN	*Control Unit Busy Enable* – Indicates that CUBSY can be immediately set upon responding to a CUI or CHI. (Program set to emulate a single thread control unit.)
	ONLINA	*On line A* – Indicates that the DX11-B has made or is making a request to go on-line to the 360 channel. (Program set.)
	CUBSY	*Control Unit Busy* – Indicates that a channel-initiated sequence will be answered by a control unit sequence. (Program and hardware set.)
	DONE	If INTEN is also set, indicates that an interrupt will be requested. Program cleared only before making an attempt to change registers. In Phases 0 or 7, if reset, LOCKO can also be reset. Can be loaded only in Phases 0 or 7. In Phase 4, TS1 is cleared to reset Interrupt Request. In Phase 4, TS2 is set with NPRX to prevent conflict with a possible program load in Phase 7. **NOTE** **A program set of DONE should only be attempted by a maintenance program as an isolation test of the DX11-B interrupt.**
	INTEN	*Interrupt Enable* – Indicates an interrupt enable condition. The bit is always writable. Should always be on before activating the on-line flip-flop. Should not extinguish until the on-line flip-flop has been cleared. (Program set and cleared only.)
	STKSTA	*Stack-Status* – If set, indicates that status was stacked. If cleared, indicates that status was accepted. Program set when presenting a suppressable status. Automatically set when the CH requires a status to be stacked and the DX11-B will attempt to present it again.
	XBA17 XBA16	*Extended Bus Address* – Two lamps that indicate the extended most significant bits of the Memory Address Register during data input/output. Program loaded and cleared. Complement on a DXBA overflow from a DXBA increment of +2 during a data transfer.

(continued on next page)

Table 3-1 (Cont)
DX11-B Controls and Indicators

Index No.	Name	Function
16 (cont)	FCTN 2 FCTN 1	*Function* – Two lamps that indicate the binary contents of the DX11-B Function Register, which represent the operation desired as follows: FCTN=0 = Reset the DX11-B FCTN=1 = Input data transfer FCTN=2 = Output data transfer FCTN=3 = Present status
	GO	Indicates that the function requested is performed. If FCTN=0, DONE is left cleared; if FCTN≠0, REQI is raised.
17	CUCR__03	*Control Unit Command Register* – Eight lamps indicate the contents of the CUCR constituting the left-hand byte of the DXCA (Command and Address Register). Contains the last command sent by the channel even if it was rejected by the DX11-B.
18	SINGLE PULSE	Momentary toggle switch. Each operation will complement MNCLKF. Two operations will step the DX11-B from one time state to the other.
19	SYSTEM/LOCAL	Two-position toggle switch. In SYSTEM position, puts the DX11-B under control of 360/370 power up sequencing. In LOCAL position, removes the DX11-B from control of 360 power sequencing.
20	POWER HOLD	One lamp that indicates completion of the 360/370 power up sequence.
21	LOCAL POWER	Two-position toggle switch that applies or removes local power to/from the DX11-B.
22	ENABLE ON LINE OFF LINE	Two position toggle switch. In ON LINE position enables the DX11-B to be put on-line by the program, and in OFF LINE position enables the DX11-B to go off-line in accordance with 360/370 protocol.
23	ON LINE ENABLED	One lamp that indicates when the DX11-B is enabled on-line.

3.3 SPECIAL OPERATING PROCEDURES

The following operating guides serve as an aid to maintenance personnel who need to do a limited amount of 360 programming to run diagnostics in the DX11-B System.

3.3.1 2848 RESPONDER Operator's Guide

The 2848 RESPONDER is a closed-loop 2848/2260 Emulator that simulates the operation of two 2260 Display Units. It requires 8K of core in the PDP-11 and a DX11-B. Optionally, a 2400 baud, TTY-compatible display can be connected to the console TTY interface for emulation of one of the two 2260 terminals. The other 2260 is emulated entirely in core (see 2848 RESPONDER abstract for loading procedure).

3.3.2 DMA4 Operator's Guide

DMA4 is an IBM diagnostic monitor used on small 360/30 models. It is loaded as a card deck and loads and executes the 2848 diagnostic from cards.

NOTE
DMA4 configured for the DX11 can be supplied by the customer.

To run the 2848 diagnostic using the DMA4, proceed as follows:

1. Set load switches to card reader.
2. Put DMA4, followed by 2848 diagnostic, in card reader. Hit START and EOF.
3. Hit LOAD. DMA4 will load and the console printer will type:

   WTE DMA4

4. Hit INTerrupt. DMA4 will type UDT configuration, then will begin loading and executing the 2848 diagnostic.

3.3.3 DME Operator's Guide

This card deck includes the DME diagnostic monitor and the 2848 diagnostic.

NOTE
DME configured for a customer installation can be supplied by the customer.

To run the 2848 diagnostic on all machines with 64K or more, proceed as follows:

1. Set 360 LOAD switches to card reader address (usually 00C).
2. Put DME/DIAGNOSTIC (MAINDEC-11-DZDXD-A-C) card deck in card reader. Hit START and EOF on card reader.
3. Ensure that any LCS memory (Large Core Storage) on 360 is off-line.
4. See that the line printer is turned on and hit the blue LOAD button on the 360 console. The DME portion of the card deck will load and the DME will print, on the line printer,

   WTE DME

5. Skip this test if DME is supplied by customer. Hit REQUEST on the console. Wait for the PROCEED light and then type a configuration command as follows:

   UAA.BB.CC.DD.EEEEEE ⟨EOB⟩

   where

   AA is the 360 model (30, 40, 50, 65, 67, or 75)

   For 370 models use the following for AA:

370 Model	AA=
135	40
145	50
155	65
165	75

(continued on next page)

5. (cont) BB=00

CC is as follows for size of 360 main storage:

Storage Size	CC=
64K	05
128K	06
256K	07
512K	08
768K	08
1024K	09

DD is as follows for 360 model:

Model	DD=
360/30	FA
360/40	FA
360/50	C8
360/65	4B
360/67	4B
360/75	19
370/135	19
370/145	19
370/155	08
370/165	08

EEEEEE is as follows for size of 360 main storage:

Storage Size	EEEEEE=
64K	00FFFF
128K	01FFFF
256K	03FFFF
512K	07FFFF
768K	0BFFFF
1024K	0FFFFF

⟨EOB⟩ is ALT CODE/5

6. On the 360 console hit INTerrupt. The configuration will print out on line printer (this printout can be ignored).

7. Hit REQUEST, wait for PROCEED, then enter the following on the console typewriter:

a70,0CDD.1B.X000DD ⟨EOB⟩
a70,0CEE.1B.X000DD ⟨EOB⟩

where

C = the channel on which the DX11-B is connected.
DD = the low-order DX11-B address without the channel number.
EE = DD+1, if the DX11-B is wired for more than one device.
X = 8 if on a Selector (SEL) Channel
0 if on a Multiplexer (MPX) Channel.

(continued on next page)

8. Enter on the console,

   L/B ⟨EOB⟩

   The 2848 diagnostic will load and execute from the card deck. Errors will print on line printer. If all line printers are off-line, error messages will print on console typewriter.

3.3.4 FRIEND Operator's Guide

FRIEND is a 360 card deck that allows the generation of CCWs from information typed in on the keyboard. Once loaded, type in the CCW in mnemonic form. FRIEND then generates CCWs for a particular device or set of devices. When GO is typed, it executes the CCWs.

Proceed as follows:

NOTE

All 360 console inputs are terminated by ⟨EOB⟩, obtained by pressing ALTmode and 5. On 370, hit END. This refers to every line entered.

1. Load FRIEND from the card deck (see Paragraph 3.3.2 for 360 load procedures).
2. FRIEND will type some initial instructions then ask for the CPU model by typing

   CPU MODEL = XX ⟨EOB⟩

3. Answer by typing 30, 40, 50, 65, 67, or 75 for correct CPU model (for all 370 models, reply with 75).
4. Next, FRIEND will ask for a device address by typing

   DEV = XXX ⟨EOB⟩

5. Answer by typing the address (including channel) of the device on which you wish to execute a channel program.
6. Next, type the channel program in the following format:

   CMD XX ⟨EOB⟩

   where

   XX is the CCW op code.

7a. If the command is a Write-type command, FRIEND will ask

   DATA=

   Type the data you wish transferred as follows:

   mmmXhh [EOB]

   where

   mmm = a decimal multiplier that specifies how many times the data is to be reproduced to form the data buffer

   hh = a series of hexadecimal digits that are reproduced mmm times, or,

   type

   mmmCee [EOB]

   where

   ee = a series of EBCDIC characters that are reproduced mmm times.

(continued on next page)

7b. If the command is a Read-type command, FRIEND will ask

DL=

Type in the decimal buffer length to be used as the CCW byte count. Then type [EOB].

8. You can then continue to type additional commands, which will be chained together. Also, you can specify more than one device by typing

Type GO [EOB]
Hit ⟨REQUEST⟩
Type DEV=XXX [EOB]
└ includes the CH address

This terminates the CCW list for the previous device and allows specification of a CCW list for device XXX.

9. When done, type

GO [EOB]

to begin CCW execution. All CCW lists previously specified will be executed simultaneously and repetitively until the REQUEST key is hit.

To initialize, type

RESET. [EOB]

To continue after REQUEST is hit, type

⟨EOB⟩.

Example:

Simultaneously execute a Write chained to a Read for device 010, and a Write for device 011.

FRIEND types

DEV=

You type 010 [EOB]

CMD 01 [EOB]
DATA=100cHI THERE [EOB] where c = character data
CMD 02 [EOB]
DL=800 [EOB]
Type GO [EOB] and hit ⟨REQUEST⟩
DEV=011 [EOB]
CMD 01 [EOB]
DATA=50xFF [EOB]
GO [EOB]

CHAPTER 4
PROGRAMMING

4.1 SCOPE

This chapter presents general programming information for software control of the DX11-B. Although a few typical program examples are included, it is beyond the scope of this manual to provide detailed programs.

This chapter is divided into four major portions:

a. 360/370 and PDP-11 Format Comparison
b. 360/370/DX11-B Communication
c. DX11-B/PDP-11 Communication
d. DX11-B Formats

4.2 IBM 360/370 AND PDP-11 FORMAT COMPARISON

As the DX11-B transfers data between 360/370 and PDP-11 memories, a comparison of their word formats is called for (Figure 4-1).

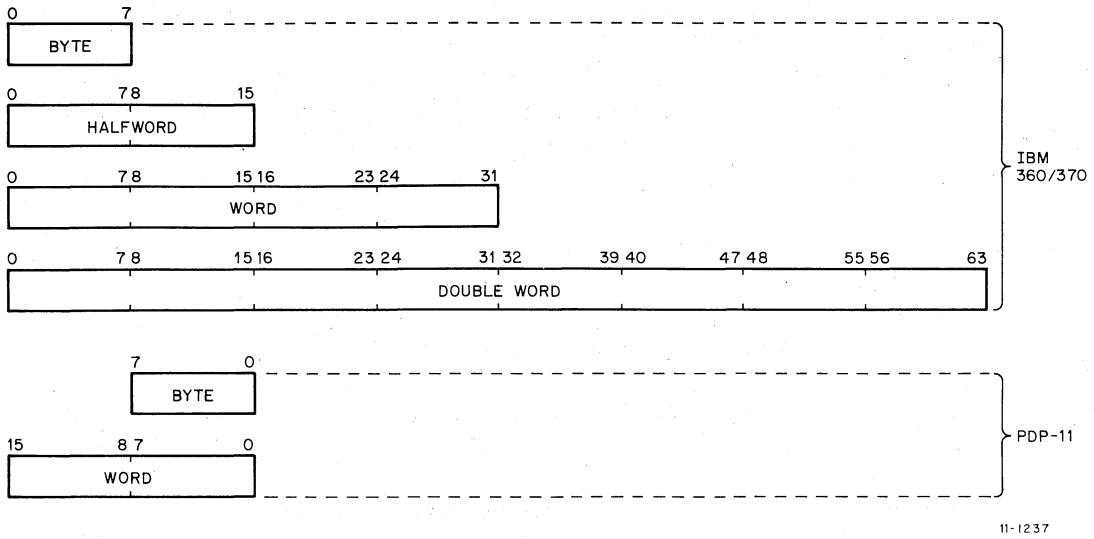


Figure 4-1 IBM 360/370/PDP-11 Data Format Comparison

In the IBM 360/370, information is handled in units of eight bits (byte). A ninth bit (parity) is generated for each byte and carries odd parity on the byte but is not included in references to size of data fields. Bits are numbered from left to right (00–07). A halfword comprises two consecutive bytes (00–15) while a word is four bytes (00–31). The location of any group of bytes is specified by the address of the left-most byte. Partitioning is based on a hexadecimal notation in which each byte represents two hexadecimal digits in 4-bit binary form (0000=0, 1111=F).

PDP-11 information is also handled in 8-bit bytes, but parity is not generated. Bits are numbered from right to left ⟨15:00⟩, with a word comprising two consecutive bytes. The right-most byte ⟨07:00⟩ is designated the low byte and is addressed in even locations. The left-most byte ⟨15:08⟩ is designated the high byte and is addressed in odd locations. Partitioning is based on an octal notation in which three binary bits represent one octal digit (000=0, 111=7). In the PDP-11, the basic unit of data is the byte; the only smaller unit recognized by the hardware is the individual bit.

4.3 IBM 360/370/DX11-B COMMUNICATION

The DX11-B functions as a control unit (CU) in the IBM 360/370 System and is in communication with a channel unit in that system. All transactions between the DX11-B and the channel (CH) are conducted over an I/O interface. The DX11-B accepts control signals from the channel, controls the timing of data transfers over the I/O interface, and provides status of the devices it controls. The interface provides the DX11-B with an information format and a sequence of signals that is common to all control units on that channel. The DX11-B decodes commands from the channel, interprets them for a particular device on the Unibus, and generates a signal sequence necessary to execute a given operation. The channel directs the flow of information between the I/O devices and 360/370 memory.

Communication between the channel and the DX11-B is on an interlocking basis, in which each signal requires a response for the sequence to proceed. The I/O interface contains two buses (Bus-In and Bus-Out), Mark and Tag lines, and Selection and Metering Control lines.

Bus-Out is used to transmit information from the channel to the DX11-B such as data, device addresses, commands, and control orders. Bus-In transmits information in the opposite direction (data, selected I/O device identification, status and sense data). Mark indicates the bus being used. Tag lines are used to identify what kind of information is placed on the bus at any one time. The selection control lines serve a scanning or selection function of attached I/O devices, while the metering lines condition usage meters on the attached units.

4.4 DX11-B/PDP-11 COMMUNICATION

The DX11-B functions as one of the devices on the PDP-11 Unibus, and is in communication with the PDP-11 Central Processor over that bus. The Unibus is similar to the 360/370 I/O Interface except that all devices, memory, and the processor are connected to the same logical set of wires. Whereas on the IBM interface the channel controls all transactions, on the Unibus any device, or the processor, can control the transfer of information.

The DX11-B communicates with the PDP-11 processor in three ways. The most important way is via Non-Processor Request (NPR) cycles for direct access to the core memory. The DX11-B uses this to retrieve device status, store sequence results, and to transfer data.

A second way is via the interrupt which causes a temporary change in the PDP-11 program. The interrupt is used to notify the program that sequence information is now available.

The third type of interaction is program changes to the DX11-B control registers. In this case, the PDP-11 processor becomes Unibus "master" and the DX11-B becomes the "slave" device.

In the IBM System, the channel is always in charge of signal sequences, as data is spoken of as from or to the channel. On the Unibus side of the DX11-B, devices can be either "master" or "slave", depending on whether or not they have initiated an operation. Direction of data transfer in the PDP-11 System is described relative to that device that has momentarily become "master". To avoid confusion, in this manual "360 read" and "360 write" are used to describe the direction of data transfer with respect to the IBM System. The terms "output" and "input" are used to describe direction of data transfer with respect to the PDP-11 System. It can be seen from this that a transfer from 360 to PDP-11 will be a "write" transfer to the DX11-B and an "input" transfer to the PDP-11 (Figure 4-2).

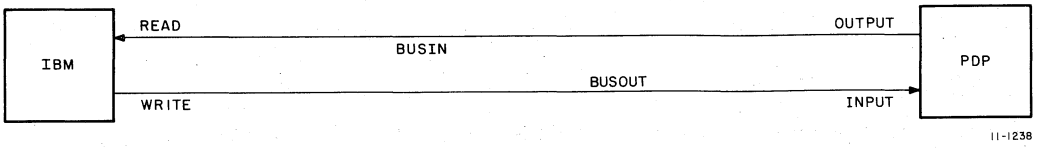


Figure 4-2 360/PDP-11 Transfer Conventions

On the Unibus, only a single sequence is involved in establishing a logical connection; but in the IBM System, data (status information and commands) must be exchanged while maintaining a sequence. Neither method uses a synchronizing clock and both have overall time limitations.

4.5 DX11-B FORMATS

This paragraph lists all of the DX11-B Unibus programmable registers in tabular form (refer to Drawing C-FD-DX11-P-04, Sheet 7 of 8). All bits can be read by the program; all registers are byte addressable. Table 4-1 lists just the registers with their byte numbers (Base Address Offset), mnemonics, word number, full register name, and pertinent information about each. Tables 4-2 through 4-9 break each register down bit by bit, giving a detailed description of each bit's function.

Table 4-1
DX11-B Registers
(Addressable as PDP-11 Memory)

Base Address Offset	Mnemonic	Word No.	Remarks
1,0	DXDS	1	Device Status (TT entry no. 1) (Read-only)
3,2	DXCA	2	Command and Address (TT entry no. 2) (Write lockout)
5,4	DXCS	3	Control Unit Status (loaded by ISR) (Write lockout)
7,6	DXOS	4	Offset and Status (loaded by ISR) (Write lockout)
11,10	DXBA	5	Bus Address (loaded by ISR) (Write lockout)
13,12	DXBC	6	Byte Count (loaded by ISR)
15,14	DXMO	7	Maint Out (simulator) (diagnostic)

(continued on next page)

Table 4-1 (Cont)
DX11-B Registers
(Addressable as PDP-11 Memory)

Base Address Offset	Mnemonic	Word No.	Remarks
17,16	DXMI	8	Maint In (simulator) (diagnostic)
21,20	DXCB	9	Control Bits (control) (diagnostic)
23,22	DXND	10	NPR Data (control) (diagnostic)
25,24	DXES1	11	Extra Signals (control) (diagnostic)
27,26	DXMOB	12	Buffered Bus Out (simulator) (diagnostic)
31,30	DXES2	13	Extra Signals (control)

4.5.1 Device Status Register (DXDS)

This register contains all the interrupt producing conditions, along with various noninterrupt producing device status flags. This register is read only and stored as the first tumble table (TT) entry for this device before reset. These bits and their functions are listed in Table 4-2 and shown in Figure 4-3.

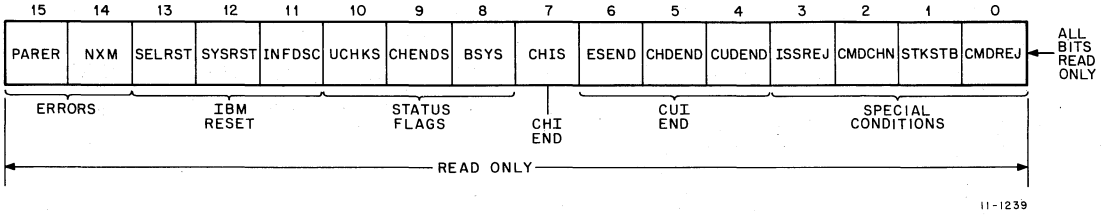


Figure 4-3 Device Status Register Bit Assignments

Table 4-2
DXDS Bit Assignments

Bit No.	Mnemonic	Description
Error Indicators		
15	PARER	*Parity Error* – This bit is set if the DX (also referred to as CU) detects an even-parity condition on the Bus-Out when either a command or data information is sent by the channel (CH) to the CU. The PARER flip-flop will be set if bad Command-Out parity is detected during an ISS or if bad (BUSO) Data-Out parity is detected. The PARER flip-flop is not set on bad Address-Out parity during an ISS. In this case, the CU simply will not recognize the address from the CH. If set during a data transfer while PARSTP is set, this will terminate the sequence by IOD=1 and CUDEND will also be set.

(continued on next page)

Table 4-2 (Cont)
DXDS Bit Assignments

Bit No.	Mnemonic	Description
Error Indicators (Cont)		
14	NXM	*Non-Existent Memory* – This bit will set when the CU takes longer than 20 μs to complete any one Non-Processor Request transaction (NPR), as a result of addressing a non-existent (memory) location. If set during a data transfer, this will terminate the sequence by IOD=1 and CUDEND will also be set. Bus time out is taken as equivalent to bus completion elsewhere, so that a sequence can proceed to its normal ending point.
IBM Reset Indicators		
13	SELRST	*Selective Reset* – This bit will be set by the CH execution of a Selective Reset sequence as described in the *CH to CU OEM Interface Manual* published by IBM. This sequence is usually a response to a malfunctioning CU/device. When set, this bit causes a Program Interrupt (PI).
12	SYSRST	*System Reset* – This bit will be set by the CH execution of a System-Reset sequence as described in the IBM document, *CH to CU OEM Interface Manual.* When set, this bit will cause a PI.
11	INFDSC	*Interface Disconnect* – This bit is set when the channel performs a disconnect operation with the CU.
CH Status Flags		
10	UCHKS	*UC Sent* – This bit is asserted when Unit Check has been presented in the status byte.
09	CHENDS	*CH End Sent* – This bit is used to notify the emulator that CHEND status was sent in a status response.
08	BSYS	*BSY Sent* – BSY status bit was sent to the channel.
CHI End Indicator		
07	CHIS	*Channel-Initiated Selection Sequence End* – This bit is set when a channel-initiated sequence has been completed with the CU. This bit becomes a 0 when the DXDS is reset after the DXDS is entered into the TT.
CUI End Indicators		
06	ESEND	*Ending Sequence End* – This bit is set when a status byte is presented to the channel as a result of a program-initiated sequence or when a stacked status is finally accepted. This bit is most commonly associated with the ending status presentation type of sequence which normally follows a Data Transfer sequence. It may, under some circumstances, occur that the CHIS bit is set as a result of a CUI-ISS contention situation, where the device address requested matches the device address selected from the channel and the CU was requesting to present status.

(continued on next page)

Table 4-2 (Cont)
DXDS Bit Assignments

Bit No.	Mnemonic	Description
CUI End Indicators (Cont)		
05	CHDEND	*Channel Data End* – This bit is set during a Data Transfer sequence when the channel byte count overflows. (Command-Out is sent in response to Service-In.) This bit, in its true state, causes the CU to terminate the Data Transfer sequence. This bit is also set when the CH terminates a Data Transfer sequence by Interface Disconnect (INFDSC=1). It also sets the IOD signal. CHDEND is cleared after being copied into the TT.
04	CUDEND	*Control Unit Data End* – This bit is set during a Data Transfer sequence when the DXBC (Byte Count) goes to its all 0 state. When set, this bit causes a PI and also causes the DONE bit to assert. It also sets the IOD signal. When this bit is asserted, it causes the CU to terminate the Data Transfer sequence.
Special Indicators		
03	ISSREJ	*Initial Selection Sequence Rejected* – This bit is set when a channel-initiated selection sequence addressed to the CU is answered by the CU with a Control Unit Busy status indication and a short Control Unit Busy sequence. This can only occur if the CUBSY bit in the DXCS is set when the CH tries to initiate an ISS and status is not pending for the addressed device. ISSREJ is not cleared with the rest of the DXDS, but remains set until CUE is accepted by the CH.
02	CMDCHN	*Command Chaining* – The bit sets if the CH has indicated that another (CCW) operation will probably follow for the CU/Device currently connected when Device End is presented. Command chaining occurs when the Status-In tag from the CU is answered by Service-Out with Suppress-Out up.
01	STKSTB	*Stack Status Copy* – This bit is set when the CH informs the CU that the status byte being presented on Bus-In cannot currently be accepted by the CH. This occurs when the CH responds to Status-In with Command-Out. STKSTA can be set by the program if the CU is not active with the CH (LOCKO=0). This is useful when initiating a CU request for status presentation. If Suppress-Out and STKSTA are both true, the CU drops its Request-In, since the status contained is suppressable (once status has been stacked the CH also defines that status as suppressable). Since this bit is a copy of the STKSTA flip-flop, it is read-only.
00	CMDREJ	*Command Reject* – This bit is set when the command sent to the DX11-B is rejected. The rejected command is stored in the CUCR.

4.5.2 Command and Address Register (DXCA)

This register contains the Control Unit Command Register, CUCR, and the Control Unit Address Register, CUAR. These two bytes are the command and address as transmitted from the channel during an initial selection sequence. This register is stored as the second tumble table entry for this device upon completion of a CH-CU interaction. These bits and their functions are listed in Table 4-3 and shown in Figure 4-4.

Table 4-3
DXCA Bit Assignments

Bit No.	Mnemonic	Description
15–08	CUCR	The left-hand byte of the DXCA Register contains the CUCR. The CUCR cannot be loaded by the program. This byte contains the last command sent by the channel (even if it was rejected by the CU).
07–00	CUAR	The right-hand byte contains the CUAR. The CUAR is loaded with the device address from Bus-Out during an initial selection sequence. The CUAR is also loaded and cleared via PDP-11 program control if LOCKO is a 0.

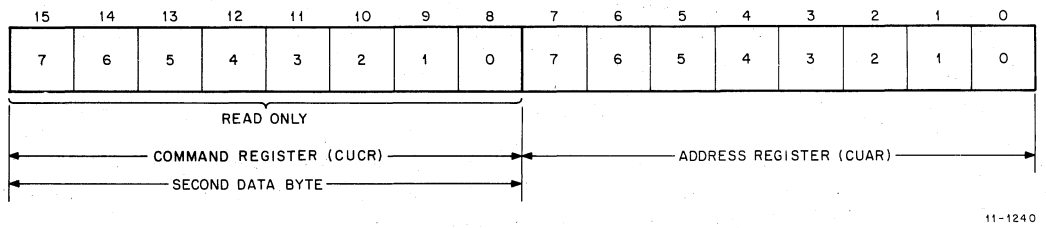


Figure 4-4 Command and Address Register Bit Assignments

4.5.3 Control Unit Status Register (DXCS)

This register contains the primary control command information bits and primary status indications for the DX11-B. The DXCS can not be modified by the PDP-11 program when LOCKO equals 1 (except for DONE and INTEN). These bits and their functions are listed in Table 4-4 and shown in Figure 4-5.

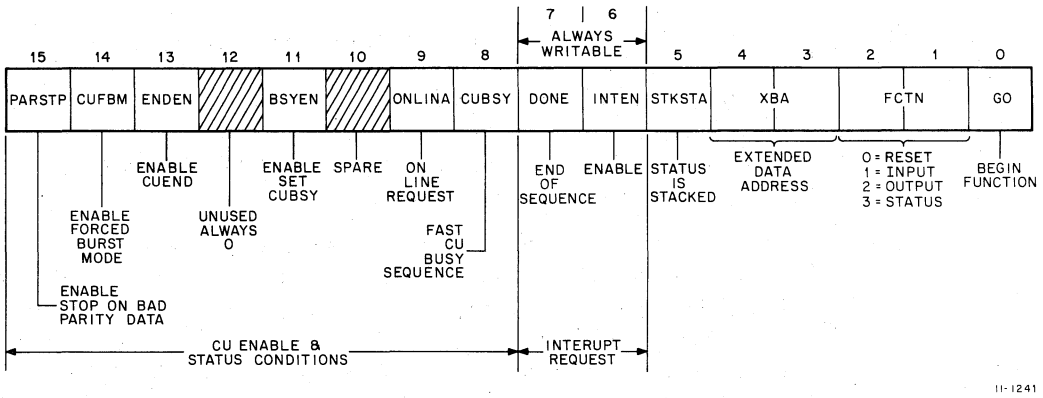


Figure 4-5 Control Unit Status Register Bit Assignments

Table 4-4
DXCS Bit Assignments

Bit No.	Mnemonic	Description
15	PARSTP	*Stop on Parity Error Enable* – If this bit is set to a 1 and a parity error occurs on Bus-Out during a Data Transfer sequence, then the sequence is terminated and CUDEND will be set. If this bit is not set, PARER can still be set but the sequence will end normally.
14	CUFBM	*Control Unit Forced Burst Mode Enable* – This bit can be set by the program when CU Forced Burst mode is desired. This condition causes the CU to hold OPL-IN up from initial selection through the presentation of Channel End (except for TIO and HIO).
13	ENDEN	*Control Unit End Enable* – This bit is set and cleared only by the program. The purpose of setting this bit is to assert CUEND in the device status presented to the channel during a CU busy sequence, i.e., where the CUBSY bit is already set.
12	Not Used	
11	BSYEN	*CUBSY Enable* – This bit enables the setting of CUBSY immediately upon responding to a CUI or CHI. This bit is set to 1 by the program for emulating a single thread control unit.
10		Reserved
09	ONLINA	*On-Line Request Enable* – This flip-flop, when set, indicates that the control unit has made or is making a request to go on-line to the 360 channel. It is a two-level flip-flop. This is the lower stage of the on-line flip-flop which is loaded or cleared by program command (ONLINA: DXCS (09)). The upper stage is the operating on-line flip-flop ONLINB (DXCB (02)). ONLINB follows the changes of ONLINA at a time when, as specified in the OEM channel manual (IBM), it is proper to make changes from on-line to off-line or from off-line to on-line. (The CH is considered on-line itself whenever operational-out is set.)
08	CUBSY	*Control Unit Busy Enable* – This bit set will cause a channel-initiated sequence to be answered by the control unit with a CU Busy sequence. This bit causes the BSY and SM bits to assert to the Bus-In during the subsequent status presentation from the CU. This bit is set and cleared by the program and by the DX hardware if so enabled by BSYEN. Program manipulation of CUBSY is not recommended.
07	DONE	*Sequence Done* – The DONE bit is the normal interrupt producing condition which the DX11 uses for its primary interrupt control (c.f., INTEN). If both DONE and INTEN are set, an interrupt will be requested. Clearing DONE is required (of the program) only before making an attempt to change registers. With DONE reset, LOCKO may also be reset if the DX11 is in either Phase 0 or Phase 7. Loading DONE is allowed only in Phases 0 or 7. In Phase 4, TS1, DONE is cleared to reset the interrupt request. In Phase 4, TS2, DONE is set (also NPRX is set) so as not to conflict with a possible program load in Phase 7. (A program set of DONE should only be attempted by a maintenance program as an isolation test of the DX11 interrupt.)

(continued on next page)

Table 4-4 (Cont)
DXCS Bit Assignments

Bit No.	Mnemonic	Description
06	INTEN	*Interrupt Enable* – This bit is always writable. It is recommended that this bit always be in its 1 state before activating the on-line flip-flop and that the on-line flip-flop be cleared prior to clearing this bit. This bit can be cleared or set by program control only.
05	STKSTA	*Stack Status (c.f., STKSTB)* – If set, STKSTA indicates that status is stacked. When cleared, it indicates that status is accepted. It can also be set voluntarily by a program that is presenting a suppressable (or low priority) status. It is also set automatically by the DX11 when the CH requires a status to be stacked and the CU will attempt to present it again.
04,03	XBA	*Extended Bus Address Bits* – These bits are the two extended, most significant bits of the memory address register during data input/output. They are loaded and cleared under program control and can be caused to complement should the DXBA overflow from a DXBA increment of +2 during a data transfer. They are used only during a data sequence.
02,01	FCTN	*Function* – These two bits make up the DX11 Function Register. They are used by the program to select the CU operations desired: FCTN=0 – reset the DX11 FCTN=1 – input data transfer FCTN=2 – output data transfer FCTN=3 – present status
00	GO	When the GO bit is set, the function requested is performed. If FCTN=0, the reset operation is done on the DX11 and the DONE bit is left cleared. If FCTN≠0 then Request-In (REQI) will be raised at the start of a CUI sequence.

4.5.4 Offset and Status Register (DXOS)

This register contains the offset address of the status pointer word (SPW), which is termed CUOR, and the status byte, termed CUSR. The contents of the CUOR are program loaded for the first (or the left-most) six bits. Bits 9 and 8 of the CUOR are unused and are always zeros.

The CUOR contains the high-order six bits of the SPW table and of the tumble table. The CUOR should be set before an on-line request is made.

The CUSR contains the status information listed in Table 4-5 and shown in Figure 4-6. These bits are transmitted to the channel.

Table 4-5
DXOS Bit Assignments

Bit No.	Mnemonic	Description
07	ATTN	Attention
06	STAMOD	Status Modifier
05	CUEND	Control Unit End
04	BSY	*Busy* – The program should not directly set this bit. This bit is set only by a CU Busy sequence or by being loaded as the status portion of the SPW.
03	CHEND	Channel End
02	DEVEND	Device End
01	UCHECK	Unit Check
00	UXCEP	Unit Exception

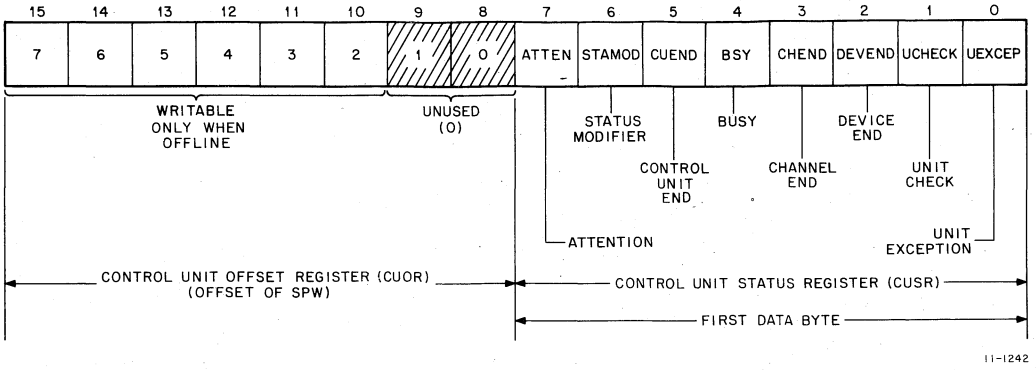


Figure 4-6 Offset and Status Register Bit Assignments

4.5.5 Bus Address Register (DXBA)

This 16-bit register, shown in Figure 4-7, can be cleared and loaded under program control if LOCKO is a 0. It is used during data transfers to point to the PDP-11 core location to or from which data will be transferred in 16-bit words at a time. During a data transfer, the DXBA is preset by the program to point to the first byte location where data is sent or stored. The DXBA is incremented by two each time a PDP-11 data word is fetched or stored in core during the data transfer process. Should the DXBA overflow, the extended memory address bits (XBA) in the DXCS will be caused to complement their states appropriately.

The DXBA is also used during channel-initiated sequences to fetch both the status pointer word and the device status byte from PDP-11 core. In addition, the DXBA is used to address the tumble table when information is to be stored there.

The low-order bit of the DXBA (DXBA 00) is normally set to 0 by program load. When this bit is placed on the Unibus address lines, it is always represented as 0.

During write operations, when data is being sent from the IBM channel to the PDP-11, starting or stopping on an odd byte boundary always puts a garble in the adjacent byte.

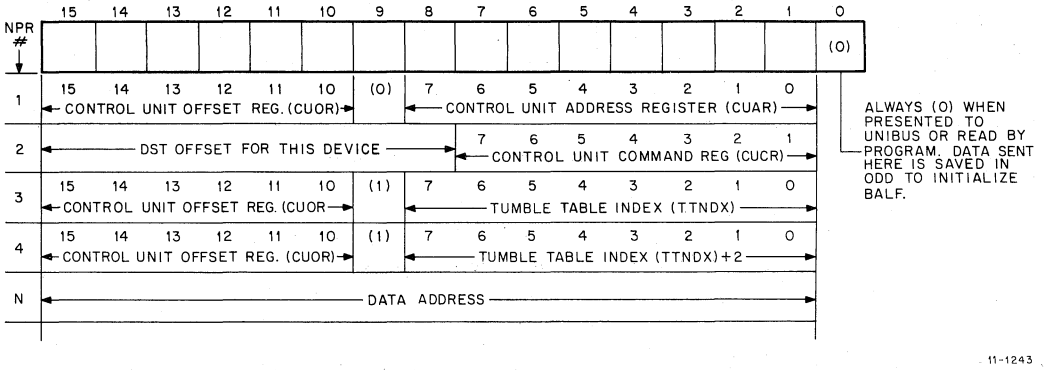


Figure 4-7 Bus Address Register Bit Assignments

4.5.6 Byte Count Register (DXBC)

This register, shown in Figure 4-8, is used only during data transfers. It is loaded and cleared under program control and is set up prior to the data transfer involved. The DXBC is set to the negative of the number of bytes desired to be transferred. As each byte is actually transferred to or from the DX11-B, the DXBC is incremented by one until all bytes are transferred, whereupon the DXBC equals 0. When the DXBC contents go to 0 during a Data Transfer sequence, the CUDEND bit of the DXDS will set and thereby terminate the Data Transfer sequence with the channel.

The program could set DXBC to -1 at any time during a data sequence (Phase 5 or 6) to terminate that sequence. Several attempts may be required in case the DXBC is incremented at the time of program load.

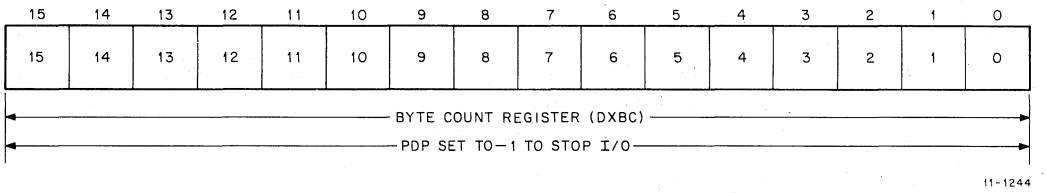


Figure 4-8 Byte Count Register Bit Assignments

4.5.7 Maintenance Out Register (DXMO)

This register, shown in Figure 4-9, is used to hold the 360 Channel Bus Out Data and Tags. This register is always directly readable by a PDP-11 program. When the DX11-B is on-line, the bits in this register are mostly the same as what appears on the Bus-Out lines (hardwired cables to the Bus-Out plug). When the DX11-B is off-line, these bits can be written directly by a PDP-11 programmed request. The programmed bits are held buffered in the DXMOB. When the DX11-B is on-line but cabled to the Bus-Out Test plug, these bits are also writable by PDP-11 programs. The On-Line Cabled mode is used to isolate the cables and Bus-Out receivers as an error source.

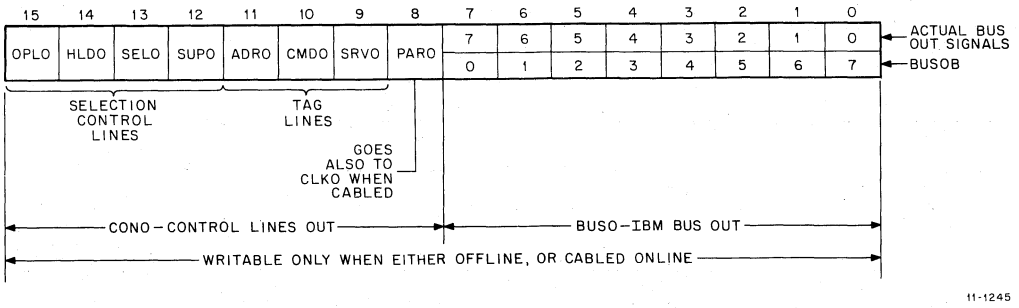


Figure 4-9 Maintenance Out Register Bit Assignments

The IBM Bus-Out Register (BUSO) bits 07–00 contain the Bus-Out data bits 0 to 7 as seen either directly from the Bus-Out cables or from BUSOB if off-line. When written by the PDP-11 program, this byte is buffered in BUSOB.

The Control Lines Out Register (CONO) bits 15–08 contain the signals listed in Table 4-6 as strobed (copied) at TP1 or TP2 from either the Bus-Out lines or from CONOB.

Table 4-6
DXMO CONO Bit Assignments

Bit No.	Mnemonic	Description
Selection Control Lines		
15	OPLO	*Operational-Out* – This line indicates that the channel is in operation. **NOTE** **Refer to IBM manual A22-6843 for a detailed description of each line of the 360 Bus.**
14	HLDO	Hold-Out
13	SELO	*Select-Out* – This flip-flop is set only if both hold out and select out are set. When set or cleared by a PDP-11 program, only the simulated select out signal is affected (see DXMOB).
12	SUPO	Suppress-Out
Tag Lines		
11	ADRO	Address-Out
10	CMDO	Command-Out
09	SRVO	Service-Out

(continued on next page)

Table 4-6 (Cont)
DXMO CONO Bit Assignments

Bit No.	Mnemonic	Description
Parity Line		
08	PARO	*Parity Out* – This bit has double duty when written by a PDP-11 program while the DX is in On-Line Cabled mode. At such times the state of the bit will be translated directly both into the state of the Simulated Clock Out line (of the Bus-Out Test plug) and into the Parity-Out line. Clock Out's primary purpose is to provide a synchronization time for the Control Units to make changes in their ON/OFF LINE state. Another function of the bit is to allow program generation of either normal (odd) Parity-Out or the generation of "bad" (even) Parity-Out. This feature is necessary to permit checking the parity generator within the main DX11 logic. The parity is determined by counting the number of ones in Bus-Out (8 bits); if the result is an even number, PARO should be set to 1. Parity is then checked by adding all 9 bits together; good parity always gives an odd sum.

4.5.8 Maintenance In Register (DXMI)

This register, shown in Figure 4-10, is used for reading the Bus-In tags and data originating from the main DX11-B logic. In this way, the register represents the channel's view of Bus-In. The output of these flip-flops is enabled to the Bus-In lines either when OPLI=1 or when a Fast CU Busy is in progress.

The register is normally read-only; but for maintenance purposes, programmed modification is permitted except for CLKO and PARI. When the DX11-B is off-line, the data read by PDP-11 programmed references comes directly from DXMI. When the DX11-B is on-line, the data read by the PDP-11 program comes from the Test-In plug. Correct data will be seen then only if the plugs are cabled together.

The Buffered Bus In Data Register (BUSI) bits 07–00 contain the data that is enabled to the Bus-In lines for transmission back to the IBM 360 Channel. The output of this byte is always input to a parity generator that produces the signal PARI.

The Buffered Control Lines In Register (CONI) bits 15–08 are listed in Table 4-7.

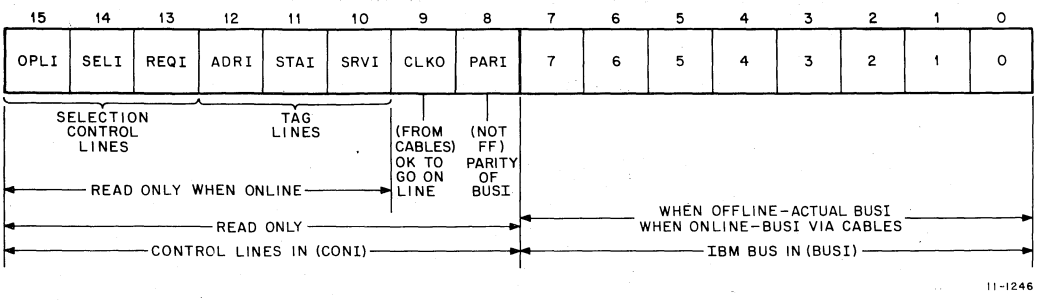


Figure 4-10 Maintenance In Register Bit Assignments

Table 4-7
DXMI CONO Bit Assignments

Bit No.	Mnemonic	Description
Selection Control Lines		
15	OPLI	Operational-In
14	SELI	Select-In (not direct cleared)
13	REQI	Request-In
Tag Lines		
12	ADRI	Address-In
11	STAI	Status-In
10	SRVI	Service-In
Test Lines		
09	CLKO	*Clock Out Signal from Bus-Out* – This line always comes from the cables, even when the DX11 is off-line.
08	PARI	Bus Parity In (not a flip-flop – output of parity generation for BUSI)

4.5.9 Control Bits Register (DXCB)

This register contains control bits used for DX11-B operation. They are read-only, and are used for diagnostic or emergency purposes only. The register is reset by INIT=1, except for ONLINB and SELI. These bits are listed in Table 4-8 and shown in Figure 4-11.

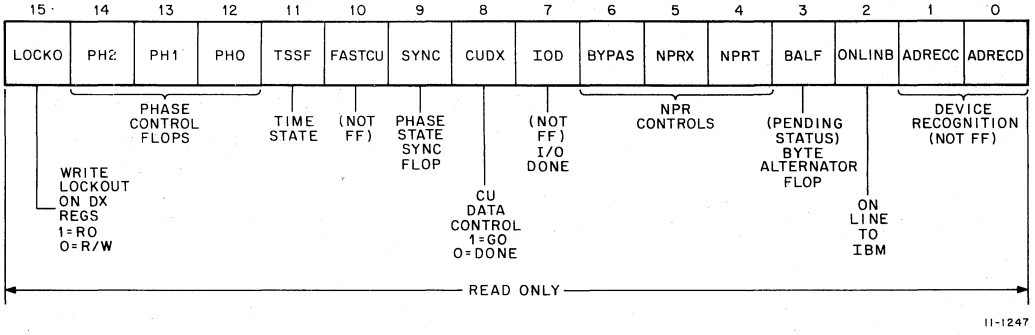


Figure 4-11 Control Bits Register Bit Assignments

Table 4-8
DXCB Bit Assignments

Bit No.	Mnemonic	Description
15	LOCKO	*Lockout* – If this bit is set to a 1, the first five programmable registers are made read-only to the PDP-11 program. It is this Lockout protection which provides the interlocking or contention control which is needed when a channel-initiated sequence conflicts with the setup of a control unit initiated sequence. The channel sequences are always given precedence, and the control unit programs are prevented from further modification of the DX11 Registers.
14,13,12		*Phase Control Bits* – These bits are used to determine the major phase of the DX11 Controller.
11	TSFF	*Time State* – This bit is the clock control: 0 = TS2 1 = TS1
10	FASTCU	This signal is available for diagnostic test purposes.
09	SYNC	*Synchronizer* – This is a phase synchronizing flip-flop that is used for miscellaneous purposes to disable certain portions of logic within any particular phase. The details of this use are spelled out in the flow diagrams.
08	CUDX	*CU Data Control* – This is a control unit data control flip-flop. If it is a 1, this means that the Control Unit data (composed of the CUCR and CUSR) can be used or is available to either the Bus-Out lines, or to the Bus-In lines, in the case of PDP-11 output transmission.
07	IOD	*Input-Output Done* – This bit is not a flip-flop, but is the Inclusive-OR of all conditions which can stop a data transfer that is taking place in Phase 5 or 6.
06	BYPAS	*Bypass* – This bit is used for a number of purposes: a. First, it is used to indicate that this is the first byte of data that is being transmitted or received, i.e., bypass suppress-out. b. Bypass to a 1 also indicates that the device status table will not be accessed by this particular channel-initiated sequence. This might occur because of a zero pointer in the left portion of the SPW, i.e., bypass DST. c. Bypass to a 1, may also be used as an indication that no address out has been received. This is a synchronizing function that is used in Phase 0 for a CUI sequence, i.e., bypass SPW and DST. d. Bypass is also used as a copy of the parity okay signal. This is used to communicate between Phases 2 and 3, i.e., bypass the command. e. Bypass is used as a copy of the status pending indication between TS1 and TS2 of Phase 0 during a CHI sequence, i.e., bypass Phase 1 and both SPW and DST. f. Bypass a 1 (and SYNC=0) is used to indicate in TS2 that a fast CU Busy response was made in TS1. This is essentially a synchronizing function, i.e., bypass CHI sequence.

(continued on next page)

Table 4-8 (Cont)
DXCB Bit Assignments

Bit No.	Mnemonic	Description
05	NPRX	*NPR Control Switch* – This flip-flop is the NPR control. If it is a 1 this means the DX is requesting a non-processor or memory-to-device data transfer from the PDP-11. When NPRX becomes a 0 this is the indication that the transfer has been completed.
04	NPRT	*NPR Transfer Direction* – If NPRT equals 0, then data (or status table information) is being taken from the PDP-11 core. If NPRT is 1, then data (or TT information) is being sent into PDP-11 core.
03	BALF	*Byte Alternator Flip-Flop* – a. If this bit is a 1, it indicates that status is pending and that the device for which status is pending has been addressed by the channel. In this case both the SPW and the DST fetches are by-passed and the CUSR is used as is, i.e., alternate logic is used. b. This bit is used during Phases 5 and 6 to keep track of which half of the data word is currently being used, i.e., alternate bytes are used.
02	ONLINB	*On Line to IBM* – The state of this flip-flop reflects the actual status of the DX11 as to whether or not it is connected to the I/O interface. ONLINB is the second level of the ONLINA flip-flop found in DXCS bit 9. ONLINB is not direct cleared when the rest of the DXCB is cleared.
01	ADRECC	*Address of Control* – This bit is not a flip-flop, but is the continuing signal from the address comparator indicating that the control unit portion of the address matches Bus-Out.
00	ADRECD	*Address of Device* – ADRECD is a similar signal indicating that the device address matches (ADRECC=1 also).

4.5.10 Non-Processor Request Data Register (DXND)

This register, shown in Figure 4-12, contains each data word that is transmitted to or from the PDP-11 core via NPRs. This word is available for use by the MAINDEC.

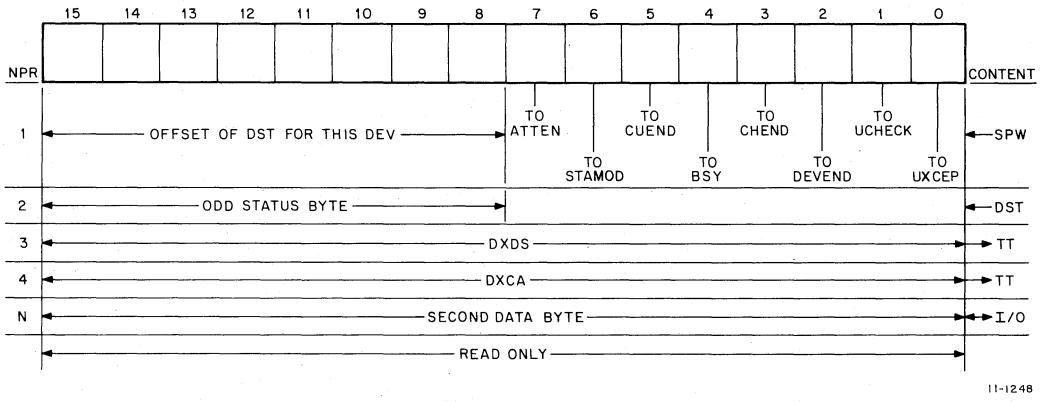


Figure 4-12 Non-Processor Request Data Register Bit Assignments

4.5.11 Extra Signals Registers (DXES1 and DXES2)

The bits assigned to this register are listed in Table 4-9 and shown in Figures 4-13 and 4-14. These two bytes are provided for future 370 controls and for diagnostic uses.

Table 4-9
DXES1 and DXES2 Bit Assignments

Bit No.	Mnemonic	Description
DXES1		
15–08	TTNDX	*Tumble Table Index Byte* – This byte is the low-order address of the TT entry to be used next. It is shifted left before being copied into the DXBA.
07–00	MISC	Miscellaneous Signals
07		Reserved
06	ODD	Copy of DXBA (00) (for future use)
05	NPRTO	NPR latency error. Bus grant not received within timeout interval.
04	DXTO	Program response latency error. While OPL-IN was up, the program did not interact for a 5 sec period.
03	TIMDIS	Set to disable DXTO during program debugging.
02	SOSIEN	Fast NPR test enable (Service-Out-Service-In-Enable). Causes simulated SRVO to follow SRVI.
01	MNCLEN	*Maintenance Clock Enable* – When this bit is set, the DX11 does not change time states until MCLKP is set.
00	MCLKP	*Maintenance Clock Pulse* – If MNCLEN is set, setting this bit causes the DX11 to enable the next time state. One normal clock pulse will be issued with each setting of MNCLKF, the Clock pulse thus generated will reset the MNCLKF. The Unibus Interface continues to run at normal speed at all times. Maintenance Clock mode cannot be entered when on-line.
DXES2		
01	DSCRSP	*Disconnect Response* – Set when IBM issues an Interface Disconnect.
00	IRS	*IBM Reset Stored* – Set in Phase 4 if an IBM Reset is issued, and used in Phase 7 to remember that an IBM reset sequence occurred.

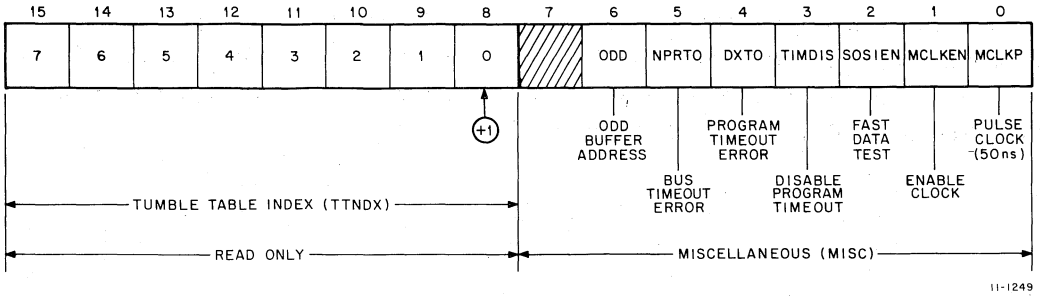


Figure 4-13 Extra Signals Register DXES1 Bit Assignments

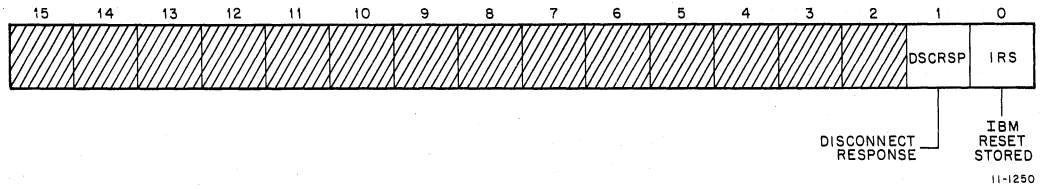


Figure 4-14 Extra Signals Register DXES2 Bit Assignments

4.5.12 Maintenance Out Buffered Register (DXMOB)

This register, shown in Figure 4-15, holds the bits written by the Unibus into DX11-B address plus 14 (program thinks that this is the DXMO). This register is needed when the DX11-B is in On-Line Cabled mode to present steady levels to the drivers. The left-hand byte is CONOB (Control Out Buffered); the right-hand byte is BUSOB (Bus Out Buffered).

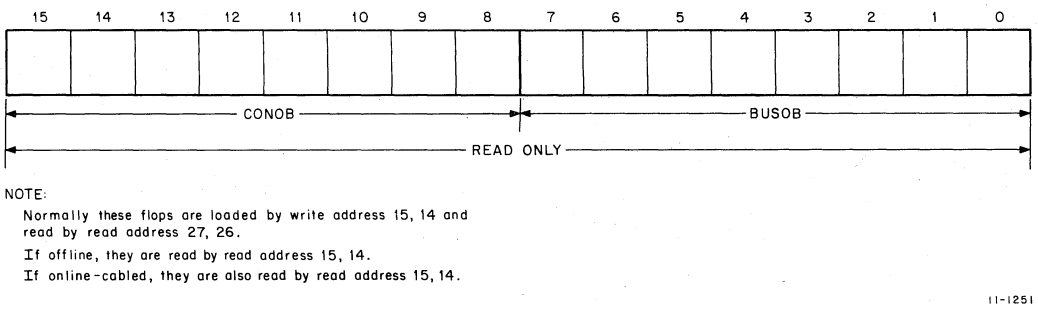


Figure 4-15 Maintenance Out Buffered Register Bit Assignments

4.6 PROGRAMMING CONSIDERATIONS

This paragraph contains a discussion of some of the general programming considerations that pertain to the DX11-B.

4.6.1 Hardware/Software Interlock

Because of a contention situation that can arise when both the channel and the control unit (through software request) attempt to use the interface, an interlock mechanism is necessary to protect information used by both parts of the system. The control unit can appear busy to channel activity when the control unit software must use the facilities. After system reset, while table initialization is in progress, the CUBSY flip-flop is set.

The general soultion is simply to let CH requests always override CU requests. This is done by the LOCKO flip-flop which prevents further changes to the DX11-B registers once a selection sequence has begun. The program can later examine the interrupt conditions to determine if the program requests must be repeated. LOCKO gates DXCS, DXCA, DXOS, and DXBA. Only DXBC remains program-writable.

4.6.2 Boundary Considerations

a. The SPW, TT firmware is 512 words long and must begin on a 2000_8 address boundary.

b. All DSTs are 256 bytes long and must begin on a 400_8 address boundary.

c. All data transfers should begin on an even boundary. On input operations, the following will occur for odd BA and odd BC, respectively:

   1. When starting an input on an odd address, the *previous* even address byte is clobbered.

   2. When ending an input on an even address, the *following* odd address byte is clobbered.

d. On output operations, as many as two words *following* the end of the data buffer can be pre-fetched. Therefore, buffers should not be assigned at the *end* of core. This prevents spurious NXMs.

4.6.3 Interrupt Request

When the DX11-B requires either a program interrupt or tumble table service, it sets the DONE bit, leaving LOCKO set. When the program is ready to try a CUI, it must clear DONE; then, if no new selection is in progress, the DX11-B will clear LOCKO.

NOTE
Clearing INTEN is discouraged during DX11-B operation.

The following rules should be followed:

a. The TT entry should be zeroed after being serviced.

b. On an INT, *all* nonzero TT entries should be serviced before dismissing INT (RTI).

c. Software should keep a pointer to current TT entry. This should follow the hardware pointer in relieving the entries the hardware places (hardware will guarantee a nonzero TT entry).

d. No software requests for data transfer or status should be made until all TT entries are serviced.

e. *Before* each TT entry is serviced, DONE should be cleared with a

   BIC # DONE, DXCS

   thus, the general INT service procedure is:

   1. Clear DONE.

   2. Service current TT entry. Update action to be performed for device whose address is in TT entry. Do not request data transfer or status at this time. (Note that this may *cancel* a previously queued request for this device.)

   3. Clear TT entry.

   4. Bump software pointer to next TT entry.

NOTE
If TT entry not 0, go back to Step 1. If it is 0, proceed.

   5. When a zero TT entry is encountered, initiate *last* action pending for each device.

   6. Dismiss interrupt (RTI).

4.6.4 Data Transfer

Data transfer sequences (DT) are always initiated by the DX11-B program. It is a software responsibility to ensure that a DT is valid at the point requested. Information supplied to the hardware includes Buffer Start Address (DXBA), Byte Count (DXBC), Device Address (CUAR), and I/O direction (FCTN – input or output). The hardware will get control of the I/O interface and transfer the data in a single burst, after which it will generate a Data End interrupt. The last bit set is the GO bit. If LOCKOUT is set at this time, the effect is a NO OP and the result is no data transfer.

Several other events can happen as follows:

a. A bus out parity error can occur, setting PARER. This will terminate data transfer with PARSTP.
b. A timeout reference can occur, setting NXM.
c. The channel can indicate I/O stop, setting CHDEND.
d. An Interface Disconnect can occur.

If a CUI is used, it could be overridden by an ISS, in which case an interrupt would occur; but different bits would be set in the DXDS and copied into the tumble table.

4.6.5 Status Presentation

There are several cases in which presentation must be initiated by the program as follows:

a. When stack status is indicated by the channel. In this case, the DX11-B will automatically request presentation of the status again, until it is subsequently relieved or overridden via an ISS (only if BSYEN=1).
b. When ending status is initially available for the device (DEVEND, etc.).
c. When asynchronous status becomes available for the device (DEVEND or ATTEN).
d. When a device that had previously been interrogated while busy becomes free (CUEND or DEVEND).
e. At the termination of a data transfer (CHEND or CHEND+DEVEND).

The program loads the status and device address and requests status presentation (FCTN=3). If the status is accepted, the device status, device address and command (if any) are loaded into the tumble table and an interrupt is generated.

4.6.6 On-line/Off-line Control

The ONLINA flip-flop is written by the program to request a change of on-line status. The ONLINB flip-flop, in DXCB, can be read to verify that the transition occurred. The program clears the ONLINA flip-flop to attempt to put the DX11-B off-line. The ONLINB flip-flop will not clear if any channel activity is in progress.

NOTE
When the program sets ONLINA, the ONLINB flip-flop will only set if the hardware ON-LINE/OFF-LINE switch is in the enable position when channel conditions permit.

An on-line/off-line request can be made at any time. If channel activity occurs at the time ONLINA is cleared, it will not clear. This allows the program a chance to reconsider the off-line request in view of the new CHIS.

NOTE
An On-line/Off-line sequence should not be attempted in an interval less than 10 ms.

CHAPTER 5
THEORY OF OPERATION

5.1 SCOPE

This chapter provides a detailed description of the DX11-B System. The discussion consists of three major parts: a) a functional description of the overall control unit operation, b) a system block diagram discussion, and c) a detailed theory of operation covering control unit logic circuits. The discussions in this chapter are supported by a complete set of engineering drawings contained in a companion volume entitled *DX11-B, System 360/370 Channel to PDP-11 Unibus Interface, Engineering Drawings.*

5.2 FUNCTIONAL DESCRIPTION

As described in Chapter 1, the DX11-B functions as a data and status transfer controller between an IBM 360/370 I/O Channel and a DEC PDP-11 Computer (Figure 5-1). When operational, it executes signal sequences, performs NPR data transfers, initiates status presentation, and handles stack status conditions, returning appropriate indications to the program that enable that program to complete the sequence. When initiated by a stored PDP-11 program, the DX11-B conducts transfers between the IBM Channel and the Unibus, automatically transferring data between its buffers and Unibus locations, participating in data transfers on the channel, and stopping the transfer when the prescribed number of bytes have been transferred. In addition, the DX11-B communicates with the PDP-11 Processor, which controls the status indications provided by the DX11-B to the 360 Channel.

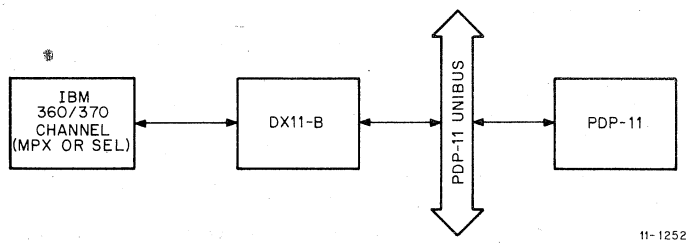


Figure 5-1 DX11-B System, Functional Block Diagram

As such, the DX11-B must have the dual capability of communicating in hexadecimal code with the IBM environment, and in octal code with the PDP-11 System. It must also make the format conversions between the bit formats of both systems. The DX11-B must be able to respond to the normal signal sequences generated by either a multiplexer or selector type channel, and must also be able to participate in Unibus dialogue – within the latency requirements of both systems.

5.3 BLOCK DIAGRAM DISCUSSION

A typical DX11-B System block diagram is shown in Figure 5-2. As can be seen from the figure, a 360/370 Central Processor is connected to several miniprocessors termed channels. These channels can be of either a multiplexer or selector variety. Each channel can accommodate several control units which, in turn, interface to several devices. The DX11-B Controller functions as a control unit in the 360 system and can be tied to either a multiplexer or a selector channel bus. The DX11-B, then, is assigned a Unibus address and is thereby put in communication with the PDP-11 processor, its memory, and any devices on the bus. Special software (emulators) placed in the PDP-11 memory adapt the hardware, causing it to appear to be any of most IBM type control units.

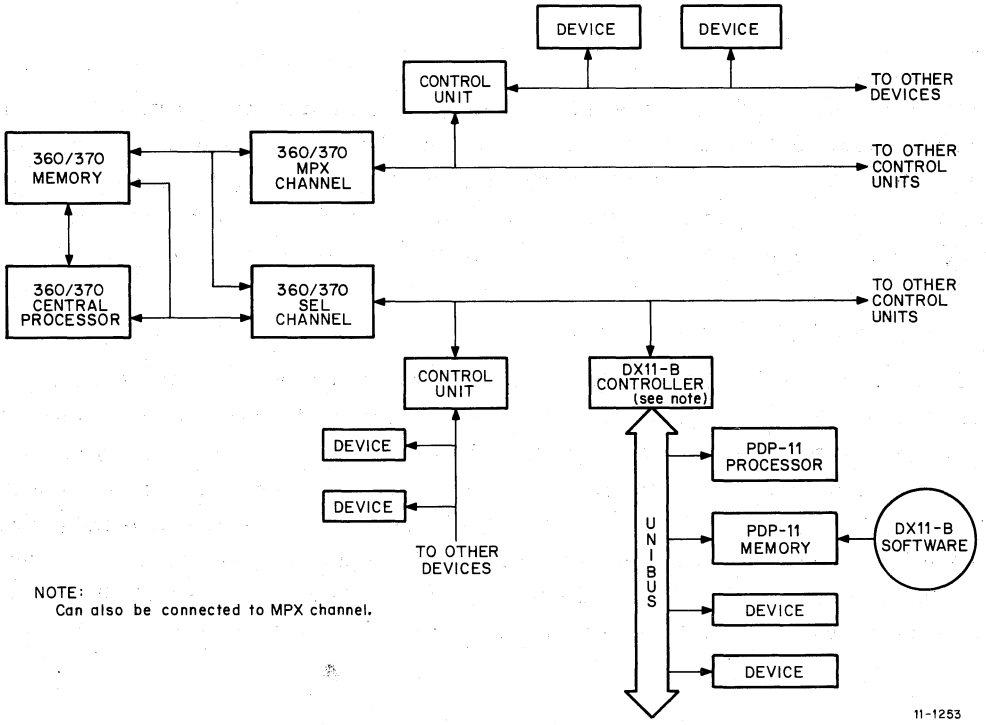


Figure 5-2 DX11-B System, Block Diagram

Although the overall system comprises hardware and software sections, the hardware consists of two parts: Data Paths and Control Logic. The data paths are in three sections: 1) IBM Control, 2) Central Control, and 3) PDP Control (Figure 5-3). The control logic is divided into nine parts: phase independent operations plus eight major phases. Finally, each major phase comprises two time states.

Internally, the DX11-B contains a data path control section which ties directly to the 360/370 Interface, a section for dealing with the PDP-11 Unibus, and a central data path section for synchronizing operations between the two hardware interfaces. The IBM control, central control, and PDP control sections are internal to the DX11-B and are not visible across the formal interfaces.

It is the control logic section, along with the support software, that provides the basic programability of the unit. The separation of the DX11-B into functional sections does not imply that they are physically separable sections.

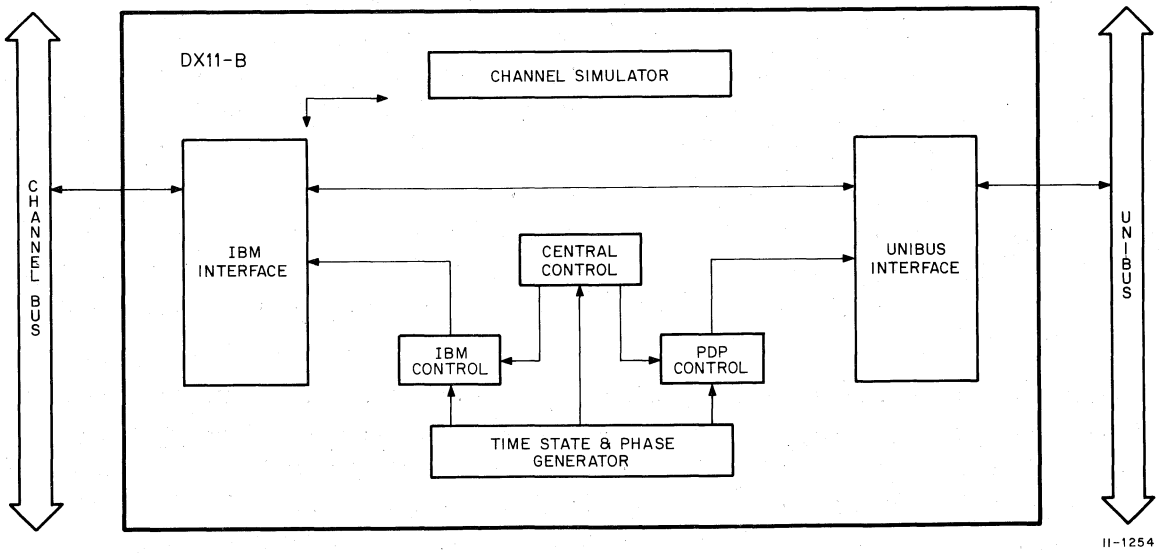


Figure 5-3 DX11-B, Block Diagram

5.3.1 IBM Interface Operation

An understanding of the operation of the IBM Interface can be gained only by becoming familiar with the operation of the 360/370 Channel. For detailed information, refer to the Introduction and Input/Output Operations sections of the *System/360 Principles of Operation,* and to the *System/360 I/O Interface, Channel to Control Unit* manuals. Although it is beyond the scope of this manual to provide exhaustive coverage of IBM architecture and facilities, this paragraph supplements the referenced information by giving a minimal description of the 360/370 I/O Interface as seen by an external device such as the DX11-B.

The I/O capability of the 360/370 Central Processor (CP) is provided by a number of limited-function I/O processors called channels (CH). All channels are attached to the CP and provide a standard I/O interface to external devices. Thus, each channel provides a single data path into memory which is shareable among up to 256 directly addressable devices per channel.

NOTE
The DX11-B can accommodate only 128 devices at a time.

Conventional I/O is handled by two types of channels: Multiplexer (MPX) and Selector (SEL).

An MPX channel provides concurrent, byte-interleaved I/O with multiple devices, and is used primarily with slower devices such as printers, card-readers, and communications multiplexers, although it *can* handle high-speed bursts of data.

An SEL channel is specifically designed for high-speed burst operation and can service only one device at a time. The typical transfer is one block or physical record at a time. SEL channels are typically used for high-speed devices such as disks, drums, and data concentrators.

Each channel represents a single path to memory, and has a single I/O interface that is, itself, logically connected to only one device at any one time. Multiple interleaved operations on one channel can be effected with the use of subchannels described later in this section.

A channel can be termed a processor in the sense that its activities are governed by stored programs using sequential fetch and execute techniques. The basic unit of a channel program is the Channel Command Word (CCW). The term "Command" is used to distinguish it from Central Processor "Instructions". A single command, Transfer In Channel (TIC), is dedicated to program control. It provides the channel with an unconditional program branch (Jump) capability. All other commands relate to device operations and cause activity on the channel/device interface. Read and Write are typical commands.

Sequential command execution is not automatic but is a function of a command chaining bit in each command. If this bit is set, execution continues with the next command. If it is not set, the program terminates. A program can also be aborted (command chaining suppressed) by detection of a malfunction or other abnormal condition in the channel or in the device.

The channel cannot initiate the execution of a program. This is done by the CP using a Start I/O (SIO) instruction. The SIO alerts the channel which then fetches the first command, establishes logical connection with the indicated device, and reads back status from the device indicating either acceptance or rejection of the command. If the command is accepted, the CP and CH can continue concurrent and asynchronous execution of their respective programs. If the command is rejected, the CH program is aborted. The CP will be aware of this at the completion of the SIO execution and can process the exceptional condition indication. Channel program termination is communicated to the CP through state words and interrupts.

The activities encountered during a typical command execution are a Selection Sequence, a Data Sequence, and an Ending Sequence. A detailed set of IBM interface timing diagrams are given in Appendix C of the IBM OEM manual (refer to the Foreword). During a command execution, the channel fetches the command that follows the SIO, establishes logical connection with the indicated device, and transmits command information. This is called an Initial Selection Sequence (ISS). If the command is accepted, a Data Sequence (DS) can follow in which data is transmitted across the interface. When I/O is complete, the device presents status that indicates the success or failure of the operation. This Ending Sequence (ES) completes command execution.

Some commands involving control operations have no data sequence. In a magnetic tape rewind command, for example, the ISS starts the rewind operation and the ES takes place when the rewind is complete. If the tape was already rewound, that would be an "immediate" command for which the ending status is presented *during* the ISS and for which there is no separate ending sequence.

Logical connection between the channel and device can be maintained throughout command execution. In this case, the device monopolizes the channel until the ES is complete. This is the only method used on the *selector* channel. On the other hand, the channel and the device can operate independently after the ISS, resynchronizing only as necessary for transferring data and status. Thus, a *multiplexer* channel can interleave command executions for different channel programs.

The channel provides a standard bus of interface signals to all devices regardless of type. Thus, each device requires a control unit to translate the standard interface sequences into those control signals that are peculiar to the specific device, and to translate the device's responses back into standard interface sequences. The DX11-B performs this function for the PDP-11 System to which it is attached. As such, the PDP-11, its memory, devices, and software, constitute a device on the IBM channel – a device, which with changes to the software, can emulate any of many standard IBM devices. In some cases, such as printers, readers, tapes and disks, the control unit (CU) is logically (and often physically) distinct from the device. The DX11-B falls into this category. In other instances, such as the channel-to-channel adapter and other wholly electronic devices, the CU is an integral part of (and indistinguishable from) the device.

Devices which are configured in groups, of which only one device need be active at a time, can share a CU. An example of this is magtapes, where four drives can share a single CU.

Each device on a channel is assigned a unique 8-bit address. Each CU is responsible for recognizing the addresses of all devices it controls. Multi-device control units are assigned blocks of contiguous addresses, the block size being some power of two with the first address being some multiple of the block size. The limit on block size is usually 16; if the CU requires more addresses, it is assigned two or more (usually contiguous) blocks. Where only one block is assigned, the high-order address bits that are common to all devices on the control unit can be regarded as the "CU Number", and the low-order bits that address the devices as the "Device Number". In cases where the address cannot be broken down in this way, one device address can be dedicated to the CU itself for commands that involve only the CU.

A system-wide unique address for each device is provided by prefixing the 3-bit "Channel Number" to the 8-bit device address. This is the full device address used by the CP in initiating channel programs.

As stated earlier, multiple interleaved operations on one channel can be effected with the use of "subchannels". The facilities required by a channel to service one device are termed a subchannel. These facilities are the command address, data address, data count, etc. The number of channel programs that a channel can execute concurrently is merely the number of subchannels it contains. The subchannel, then, is just some storage area in the channel where it maintains status information on the particular device or devices that are active. Note that this storage is *not* in the CU but in the channel itself.

A selector channel (SEL) has only one subchannel. From the initiation of a channel program for a particular device until that device releases the channel at the end of the last command, channel facilities are monopolized by that device; during this period, other devices on that channel are restricted to operations that do not require use of the channel.

A multiplexer channel (MPX) has several subchannels; at least one for each currently active device. It is monopolized by any device a) during a selection sequence, b) during part of a data sequence for transferring one byte or burst of bytes, or c) during a status transfer. In other words, it is monopolized only during times in which the device has an active, immediate need for channel services. The multiplexer channel thus appears to each program that it executes as a dedicated selector channel.

Because the subchannel used for any device is fixed in any given installation, the channel determines automatically, from the device address, which subchannel to use. Some subchannels are shared among a group of devices with contiguous addresses, while others are dedicated to single devices. Shared subchannels can service only one of their associated devices at a time (the selector subchannel is, in effect, a shared subchannel). Shared subchannels are particularly applicable to shared CU devices that automatically satisfy the one-at-a-time requirement.

The use of multiple subchannels provides the MPX with its basic multiplex mode, an interleaved half-duplex byte stream. However, the device can request the channel to operate in burst mode (a transfer in which the device retains monopoly of the channel for more than about 100 μs). The burst can involve anything from a few bytes to a complete command execution. This is called Control Unit Forced Burst Mode (CUFBM).

The selector channel, by contrast, operates only in burst mode. In discussing devices designed to operate on either type of channel, this is called Channel Forced Burst Mode (CHFBM).

The number, type, and subchannel configuration of channels that can be included in a system are a function of the CP model. A typical configuration has a single MPX and one or more SEL channels. In addition, the width (in bytes) of a channel's access to memory, and the degree to which it shares facilities with the CP are functions of the CP model number. Because of this, it is impossible to make any but the most general statement about channel capabilities. IBM provides a guide for each CP model on the intricate calculations required to determine channel loading. The principal limiting factors are a) shared facilities with the CP, and b) the access width.

In summary, the MPX channel is suitable for operations involving frequent but short data transfers, while the SEL channel is suitable for infrequent but large data transfers. The respective ranges of suitability are from telegraph adaptors to slower magtapes for the multiplexer, and from magtapes to high-speed disks and drums for the selector.

The architecture of 360 channels allows them to accommodate devices within a wide range of data rates and response times. This is achieved through the use of edge-interlocked signal sequences in which, typically, only one outbound signal from the channel and one inbound signal from the control unit participate. One signal can rise only when the other has risen, and must then remain up until the other has fallen. This is similar to the MSYN/SSYN relationship on the PDP-11 Unibus. This requirement and the use of control unit initiated sequences allow the CU to present data and/or status information at times and rates appropriate to the specific device.

Despite its ability to operate at almost dc rates, the channel imposes certain timing constraints to guarantee that no device ever requires excessive time to complete a signal sequence. This is done to prevent one device from jeopardizing the operation of other devices on the channel. In particular, device selection sequences are required to be performed within 32 μs, start to finish. A channel timeout monitors this performance. Other timeouts related to data transfers and to interface activity, in general, are scaled to appropriate limits ranging from 1/2 sec to 30 sec. I/O under System 360/370 is under the overall control of the CP, which has instructions for starting channel programs (SIO), testing their current operational status (TIO), and aborting them (HIO).

The Start I/O instruction (SIO) initiates the execution of the first command of a channel program. Before the CP executes an SIO, it must first set up a pointer called the Channel Address Word (CAW). This word points to the first Channel Command Word (CCW) of the channel program to be executed. The CAW is at a fixed core location and, since its contents are required by a channel only during SIO execution, only one is required to service all channels. The SIO instruction itself identifies the channel and device for which the channel program is to be executed. A condition code, returned by the channel, indicates whether or not the channel was able to select the addressed device, and, if so, whether or not the device accepted the command. The execution of an SIO instruction is not complete until the condition code is available. Successful completion of an SIO execution does not guarantee that a channel program (or indeed even the first command of the program) will be completed successfully, only that the program was started successfully.

Because the IBM diagnostics utilize these codes to inform the programmer of malfunctions, a discussion of their meanings is in order. These condition codes are set in the 360/370 and can be interrogated by the program in a similar fashion to the Branch On Condition instruction in the PDP-11. There are four sets of condition codes expressed by two bits in the 360/370 Program Status Word. Their meanings for an SIO instruction are as follows:

0 = The device that was selected has accepted and is now beginning to process. Does not say that the process was completed.

1 = The device that was selected did not accept. Does not say why. Implies that the device was there and did not want to start.

2 = The channel or subchannel was busy. Occurs on an SEL channel if one device has been started, is still active, and a start I/O is issued to that or another device. On an MPX channel, occurs if two SIOs are issued in a row to the same device.

3 = No CU recognized the device address that was put on the bus. The assumption is that the CU is off-line. Note that if the device was not there, the I/O would not be accepted and a condition code 1 would be indicated.

Between SIO executions, all channel-to-CP communication is done through the Channel Status Word (CSW) and via I/O interruptions. An I/O interrupt signifies that the channel in question has new information to store in the CSW. Such information typically contains channel status, device status, the address of the last CSW executed, and the residual data count (if appropriate for the last operation). The usual function of I/O interrupts is to indicate termination of a channel program, either because the last chained CCW has been executed, or because an exceptional condition has caused the I/O program to be aborted.

In addition to using the I/O interrupt facility, the CP can obtain information from channels through the "Test I/O" (TIO) instruction. This instruction appears on the interface as a TIO command. Its function is to test control units and devices and to relieve them of status conditions that might otherwise cause an I/O interrupt. A TIO can also be issued by the channel itself under some circumstances where it is desirable to obtain status quickly. As with all instructions, there are four sets of condition codes expressed as follows:

0 = The path to the device is free.

1 = The device attempted an I/O interrupt because of the completion of its activity but was prevented from doing so because the CH was disabled. Relieves that interrupt pending condition without actually causing an I/O interrupt. This is similar to a PDP-11 System turning off INTEN and spinning on the DONE flag, or raising the priority to 7.

2 = The CH was busy.

3 = The CU is off-line. Did not recognize the address selection of the TIO.

A "Halt I/O" (HIO) instruction is used to terminate I/O activity for a particular device. This appears on the interface as a special signal *sequence* called "Interface Disconnect". Depending on the device and the current activity on the interface, the channel can first be required to select the affected device before it can execute the disconnect sequence. If the channel is operating in burst mode at the time HIO is executed, Interface Disconnect is signaled to the currently active device, regardless of which device was actually addressed by HIO. This instruction is typically used to clear channels for higher priority operations and to terminate operations involving non-passive devices, such as communication devices, for which the CP program cannot determine a prior termination condition.

The four sets of condition codes associated with an HIO are as follows:

0 = The device has been signaled to stop.
1 = An I/O interrupt for that device is pending.
2 = A burst mode transfer was stopped.
3 = The CU is off-line.

NOTE

A condition 2 instead of condition 0 will be returned on an HIO when it terminates a transfer on either an SEL or MPX channel in burst mode.

The command structure of 360 channels provides five basic I/O commands. These are Read, Read Backwards, Write, Sense, and Control. Variations on the basic commands are obtained by microcoding command-byte bits that are not used by the basic commands. The command-byte formats for the basic commands (and for test I/O which looks like a command to the external device) are given in Table 5-1. The bits marked "m" are the command modifiers. For basic operations, they are 0. For other operations, they are microcoded according to the requirements of the particular application.

Table 5-1
I/O Command Byte Formats

Command Name	Bit Number							
	0	1	2	3	4	5	6	7
Read	m	m	m	m	m	m	1	0
Read Backward	m	m	m	m	1	1	0	0
Write	m	m	m	m	m	m	0	1
Sense	m	m	m	m	0	1	0	0
Control	m	m	m	m	m	m	1	1
Test I/O	0	0	0	0	0	0	0	0

The I/O command Read transfers data from the device to 360/370 core, where it is stored in ascending byte locations starting at the address contained in the CCW for the read command.

Read Backward is similar to Read except that data is stored in descending locations from the starting address.

The Write command transfers data from ascending core locations to the device starting with the address contained in the CCW.

Sense is essentially a "read" command except that the data transferred pertains to exceptional conditions in the device and/or CU. The sense bytes identify, in detail, what caused the CU to present a unit check indication.

The Control command transfers control data to the device. The modifier bits can provide enough data to cover all control functions, or the Control command can send one or more bytes providing any additional information required. For example, a magnetic tape rewind can be signaled through the modifier bits alone, while a communications device "control" command, such as a 2701 Set Mode command, can require two or more bytes of control information to be transmitted. A control command with all modifier bits set to 0 is considered an I/O NO OP command.

A channel has two means of establishing logical connection with a device. It can address a device directly or it can scan (poll) for devices requiring service. *Addressing* is the start of the initial selection sequence (ISS) for a command. It can also occur due to the execution of a TIO or HIO instruction. It is achieved by first sending the address to *all* devices common to the bus, and then requesting the addressed device to acknowledge. *Polling,* on the other hand, is performed in response to a request from one or more currently active devices. Since Request-In is a common bus line, the channel can only determine which device is requesting by polling all devices on the channel. This is accomplished by raising its selection lines, Select-Out. Unlike the other channel interface lines, Select-Out is a serial line. The fact that it *is* serial allows allocation of service priorities and avoids contention conflicts. With this agreement, each CU, in turn, has a chance to obtain selection. If it does not require service, it can propagate Select-Out to the next CU down the chain. Thus, only one CU at a time can make a selection decision. CUs which are closer to the channel on Select-Out are consequently selected preferentially and have higher priority than CUs logically further away from the CH.

If Request-In is up, the channel will normally conduct a polling operation before starting an ISS. This gives currently active devices preference for service over the initiation of activity for some currently idle device. This reduces the chance of overruns or loss of synchronization for the active devices.

In order to obtain selection during a polling sequence, the requesting CU blocks the propagation of Select-Out and executes a Control Unit Initiated (CUI) selection sequence. The CUI sequence terminates polling during its execution; requesting CUs further down the selection chain do not see Select-Out until the channel resumes polling after that CUI sequence is complete.

As shown in Figure 5-4, the physical interface consists of 31 lines (16 Out and 15 In). Each line group consists of a 9-bit data-plus-parity bus, a number of tag lines that are used to identify what is being presented on the bus, and a number of selection control lines. The functions of these lines are summarized in Table 5-2. Three additional lines are used for device metering but are not included in the table.

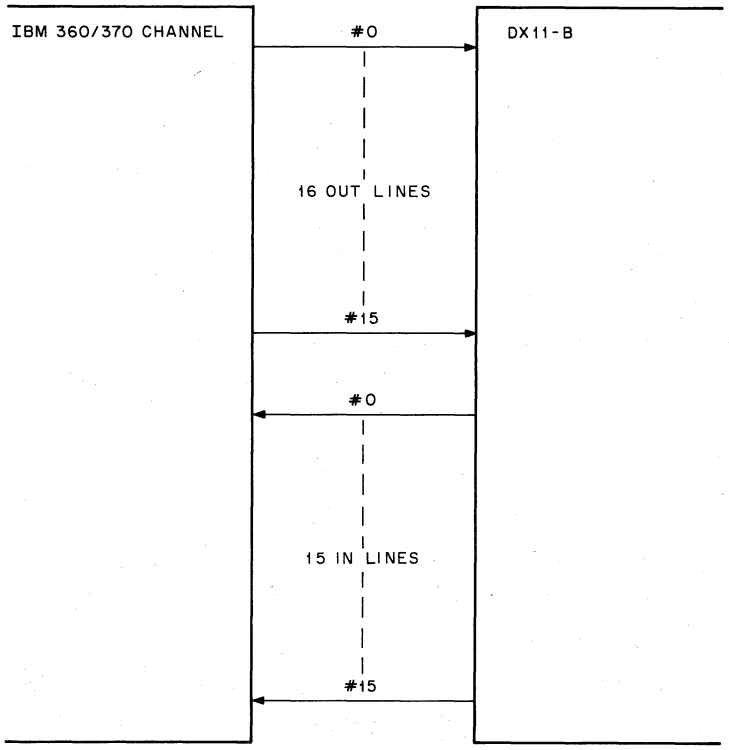


Figure 5-4 Channel/DX11-B Interface

Operational-Out is normally up for an active channel. The other interface lines, in general, are significant only when this line is up. It is dropped only during certain reset sequences and when the channel is deactivated for diagnostic purposes.

Hold-Out is used to condition Select-Out and is functionally identical to Select-Out at the channel. Select-Out, however, is a serial line while Hold-Out is distributed to all CUs in parallel. As such, Hold-Out is not subject to the propagation delays inherent in Select-Out. The fall of Hold-Out indicates the impending fall of Select-Out and is used to purge the select line chain of Select-Out directly, thereby eliminating the delays that would be entailed in propagating the trailing edge of Select-Out through the selection chain.

Table 5-2
Channel/DX11-B Interface Lines

Name	Function
Bus-Out	Output data bus: 8 data bits + 1 parity bit.
Bus-In	Input data bus: 8 data bits + 1 parity bit.
Operational-Out Hold-Out Select-Out Suppress-Out Operational-In Select-In Request-In	Selection control lines, used to select, scan, or disconnect devices.
Address-Out Command-Out Service-Out Address-In Status-In Service-In	Tag lines, used to identify data being presented on bus lines, to interlock the data transfer, and to signal special interface sequences.

Select-Out is the interface select line. It is used to indicate that addressed or polled devices can respond. The selected device remains selected at least as long as Select-Out is up. This allows Select-Out to cause channel-forced burst from devices attached to SEL channels.

Suppress-Out is a line used by the channel (when it cannot immediately handle an interrupt status) to temporarily suppress the presentation of data and status, and to signal certain reset sequences.

Operational-In is a signal from the CU that indicates either that it recognizes the address on Bus-Out during an attempted ISS, and is available, or that it desires service during a polling sequence. In general, in-tag and data lines are significant only while Operational-In is up.

Select-In is a signal returned to the CH by the last CU in the selection loop (Figure 5-5). The rise of Select-In at the channel indicates that no CU recognized the address on Bus-Out during an attempted ISS, or no device required service during a polling sequence. Every CU in the chain propagated Select-Out.

Request-In is raised by a CU to indicate the need to transfer data or status. Request-In causes the CH to perform a polling sequence as soon as feasible. Request-In is normally dropped when the CU obtains selection, i.e., when it raises Operational-In.

Address-Out is normally used to indicate the presence of a device address on Bus-Out during an attempted ISS. It is also used during the Interface Disconnect Sequence.

Command-Out is normally used to indicate the presence of a command byte on Bus-Out. It is also used either as an "I/O Stop" or as "Stack Status" indicator.

Service-Out is used as a data-interlock line. It indicates data on Bus-Out during output operation, and acknowledges the data on Bus-In during input operations.

Address-In is used to indicate the presence of a device address on Bus-In during an ISS or polling sequence.

Status-In is used to indicate the presence of status information on Bus-In.

Service-In is the input analog of Service-Out. It indicates data on Bus-In during input operations and requests data on Bus-Out during output operations.

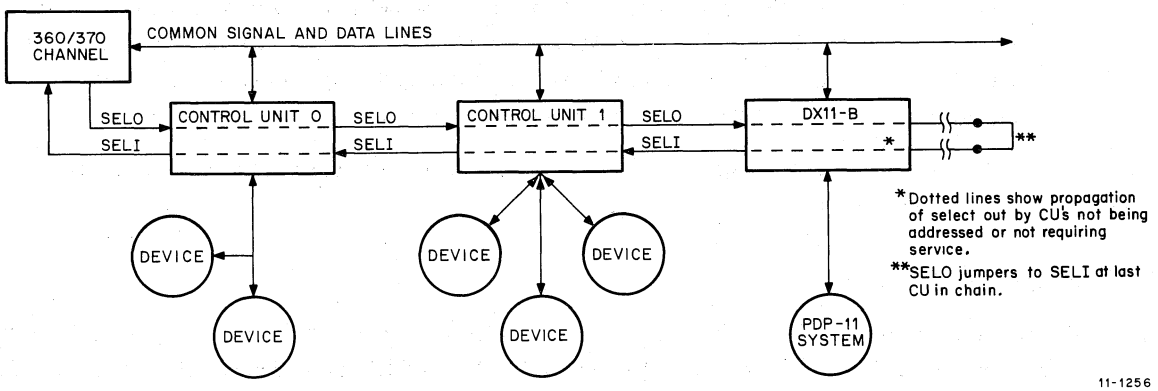


Figure 5-5 Channel Selection Chain

These are the interfacing signals between the IBM Channel and the DX11-B Control Unit. All "Out" signals are generated by the channel and issued to the DX11, while the DX11 must respond to the channel by generating all "In" signals. The timing and relationship of these signals, one with the other, have been explained briefly. For a detailed discussion of their interlocking relationships, refer to Paragraph 5.4 in which the various sequences of operation are discussed.

5.3.2 IBM Control Operation

The DX11-B can present one of three states to the channel. These are *available, busy,* or *not operational.* The DX11 is defined as *available* if it can respond to a channel-initiated selection sequence (CHIS) addressed to it, decode the command presented by the channel, and present appropriate status to the channel. The DX11 is *busy* if it cannot accept the CHIS because a previous operation is still underway, or because status from a previous operation is available in the DX11 but has not yet been accepted by the channel. The DX11 is *not operational* if it does not respond in any way to a CHIS. This can be due to either a power down in the DX11, or if it is powered up and off-line.

The DX11-B can operate with either an SEL or MPX channel. It can be attached to a Block Multiplexer Channel as well. The DX11-B can operate in one or more of several possible modes that are commonly designated as shared, unshared, burst, and byte by IBM. A more descriptive delineation defines these modes as single-thread (shared), multi-thread (unshared), and burst.

In burst mode, once selected by an ISS, the DX11 will force Operational-In to remain up until Channel-End is presented, regardless of whether Select-Out remains up during this period or not.

In single-thread mode, when connected to an SEL channel, the DX11-B operates in channel-forced burst mode and is operationally indistinguishable from a burst mode control unit. When connected to an MPX channel, however, the DX11 operates in either byte or burst data mode as determined by program selection. When in byte mode, it must deselect and reselect periodically to maintain the byte mode operation. It can service activity for only one attached device at a time and, therefore, is busy to selections addressed to all other devices it services from the time one device is selected (via an ISS) until Channel-End is presented for that device. Selections during this period receive the Fast CU Busy sequence described in Paragraph 5.4.1.4.

NOTE
The use of the term "device" in these discussions is intended to be synonomous with the term "360 Device Address".

In multi-thread mode, the DX11 normally operates only with an MPX channel and only in byte mode. It will accept a number of selections and maintain several active devices at once. While a CU Busy sequence can be executed, it is usually the result of a transient condition and thus the "busy" and "status modifier" status is accompanied by Control-Unit-End, causing the CP to immediately retry the selection.

The type of channel to which the DX11-B is to be attached is a consideration of the emulator software. The DX11-B hardware makes no distinction.

5.3.3 Unibus Interface Operation

The DX11-B services the PDP-11 Unibus with standard DEC modules which are, in turn, controlled by the DX11-B through simple synchronized flip-flops. The interrupt is controlled by a DONE indication and INTEN. The NPR is controlled by a direction flip-flop (NPRT) and by a flip-flop which the DX11-B turns on to request (NPRX) and the Unibus turns off when complete.

The NPR and BR operation of the DX11-B with the PDP-11 Unibus is standard and will not be discussed in great detail here. The reader is referred to the various PDP-11 interface manuals for more information.

5.3.4 PDP-11 Control Operation

The function of the PDP-11 Control is to allow the DX11-B to operate as a conventional device on the Unibus. Since NPR and BR operations are adequately documented in the *PDP-11 Handbook* and *PDP-11 Unibus Interface Manual,* only their use will be defined in this paragraph.

During selection sequences, NPRs are used in the fetching of ISS status and in all data transfer operations. The slave address and data direction are controlled by the synchronization section current state of the DX11-B.

BR sequences are initiated to transmit command information (i.e., the command byte) and the IBM device address to the PDP-11 during the ISS and to signal exceptional conditions (e.g., reset sequences) on the IBM interface.

All NPR and BR sequences are executed under control of the synchronization section. See discussion in the control flows discussion in Paragraph 5.4.2.3.

During data transfer sequences, the DX11-B maintains a data address and data count. Data transfers are made continuously using NPR sequences (not back-to-back) until either the count is exhausted or the IBM interface indicates I/O Stop, at which time the BR facility and device status words are used to notify the PDP-11 of the completion.

The DX11-B maintains information for each emulated device. There are three types of status involved: a) the status byte presented to the channel, b) the status (condition) of a particular device, and c) the status of the DX11-B.

The DX11-B status is always available to the PDP-11 via the special register within the DX11-B called the DXCS (Control Unit Status). This register is loaded by the service routine.

Device status is generated during an ISS and is kept in the DX11-B DXDS Register until stored in a circular buffer list in PDP-11 memory called the Tumble Table (TT), where it is kept for use by the program. The TT contains information concerning event notification. A single entry consists of DXDS (Device Status Word) and DXCA (Command and Address Word), also located within the DX11-B. The program must zero the entry after use. All entries are guaranteed to be nonzero.

The channel status is obtained from either of two tables held in PDP-11 core. These are a word table (16-bits wide) called the Status Pointer Words (SPW) and the Device Status Table (DST), a byte table 8-bits wide. A single SPW exists for each device on the DX11-B.

5.3.5 Central Control Operation

Most DX11-B hardware is designed into a matrix of time states and logic phases. Each logical operation, gating, or enabling condition is valid only during particular time states; i.e., if all conditions are met to perform an operation then that operation will actually take place when the appropriate time pulse occurs.

A phase is an arbitrary but logical subdivision of the sequences which the DX11-B performs. Each phase has both a name and a number between 0 and 7 with the name describing the main operational function or functions performed during that phase. The sequence of phases entered varies depending on the function or operation to be performed.

A time state, on the other hand, is a fixed division of time that is generated by a free-running clock. There are two time states (1 and 2) and these states pertain to all phases. The DX11-B can process from Phase X/Time State 1 through Time State 2 of that same phase to Time State 1 of the next phase with no delays; but it will more often remain in some phase and step regularly between the two time states of that phase until the proper conditions are met (in Time State 2 (TS2)) to enter the next phase called for.

The use of phase-controlled logic in the DX11-B simplifies the control and results in a better understanding of the sequences involved. Optimum subdivision also results in logic savings when a phase can be used in several different types of sequences.

Drawing C-FD-DX11-B-04, Sheet 1 is a major phase diagram showing the transitions made by the DX11-B logic. On that drawing, the phase number is indicated in the upper right-hand corner of each functional block.

Phase 0, the Idle or Requesting phase, is used for the following functions:

a. Selection Bypassed

   1. Propagate Select-Out
   2. Fast CU Busy Sequence

b. Selection Sequences

   1. ISS in Progress
   2. Polling Sequence
   3. Status Pending

c. Asynchronous Operations

   1. System Reset
   2. PDP-11 Interrupt Granted

d. Programmed Interactions

   1. DX11 Reset
   2. Request Made

Phase 1, the Address Response phase, uses the address from the BUSO (Bus-Out) flip-flops and the Offset Register (CUOR) to obtain the Status Pointer Word (SPW) from PDP-11 core. The information thus obtained is used to set certain control flip-flops. Control of the DX11-B is then transferred to Phase 2.

Phase 2, the Status Preparation phase, is used to prepare for status presentation or, if a CU request has been granted, to transfer to the appropriate phase (3, 5, or 6) from which the DX11-B proceeds with the request. If a command has been accepted, then the Device Status Table (DST) is accessed. If the command was rejected for some reason (e.g., bad parity), then the DST is bypassed. During an ISS, Phase 3 is used next. If any IBM Reset condition is issued during Phase 2, then Phase 4 is used next.

Phase 3 is the Status Presentation phase. The logic controlled by TS1 of this phase governs the transfer of information into BUSI (Bus-In) flip-flops and then onto the Bus-In lines. The logic of TS2 of this phase makes the appropriate responses, depending upon whether status was accepted or stacked.

Phase 4, or Mark phase, stores the device information resulting from the I/O interface transaction into a tumble table of data in PDP-11 core. The information consists of the DXDS (two bytes of device status indicators) and the CUCR (command byte) plus the CUAR (address byte).

Phase 5, the Input phase, governs the transfer of information from the channel into the DX11-B. It contains two separate logic paths which run independently. One path synchronizes NPR word transfers to the PDP-11 with the transfer of data bytes from the I/O interface. The other path controls the signals required by the channel-to-DX11 interaction.

Phase 6 is the Output phase. This phase is similar to the Input phase except that data is transferred out from the DX11-B to the channel.

Phase 7 is the Done phase. It synchronizes the termination of the signal sequences and makes the appropriate response either to an SEL or MPX channel, depending on which is attached to the DX11-B.

There are certain phase-independent operations performed by the DX11-B. These operations synchronize a) the various signals that could occur either independently of the DX11-B clock and/or during any phase, or b) flip-flop changes that must be synchronized with the DX11-B clock.

5.3.6 Channel Simulator Operation

The Channel Simulator comprises built-in diagnostic hardware which is used in conjunction with a PDP-11 diagnostic program to test all DX11-B responses as seen and simulated by the channel. These are the programmable registers DXMO and DXMI. These are 16-bit writable registers to give off-line checks. This block is shown alone in Figure 5-3, as it normally does not function as part of the DX11-B but is put into operation by relocation of channel bus and tag cables to connectors associated with these registers.

The feature includes a set of test receivers and drivers so that tests can be made in on-line mode to check the entire DX11-B hardware without the need for the IBM system with which it normally operates. Channel tags in 360 sequence can be simulated by this hardware, as well as all control signals and data bits.

In off-line mode, all logic *except* the 360/370 bus receivers and drivers are tested.

5.4 SEQUENCES OF OPERATION

As previously explained, the DX11-B is a static piece of hardware when divorced from its specially designed software. It will do very little without program intervention at almost every step of the way from start of an operation to the conclusion of that operation's execution. As such, the sequences of operation are of prime importance in understanding the functioning of the hardware. There are many different sequences depending on both the operation to be performed and the way that the programmer wishes to perfrom them; often times, many sequences occur simultaneously and independently in the IBM interface, the DX11-B itself, and in the PDP-11 interface. There are timing constraints imposed by both the 360/370 channel and the PDP-11 System; in most cases, the time limitations must be satisfied for all three portions of the operational system.

In this paragraph, the sequences with their attendant timing diagrams are divided into two main categories: a) those on the IBM interface, and b) those in the DX11-B, including the interaction with normal PDP-11 interchange.

5.4.1 IBM Sequences

In dealing with the IBM channel, the DX11-B takes part in several sequences of operation. These involve a) initiation of activity either by the channel or by the DX11-B, b) the transfer of data either to or from the 360/370, c) presentation of status to the channel by the DX11-B, and d) reset sequences from the channel to the DX11-B.

In these subparagraphs, detailed descriptions of sequences are arranged in narrative order, in an attempt to give a feel for overall operation. The particulars pertain primarily to operation with an SEL channel since operation with an MPX channel is very similar. Following this, a subparagraph is devoted to the specific differences in the operation of the DX11-B with an MPX channel. Finally, IBM sequences are summarized as to contrasting conditions to give a complete understanding of this very important part of DX11-B operation.

5.4.1.1 Selector Channel-Initiated Sequences (CHI) – In the execution of a typical Channel Command Word (CCW) involving data transfer such as read, write or sense, the channel, in executing a CCW, first selects the device, then performs the data transfer, and finally terminates the CCW by going through an ending status sequence.

The sequence for selecting the device begins with the raising of Operational-Out with the 8-bit device address being placed on the bus by the channel, and with the channel raising Address-Out (Figure 5-6). After this, the channel raises Select-Out and Hold-Out (the latter not shown on the diagram).

NOTE
In these figures the arrows denote sequence only and are not intended to imply preconditions.

At this point each CU on the bus, in turn, propagates Select-Out after comparing the device address with its range of device addresses and finding no comparison. This propagation continues until the CU (e.g., the DX11-B) that is being addressed recognizes the address and raises Operational-In. This prevents any further propagation of Select-Out to any CUs further down the bus and allows the channel to drop the information on Bus-Out, and to drop Address-Out.

NOTE
The latter part of this sequence is somewhat different from all other sequences since normally bus information must remain on the bus as long as the qualifier tag is up.

When the channel drops Address-Out, the DX11-B echoes the device address on Bus-In and raises Address-In.

NOTE
Bus-Out (BUSO) refers to a group of 9 lines comprising 8 data bit lines and 1 parity bit line.

When the device address is echoed on Bus-In by the DX11-B, the channel puts the op code portion of the CCW on Bus-Out and raises Command-Out. When Command-Out rises, the DX11 is permitted to drop Address-In, at which time it waits for Command-Out from the channel to fall.

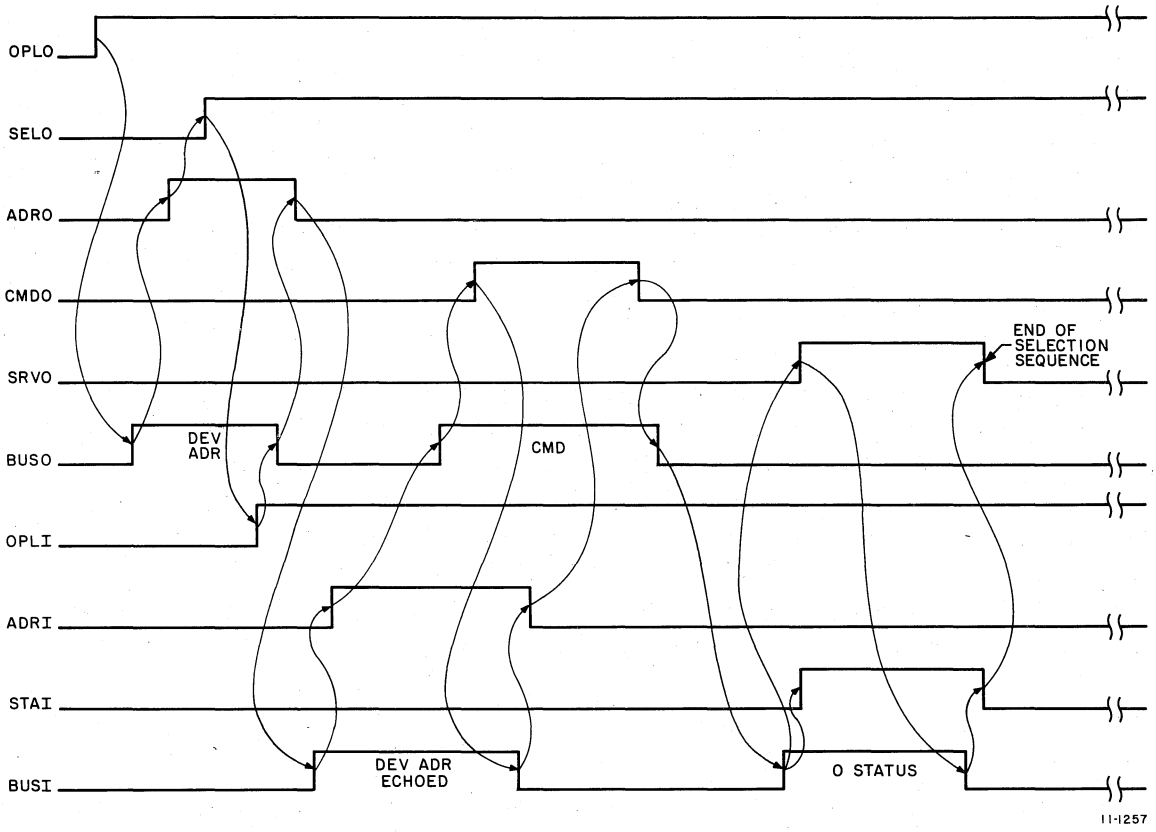


Figure 5-6 Initial Selection Sequence (ISS) Selector Channel Initiated

At this point, the DX11-B can accept or reject the command on the bus. If it accepts it, the DX11 presents a 0 status byte on Bus-In and raises Status-In.

The channel acknowledges receipt of that status by raising Service-Out which, in turn, allows the DX11-B to drop Status-In and take the status byte off of Bus-In. When the channel drops Service-Out the DX11-B is officially selected and the selection sequence is ended. The command is now stored in the DX11-B along with the address of the device commanded ("ordered" in IBM terminology), and the DX11 can now proceed to do what is necessary to get that device going, including interrupting the PDP-11 program so that the data transfers can start.

The ensuing data transfer sequence is shown in Figure 5-7. Assuming that the command, just stored as a result of an ISS, was a Read command, the DX11-B must then put data on Bus-In.

NOTE

Although the convention has been established to designate "read" and "write" operations with respect to IBM, and comparable "out" and "in" operations for PDP-11, the Bus designations *must* remain IN for an IBM read and OUT for an IBM write.

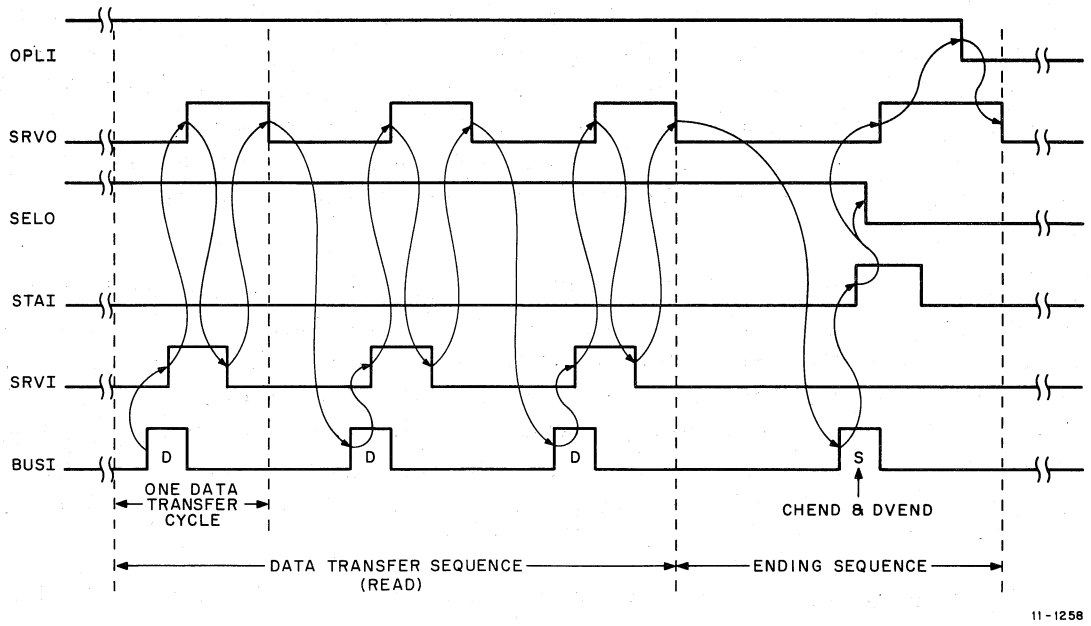


Figure 5-7a Data (Read) Transfer and Ending Sequences with Selector Channel

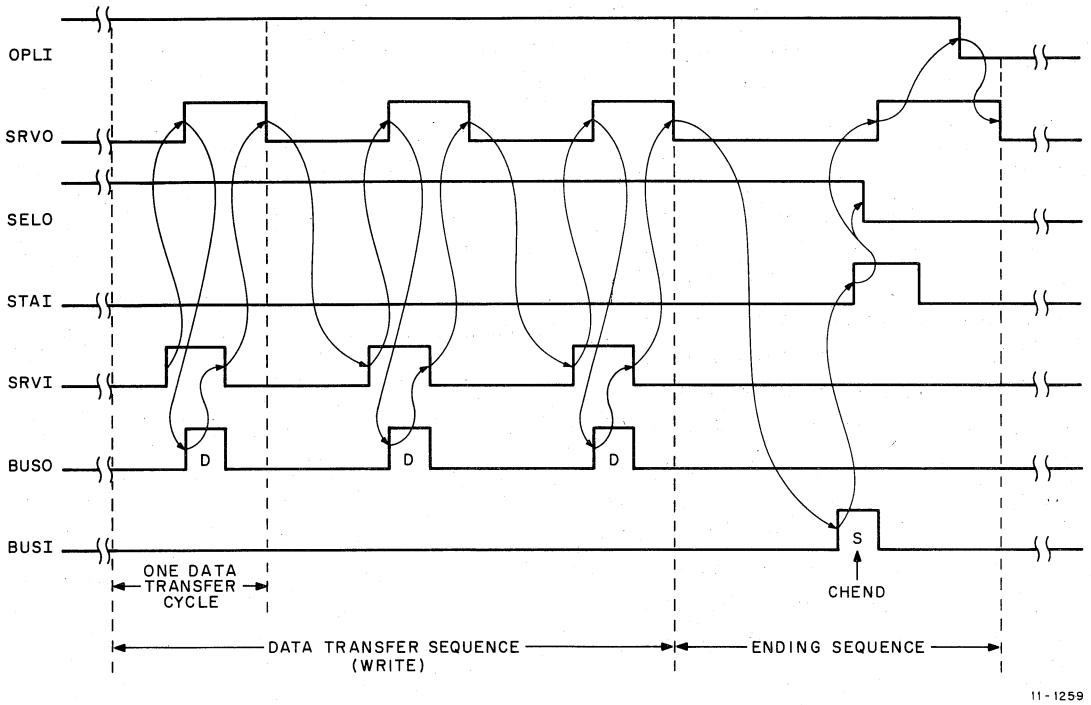


Figure 5-7b Data (Write) Transfer and Ending Sequences with Selector Channel

With data on Bus-In, the DX11-B raises Service-In. The channel acknowledges the receipt of that data by raising Service-Out. At that time the DX11 drops Service-In, takes the data off of Bus-In, and the channel drops Service-Out. This constitutes just one cycle in a data transfer sequence with the same events repeating for each byte of data transferred. Notice that Operational-In remains up throughout the recurring period, as does Operational-Out, Hold-Out, and Select-Out. The DX11-B remains on the channel during this period until it has transferred all the data it has to transfer. If the channel runs out of storage space first, an IO-Stop sequence is initiated. This special sequence is discussed later.

Assuming a normal transfer with no IO-Stop, when the DX11-B has transferred all its data, it ends the transfer sequence by entering an Ending Sequence. This is indicated to the channel by the DX11 presenting ending status instead of data on the Bus-In and then raising Status-In. If there were no errors during the transfer, that ending status will consist of Channel-End (CHEND) and Device-End (DEVEND). The channel accepts that status in the same way that it accepts data, by dropping Select-Out and raising Service-Out. The DX11 responds by dropping Status-In and Operational-In (the DX11 cannot drop Operational-In until Select-Out is down), and the channel comes back with a drop of Service-Out. This concludes the sequence, the DX11-B is off the channel and the tags are down.

Figure 5-7b illustrates the sequence for a write in which the channel puts the data on Bus-Out. As with a Read command, the DX11 raises Service-In and waits for Service-Out to rise. The DX11 then strobes the data on Bus-Out and acknowledges receipt by dropping Service-In. This cycle repeats until the data transfer is complete. At this point, the DX11 places status on Bus-In and raises Status-In and proceeds with a normal ending sequence.

Note again that an NPR is executed for every two bytes of data and to retrieve initial ISS status. A program interrupt is generated at the end of the ISS, at the end of the complete data transfer, and at the end of every status cycle.

Considering the eventuality, mentioned earlier, that the CH might run out of storage area because the size of the buffer (360) was smaller than the amount of PDP-11 data to be transferred to the CH, an additional signal comes in to play, as shown in Figure 5-8. In this case, the CH, instead of acknowledging the receipt of the data by raising Service-Out, raises Command-Out instead. This indicates to the DX11-B that the CH did not accept the data that was presented on Bus-In, and that it should not raise Service-In again. It also requires the DX11 to raise Status-In as its next In-Tag to end the sequence. The status is presented on Bus-In. Raising Status-In causes the channel to drop Select-Out and to raise Service-Out. When the DX11 sees this, it drops Operational-In and Status-In; the channel then drops Service-Out. The sequence is ended and the proper interrupts have been made along the way to the PDP-11 program.

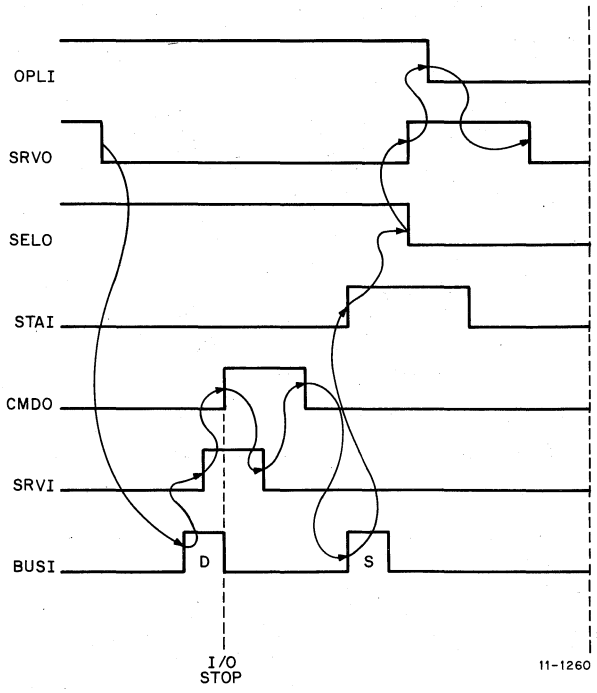


Figure 5-8 I/O Stop Sequence on Selector Channel

If command chaining is indicated in the CCW, the DX11-B is required to do nothing but wait for the next selection to occur immediately. This condition is indicated to the DX11, when ending status is presented, by the channel raising both Suppress-Out and Service-Out.

In this sequence, it may be possible that the channel cannot accept status when the DX11-B wishes to present it because it has no storage available into which to put it. In this event, the DX11 must be prepared to "stack" the status. This sequence is shown briefly in Figure 5-9, in which the DX11 is on the channel with Select-Out and Operational-In both up. When the DX11-B places the status byte on Bus-In and raises Status-In, the channel immediately raises Command-Out and drops Select-Out. This indicates to the DX11 that it must get off the channel by dropping Status-In, by taking the status off of Bus-In, and by dropping Operational-In. The channel then drops Command-Out and the sequence ends with the DX11-B not having presented status for the transfer completed.

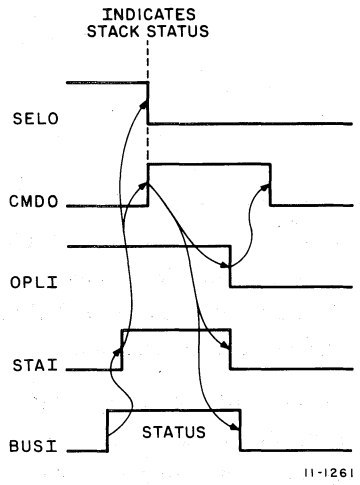


Figure 5-9
Stack Status Sequence

The above conditions represent unfinished business for the DX11-B. It has the status it needs to present, and, in order to present it, the DX11-B must get back on the channel. If the DX11 at this time were to arbitrarily raise Status-In again, the channel would have no way of knowing the source. The sequence for finishing this condition is explained in Paragraph 5.4.1.2 in which a Control Unit Initiated Sequence (CUIS) is described.

In the sequences just discussed, where no data transfer results as part of the ISS, status is generated as a function of the command received unless status was already pending for that device. If no status was pending but parity on the command was bad, then Unit Check (UCHECK) is presented to reject the command. The command is also rejected if the addressed device is off-line or if the command is illegal for the particular device.

When the DX11-B is not operational because it is either powered-down or is powered-up and in off-line state, Select-Out is propagated automatically: all in-tag, in-control and bus-in functions are logically disconnected from the channel interface. An attempt by the channel to address the DX11-B in this instance will cause Select-In to rise (eventually) back at the channel (and via the channel to the CP program), indicating that the DX11 is logically not on the interface.

5.4.1.2 Control Unit Initiated Sequences (CUI) – The DX11-B initiates selection sequences to transfer data to, request data from, or transfer status to the channel; the latter being the purpose carried over from the previous paragraph as just one example. A status transfer can arise because emulator status has just become available, or because a previous status transfer was stacked by the channel, as is the case in this running description. On an SEL channel, only status is actually transferred by a CUI selection sequence. On an MPX channel, the transfer can consist of one or more data bytes and/or status bytes.

As shown in Figure 5-10, the DX11-B raises Request-In to execute a CUI. This is a parallel line that indicates to the channel that some CU on the bus desires to use the channel. The channel, when it is free, responds by raising Select-Out (and Hold-Out). Select-Out is propagated in the same manner that it was by CHIS, but this time the DX11-B does not do an address compare. The receipt of Select-Out is all that is necessary for it to raise

Operational-In. Now that Operational-In is raised, the DX11 can drop Request-In and at this point it puts its address on Bus-In to identify itself to the channel. When the channel decodes the address, it acknowledges by dropping Select-Out and raising Command-Out.

NOTE

This is a special use of Command-Out. There is no command on the bus. In this instance it is used as part of the dialogue to acknowledge receipt of the initiating address, i.e., that a polling sequence has been completed.

When the DX11-B sees acknowledgment of its address, it drops Address-In and takes its address off of Bus-In. Since the channel will have dropped Command-Out, and Operational-In is up, the DX11-B is now on the channel and the sequence is complete.

Continuing with the example started in the previous paragraph, now that the DX11 is back on the channel, it can attempt to present status once again. There is no guarantee that the channel will accept it, particularly if it still has no room. Since there is no harm in trying, the DX11-B proceeds as it did before to present status by putting the status byte on Bus-In and raising Status-In. Assuming that the channel accepts the status this time, it raises Service-Out instead of Command-Out, the DX11 drops Operational-In, and the sequence terminates normally.

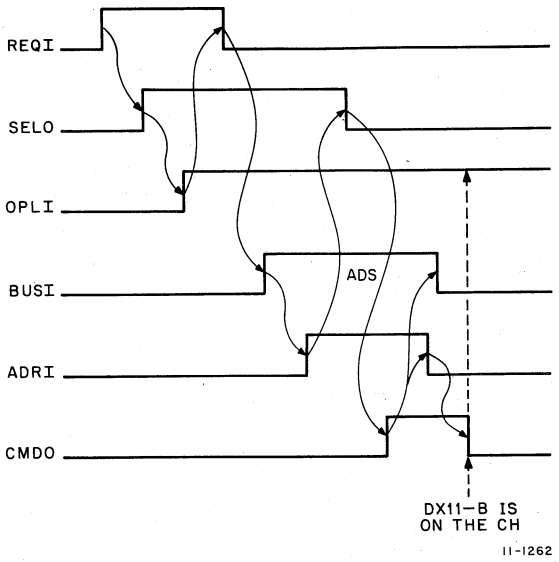


Figure 5-10 CUI Sequence to Selector Channel

5.4.1.3 Reset Sequences – There are three reset sequences initiated by the channel to the DX11-B. They are: a) System Reset, b) Selective Reset, and c) Interface Disconnect.

An *interface disconnect* is an asynchronous event that can occur at any time as a result of the 360 program executing an HIO instruction. An HIO can occur under three possible conditions: a) when the device is idle, so consequently the DX11-B is off the channel and Operational-In is down; b) when the DX11-B is on the channel but is currently between sequences with Operational-In and Select-Out both up, but all other tags are down; or c) during a data transfer with Select-Out and Operational-In both up and Service-In is raised. A Halt-IO (HIO) is defined as a static condition in which Select-Out=0 and Address-Out=1.

The sequence for case a is shown in Figure 5-11. The DX11-B is off the channel presently idle. When the program issues an HIO, the channel must first select the

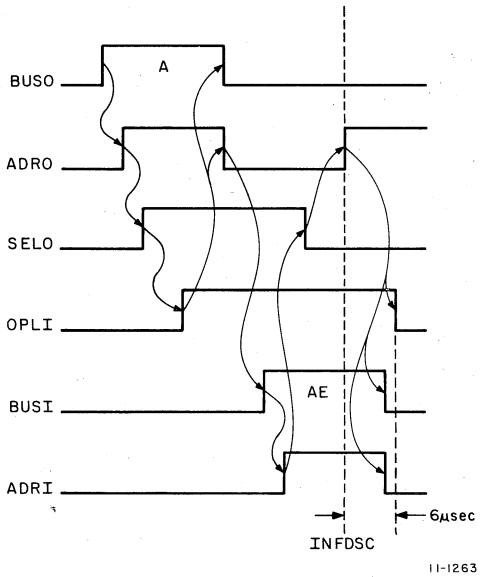


Figure 5-11 Interface Disconnect Sequence to CU Idle *off* the Channel

DX11 in order to tell it to deselect. This is done by putting the DX11 address on Bus-Out, then raising Address-Out and Select-Out. Select-Out is propagated until seen and recognized by the DX11-B. At that point, the DX11 raises Operational-In which drops Address-Out. Then the DX11 puts its address on Bus-In and raises Address-In, which causes Select-Out to fall and Address-Out to rise. With the conditions for interface disconnect now met, the DX11-B has 6 μs to take everything off the bus and drop Operational-In. This 6 μs limitation to get off the bus is extremely important in dealing with the 360/370, particularly when operating with an MPX channel, which will hang if the time limit is not met and will, in turn, hang the whole system. The DX11-B is equipped with complex logic to ensure that the 6 μs limit is met.

The sequence for case b is shown in Figure 5-12. The DX11-B is on the channel presently idle, as it would be between bytes of a data transfer, or if it had just been selected and had not begun transferring data as yet, or if all the data had been transferred and it was presently generating status that had not as yet been presented. Both Select-Out and Operational-In are up when the program issues an HIO. The channel drops Select-Out at this point, which causes the DX11 to do nothing, then raises Address-Out. This is the interface disconnect combination. The DX11-B then drops Operational-In and Address-Out drops. This concludes the sequence and the 6 μs condition prevails as before.

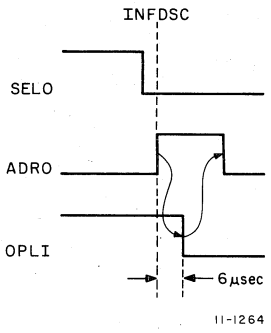


Figure 5-12 Interface Disconnect Sequence to CU Idle *on* the Channel

The sequence for case c is shown in Figure 5-13. The DX11-B is on the channel performing a data transfer when the program issues an HIO. Operational-In and Select-Out are both up, and the DX11 raises Service-In. Instead of responding with Service-Out or Command-Out, the channel drops Select-Out and raises Address-Out, creating the conditions for an interface disconnect. Seeing this, the DX11-B drops both Service-In and Operational-In and the sequence is complete.

Note that in these cases, once again IBM uses a tag for a purpose other than what its mnemonic would indicate. As with Command-Out in the CUI sequence, Address-Out is used here, with no address on the bus, to combine with Select-Out in creating an interface disconnect.

In summary, on an interface disconnect from the channel, the DX11-B removes all signals to the 360/370 bus. The DX11 notifies the emulator of the condition; subsequent DX11 actions are a function of the emulator.

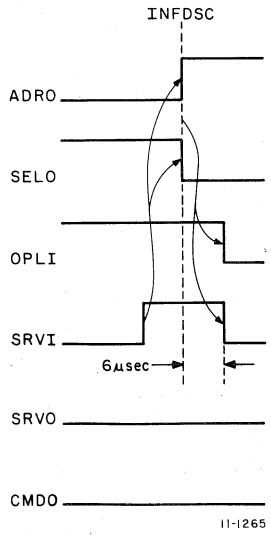


Figure 5-13 Interface Disconnect Sequence to CU Active *on* the Channel

Selective Reset is indicated when Operational-In is down and Suppress-Out is up when the DX11-B is concurrently operational with the channel. When these conditions exist, the DX11 drops all in-tags and notifies the emulator of the condition. Selective Reset affects only the currently active device.

System Reset is indicated when Operational-Out and Suppress-Out are both down concurrently. The emulator is

notified of the condition. The emulator clears all indicators and registers in the IBM Control Section, except the CU Busy indicator, which is set. The DX11 must drop all in-tags when system reset is indicated. Subsequent DX11 actions are a function of the emulator.

5.4.1.4 Multiplexer Channel Differences – Figures 5-14 through 5-16 illustrate the normal sequences when the DX11-B is attached to an MPX channel. Comparison of these diagrams with those for an SEL channel will show the similarities in the sequences. The principle difference is that Select-Out does not remain up during the sequence. On the SEL channel it is the fact that Select-Out remains up that forces channel forced burst mode. On that type of channel, Operational-In must remain up while Select-Out is up.

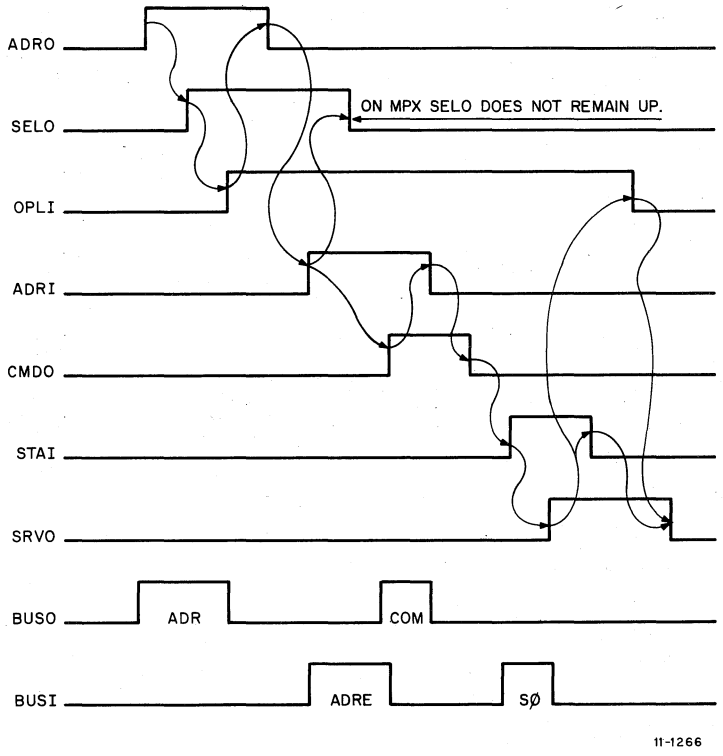


Figure 5-14 ISS, Multiplexer Channel Initiated

During an ISS on an MPX channel, Select-Out is allowed to fall shortly after Operational-In rises and after Address-In rises. After an ISS is completed, the DX11 drops Operational-In (unless forcing burst mode) because selection has already taken place, the command is known, and the device address has been identified. When that device has data to transfer (Figure 5-15), a CUIS is executed to get back on the channel. Once back, the DX11 has Operational-In up again and can raise Service-In, as required, to transfer the data. When the data is transferred, the DX11 drops Operational-In.

When the DX11 has more data to transfer, it goes through another CUIS. When a data transfer is finally complete, another CUIS must be executed in order to present status (on another page) (Figure 5-16).

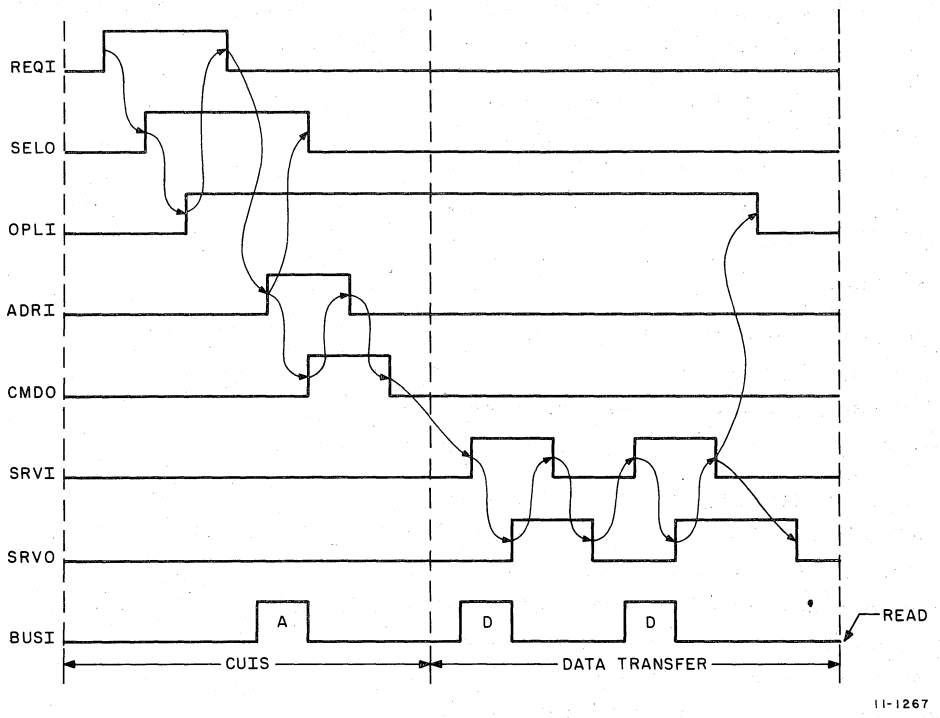


Figure 5-15 CUIS and Data Transfer, Multiplexer Channel

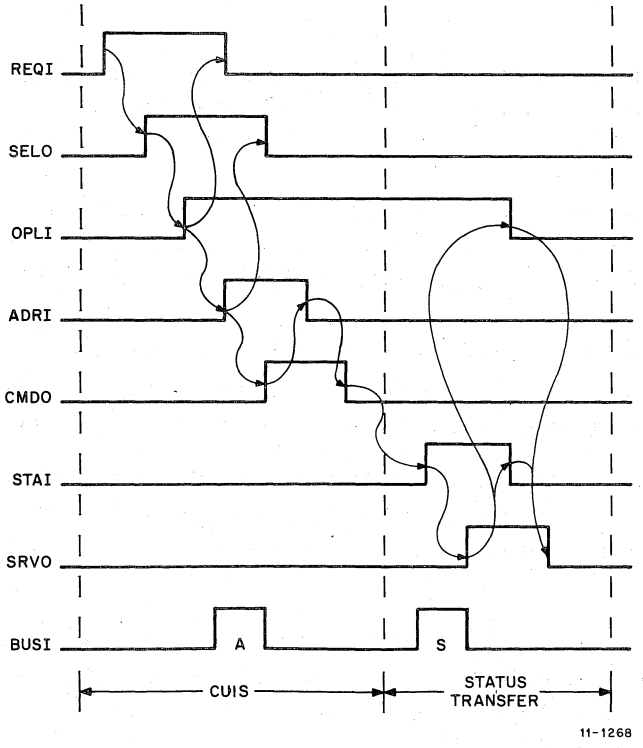


Figure 5-16 CUIS and Status Transfer, Multiplexer Channel

The reason for getting on and off in this fashion is to allow interleaving of data from different devices on the MPX channel. This allows several relatively slow devices to be run simultaneously, each transferring a little data when they have it ready without holding up any other device on the channel.

There is special logic in the DX11-B to deal with a situation on an MPX channel (shared CU) when the channel tries to select a device when that CU is busy servicing another device. Because the CU is shared, it can only service the first device, but must respond to the selection for the second. This response is called a Fast CU Busy sequence.

Figure 5-17 illustrates this sequence. The selection starts normally since the CU is off the channel, but the difference here is that the CU is busy and is in between sequences on the channel for the device it is servicing.

The channel begins by placing an address for the second device on Bus-Out and raising both Address-Out and Select-Out. The DX11 recognizes that the address is within its range of addresses, so it cannot propagate Select-Out. If it were not busy it would raise Operational-In, but it cannot accept this selection sequence so it places a special status on Bus-In (Status Modifier + Busy and optionally CUEND) and raises Status-In. This indicates to the channel that the DX11 has recognized its address selection but cannot accept it at this time. The channel then drops its tags to abort the selection and the DX11 responds by dropping *its* tags. This completes the sequence.

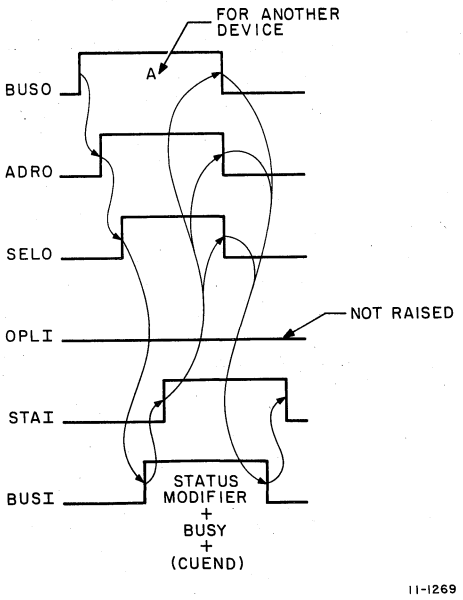


Figure 5-17 Fast CU Busy Sequence, Multiplexer Channel

NOTE

The status CUEND is not presented for 2848 emulation, but some other emulators will present it along with the other status.

5.4.1.5 IBM Sequence Summary – There are two types of selection sequences: Channel Initiated (CHI), and Control Unit Initiated (CUI). On a CHI, when Address-Out rises, the DX11 decodes the address being presented on Bus-Out, including parity, then waits for Select-Out. When Select-Out rises, the DX11 response is a function of the address and parity on Bus-Out, as well as the busy state of the DX11. If the DX11 does not recognize the address, or if the parity is incorrect, the DX11 propagates Select-Out. It will continue to do this until the incoming Select-Out falls. If the DX11 *does* recognize the address and if the parity *is* correct, the DX11 blocks propagation of Select-Out and raises Operational-In or Status-In. It then waits for Address-Out to fall.

Status-In is raised if the DX11 is operating in Single-Thread mode and a previous selection is still active for another device address. Busy and Status-Modifier are placed on the bus as status. When Address-Out falls, Status-In is dropped, ending the sequence. Operational-In is raised if the DX11 can accept the selection or if the device addressed has status pending.

When Address-Out falls, the DX11 gates the recognized address on to Bus-In and raises Address-In. When Command-Out rises in response to Address-In, the DX11 decodes the command being presented on Bus-Out, including parity. When the DX11 no longer requires the data on Bus-Out, it drops Address-In, removes the address from Bus-In and waits for Command-Out to fall.

When Command-Out falls, and the DX11 is ready to present initial status, it gates the status byte to Bus-In and raises Status-In. Status is generated as a function of the command received unless status was already pending for that device. If no status was pending but parity on the command was bad, then Unit-Check is presented to reject the command. The command is also rejected if the addressed device is off-line or if the command is illegal for that particular device.

The channel accepts the presented status by raising Service-Out. Alternatively, the channel can stack the status (for presentation by the DX11 at a later time) by raising Command-Out. In either case, the DX11 removes the status data from Bus-In, drops Status-In, and waits for the out-tag to fall.

When the DX11 is not operational because it is powered-down or if it is powered-up and in the off-line state, Select-Out is propagated automatically. All in-tag, in-control, and bus-in functions are logically disconnected from the channel interface.

During a CUI, the DX11 raises Request-In at the interface unless it is already connected (Operational-In is up). When the channel initiates a polling sequence by raising Select-Out while Address-Out is down, the rise of Select-Out at the DX11 will cause the DX11 to raise Operational-In and drop Request-In. The DX11 gates the address to Bus-In, raises Address-In, and waits for Command-Out. The rise of Command-Out signals the acceptance of the address, the DX11 removes the data from Bus-In and drops Address-In, then waits for Command-Out to fall. The DX11-B response to this depends on the type of transfer. For data output to the channel, the DX11 gates the data byte to Bus-In. For data input or output, DX11 raises Service-In. For status, the DX11 gates the status byte to Bus-In and raises Status-In. In all cases, the DX11 then waits for an out-tag to rise.

If the channel raises Service-Out, the DX11 response depends on the transfer type. For input to the channel, it removes the data from Bus-In and drops Service-In. For output from the channel, it transfers the data on Bus-Out to the device, then drops Service-In. For status transfers, the DX11 removes the status from Bus-In and drops Status-In before initiating another action. In all cases, the DX11 waits for Service-Out to drop. Unless Control Unit Forced Burst mode is in effect, the DX11 will drop Operational-In after the rise of Service-Out, if the DX11 was transferring a single byte.

If the channel raises Command-Out, the DX11 response once again depends on the type of transfer. For input to the channel, response is as for Service-Out, except that no further data transfers are requested. Status may or may not be presented at this time, depending on whether or not Burst mode is in effect and whether status is available from the emulator. For output, a termination indication is transferred to the emulator, the data on Bus-Out is 0, but parity is not significant. No further data transfers are requested. For status, if the DX11 is operating in single-thread mode, the DX11 places the status in the stacked state and prepares for a new CUIS under control of Suppress-Out. If in multi-thread mode, the emulator must recognize the stacked condition and attempt to re-present the status at a convenient time.

After status presentation, the DX11 drops the in-tag and waits for Command-Out to fall. The DX11 also drops Operational-In and, if status has been stacked, the DX11 can raise Request-In to prepare for a new CUIS.

Data transfers can be caused by the execution of any one command but will occur in only one direction across the interface determined by the emulator at the time it initializes for the first byte transfer. Each device then retains that direction until Channel-End has been presented. Read-type commands, including Sense, send data *to* the channel; while Write commands, including some control commands, input data *from* the channel.

Upon a read command, the DX11 gates the data to Bus-In and raises Service-In, waiting then for Service-Out to rise. When it does, the DX11 drops Service-In and removes the data from Bus-In, waiting then for Service-Out to fall.

For a Write command, the DX11 raises Service-In and waits for Service-Out to rise. When it does, the DX11 strobes the data on Bus-Out. When data is no longer needed, the DX11 drops Service-In and waits for Service-Out to fall.

For both Read and Write commands, Service-Out must be down and Operational-In must be up before Service-In can rise. The data on Bus-In is loaded 200 ns before the rise of Status-In or Service-In.

The distribution of data cycles relative to other signal sequences is a function of the data transfer mode, either "burst" or "byte". In CH Forced Burst mode, Select-Out remains up from the ISS through the presentation of Channel-End. Operational-In remains up until Select-Out falls. Thus, the data cycles are all contiguous and are bracketed by ISS and ES.

Normally, CU forced burst means that Operational-In is held up from the ISS through the presentation of Channel-End. Use of burst mode on the MPX channel must be explicitly requested by the emulator before the CH initiates the associated selection sequence.

The data sequence can be terminated by either the channel or the DX11-B. The channel terminates a data sequence by raising Command-Out instead of Service-Out during a data cycle. During input, the data on Bus-In is ignored. During output, no data is presented on Bus-Out. The DX11 indicates the termination condition to the emulator.

The DX11, at the request of the emulator, terminates the operation with the channel by gating status to Bus-In and raising Status-In. Channel-End is presented at this time along with any other status conditions, including Device-End if it is available. If Device-End is not presented, a separate status cycle is required to present Device-End when it does become available.

The channel can attempt to suppress data cycles by raising Suppress-Out while Service-Out is up in response to Service-In. When this occurs, the DX11 will not raise Service-In again (except for the first byte after a CUIS) until Suppress-Out falls. If, however, the device requests a termination at this time, Status-In can be raised, regardless of the condition of Suppress-Out.

A status cycle can begin only when Operational-In is up and all out-tags are down. The DX11 places the status on Bus-In, raises Status-In, and waits for an out-tag to rise. For a normal status cycle, the channel raises Service-Out when the status has been read. The DX11 drops Status-In and waits for Service-Out to fall. If the channel raises Command-Out in response to Status-In, the DX11 enters the Stacked Status condition.

The occurrence of status cycles relative to other signal sequences is a function of the availability and type of status involved. However, between the time that the DX11 has been selected and the time that either Device-End or some other status that precludes presentation of Device-End has been accepted by the channel, the DX11 is in the "working" condition. The DX11 will execute as many status cycles as are required to clear "working". At most, this consists of three cycles, as for the typical I/O operation having the following cycles:

a. An ISS 0-status cycle (pre-I/O status, part of the ISS)
b. An ES Channel-End cycle
c. An ES Device-End cycle

If the channel indicates Stack-Status, the DX11-B retains the status and will attempt to present it again by a CUIS. This status indication is retained by the DX11 for the device in question until subsequently accepted by the channel.

If the channel raises Suppress-Out before raising Select-Out in response to Request-In, it is requesting the DX11 to suppress status. If Request-In is up to request a data transfer, or the transfer of status that has not been

stacked, the DX11 will proceed with the CUIS regardless of Suppress-Out. However, if the status has been stacked, the DX11 will respond to Suppress-Status by dropping Request-In. It will not raise Request-In again for the stacked status until Suppress-Out falls.

Command chaining is indicated when the channel raises Suppress-Out before raising Service-Out in response to Status-In. The command-chaining indication for the affected device is retained until the device is again selected by the channel. If command chaining is indicated in response to Channel-End (Device-End not yet available), the DX11 must present Device-End for the affected device before any pending Device-End indications for other devices unless, in the interim, the DX11 is addressed again for another device.

Most status is emulator-provided, either from preset tables (for ISS status) or from status transmitted directly by the emulator (for ending status). However, under certain circumstances, the DX11 itself can affect status and sense data. These circumstances involve malfunctions or exceptional conditions to which *all* control units respond identically. One of these conditions is incorrect parity on Bus-Out during Command-Out. The DX11 will present Unit Check alone during ISS status presentation. During the subsequent Sense operation, the emulator presents Bus-Out-Check. Another condition occurs during Control Unit Busy sequences, when the DX11 will present Busy and Status-Modifier in response to the selection attempt. Control Unit End can be presented *if* enabled by the device.

If the channel attempts to address a nonexistent device, within the range of device numbers assigned to the DX11, another example is illustrated as the DX11 responds with Unit Check alone during ISS status presentation. During the subsequent Sense operation, the emulator will present Intervention-Required.

If the DX11 attempts to address nonexistent memory, the DX11 responds with Unit Check alone during ISS status presentation and indicates NXM on its indicator panel. During the subsequent Sense operation, the emulator will present Equipment-Check.

5.4.2 DX11-B Sequences

The DX11-B sequences are described in Drawing D-FD-DX11-B-04, Sheets 1 through 9. The symbology used is unique to this equipment and requires some explanation. The technique used in these flow diagrams is such that a complete understanding of the entire operation of the DX11-B can be gained by following through the sequences. Termed "Design Flows", the DX11 was designed directly from these flows. Because of their completeness, the sequence discussions that follow will constitute the bulk of operational discussions. No gate-by-gate discussions are necessary in this manual.

In this paragraph, the basic data paths are first discussed. This is followed by an explanation of the conventions used in the flows and a detailed discussion of the design flows.

5.4.2.1 Register Organization – Sheet 9 of Drawing 04 contains the DX11-B data paths. The registers on Sheet 9 are shown in detail on Sheet 8. Refer to that sheet for an explanation of the programmable registers, their mnemonics, and uses. The conventions used on this diagram are explained in a legend at the bottom of the page. Note that as part of this legend the symbol for data gate number is given as an arrow intersected by a dotted line and a number. These numbers refer to a list of signals on the right-hand side of the page. They appear in the block schematics and on the control flows.

The significance of the mnemonics assigned may not always be obvious, but they do follow a loose convention. Gate 1, for example, pertains to the signal ARSO. For the most part, these signals were derived by dropping the prefix from a register (either CU or DX) and combining the last two letters of both registers (or signals) involved in that transaction. The mnemonic ARSO means the Control Unit Address Register (CU*AR*) gets the contents of

Bus-Out (BU*SO*). The mnemonic BAAR means the DX11 Bus Address Register (DX*BA*) gets the shifted contents of the Control Unit Address Register (CU*AR*). There are exceptions as with the signal for gate number 4 (BIAR), which means that *B*us-*I*n gets the contents of CU*AR*. When confusion arises as to the meaning of a mnemonic, refer to Appendix A of this manual, which contains a glossary of signals used in the DX11-B.

Note that the drawing is divided into three vertical areas segregating the registers and data paths associated with IBM Control, Central Control, and PDP-11 Control. Along the top edge of this sheet are four ellipses which represent external connections to the IBM channel and DX11-B Channel Simulator. The left-hand plug receives the lines that come from the IBM Channel; this is termed the Bus-Out plug. The Select-Out line of this plug can be connected to a hardware element (shown in a small ellipse which bypasses the Select-Out signal) either when off-line to the channel, or when the PDP-11 is powered down. The signals from the Bus-Out plug also go to the Bus-Out receivers. After the signals have been amplified by the receivers, the clock out signal is sent to the CLKO location contained in DXMI. The other signals from the receivers are gated by whether or not the DX11 is on-line.

If the DX11 *is* on-line, these signals are presented to the inputs of the CONO flip-flops. These flip-flops are clocked at either T1 or T2. Note that the Select-In tag, which is held in CONI, is cleared if either Hold-Out or Select-Out fall. The plug data is not clocked into flip-flops but is presented continually as BUSO to both a parity check and to a comparison circuit. The parity check circuit produces a signal called parity okay (PAROK) which, when a 1, indicates that the proper parity has been seen on Bus-Out. If the BUSO lines contain an address, gate number 1 will be activated by the control flow logic, thereby copying the contents of BUSO into the CUAR.

All of the data paths indicated on this diagram are activated by the logic control flows discussed later in Paragraph 5.4.2.3.

Gates 1 through 4 are involved in *Address Response.* Gate number 1 (ASRO) is hit when an address is being copied from the BUSO as the CUAR gets the contents of BUSO. Gate number 2 (BAAR) is used to read the shifted contents of the CUAR into the DXBA in preparation for a fetch of the status pointer word (SPW). At the same time that gate 2 is activated, gate 3 (BAOR) is used to bring in the contents of the Control Unit Offset Register (CUOR). Gate number 4 (BIAR), shown at the upper middle of the page, is used to read the contents of the CUAR and place it on the BUSI lines for presentation to the Bus-In plug.

Gates 5 through 9 are involved in *Status Preparation.* After the SPW has been fetched, gate number 5 (BANDSO) is used to read the pointer part of the SPW from the DXND to the left half of the DXBA. At the same time, gates 6 and 7 are used to copy the command (which is now in the BUSO line) into both the CUCR and the right half of the DXBA.

Gate 8 (SRNDH) or 9a (SRNDL), which is to the right under CUSR, is used to copy the status byte that is subsequently fetched from the DST into the CUSR. Actually, either gate 8 or gate 9 (SIG3) is used for this purpose; they are mutually exclusive.

Gates 9 through 15 are used for *Status Presentation.* Gate 9 (SIG3) can be used to read the status half of the DXND when it contains the contents of a DST. Status bits that come from the SPW and are presented to BUSI are available for the program to read via CUSR. All bits that are (or were) presented to Bus-In are available to the program by reading BUSI. Gate 10 (BISR), located under BUSI, is used to strobe the contents of the status register into the BUSI flip-flops from which it is represented to the Bus-In plug if on-line and enabled.

Gate 11 (SIG8) is used along with gate 12 (also SIG8) to read the contents of both the CUCR and the CUAR (the DXCA) into the DXND in preparation for the first tumble table entry. The second tumble table entry, namely the DXDS, is gated to the DXND through gates 14 (SIG7) and 15 (also SIG7). Gate 13 (BANX) is used to gate the shifted contents of the TT*NDX* into the DX*BA* to advance the tumble table address. It is shifted left one position because A00 on the PDP-11 Unibus is not used for an address bit but is rather a byte pointer. On the DX11-B interface, this bit is permanently grounded to force this bit to always be 0.

Gates 16 through 23 are used for *Data Transfer.* Gate numbers 16 (CRSO2) and 17 (SRSO) are used to gate the second and first bytes of the data into CUCR and CUSR, respectively. Their contents are transmitted to the DXND by gates 18 (NDCRSR) and 19 (also NDCRSR). Note that gate 18 is identical in function to gate 11 though used for different kinds of information.

The contents of the DXND, on the other hand, are transmitted to the CUCR and CUSR during a PDP-11 output transmission by means of gates 22 (CRSRND) and 23 (also CRSRND). Gates 20 (SISR) and 21 (SICR) enable that same data, byte by byte, to the BUSI.

Note that BUSO is continually gated to both the parity check and the address comparison. BUSO also goes to gates 7, 16, 1, 17 and 6. The BUSO data can also be read by means of a PDP-11 programmed read. This is indicated by the dark arrow coming out of the BUSO register.

5.4.2.2 Flow Diagram Conventions – Referring to Sheet 1 of this set, a "Flow Diagram Legend" is given containing the various symbols used in the flows. For example, a condition followed by a curved line to which a right arrow is appended, implies that the condition (or conditions) contained to the left of the curved line will produce the action to the right of the vertical line to which the arrow points. A condition may be a Boolean expression containing logical AND, logical OR, the state of the flip-flops, and/or the condition of the signals. The AND is symbolized by a dot and is given higher precedence than a logical OR operation, which is symbolized by a + sign. Thus, when the OR of two conditions is itself to be ANDed with other conditions, the entire OR expression is enclosed in parentheses.

An action or result can be one of four things:

a. A signal level signified by an equals sign.

b. A flip-flop transitioned to a particular state indicated by a left arrow meaning the function "gets".

c. A gating operation in which one byte or a two byte register is given the contents of another byte or register represented by a double left arrow.

d. A reset operation in which one byte or register is cleared, shown as a double left arrow followed by a 0.

Certain signals that can change in a completely asynchronous fashion (e.g., the on-line enable switch) or that can be slightly out of phase with other signal transitions are enclosed in brackets.

When a condition is used as an enabling condition, there are three ways in which it can be portrayed. The first two have the same meaning:

a. Vertical line and condition and right arrow starts off an additional line of logic (sometimes used to indicate parallel sequences splitting off or simply to show additional enabling conditions to perform some action).

b. Right arrow and condition in parentheses and another right arrow used when a single condition is necessary to continue and usually when a symmetric split is made in the flow.

c. An enabling path broken by a double squiggly line. This means that if the line above the break was enabled, control will eventually get to this point. To proceed, however, the condition to the left of the right arrow below the break must be met. This is the full, necessary condition for proceeding. This provides the method of consolidating signal sequences. The break implies that there could be a wait until the condition is met. A number appearing in this break indicates approximately how many time state cycles will be made before the condition is satisfied. A 0 between the break lines means that the condition is normally met immediately.

When a logic line terminates at a circle with a letter within it, a page connector is indicated. This is for informational purposes and serves no logical function. The matching circle will be found at the top of another logic path, either on the same page or another. Occasionally two page connectors will be found connected to a single logic line. This indicates that either or both paths may have become enabled.

Beneath each reference to a page connector is a short descriptive comment, in parentheses, on the reason for the reference. Similar comments at the top of a page describe categories of the logic paths that follow.

A downward pointing triangle indicates that a phase transition will be made. All phase transitions are made by changing the appropriate bits in the phase control flip-flop when TP2 appears, and when the conditions required during TS2 have been met. When the phase transition occurs, only the conditions in the new phase will be enabled and the conditions in the old phase will be disabled completely.

When a logic path has been enabled, many times a signal name is assigned to that line and printed vertically along the logic path. These signals include phase and time state conditions.

Certain synchronization operations are phase independent; they are also shown on Sheet 1. For example, examining the seventh action in the center list titled SYNCHRONIZATION OPERATION, when both IBM signals, HLDOUT and SELOUT, equal 1, then the flip-flop SELO will get a 1. All phase independent actions are accomplished by either TP1 or TP2. This sheet also shows various special signals that are generated. These signals contain no phase or time state information. This is also an index of the major phase interactions, with the sheet number on which that phase can be found. These are shown as part of the diagram at the right-top titled MAJOR PHASE DIAGRAM AND INDEX.

Program interventions that trigger special responses are emphasized by a pointing hand symbol.

Each gate has a footnote number to correlate it with the logic signal name used internally to perform the operation. These number-name pairs appear at the right or bottom and are the same as used in the control flows and the data paths.

5.4.2.3 Design Flows – This paragraph refers to Sheets 2 through 7 of Drawing 04. These pages are arranged in phases of the DX11-B, with each page divided into two sectors (top and bottom) representing the two possible time states (TS1 and TS2) of each phase. As described earlier, a sequence will proceed through the phases but not necessarily in order. In some instances, the sequence will repeat an earlier phase as with Phase 2 during which status is prepared. Depending upon whether the command from the channel is odd or even, the sequence will proceed from Phase 2 to either Phase 5 for input (360 write), or Phase 6 for output (360 read). The phases, as shown on the Major Phase Diagram and Index on Sheet 1 are:

a. Phase 0, Idle or Requesting
b. Phase 1, Address Response
c. Phase 2, Status Preparation
d. Phase 3, Status Presentation
e. Phase 4, Mark
f. Phase 5, Input
g. Phase 6, Output, and
h. Phase 7, Done.

As Drawing 04 indicates, there are many combinations of phase rotation depending upon the function being performed. Note that Phase 4, the Mark phase, is automatically entered when any recordable event takes place. Upon entering this phase, the DX11-B performs two NPRs to enter the activity in the tumble table. All four reset functions also require the DX11 to go to Mark before returning to the Idle state.

a. Phase 0 – Idle or Requesting

Referring to Sheet 1 of Drawing 04, in Phase 0, the DX11-B is either idle with nothing to do, or it is doing one of four main tasks: Selection Bypass, Selection Sequences, Asynchronous Operations, and/or Program Interactions. Neither of the first two can become active unless Operational-Out is high, Select-Out is high, Select-In is low, and Status-In is low. The Select-In low signal implies that the Select-Out signal is not being propagated. The Status-In low means that the DX11 is not presently presenting status by means of the Fast CU Busy sequence.

Assuming that Request-In is down, and either Address-Out is 0 or the DX11 does not match the control unit address, i.e., ADRECC=0, then the DX11 will follow the left-most branch on this diagram and Select-In will be enabled to be set when the clock ticks at the end of TS1. This will cause Select-Out to be propagated to the next CU on the bus.

If the address on Bus-Out is recognized (ADRECC=1), but the DX11 is busy with another device (CUBSY=1), the DX11 follows the Fast CU Busy line which terminates in page connector L2. When the clock ticks at the end of TS1, the signal FASTCU will perform all of the operations which are listed beside it, namely, ISSREJ will be set to 1, BYPAS gets a 1, and the BUSI flip-flops will be loaded with the appropriate status bits.

In TS2 of line L2, the "necessary and sufficient conditions" to set Status-In are that BYPAS=1 (it is), SELO=1 (it is), and that SYNC=0 (which is true because the selection sequences have been bypassed). Therefore, Status-In is set to a 1 at TP2.

Notice that Select-Out=0 is a requirement for clearing Status-In and nothing further will be enabled during TS2. Therefore, the clock continues to tick through TS1 and TS2, but nothing happens until the Select-Out line is strobed to the 0 state at TS1, or TS2, or if a System-Reset is issued by the channel. The latter condition is illustrated as the left of two asynchronous operations in the center of the page. When the Select-Out line is strobed to 0, this will now enable the clearing of Status-In and the transition of BYPAS to 0. At this time, the DX11-B is once again idle.

Assuming that ADRECC matches on an initial selection sequence, and that the DX11 is *not* busy (CUBSY=0), the first thing done is to set LOCKO and SYNC to 1s by signal SIG1. If either ADRECD or BSYEN are false, the FCTN bits are cleared (so that ESEND will not be set in Phase 3).

This occurs at TS1. Note that LOCKO set to a 1 from now on prevents the programmed interactions that terminate in nodes L4 and L5. This additional synchronization is required to protect a 200 ns window against the program initialization of a control unit function at the same time as we have begun or are about to begin acceptance of a channel initiated function. Lock out will not appear in the remaining phases, but is should be remembered that as long as LOCKO is a 1, the first five programmable registers of the DX11-B cannot be changed by the program.

Setting SYNC to a 1 prevents either asynchronous operation from being enabled during TS2. This is needed so that a phase change does not occur and/or so that LOCKO will not be cleared even if DONE were just cleared or SYSRST just set.

If the DX11-B is in "single-thread" mode (BSYEN=1), then any pending status must be relieved by this ISS. If a pending status condition is recognized (SIG1B=1), BYPAS is set to bypass the SPW and DST fetches. The BALF flip-flop is used to remember the state of SIG1B for use in Phase 3.

Continuing down the middle of the page in TS2, both the channel-initiated sequence and the polling sequence merge briefly but set the CHIS flip-flop to the contents of Address-Out (1 or 0). As shown on Sheet 8 of Drawing 04, CHIS is set if Address-Out is a 1. This is considered a channel-initiated sequence. On the other hand, if

Address-Out were 0 and the DX11 were responding, then it would be a response to polling and that would be a control unit initiated sequence and one of bits 6 (ESEND), 5 (CHDEND), or 4 (CUDEND) of the DXDS would be set.

The DX11 also sets Operational-In and clears Request-In. At this time the 5 sec timeout (DXTO) is started since Operational-In sets at this time. If BSYEN is set, CUBSY is also set; and SYNC is cleared in preparation for a phase change (this happens on every phase change). If BYPAS were still 0, this would be the normal sequence of events and the contents of the Bus-Out register would be gated to become the new contents of the control unit address register (CUAR); and a change to Phase 1 would also be enabled. When the clock ticks at the end of TS2 these operations are performed and control is switched to Phase 1, page connector M on sheet three. On the other hand, if BYPAS were 1, then the CUAR address is placed on BUSI to identify the device to the channel as the DX11 begins either a response to poll or a pending status presentation and the DX11 enters Phase 2 on Sheet 3, page connector N. This would also be done if REQI=1 showing a request by the DX11. No NPR will be done for either SPW or DST.

If the DX11-B is busy, then the fast busy response is used unless this selection sequence is for the same device that is causing the busy condition and that device has status pending (FCTN=3). CUBSY can only be set in single-thread mode or immediately after System Reset.

In Phase 0 if a system reset from the CH is recognized (SYSRST=1) and if an ISS is not in progress (SYNC=0), then LOCKO (Phase 0) and CUBSY (synchronization operation, Sheet 1) are set and the results are stored in the tumble table for use by the emulator (Phase 4).

If DONE=0, this indicates an entry in the tumble table has been serviced by the program; if INTEN=1, an interrupt may have been made to the PDP-11. Either way, the program, by clearing DONE, has indicated that it has serviced the interrupt and is about to load the DX11 registers. Therefore, the DX11 must clear LOCKO. At least one instruction time must be allowed after clearing DONE before loading a programmed register. These conditions will be enabled repeatedly at every TP1 rather than only once.

NOTE

Programmed load (set/reset) of DONE is permitted only in Phases 0 and 7. Furthermore, only a diagnostic should attempt to set DONE.

Referring to programmed interactions at page connector L, in the idle phase a nonzero FCTN code and the GO bit cause Request-In to be set. If GO is a 1 and FCTN code is 0, then the DX11 is immediately reset.

b. Phase 1 – Address Response

Referring to Sheet 2 of Drawing 04, the DX11-B enters the Address Response phase only from Phase 0 when an ISS is in progress. As Phase 11 is entered (Phase 11–Phase 1, TS1), SYNC is a 0 as it always is on entrance to a phase, and therefore gates 2 and 3 are enabled. An automatic shift left and transfer of CUAR bits ⟨07:00⟩ are enabled to DXBA bits ⟨08:01⟩ placing the device address, recognized from the channel, in the DXBA as a word address. Note that DXBA00 gets a 0. Actually, this bit cannot be loaded since it is wired to ground. Also, bits ⟨15:10⟩ of the offset register (DXOR) are enabled to bits ⟨15:10⟩ of the DXBA. DXBA09 gets a 0. When loaded by the transition of time states, this places an offset base address for the recognized device into the DXBA. All of this is in preparation for fetching device status on the addressed device. In reality, this operation is to fetch the

pointer word for that status word (SPW). When the clock ticks, the following occur: an NPR request is made, and SYNC is set to a 1 (so that these operations will not be repeated while still in this phase).

NOTE

In the DX11-B, SYNC functions as a general synchronizer within phases as a flag to indicate that some function has been done and thereby prevents it from being done again while cycling between time states.

Note that during TS2 nothing will be enabled until the NPR has been finished (NPRX=0). Since SYNC was set to a 1 in TS1 by BAAR, the operations indicated (e.g., loading CUSR) in TS2 will then be performed on the next clock tick. This includes putting the address register (echo) into Bus-In (gate 4) in preparation for the address response to the ISS. Transfer is made at this time to Phase 2 (Status Preparation). Refer to the DXND format on Sheet 9 and note, as indicated on the flow diagram for Phase State 22, that if the right byte of the SPW is nonzero or if the left byte of the SPW (offset of DST for this device) is zero, BYPAS will be set to 1 to indicate that no NPR will be done for this DST. Note also that CUSR is cleared if the right-hand byte of the SPW is zero.

If any IBM reset has been recognized at this point, control will be sent to Phase 4 to mark the fact in the tumble table. Otherwise (IBMRST=0), the DX11 will continue to Phase 2, same sheet, page connector N, and the SPW fetch is done.

c. Phase 2 – Status Preparation (Part 1)

The DX11-B can enter Phase 2 from either Phase 0 or Phase 1. If Address-Out does not drop in Phase 2, the DX11 will wait, unless an interface disconnect is signaled by the fall of Select-Out or a reset signal is generated. The top of Phase 2 (node N) raises Address-In as the tag for data placed on BUSI in either Phase 0 or Phase 1. Then the DX11 waits for Command-Out (CMDO) to rise. Note that no operation is as yet enabled in TS2. When CMDO=1, the command at BUSO is copied into the command register (CUCR) and the parity OK signal is saved in BYPAS for use later in Phase 3.

If this is a polling sequence (or release of pending status) for which no DST fetch is needed, then BYPAS will be a 1 and the right-hand path will be taken. This path is also taken if the command has bad parity.

In TS2 of Phase 2, the decision is made as to what phase to transfer to next, depending on the function code specified when the sequence began (not necessarily when the program initially hit the GO bit). At this point, Phase 3, 5, or 6 can be entered.

NOTE

The function codes can be changed even after the GO bit is set but not after a sequence has begun. Thus it is possible for the program to clear the GO flag before a response is granted.

If a command status byte is needed from the device status table, then the left-hand path is taken (signal BANDSO). The address is composed in the DXBA (gates 5 and 6) and an NPR request is made (NPRX=1). SYNC is set to enable TS2 (above node H).

NOTE

BANDSO = DX*BA* gets DX*ND* · BU*SO*.

After the data is in DXND (NPRX=0), either the left or the right byte is loaded into the CUSR status bits. This result will be sent to the channel in Phase 3.

d. Phase 2, Part 2

Having entered Phase 2 at the preceding TP2, and at that time, in either Phase 0 or Phase 1, having loaded the address into Bus-In via the BUSI flip-flop, 200 ns later (one time cycle) the DX11 can be sure that the signals on the Bus-In lines will have settled down and the DX11 is ready to raise the tag line Address-In (ADRI). This is done immediately because the conditions ADRO=0, CMDO=0, SYNC=0 exist. Note that further activity in TS1 of Phase 2 is disabled, for the moment, because Command-Out is not a 1.

The next question is whether anything in TS2 is going to be active. Notice that NPRX is 0 now (it was 0 in order to have proceeded out of Phase 1). Also, SYNC is a 0, but Address-In is a 1 so that SIG4 is not true. Thus, no action is enabled yet in TS2. Address-Out=0 is part of the TS2 conditions so that only one phase change would be possible if an interface disconnect were encountered during Phase 2.

In TS1, the DX11 will again set a 1 to Address-In (this has no effect and is a correct procedure) and wait until Command-Out becomes a 1 indicating that the channel has loaded the command onto the Bus-Out (BUSO) lines. Now the DX11 reads Bus-Out into the Command register and responds by lowering the in-tag Address-In.

The parity okay signal is copied into the bypass flip-flop at this time. If command parity is bad, then the DX11 does not respond to the command. Instead, it ignores the command by taking the right-hand path. The DST is bypassed if Operational-Out has dropped because of a selective or system reset. The DST is also bypassed if BYPAS=1 because of conditions set in Phase 1. However, if the command is okay (PAROK=1) and bypass is 0, then the DX11 takes the left-hand branch and moves the NPR data into the left half of the DXBA; the right half of the DXBA is loaded with the command, itself, as taken from BUSO. Also, the DX11 puts a 1 to both NPRX and SYNC. At this point in TS2, with no further action transpiring, NPRX is not a 0, so nothing happens above node H. SYNC is a 1 so that nothing happens on the right side of TS2.

Back in TS1 again, the DX11 determines whether anything is enabled. Address-Out is a 0, Command-Out could be dropped to a 0 now that the tag-in low response has been given, but SYNC is a 1; now Address-In will not be raised and parity okay will not again be copied into bypass.

Nothing further happens in Phase 2 until the NPR response is indicated by NPRX going to a 0. Assume that it is strobed to a 0 at TS1. Therefore, TS2 is enabled. The DX11 will set SYNC to a 0 in preparation for a change to Phase 3, and the appropriate byte that was in core will be loaded into the control unit status register (CUSR). The low-order bit of the Command register is used to decide whether this was an odd or an even command. The DX11 now proceeds with the status presentation.

e. Phase 3 – Status Presentation

Referring to Sheet 4 of Drawing 04, Status Presentation, this phase is entered from either Phase 2 or 7. Upon entering Phase 3, Status-In (STAI) is 0. At this point, the logic must be able to make many determinations as illustrated by the complex logic expressions stated to the right side of the main path. These determinations are based on the following considerations:

1. Did the command have good parity (BYPAS=1)?
2. Was the SPW and/or the DST fetched from core without a nonexistent memory reference (NXM=0)?
3. Is this a test IO instruction (TESTIO=1)?
4. Is this a status pending sequence (BALF=1)?
5. Is this a sense command (SENSE=1)?
6. Is this an illegal command (UCHECK=1)?
7. Is this a channel initiated sequence (CHIS=1)?

If this is a control unit initiated sequence (CHIS=0) then the DX11 sends the contents of CUSR to the BUSI. Otherwise, if parity is okay, if there is no timeout error, if this is not a sense command, and if this is either a) not status pending or, b) a test IO instruction, then the DX11-B moves the status register to Bus-In using signal BISR.

If parity is okay, if status is pending (BALF=1), and if this is not a test IO (SIG5B), then the Busy bit is sent to BUSI04 and the rest of BUSI is loaded from the CUSR. Busy is also put into CUSR to ensure that the CUSR always reflects the status presented. If this is either bad parity or if the DX11 has failed to get proper time response either during the device status table fetch or during the SPW fetch (causes PDP-11 timeout to be a 1) and if this is not a status pending situation, then only the unit check bit is sent to BUSI and CUSR as the status response. If a hardware error prevented command interpretation, but a pending status exists for this device, then that status is presented.

In TS2, Command-Out could still be a 1; since Status-In is 0, this would prevent further operation. Assuming that Command-Out becomes a 0 or has become 0 by this time, then, with Service-Out a 0, the DX11 is able to set Status-In to a 1. If an interface disconnect, system reset, or selective reset were present here, then Operational-Out would be a 0 and this sequence would quickly conclude. The DX11 now waits for either a Command-Out or a Service-Out response with Status-In a 1. Notice that as long as Status-In is a 1 no activity is enabled during TS1.

When a response is given to the status presented, the DX11 lowers the Status-In tag at TS2, makes a phase change to Phase 4, and recognizes either Stack-Status or Status-Accepted. If status is stacked, the Stack-Status bit becomes a 1. (Note that this will not be done by the channel if a 0 status is given to other than a test IO instruction.)

If status is accepted, stack status is reset to 0. If Suppress-Out is a 1 at this time, the DX11 recognizes Command-Chaining. If the function code is three then ESEND is set because this is either a presentation of the previously stacked status or is, in fact, an ending sequence presentation.

Finally, if Control Unit End status is accepted, the ISSREJ flip-flop must be cleared, since a CUEND is no longer outstanding.

f. Phase 4 – Mark

Phase 4, the Mark phase, is also located on Sheet 4 of Drawing 04. This phase can be entered from any other phase since this phase is used to enter data into the tumble table. On entering the Mark phase, SYNC is 0 and NPRX is 0 unless there is an NPR already in progress (and interrupted by a reset condition). A number of operations are done in TS1 with the objective of making the first entry into the tumble table. The tumble table address is composed in the DXBA by combining the contents of the DXOR high-order byte, the tumble table index byte times two, and by putting 0 in the lower bit of the DXBA. For the data that will be stored, the contents of the DXDS (the device status) register are placed in the DXND.

At the clock tick, NPRX is set to 1 and the data transfer proceeds. At this time, SYNC is also set so that none of the TS1 operations can be enabled again, and the DX11 proceeds to count the tumble table index by +1. In TS2 nothing happens until the first NPR has completed, whereupon the Command and Address (DXCA) is put into the NPR data register (DXND). The DX11 again copies the tumble table address into the DXBA and loads one to NPRX to request the second NPR store cycle for the tumble table data. At this time, the DX11 clears SYNC, increments the tumble table index, and switches to Phase 7 with the NPR in progress.

When the DXDS is copied for the tumble table store operation, a special flip-flop, IRS, must be set to remember if an IBM reset condition was stored in the tumble table. IRS is then used in Phase 7 (where DXDS is cleared) to determine if a reset sequence occurred between Phases 4 and 7.

g. Phase 5 – Input (360 Write)

Referring to Phase 5, Input, located on the same sheet of Drawing 04, the DX11 enters at node Y. The transition to Phase 5 from either Phase 2 or 7 is initialized by setting BALF to 0, BYPAS to 1, and CUDX to 1. With CUDX=1, data okay a 1, Service-In a 0, and IOD a 0, the resultant SIG9 is generated, which raises Service-In delayed by BLLM.

Later the DX11 will zero the BYPAS flip-flop. The use of the BYPAS flip-flop here is simply a reflection of its appearance in the data okay expression to check for first data byte. Note that nothing will be active under node Z because SYNC is not a 1. The entire Phase 5 is similar to Phase 6. Note the symmetry in the two flow diagrams.

The next actions in Phase 5 occur in TS2. If the channel responds with Service-Out to Service-In, the DX11 lowers Service-In, at the same time copying the content of BUSO into either CUCR or CUSR. The destination is determined by the present value of the BALF flip-flop. If BALF=1, then the CUCR is filled and CUDX is reset to 0, indicating that path Y activity is no longer in progress.

NOTE

If Command-Out is raised in response to Service-In (IO Stop) then BUSO will contain zeros. Thus if only the low-order byte contains data, the high-order byte will contain zero. Similarly, if CUDEND is to be set, and BALF is 0, then CUCR is cleared so that the last byte entered into PDP-11 core will be zero.

In addition, for each cycle through this sequence, i..e, for each BUSO copy, 1 is added to the byte count. Data transfer will cease when, 1) the DXBC is minus one, 2) a timeout error occurs, or 3) a parity stop is indicated. The latter is a special feature that allows the option of arresting the data sequence when a parity error occurs on Bus-Out. In byte-multiplex mode this permits deletion of the bad byte and resumption of the operation.

There is still no activity below node Z, Phase 5, TS1, because SYNC is still not a 1; but in TS2, under node Z, activity is now possible because the three lower conditions are now present (SYNC=0, NPRX=0, and CUDX=0). The DX11 now takes the left-hand branch and copies CUCR and CUSR into DXND in preparation for transfer of the two bytes of data into PDP-11 core. The transfer is started by NPRX=1 and the direction is controlled by NPRT=1. CUDX is also set to a 1 to indicate that activity is again valid in regard to CU data.

Finally, SYNC is set to 1, enabling the operations under node Z, TS1. This happens when NPRX is 0. The operations are clear SYNC and increment DXBA by two.

NOTE

Incrementation is possible at this point because it follows the completion of the NPR cycle and another NPR cycle will not be made for a period of time, allowing plenty of time for the DXBA to carry, even into its extended address bits.

When IOD is finally set to 1 and activity is no longer possible on the CU data, and when the NPR transfer is complete, the top three conditions exist under column Z, Phase 5, TS2. These are IOD=1, NPRX=0, and CUDX=1. The DX11 now follows the right-hand path to clear SYNC and switch to Phase 4, marking the termination of this particular transfer.

h. Phase 6 – Output (360 Read)

Referring to Sheet 5 of Drawing 04, the Output phase, the DX11 will have arrived here from either Phase 7 (done) or Phase 2 (status presentation). Upon entering this phase at node A, SYNC is already 0; in TS1, this will now enable the setting of SYNC; and with IOD being a zero, SYNC=0 will also enable the setting of NPRX. This will not occur until the clock tick to TS2; but, when the tick occurs, SYNC=0 will first allow NPRX to be set and will then set itself. The transition to TS2 will perform the first fetch of data from the PDP-11 core, assuming that the program has correctly initialized the DX11 Bus Address Register (DXBA) and the Byte Count (DXBC).

When the NPR is completed, NPRX is reset to 0 and the conditions to enable operations in TS2 are met. SYNC is reset to 0. Assuming that IOD is still a 0, the DX11 increments the DXBA by two. Simultaneously, the DX11 copies the data from the DXND into the CU data registers (CUCR and CUSR are used for this purpose) and sends a 1 to CUDX indicating that the IBM buffer is ready.

At the next tick, the DX11 reverts to TS1, but this time both page connectors, A and B, are activated. At this point, SYNC is zero once again, so it and NPRX will set to 1 on the next transition of states. The NPR direction control (NPRT) was already reset to 0 when this phase was entered; now, both NPRX and CUDX are in the 1 state or active.

Inspecting the logic path begun at connector node B, CUDX is a 1, the data okay signal is a 1 (see phase independent sequences on Sheet 1), Service-In is assumed to be 0, and IOD is also assumed to be 0. If BALF is 0, then the DX11 will load CUSR into the BUSI flip-flops. From there the 8 bits of data plus parity will be presented to the Bus-In lines. At this time, the DX11 sets the Bus Load Limiting flip-flop (BLLF) which, at the next TP2, will set Service-In (SRVI) to a 1. This setting is delayed by the BLLF to give a one time pulse difference between loading the BUSI and setting the appropriate in-tag, thereby ensuring that the parity line (PARI) will have time to settle (its data is the output of the BUSI flip-flops).

Repetitive SRVI sequences are delayed by BLLM and BLLF to limit the rate of data transfer to the channel and DX11. The rate is set to 4 μs when shipped. This is an installation parameter dependent on system configuration.

NOTE

It is possible for both logic paths A and B to have TS1 or TS2 elements enabled simultaneously. This is to allow simultaneous operation of both the I/O interface bus and the PDP-11 Unibus synchronized to the DX11 clock. At TS2, the DX11 will also set BYPAS to a 0 to mark the end of the first data transfer.

When Command-Out is raised in response to Service-In being asserted, the DX11 will respond by lowering Service-In and taking the left-hand branch of Phase 6, node B, TS2. At this point, the DX11 recognizes that the channel has indicated data transmission end. This will set a bit in the device status (CHDEND, which also causes IOD to assert) at TS2. In addition, this clears the CUDX flip-flop so that no further activity is enabled under path B. If, however, Service-Out is raised, indicating acceptance of the data, the DX11 takes the right-hand branch, complements the BALF flip-flop, and increments the byte count (DXBC). If the byte count is minus 1 at this time (i.e., about to go to 0), then the DX11 sets control unit data end (CUDEND), which also asserts the IOD line and clears the CUDX flip-flop. The same task would be done if PDPTO were set, indicating an NPR failure.

From node A in TS2, assuming that CUDX is 0 (or was made 0 at prior TS2), the same cycle as described above would be repeated. If, however, both IOD and Interface Disconnect = 1, CHDEND will be set to indicate that the channel terminated the data transfer by means of an interface disconnect. In any case, if IOD is a 1, the DX11 will simply switch to Phase 4 (Mark) to record the data end indication bits into the tumble table and then proceed to Phase 7 (Done).

i. Phase 7, Done

Referring to Sheet 6 of Drawing 04, to enable any condition in the done phase, node J, the last NPR should be completed (NPRX=0). SYNC will be 0 but the DX11 must also wait for either Select-Out or Stack-Status to become 0. When this happens, the actions listed to the right of the line are enabled to occur on the next clock tick. SYNC will be set to a 1, the FCTN and GO bits will be cleared, the device status register (DXDS) will be cleared (Note, except bits 3, 11, 12, and 13), and both the BALF flip-flop and the NPR transfer flip-flop (NPRT) will be reset.

Now, control will go in one of two paths. Notice that, with the exception of the top three quantities in the right-hand gate, the conditions for right- and left-hand directions are complements. An SEL channel will take the left path (Forced Burst mode) while an MPX channel will take the right path (Byte Multiplex mode).

When operating with an SEL channel, Select-Out is up, the DX11 takes the left-hand path and drops into Phase State 72. If an IBM reset is issued at this time (IBMRST=1), SYNC=1 and OPLI is still 1, the DX11 will take page connector T to reset SYNC with the signal BLAST, enter Phase 4 to mark the fact, come back to Phase 7 and take the right-hand path to get off the channel.

If no reset is received, however, Operational-In is still up, and the DX11-B will just set SYNC on the time state transition. Lockout was set in Phase 0; the only condition remaining is the requirement that the program clear DONE.

When the program clears DONE, Lockout is disenabled and the DX11 drops into the next gating condition (provided an IBMRST is not issued, see above). To enable SIG17, the only requirement needed now is for the program to set GO, since OPLI and SYNC are both set. The program will set GO simultaneously with setting the function.

The function code that is set at this time depends on the intention of the program. If FCTN is set to 0, the DX11 will enter Phase 0 at node L and perform a DX11 RESET operation (resets the world).

If this is an input operation, FCTN is set to 1 and the DX11 enters Phase 5 at node Y. It will also set Lockout, BYPAS, and CUDX; the ODD flip-flop will be copied into the BALF flip-flop. Upon returning from Phase 5 (after having gone to Phase 4 to store the device status, command, and address in the tumble table, and setting DONE), the DX11 takes the normal path and does all the things it did during the last pass. When it drops to TS2, this time it sets ending status by setting FCTN to 3 and setting GO. This time the DX11 exits at page connector H, goes to Phase 3, then to Phase 4, and back to Phase 7.

If this is an output operation, the program would set FCTN to 2, go to Phase 6, then to Phase 4, then to Phase 7, and back to Phases 3, 4, and 7.

Once data and/or status has been presented, the channel drops Select-Out. This enables the DX11 to take the right-hand path in TS1 of Phase 7. SELO=0 yields the signal IRCCLR. This drops Operational-In on the next clock tick; with SYNC=1, SIG16 clears everything listed to the right of the line. This restores the flow to the idle state, Phase 0. If status has not been presented (STKSTA=1) and BSYEN=1, FCTN is automatically set to 3 and Request-In is raised to try a CUI.

j. IBM Reset Stored Sequence

The purpose of the IRS flip-flop, located in the DXDS Register, is to remember the fact that IBMRST has been issued to store that fact in the tumble table by a forced additional tumble table entry.

Referring to the flow diagram for Phase 4, Sheet 4 of Drawing 04, SIG7 is generated, which on the clock tick to TS2 sets the NPRX flip-flop and transfers the contents of DXDS to the DXND. Once transferred, that data is ready to be sent to the tumble table in the next microsecond or two, when the present NPR cycle is through. At that instant, before the second NPR is done (DXCA store), if IBMRST is issued, IRS is set and the condition that caused the IBMRST is stored in the DXDS. These can be a System Reset (SYSRST), a Selective Reset (SELRST) or an Interface Disconnect (INFDSC) (see "Special Signals" on Sheet 1 of Drawing 04).

The DX11 then drops through TS2 of Phase 4 and enters Phase 7, where the DXDS is cleared except for certain bits containing the reason for the IBMRST. The DX11 then attempts to take the right-hand path and although IRS=1, Operational-Out is now a 0 (one of the results of IBMRST). This forces the DX11 to take the left-hand path (not zero the reset conditions), and to fall out at page connector T, where it finds the conditions right to generate the signal BLAST. This zeros SYNC and causes the DX11 to return to Phase 4 where it can do another tumble table entry.

The uncleared reset conditions in DXDS are now transferred to the DXND and stored in the tumble table. The DX11 goes through the second phase of storing the address (this is meaningless at this point) and returns once again to Phase 7. This time the right-hand path conditions *are* met (IRS=1, SELO=0) enabling IRCCLR (IBM Reset Conditions Clear). At the tick to TS2, SYSRST, SELRST, and OPLI are all cleared, all other Phase 7 functions are performed, and the DX11 returns to Phase 0.

Note that since on an IBMRST OPLI→0, the setting of IRS (an ORed condition) provides an additional way to enable IRCCLR and a return to Phase 0.

k. Interface Disconnect Sequence

Interface Disconnect (INFDSC) is a special situation indicated by the combination of interfacing signals comprising Select-Out=0, Operational-In=1, and Address-Out=1 (see Synchronization Operation on Sheet 1 of Drawing 04). When the DX11-B has Operational-In raised, if the channel raises Address-Out, puts the DX11 address on the bus and then drops Select-Out, it is informing the DX11-B to go away. In IBM parlance this is called an Interface Disconnect; IBM requires that the DX11 drop Operational-In within 6 μs after receiving an INFDSC. Since the DX11, upon receipt of a disconnect, could be involved in internal bookkeeping and in transactions with the PDP-11 that could take longer than 6 μs, it provides hardware to satisfy the IBM requirement while allowing its own operations to proceed. Further, since once the DX11 has responded to the disconnect a subsequent selection from the channel could occur before processing of the disconnect was completed, this circuitry permits proper interchange with the channel to occur without hindering the successful completion of its internal operations.

Referring to the note on the flow diagram on Sheet 7 of Drawing 04, this sequence can be enabled in either Phase 4 or Phase 7, immediately following receipt of INFDSC and functions simultaneously with these phases. In actuality, it uses the hardware designated for Select-Out propagation and for the Fast-CU-Busy sequence of Phase 0.

If an INFDSC is received during some data phase, the DX11 will immediately terminate that phase and go to Phase 4. It will perform all normal functions in this phase, including the setting of DXDS bit 11 (INFDSC). When this bit sets it causes the Disconnect Response flip-flop to set which, 200 ns later, causes OPLI, ADRI, and SRVI to drop, and enables the operation of the Disconnect Response flow. Meanwhile, the DX11 continues to complete the tumble table entry and then goes to Phase 7, as it would if a disconnect had *not* occurred.

In this case, as the DX11 enters Phase 7, DSCRSP=1, which enables SIG14. It then takes the right-hand path finding both IRS and INFDSC set, sufficient conditions for IRCCLR. This results in the dropping of SYSRST, SELRST, and OPLI, the latter being the IBM requirement to release the channel. The DX11 then continues

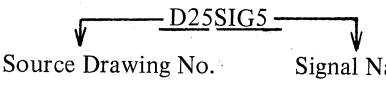
through Phase State 72, clearing all conditions to the right of the line. Since this could take considerable time to complete as the DX11 does its NPR cycles (as much as 10 to 15 μs each), and since the channel could become selective again in that time, the logic described in the Disconnect Response flow diagram is provided to take care of that eventuality. Remember, events in this flow occur simultaneously with events in Phases 4 or 7.

If the channel becomes operational while the DX11-B is still processing the INFDSC; then as the DX11 enters the response sequence OPLO will be a 1, SELO=1, and STAI=0. At this point in the flow the determination is made as to whether or not the channel's reselect is addressed to the DX11 or to another CU on the bus. If the select is for another CU, ADRECC=0 (no compare) and the left-hand path will be taken. In this instance, SELI will be set and the Select signal is propagated down the line.

If, however, the Select is for the DX11-B, ADRECC=1 and the right-hand flow is taken. In this event, the DX11 is still busy and cannot accept the selection so the FAST CU response is given. The DX11 rejects the ISS (ISSREJ=1) and records the fact in the tumble table. BYPAS is set to enable Bus-In. Bus-In is cleared but bits 04 (BUSY), 05 (CHEND), and 06 (STAMOD) are force-set. When the clock ticks to TS2, Status-In is raised to put the status on the bus. When the channel responds with a Select-Out drop, Status-In is dropped and BYPAS is cleared. The DX11 then proceeds with the FASTX logic described at that point in Phase 0. Meanwhile, the DX11 has processed through Phase 7 and terminates its flow in Phase 0.

5.5 CIRCUIT DESCRIPTIONS

As mentioned at the beginning of the last paragraph, the flow diagrams are such that a gate-chasing discussion is not required. In this paragraph, therefore, only certain subtle circuitry characteristics are discussed. In these block schematics, each signal is preceded by a source code to identify the drawing number on which that signal is generated. For example:



means SIG5, generated on Drawing DX11-B-25. In most cases, the location designation of a flip-flop bears a numerical relationship to its bit position in the register format. For example, on Drawing 05 (DXDS), the M205 at the top left-hand side of the page is located at C25. Its position in the DXDS format is bit 15. In most cases, if 10 is subtracted from the hardware locations, the register bit position can be determined. Another example is the SYSRST flip-flop directly below, located at C22 and situated in the register at bit position 12 (22-10=12). There are exceptions to this convention, e.g., the IRS flip-flop shown as part of the DXDS at location C24 is not a functional part of that register. Bit 14 of the DXDS is NXM, which is located in the NPR and INT Control. IRS is displayed as bit 00 of the DXES2 Register.

In this paragraph, the circuits are described in drawing number order, excluding the functional flows.

5.5.1 DX11 Device Status Register (DXDS)

The DXDS is shown on Drawing No. D-BS-DX11-B-05, Sheets 1 and 2. This register contains the bits stored in the DXDS, except the M205 at C24. This is IRS which is not part of this register but is part of the DXES1 register bit 00. Bit 14 in the DXDS is NXM set by PDPTO which comes from a timer in the NPR control. This register is the first of two tumble table entries described in the sequences (Status, and Command & Address).

A typical setting arrangement can be described for the register by using PARER as an example. Enabling conditions are AND/NORed to the data input of the flip-flop and this result is ANDed with CLKN to the clock input. CLKN2 ticks at pin B1 of the M112 every 200 ns, but nothing happens until the data gate is activated, e.g., the top condition is SIG5ΛBYPAS (0). As shown on Drawing 25, SIG5 implies Phase 3, TS1. If BYPAS=0, that gate will be enabled on the phase change and will go low at pin A1 of the M112 and at pin K1 of the flip-flop. The flip-flop will set on the leading edge of CLKN2.

Setting of all other flip-flops in this register is similar with the gating conditions being more or less complex. Note that SYSRST can only be direct-set, however, as with ISSREJ. The latter cannot be direct-cleared, but is reset by clocking against a grounded data input.

Note that all flip-flops in this register are "set-only". They cannot be cleared from the gates. With the exception of INFDSC (bit 11), SYSRST (bit 12), SELRST (bit 13), and ISSREJ (bit 03), they are all direct-cleared by CLR DXDS or CLR DXDSA. This signal is the product of four OR conditions and CLKN. The M606 is a PA that, when enabled, produces a low-going edge (one pulse 30 to 100 ns wide).

5.5.2 Control Unit Status Register (DXCS)

The DXCS is shown on Drawing No. D-BS-DX11-B-06, Sheets 1 and 2. In this register, all flip-flops are set or reset in fundamentally the same way, just the gating complexity varies. Most are direct-cleared by CLR DXCS, either alone or in conditioned combination.

The exceptions are CUBSY, which has no direct-clear, and the two function flip-flops which are cleared by CLR FCTN, but here this signal is a function of CLR DXCS when combined with CLKP2ΛSIG14; the latter generated in Phase 7, TS1. CLR DXCS is generated by CLKN2 for either a DXRESET or an INIT. This is seen on Sheet 1 at coordinates C2.

All flip-flops are program-loadable since data inputs to all flip-flops contain, as their inputs, the appropriate DBUS bit for that flip-flop, either directly or once again in combination with appropriate gating conditions.

All flip-flops are clocked by some combination of positive or negative clock depending upon gate polarity requirements. Clocking is usually enabled by DXCS CLKEN (the ANDing of ADRS04, DATO EN HIGH, and LOCK (0)). When clocked, all flip-flops will set on the presence of data and reset on the absence of data.

Most flip-flops cannot be loaded in Maintenance Clock mode; note that the FCTN and GO flip-flops *can* be loaded since their clock pulse originates at the free-running source in the clock generator. The FCTN flip-flops use PCLK while the GO flip-flop uses NCLK. All other flip-flops use the gated versions CLKN or CLKP.

5.5.3 DX11 Bus Address Register (DXBA)

The DXBA is shown on Drawing No. D-BS-DX11-B-07. This is the register in which the tumble table address is composed for each NPR.

The register comprises a loadable 16-stage M238 Counter that keeps track of the next tumble table entry address, and an M169 Multiplexer to gate the various source addresses into the register. Clocking is accomplished by an M606 that is fed by the OR of all input gating conditions. All flip-flops are cleared by DXRESET.

Pin S1, the 00 input, on the M238 is grounded so that on each and every register load (LOAD DXBA) the register is counted by +1 as triggered by CLKP1 (COUNT BA). The register will be counted by +2 during Phase 5, TS1, when SYNC=1 and NPRX=0. This incrementation is done following the NPR cycle, at a time when another NPR cycle will not occur for a period of time long enough to permit carrying to occur (DXBA CRY). CRSRND is used to increment the address during an output transmission.

When ADRS10 HΛOUT HI HΛLOCKO (0) H are all true, BADB1 and 2 are generated on Drawing 25. BADB1 enables the contents of DBUS ⟨15:01⟩ to the register, while BADB2 generates LOAD DXBA.

In Phase 1, TS1, BAAR loads the shifted contents of the CUAR into the register in preparation for a fetch of the SPW. At the same time, BAOR enables the contents of the CUOR into the register as an offset.

In Phase 2, TS1, after the SPW has been fetched, BANDSO reads the pointer part of the SPW from the DXND to the left half of this register, and copies the command from the BUSO lines into the right half of this register.

During Phase 4, BANX is used to gate the shifted contents of the TTNDX into this register to advance the TT address.

5.5.4 DX11 Byte Counter (DXBC)

The DXBC is shown on Drawing No. D-BS-DX11-B-08. The program loads the negative byte count (2's complement) into this register (LOAD DXBC) and the register counts on the AND of CLKP1 and either SIG10 (during input operations) or SIG13 (during output operations).

When the count is about to be exhausted (about to go to 0), the signal BCMO (Byte Count Minus One), is generated, which ANDs with either SIG10 or SIG13 on Drawing 05 to set CUDEND. This, in turn, asserts IOD and clears CUDX on Drawing 11 indicating DONE.

A DX RESET generates CLR DXBC to reset the register.

5.5.5 Control Unit Status Register (CUSR) Part of DXOS

The CUSR is shown on Drawing No. D-BS-DX11-B-09, Sheets 1 and 2. The bits stored in this register are shown in Figure 4-6. These are the Control Unit Status bits loaded either from PDP-11 core during Phase 2, TS2 (SRDB), or from multiplexed sources as described below.

When ADRS06 HΛOUT LOW HΛLOCKO 0 H are true, SRDB (CU*SR*←*D*ata *B*us) is generated allowing the Data Bus (DBUS ⟨07:00⟩) to enable the flip-flops in the register. This loads the register with the status bits from core.

In Phase 1, TS2, SRNDEN (CU*SR*←DX*ND En*able) is generated when the NPR has been granted to enable the status (DXND ⟨07:00⟩) into this register.

In Phase 2, TS2, one of two signals will be generated on Drawing 25, depending on whether the command stored in the CUCR is odd or even. If CUCR00 is a 1, SRNDH will be generated and enable DXND ⟨15:08⟩ into this register. If CUCR00 is a 0, the low byte (DXND ⟨07:00⟩) will be enabled to this register.

NOTE
IBM commands are odd for "out" transfers (IBM Read) and even for "in" transfers (IBM Write).

In Phase 5, TS2, SRSO is generated to gate the second data byte from the Bus-Out into this register.

In Phase 6, TS2, CRSRND is generated to enable the contents of the DXND low byte to this register during PDP-11 output operation.

Clocking for all flip-flops is essentially the same. The signal STB CUSR is exactly the same as STBA CUSR; both are based on CLKN2ΛCUSR CLK ENA, the latter being the OR of all multiplexer gate enables.

The register is cleared by CLR CUSR, the AND of DX RESET and CLKN2.

Note that on Drawing 24, when any of these bits are set during a channel initiated sequence (CHIS (1)) or when a status operation is underway (FUNEQ3), the signal CUSRNZ is generated to indicate that the CUSR is nonzero.

The CUOR, also a part of this DXOS register, is represented by two M238 Modules on Sheet 2. Note that this register is writable only when in off-line mode. Bits 09 and 08 are not used in this register.

5.5.6 NPR Data Register (DXND)

The DXND is shown on Drawing No. D-BS-DX11-B-10, Sheets 1 and 2. As shown in Figure 4-12, this register is used to hold the contents of two other registers and the contents of the data bus for sequential NPRs.

NOTE
The bit position/module location convention does not apply to this register.

The data inputs of all flip-flops are fed by M169s; they are four-way multiplexers. On Sheet 1, DXND ⟨15:08⟩, only three of the four inputs are used. On Sheet 2, DXND ⟨07:00⟩, all four inputs are used. The enables for each set of inputs are generated on Drawings 24 and 25 to determine which input set is gated into the register.

The enable NDDB (DX*ND*←*D*ata *B*us) is generated on Drawing 24 as the AND of NPRX (1) (request has been made) and NPRT (0) (transfer is *from* PDP-11). When raised, it enables the contents of the Data Bus into this register.

The enable SIG7 is generated on Drawing 25 during Phase 4, TS1, when both NPRX and SYNC = 0. When raised, it enables the contents of the DXDS into this register.

The enable SIG8 OR NDCRSR (DX*ND*←CU*CR*·CU*SR*) is also generated on Drawing 25 during Phase 5, TS2, and Phase 4, TS2. When raised, NDCRSR enables the contents of CUCR ⟨07:00⟩ into DXND ⟨15:08⟩. During Phase 5, TS2, it enables the contents of CUSR into DXND ⟨07:00⟩, and during Phase 4, TS2, SIG8 enables the contents of CUAR ⟨07:00⟩ into DXND ⟨07:00⟩.

The clock for this register is made up of the OR of all data enabling conditions and CLKP1, in turn ORed as DXND CLK or DXND STB. The left-hand byte is additionally enabled for clocking by INIT.

Clearing is generated on Drawing 21 as the AND of DX RESET OR INIT∧CLKN2. Note that four DX11 clear pulses are generated and fed to sets of six flip-flops each. This is accomplished by internal wiring on the M216, which contains six flip-flops per module. All flip-flops on the same module are cleared simultaneously (see footnote on Drawing 21).

5.5.7 DX11 Control Bits Register (DXCB)

The DXCB is shown on Drawing No. D-BS-DX11-B-11, Sheets 1 through 4. This register holds some of the DX11 control bits shown in Figure 4-11. Exceptions are bits 11, 10, 01, and 00, generated elsewhere, and signals generated here but displayed elsewhere. These are DSCRSP (bit 2 of DXES1) and ODD (bit 6 of DXES). NPTDONE on Sheet 2 is not displayed at all.

On Sheet 1, LOCKO (bit 15) is clocked by CLKN1 when enabled for clocking by either DONE (1)∧SYNC (1) in Phase 0, TS2; or OPLI(1)∧SYNC(1)∧DONE(0) in Phase 7, TS2. It will be enabled for setting on either SIG17, when the function is set to anything but 00, or on an IBMRST or an error status condition.

The phase control flip-flops 0, 1, and 2 represent a binary arrangement, the combined states of which determine which of eight possible states the DX11-B is in.

Clocking for these three flip-flops is the OR of all possible enabling conditions. The SET PHx signals are wired to provide the proper coded combination for each phase. CUI5 yields a 1-0-1 combination or Phase 5. CUI6, in like fashion, generates a Phase 6 condition, while SIG8 results in a Phase 7 setting. All three flip-flops, as with most in this register, are direct-cleared by CLR DXCB generated on Sheet 4. Exceptions are DSCRSP, ONLINB, and ODD, which appear on Sheet 4.

Notice that the data enable/clocking scheme in these flip-flops is similar to most other flip-flops in the DX11, in which the data input to the flip-flop is used as one possible condition for clocking such that if the data conditions are not present, the flip-flop, when clocked, will reset.

On Sheet 4, ONLINB (a function of ONLINA) is fed to M611 inverters that function as master gates to run all receivers and drivers in the IBM interface.

The various conditions for the rest of the flip-flops in this register are self-evident and will not be detailed here.

5.5.8 Control Lines Out Register (CONO) Part of DXMO

The CONO byte of the DXMO is shown on Drawing No. D-BS-DX11-B-12. It stores the states of the IBM Selection Control lines and Tag lines as received at the interface on Drawing 27, and displayed as shown in Figure 4-9. PARO is also displayed but is not stored in this register. Note that although HLDO is displayed, SELO is made a condition of HLDO by ANDing on its data input.

This register is writable only when either off-line or on-line and cabled. It is used by the simulator program only.

All flip-flops are clocked by CONO CLK, a function of CLKN, and all are direct-cleared by DX RESET OR INIT as generated on Drawing 24.

The low-order byte of this register (DXMO), as displayed, is not stored in flip-flops but rather displays either the actual Bus-Out signals (BUSO ⟨07:00⟩ or the buffered versions (BUSOB ⟨00:07⟩).

5.5.9 Control Lines In Register (CONI) Part of DXMI

The CONI byte of the DXMI is shown on Drawing No. D-BS-DX11-B-13, Sheet 1. This register is used for maintenance purposes to read the Bus-In tags originating in the DX11-B logic. As such, it gives a representative view of the bus as seen by the channel. The register is usually read-only but, in Maintenance mode, can be modified by the program. When off-line, the PDP-11 program reads this register (if ENABLE=1); when on-line, it reads the test-in plug. The output of these flip-flops is enabled to the IBM Bus-In interface on Drawing 27.

The byte is loaded from DBUS ⟨15:10⟩ with LOAD BUS DXMI, which is generated by the AND of ADS16ΛDATO EN HIGHΛ-ONLINE. There are other loading combinations that can be read from the print while referring to the flows.

Clocking is accomplished by CLKN3 in various combinations as shown. In all cases, one of the conditions for clocking is the presence or absence of data.

All flip-flops except SELI are direct-cleared by CLR DXMI, the AND of CLKN3 with the OR of INIT, ON, or DX RESET.

Also displayed in this register, but not stored in it, are CLKO (bit 09) which comes from the cables even when off-line, and PARI (bit 08) generated on Drawing 26 as a comparison of the bits described in Paragraph 5.5.10.

5.5.10 IBM Bus-In Lines (BUSI) Part of DXMI

The BUSI byte of the DXMI is shown on Drawing No. D-BS-DX11-B-13, Sheet 2. This register contains the data that is enabled to Bus-In for transmission back to the channel. The output of this byte is sent to the parity generator on Drawing 26 to produce PARI.

The flip-flops are all identical with multiplexed data inputs. Clocking is achieved by CLKN3 ANDed with either the output of the multiplexer or with BUDATA, which is made up in the center of the page of either all multiplexer gate conditions or the OR of ON, SIG5A, or SIG5D.

In Phase 0, TS1, FASTCU is generated on Drawing 24. At this time a 1 will be sent to BUSI06, ENDEN (STAMOD) will be set in BUSI05, and a 1 will be put in BUSI04 (CUEND); all other bits will be reset to 0. This is the first portion of a disconnect response.

During Phase 1, TS2, BIAR performs a ones transfer of CUAR to this register as the address echo for transmission to the Bus-In lines.

In Phase 6, TS1, if BALF is a 0, SISREN will be generated to enable the transfer of the contents of the CUSR (first data byte) into this register for presentation to the Bus-In lines. If BALF is a 1, SICR will be generated to transfer the contents of the CUCR (second data byte) into this register and then to the Bus-In lines.

5.5.11 Tumble Table Index (TTNDX)

The TTNDX is shown on Drawing No. D-BS-DX11-B-14. It holds the low-order address of the next tumble table entry. It is shifted left before being copied into the DXBA.

M238s are used to form this incrementing counter. It is incremented by COUNT TTNDX, the AND of BANX and CLKP1. The register can be cleared by either ON or the signal DXRESET OR INIT, which forms the signal TTNDX CLR.

5.5.12 Control Unit Address Register (CUAR), Part of DXCA

The CUAR byte of the DXCA is shown on Drawing No. D-BS-DX11-B-15. The register holds either the contents of DBUS ⟨07:00⟩ or BUSO ⟨07:00⟩.

The data inputs of each flip-flop are multiplexed by M121s so that in Phase 0, TS2, on an ISS when BYPAS=0, ARSO is generated to copy the address on Bus-Out into this register. When DATO EN LOW∧LOCKO (0)∧ADRS02 are true, CUAR DATA LOAD transfers the DBUS contents to this register.

Clocking is the OR of both data conditions, ANDed with CLKN1. The flip-flops are direct-cleared by CLR CUAR, the AND of CLKN1 and DX RESET.

5.5.13 Control Unit Command Register (CUCR), Part of DXCA

The CUCR byte of the DXCA is shown on Drawing No. D-BS-DX11-B-16. This register holds the last command byte as transmitted from the channel during an ISS, even if it was rejected by the DX11-B. It also holds the second data byte received from the DBUS, which is stored as the second tumble table entry for a specific device upon completion of the CH/CU transaction. This register is read-only; it cannot be loaded by program.

The data inputs of each flip-flop are multiplexed by M121s. In Phase 2, TS1, CRSO1 is generated to transfer the command on Bus-Out to this register and simultaneously drop Address-In.

In Phase 5, TS2, if BALF is a 1, CRSO2 is generated to transfer BUSO into the odd byte of this register.

In Phase 6, TS2, CRSRND is generated to transfer the high byte of DXND into this register during a PDP-11 output operation.

As in other DX11 registers, clocking is a function of data gate enables ANDed with CLKN1. Direct-clearing occurs on DX RESET∧CLKN1.

5.5.14 Unibus Interface

The Unibus interface is shown on Drawing No. D-BS-DX11-B-17, Sheets 1 and 2. Referring to Sheet 1, there are some differences in this interface from the standard configuration, particularly in the way the M105 is used. In this arrangement, Slave-Sync is not taken from this chip, and the Select 0, 2, 4, and 6 outputs are also ignored.

NOTE
These outputs *are* here and can be used for maintenance purposes, if desired, but the pins do not appear on the wire list.

In the DX11-B, test point L2 is used to yield ADRS SELD L, which indicates that Master-Sync and all the bits are compared with the jumpered combination. Signals OUT LOW H, OUT HIGH H, and IN H are used for their normal functions. Note that BUS A ⟨04:01⟩ are not applied to the M105, but are received directly from the bus and buffered by the receivers on Drawing 17.

This device recognizes blocks of 16 addresses. When jumpered, 3, 4, and 5 must match all the time. The module is jumpered for the address 7762xx or 00–37.

The circuit is synchronized with high clock, i.e., the M205 is clocked with negative clock using the trailing edge or the rise of NCLK1 to high.

Operation can be traced by referring to the timing diagram provided at the bottom of the page. As soon as an address match is seen, ADRS SELD L is asserted, putting a low on the SEL SYNC flip-flop data input, and, by inversion, removing direct-clears on both flip-flops. At the next clock pulse, SEL SYNC will set and wait for the next sequential clock. When it arrives, the STROBE-SYNC flip-flop will set, putting Slave-Sync out on the bus.

Data is loaded at the spike shown on DATO EN coincident with the leading edge of the clock pulse. This scheme locks the Unibus to the DX11-B clock for strobe, eliminating a need to change clocks. The IBM interface is also strobed with this clock so that everything happens at the same clock and contention problems are eliminated.

ADRS SELD L is also used to enable a utility M155 in the center of the page where the four address bits A ⟨04:01⟩ are again decoded to provide address lines for use throughout the machine. They go to various gates to load registers.

The rest of the operation is standard PDP-11 interchange.

Sheet 2 shows the drivers and receivers on the Unibus. This is standard with two sets of drivers (BUS A ⟨15:00⟩ and BUS D ⟨15:00⟩), and two sets of receivers (DBUS ⟨15:00⟩ and A ⟨04:01⟩). Inputs are from the DXBA, DXND, and MUXD lines and are described in Paragraph 5.5.16. Enables (ADRS TO BUS H and DATA TO BUS H) are generated on the NPR and INT Control on Drawing 18. Extended address bits BUS A ⟨17:16⟩ are generated in the CUSR as a result of DXBA CRY created on Drawing 07.

5.5.15 NPR and INT Control

The NPR and INT control is shown on Drawing No. D-BS-DX11-B-13. This is the classic version of the PDP-11 control and, as such, will not be discussed here. For more information, refer to *PDP-11 Unibus Interface Manual,* DEC-11-HIAB-D. The priority is usually set at 4.

5.5.16 Data Bus Multiplexer

The data bus multiplexer is shown on Drawing No. D-BS-DX11-B-19. This circuit comprises sixteen M1713 modules which are 16-to-1 line data multiplexers. The outputs, labeled MUXD ⟨15:00⟩ L, are fed directly to the Unibus interface on Drawing 17. Note, once again, that by subtracting 10 from the location number the bit number can be derived.

Address bits A ⟨04:01⟩, received and buffered from the Unibus, are applied to all multiplexers as enabled by EN DATA TO BUS L. This arrangement takes each and every register in the DX11-B and gives back data to the Unibus. Each multiplexer represents a bit slice of those registers.

These M1713s are open-collector out, feeding directly to the M798s on the interface. Since the M798 contains a 1K resistor to +5V on the board, they can be plugged directly together.

5.5.17 Clock, Phase and Time State Generator

The Clock, Phase and Time State Generator is shown on Drawing No. D-BS-DX11-B-20. In this circuit, a basic 200 ns clock is generated by an M405 at B31, producing three free-running pulse trains, NCLK1, NCLK2 and PCLK1. The NCLK is also fed through an AND gate to produce CLKP1 and CLKP2. This is further inverted to produce gated CLKN signals.

The AND gate producing the CLKP and CLKN signals is used to provide a Maintenance Clock mode, and is enabled only by CLK GATE a function of the Maintenance Mode, made up of two flip-flops: a) Maintenance Clock Enable (DXES1 01), and b) Maintenance Clock flip-flop (DXES1 00).

Under normal circumstances, the 5 MHz clock NCLK and PCLK is free-running throughout the DX11-B; but when the DX11 is placed in Maintenance Clock mode by setting bit 01 in register 14 (DXES), the normal clock is disabled. As long as that bit is set, a clock pulse (CLKP and CLKN) is generated only when the Maintenance Clock flip-flop is set by DBUS00 (in DXES). This flip-flop is arranged to reset when it sets, producing a single pulse. The switch on the left-hand side of the drawing is located on the EPO panel and designated SINGLE STEP. Pulsing this switch will produce the same result. Note that the SSTF flip-flop will complement on each closure as its (0) L side is recirculated. The setting of SSTF will AND with PCLK1 to produce SSP which, in turn, can also clock the Maintenance Clock flip-flop. The Maintenance Clock flip-flop will then clear SSTF.

The rest of the circuit is involved in setting time states and phases. As described in the flows, it is these states that control the operating modes of the entire DX11-B. The free-running clock output is used to complement the time state flip-flop (TSFF) which is then used to condition a 16 output decoder (M155B24). The binary combinations of the three signals PH0, 1, and 2 determine the eight possible phases, while TSFF determines which of two possible states the machine is in. This results in a 16-time-compartmented structure expressed in PHS terms (Phase/State) to control the machine.

TSFF complements every 200 ns from a clock pulse width of 60 ns or less. The indicator for this function looks at TSFF (1) or TS1. When the indicator is out, the machine is in TS2. This is observable only when in Maintenance mode.

5.5.18 Maintenance Out Buffered Register (DXMOB)

The DXMOB is shown on Drawing No. D-BS-DX11-B-21, Sheets 1 and 2. This register holds the bits from the Unibus (DBUS ⟨15:00⟩) when addressing the DXMO while in on-line cabled mode. It is the function of these flip-flops to provide steady levels to the drivers when in this mode.

The left-hand byte is loaded with CONO ⟨07:00⟩, while the right-hand byte gets BUSO ⟨07:00⟩.

Clocking is accomplished by CONOB CLK, which is the AND of ADRS14ΛDATO EN HIGH combined with NCLK2 for the CONO half of the register. Clocking of the right half is identical using DATO EN LOW as part of the equation.

Clearing is a function of DX RESET OR INITΛCLKN2 yielding four identical DXCLRP signals which direct-clear four flip-flops each for loading purposes (actually six per board, see note on print).

At the bottom of Sheet 1 of this drawing notice a special maintenance flip-flop (SOSIEN) which eventually results in a signal called FASTSRV. This is used in Maintenance mode to create a fast Service-Out response to a programmed Service-In when simulating, diagnostically, an IBM sequence.

When the program loads DBUS02 and clocks it with LDADRL24 (Load Address Location 24), this flip-flop will set yielding SOSIEN (*S*ervice-*O*ut-*S*ervice-*I*n *En*able). Now when Service-In rises it ANDs to generate FASTSRV (CONOB01 is CLKO when cabled). FASTSRV then goes to the IBM Interface, Drawing 27, Sheet 1, where at location 4B it ANDs with OFFLINE to immediately generate Service-Out. Note that -FASTSRV H is equivalent to FASTSRV L.

A special condition exists on the CONOB07 flip-flop. Its direct-clear is also used to direct-set that flip-flop. The two gate delays (M111) on DXCLRP1 permit the direct-setting of that flip-flop after the pulse has gone away. Therefore, upon every clear of this register, this flip-flop is set to signal OPLO.

5.5.19 General Controls

The general controls are shown on Drawing No. D-BS-DX11-B-24. This circuit comprises the control signals for the DX11-B. The logic is straightforward and can be reviewed with the flow diagrams discussion in Paragraph 5.4.2.3. Because of this, the logic will not be discussed in great detail. Some signal mnemonics are obscure, such as the signal BLAST, generated at coordinates B1 and used as one possible condition for generating SET PH4 (coordinates C3). This signal is one of the IBM reset conditions which tells the DX11 to get off the bus.

Relating one signal on Drawing 24 to the flow diagram Drawing 04, Sheets 6 and 2, DX RESET (Drawing 24, coordinate B3) can be generated by SIG17Λ-STOP. This is verified on Drawing 04, Sheet 6, coordinate C7, where SIG17 implies Phase 7, TS2.

Consider also the signal ON OFFT at B6. This is an on/off transition signal used to get off the channel. To accomplish this transition in accordance with IBM protocol, LOCKOUT must be down, with no outstanding stack status, no command chaining indicated, and in-tag lines cleared. This insures that the DX11 is not doing a thing when the on/off transition occurs. Note that on Drawing 11, Sheet 4, if the program raises ONLINA (an on-line request) it ANDs with ON OFFT to put the DX11-B on-line (ONLINB).

All other signals in this circuit can be so related by cross reference.

5.5.20 Signals and Gates

The Signals and Gates are shown on Drawing No. D-BS-DX11-B-25, Sheets 1 through 3. As with Drawing 24, the signals generated on this drawing can be correlated in the flows. Some mnemonics may be obscure, such as BANX, which means the DX*BA*←TT*ND*X, or BANDSO which means the DX*BA*←DX*ND* and BU*SO*, or NDCRSR which means the DX*ND*←CU*CR* and CU*SR*; for the most part, the logic is straightforward and does not demand explanation.

Sheet 3 illustrates the means whereby the DX11 timeout is set. The M908 connector at F02 is provided for the installation of external capacitors for this purpose.

By installing three 180 μF caps across D1/L1, C1/M1, and B1/N1, an effective 540 μF of capacity is added to the M306 to set that oneshot to approximately 5 sec. This functions to take the DX11-B off of the channel arbitrarily if the DX11 holds the channel with Operational-In raised for longer than that time period. The disable flip-flop for DXTMO is clocked by LDADRL24 with DBUS03 as data. The DXTMO flip-flop is cleared by DX11 RESET OR INIT.

5.5.21 Parity and Address Compare

The Parity and Address Compare is shown on Drawing No. D-BS-DX11-B-26. This circuitry provides the means whereby the DX11-B recognizes its CU and DEV address and checks the parity of that address.

The M155 Decoder at A19 is enabled by ADRO (1) and PAROK. This means that address recognition will not be made unless an address is actually tagged on the BUSO and then only when it has been determined that the parity on that address is "odd". If parity compares to be "even" after addition of the parity bit, PAROK will not be generated and the address decoder module will *not* be enabled.

The parity comparators are standard modules. Notice that parity is generated on both Bus-In and Bus-Out. On output the EVEN generated by a compare and PARO, which the channel sends, produce PAROK. This represents "odd" parity.

The parity comparators utilize the eight bits of the IBM hexadecimal address. Although there are IBM bits 0–7, the DX11-B designates them as bits ⟨07:00⟩ (see the conversion table at the top-left of Drawing 27). This hard-wired conversion is made in the DX11-B/IBM Interface on that drawing.

Address decoding is split into two parts: a) Control Unit Address (CUA) utilizing the four IBM bits 0–3 (BUSO ⟨07:04⟩), and b) Device Address derived from IBM bits 4–7 (BUSO ⟨03:00⟩). The first portion of this module is hardwired, while the second portion can be either hardwired or programmed.

The M155 at A19 is capable of recognizing up to 16 possible CU addresses (0-F in hexadecimal notation) as set by BUSO ⟨07:04⟩. These outputs are all made available at an M908 jumper board at A20 which limits the total number of addresses recognized to 8 of the possible 16.

If 8 CU addresses are to be recognized, jumpers are installed in alphabetical order from any 8 pins (A1 through V1) to V2 through M2 shown as part of the second appearance of A20 in the center of the page (see sample at left-hand corner of Drawing 26).

If less than 8 addresses are required, all unused pins beyond the last one used are wired to that last pin. The example illustrates jumpering for just one CU address (1X).

This arrangement responds to just the first digit of a two-digit address. Note that these decoded outputs are ORed at M119B21 as one leg of an AND gate which yields ADRECC (Control Unit Address Compare).

The second digit can be ignored if desired by grounding pins B2, E2, H2, and K2 as shown in the example. If more than one device address is involved in the installation, this ground can be removed and pins L2 through J2 wired to B2 through K2, allowing recognition of up to 16 device addresses for each of up to 8 control unit addresses.

This is done by allowing the program to compare the low-order bits in the CUAR with BUSO ⟨03:00⟩.

These bits now AND to form the second leg of the AND at B18 that yields ADRECC. They are also ANDed as device compare (DEVCOM) with ADRECC to yield ADRECD. These two outputs then comprise the entire address recognition in Phases 1 and 2 to process the DX11-B to its next called for phase.

5.5.22 IBM Interface

The IBM Interface is shown on Drawing No. D-BS-DX11-B-27, Sheets 1 through 6. It is here that bit numbering and assertion levels are converted from IBM standard to DEC standard.

The table in the upper right-hand corner of Sheet 1 gives the conversion of bit numbering. Note that no bits are shifted or reversed. Bit streams are not altered here, just renumbered to conform with PDP-11 format.

NOTE

The fact that the PDP-11 numbers bits from right to left does not in any way alter the significance of any bit with respect to other numbering schemes.

Sheet 1 of this drawing lists all signals received *from* the channel, and Sheet 2 shows all signals the DX11 sends *to* the channel. Sheet 3 provides the cable connectors for these signals. Sheets 4 and 5 are devoted to the test bus signals, with Sheet 6 containing those connectors.

The M597 module, used on Sheet 1, contains an added inverter that performs a double inversion function on BUSO signals. These signals are high when true. When ONLINE 1 is high and when BUS OUT 0 is high (true) a logic mismatch occurs at the input to the OR yielding BUSO 07 H as a default triggering of the logic.

There is no logic mismatch in the lower gate in which the buffered versions can be high when OFFLINE is also high.

This pattern of logic repeats on all receivers of this interface. Note that it is at this point that the format numbering is converted.

5.5.23 Select Bypass and Power Fail

Select bypass and power fail circuits are shown on Drawing D-BS-DX11-B-29. This is a composite drawing showing the sources for the various signals used by this G890 module at location F01.

In the IBM 360/370 System, selection of all control units (the DX11-B being one) for communication with the channel is controlled by the signal Select-Out. This signal is routed serially through the "select" circuitry of all CUs, permitting each to respond to it sequentially if that CU recognizes its address, decoded elsewhere in the CU.

If recognition occurs, the CU seizes the signal and prevents it from being propagated to the next CU on the line. If recognition does not occur, the signal is passed on down the line. If no CU recognizes the address sent, Select-Out becomes Select-In to the channel, indicating to the channel that no CU responded. Figure 2 of the IBM OEMI manual illustrates this scheme.

Additionally, if any CU is powered down, if power has failed, or if the CU is off line, the Select line is automatically routed through back contacts of the relay and the "select-seize" logic is bypassed.

Note from Drawing 29 that if the DX11 is on-line (-OFFLINE L) and power is up (-DCLO L), the Select-Out relay will pick allowing the SEL OUT signal to enable the data input of the SELO flip-flop. Note also that this is, in turn, latched by the signal Hold-Out (HLDO). This is a parallel signal to all CUs to permit an immediate drop of the select bypassing relay when Select-Out from the CH drops.

5.5.24 "Bit Slice" and "Select" Loop Trace in Off-Line Cabled Mode

This paragraph is devoted to tracing the Select-Out signal and one bit through the DX11-B when it is operating in off-line cabled mode. The purpose is to provide familiarity with the general routing of these signals so that any bit can be traced if needed. This mode of operation is chosen so that all DX11-B circuitry can be covered, including the connectors to and from the IBM channel.

Figure 5-18 illustrates the path for BUS OUT3. Note the bit number conversion as shown on Drawing 27. Figure 5-19 illustrates the routing of the Select-Out signal as received from the channel bus on Drawing 27. Note that SEL OUT A, after passing through the G890 at F0, becomes SEL OUT B out to the system, through the jumper on the last unit. It then returns as SEL IN B and is returned to the channel as SEL IN A.

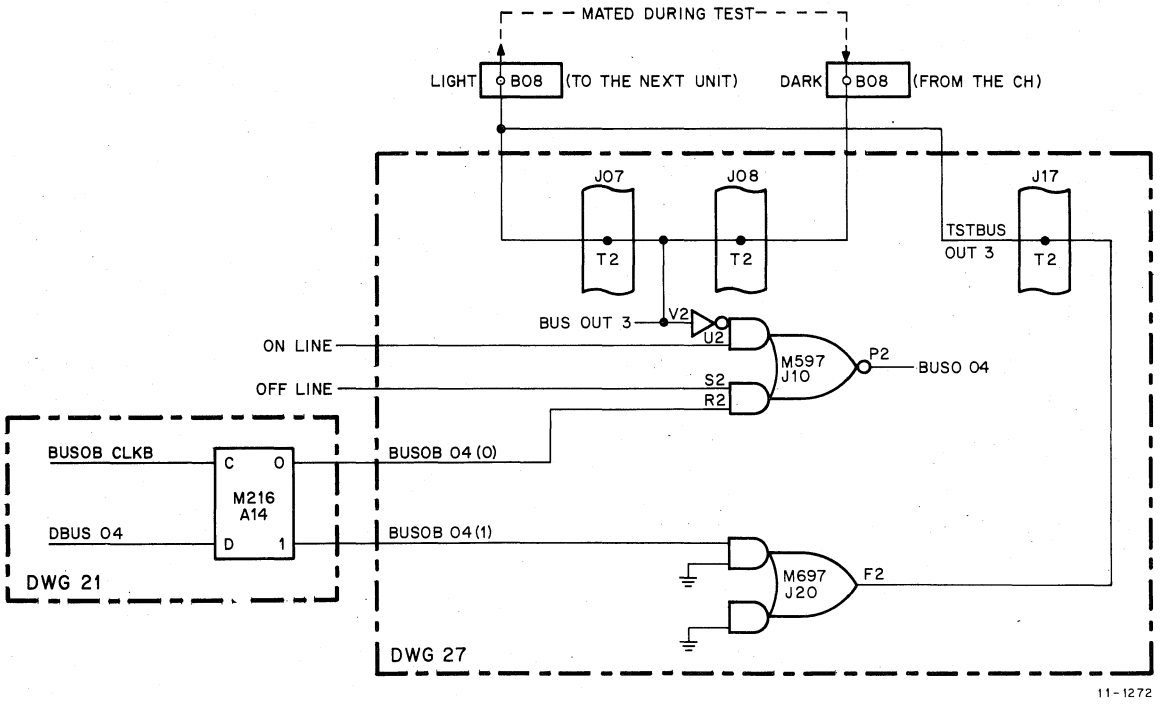


Figure 5-18 One Bit Slice Loop Trace

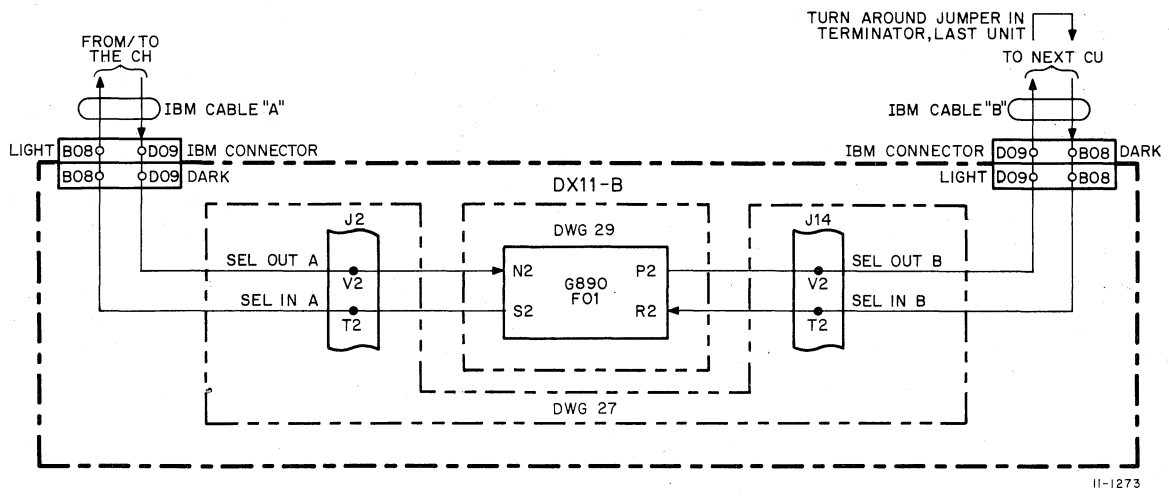


Figure 5-19 Select-Out/Select-In Loop Trace

5.5.25 EPO Panel Operation

The DX11-B miscellaneous control and EPO panel is shown schematically on Drawing D-CS-7008792-0-1, and is illustrated in Figure 3-1. The panel provides the only operating controls on the DX11-B, with the exception of the Power Contactor on the H720e Power Supply, and operates in conjunction with the standard IBM Power Control Interface. A general description of all IBM Power Interfaces is given in IBM Document File No. S360-19 GA22-6906-0 "IBM System/360 Power Control Interface Original Equipment Manufacturers' Information". The operation of the DX11-B interface conforms to the requirements described in that document.

In IBM systems, all units are powered from the main CP of the system. In addition, application of power to these elements is not applied at once, but rather is applied sequentially to each element thereby reducing overall power requirements and minimizing noise generation from power surges. Each element is equipped with a set of standard relays, switches, and indicators, together with a unit power source; and all elements usually operate in some mode of remote control although most can also operate in a local mode. Finally, each unit can be disabled immediately by operation of the Emergency Power Off (EPO) button located on the front panel of the 360/370 Console. Operation of this button is unlikely except in cases of extreme emergency.

Referring to the diagram in Drawing 7008792, the DX11-B Unit Power Source (approximately 12.6 Vdc) is provided by T1 feeding an MDA 980-1. Note that this supply is always powered as long as the DX11 power cord is connected to ac power. When operating on 120 Vac, pin 1 of TB2 is left floating. The diagram entitled EPO TIMING on the drawing describes the sequence of operation.

With the LOCAL PWR ON switch closed, power is complete to the H720e Power Supply through pins 7 and 8 of TB2.

With the LOCAL/SYSTEM switch in LOCAL position, power from the unit power source is complete through pin 1 of EPO connector J1, through the contacts of the EPO switch on the IBM console to the NC contacts of the LOCAL/SYSTEM switch, through the coil of K3 to -V. This closes contacts 8 and 6 of that relay to apply power to the H720e Power Supply. The upper switch contacts provide a "step by" capability of the IBM stepping switch if power sequencing is initated by the 360 when the DX11 is in LOCAL mode.

With the LOCAL/SYSTEM switch in SYSTEM position, unit source power (always powered) is complete through the EPO switch, through pin 2 of J1 (EPO CONTROL) to -V, through the coil of K1 which closes its contacts 1 and 3. When the SYSTEM PWR ON switch is closed, and when the step control switch is advanced to the DX11-B position, unit source power is complete through pin 6 of J1 (PWR PICK) and the now closed contacts of K1 to -V through the coils of K2 and K3. When contacts 1 and 3 of K2 close, system source is complete through pin 5 of J1 (PWR HOLD) and the PWR HOLD indicator on the EPO panel is complete through contacts 8 and 6 of K2.

When contacts 1 and 3 of K3 close, stepping relay power is complete through pin 3 of J1 (SYSTEM SRC) and returns through pin 4 (PWRING COMP) to step the switch to the next unit in the sequence. Power on the DX11-B is now held on by the unit source through contacts 1 and 3 of K2, and the coil of K3.

Once this sequence is complete, power is removed if either the SYSTEM POWER ON switch is opened or if the EPO contacts are opened.

If SYSTEM PWR is opened, PWR HOLD is lost dropping both K2 and K3. K1 remains energized, ready for another power-up sequence.

If the EPO contacts are opened, K1 also drops and EPO CONTROL is lost. Under these circumstances, remote powering of the DX11-B cannot be accomplished until they are once again closed.

The other controls on this panel are all interconnected by P1 and TB1. Pin C2 is ground. The SINGLE PULSE switch interconnects with Drawing 20 where it is used to activate two AND gates in cross-conjunction which, in turn, clock the SSTF flip-flop.

ACLO and DCLO sensing from the H720e Power Supply is sent to Drawing 18 to yield the signals ACLO and DCLO.

The ONLINE ENABLED switch sends ground to Drawing 29 where it activates a cross-conjunction flip-flop to yield ONLINE EN SW H and L. In the enable position, it causes the ONLINE ENABLE indicator on this panel to glow.

CHAPTER 6
MAINTENANCE

6.1 INTRODUCTION

Maintenance philosophy of the DX11-B is based on the premise that an optimum amount of preventive procedures, performed regularly, can eliminate many costly equipment breakdowns and can forecast failures before they occur. The design is such that, in the event of failure of a specific item, module replacement can restore the equipment to service in a minimum of time. This chapter contains both preventive and corrective maintenance procedures.

6.2 PREVENTIVE MAINTENANCE

Preventive maintenance consists of tasks performed at periodic intervals to ensure proper equipment operation and minimum unscheduled downtime. These tasks consist of running diagnostics, visual inspection, operational checks, adjustment, and replacement of marginal components.

The preventive maintenance schedule depends on the environmental and operating conditions that exist at the installation site. Under normal conditions, recommended preventive maintenance consists of inspection and cleaning every 600 hours of operation or every 4 months, whichever occurs first. However, relatively extreme conditions of temperature, humidity, dust, and/or abnormally heavy work loads demand more frequent maintenance. It is recommended that the IBM 2848 Diagnostic be run once a week as part of the normal preventive maintenance schedule.

6.2.1 Mechanical Checks

Inspect the DX11-B periodically as follows:

a. Visually inspect the unit for general condition.

b. Clean the interior and exterior of the rack using a vacuum cleaner or a clean cloth moistened in non-flammable solvent.

c. Inspect all wiring and cables for cuts, breaks, frays, deterioration, kinks, strain, and mechanical security. Tape, solder, or replace any defective wiring or cable covering.

6.2.2 Test Equipment Required

Maintenance activities for the DX11-B require the standard test equipment and diagnostic programs listed in Table 6-1, in addition to standard hand tools, cleaners, test cables, and probes. Special test equipment required for any adjustments are given as part of the adjustment procedure.

Table 6-1
Test Equipment Required

Equipment	Manufacturer	Designation
Multimeter	Triplett or Simpson	Model 630-NA or 260
Oscilloscope	Tektronix	Type 453
X10 Probes (2)	Tektronix	P6008
Module Extender	DEC	Type W980
Diagnostic Self-Test Routines	DEC	MAINDEC-11-DZDXA MAINDEC-11-DZDXB MAINDEC-11-DZDXC MAINDEC-11-DZDXD MAINDEC-11-DZDXE

6.3 CORRECTIVE MAINTENANCE

The procedures that follow are based on standard troubleshooting techniques. Once the defective module has been located, replace it with a spare module and return the defective module to DEC for repair or replacement.

6.3.1 General Corrective Procedures

Before beginning troubleshooting procedures, ensure that the equipment external to the DX11-B is operating correctly. Check with IBM CE and refer to PDP-11 maintenance manuals to determine status. Also, examine the DX11-B Maintenance Log to determine if the fault has occurred before and, if so, note what steps were taken to correct the condition.

NOTE
A recurring fault can indicate a common uncorrected fault elsewhere.

Visually inspect the physical and electrical security of all cables, connectors, modules, and wiring. In particular, check the security of ground connections between racks. Faulty grounds can produce a variety of faults.

6.3.2 Diagnostic Testing

DEC provides special diagnostic programs (MAINDECs) to assist in localizing faults within the equipment. Functionally, the programs fall into two categories: test and reliability. Test programs isolate genuine go/no-go type hardware failures that are easily recognizable; reliability programs isolate failures that are more difficult to detect, because they are marginal in nature and/or occur infrequently or sporadically. The family of test programs are written so that, when run successively, they test the equipment beginning with small portions of the hardware then gradually expand until they involve the entire machine. To accomplish this, they are built around instructions and portions of instructions whose demands on equipment capabilities progress from simple operations to the most involved manipulations. As portions of the system are proven operable, they become available to succeeding tests for use in checking unproven portions of the machine.

In the DX11-B there are six programs comprising the diagnostic package. They include:

a. The CTP that operates in two modes. In OFFLINE mode it exercises all the communication devices and the DX11-B in test mode. In ONLINE mode it runs all the communication devices with the DX11-B driven by the 360 running, e.g., FRIEND.

b. The DX11-B Diagnostic that provides a detailed off-line logic checkout of the DX11-B hardware.

c. The GTP Systems Exercisor, an off-line bus latency checker with GTP-supported devices.

d. A RESPONDER for the IBM 2848 diagnostic.

e. A 360 program to checkout the DX11-B hardware and software in the 2848 mode of operation.

f. FRIEND, a 360 CCW generator program that generates arbitrary CCW strings to check operator requested functions of any IBM CU.

6.3.3 Vibration Tests

Intermittent malfunctions can be located usually by performing a vibration test. This aggravates a condition that otherwise might not be located. To perform a vibration test, proceed as follows:

1. Check switches for immunity from vibration and shock by wiggling them and tapping them with the fingers.

2. Check modules for immunity from vibration and shock in two planes. To check the plane perpendicular to the module mounting plane, tap each module handle with the fingers. To check the plane parallel to the module mounting plane, slide a Teflon rod horizontally across the modules. This should be done slowly and twice in each direction. The Teflon rod should be approximately 8-in. long, 3/8-in. in diameter, and should be held between the fingers 6 in. from the end that is applied to the modules. This test will indicate bad components and poor solder joints.

CAUTION

If vibration tests are applied too vigorously, damage to modules could result. Do NOT vibrate the IBM connectors or the G890F1 module.

After localizing the fault to within a functional logic element, run the diagnostic which uses all functions of that element. Trace signal flow through the suspected element with an oscilloscope by synchronizing the oscilloscope sweep with control signals or clock pulses. Check for proper levels, durations, rise and fall times, and timing of all input and output signals.

CHAPTER 7
ENGINEERING DRAWINGS

7.1 GENERAL

Table 7-1 lists those engineering drawings called out in text. The listing is alphanumeric by drawing number and does not include all drawings in the manufacturing set.

Table 7-1
DX11-B Engineering Drawings

Number	Title
D-FD-DX11-B-04	Flow Diagram (9 sheets)
	Sheet 1 – Phase Independent Sequences
	Sheet 2 – Phase 0 Idle or Requesting
	Sheet 3 – Phase 1 Address Response
	Phase 2 Status Preparation
	Sheet 4 – Phase 3 Status Presentation
	Phase 4 Mark
	Sheet 5 – Phase 5 Input
	Phase 6 Output
	Sheet 6 – Phase 7 Done
	Sheet 7 – Disconnect Response
	Sheet 8 – Programmed Registers
	Sheet 9 – Data Paths
D-BS-DX11-B-05	DXDS DX Device Status (2 sheets)
06	DXCS Control Unit Status (2 sheets)
07	DXBA DX Bus Address
08	DXBC DX Byte Counter
09	CUSR (Control Unit Status Register) (2 sheets)
10	DXND NPR Data Register (2 sheets)
11	DXCB DX Control Bits (4 sheets)
12	DXMO/CONO
13	DXMI Control In/Bus In (2 sheets)
14	TTNDX Tumble Table Index
15	CUAR Control Unit Address Register
16	CUCR Control Unit Control Register
17	Unibus Interface (2 sheets)
18	NPR and INT Control
19	Data Bus Mux.
20	Clock, Phase & Time State Gen.
21	DXMOB – CONO BUFF/BUSO BUFF (2 sheets)

(continued on next page)

Table 7-1 (Cont)
DX11-B Engineering Drawings

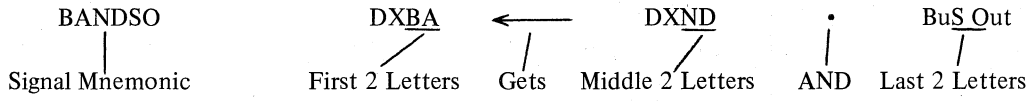
Number	Title
D-IC-DX11-B-22	Indicator Conn
D-BS-DX11-B-24	General Controls
25	SIG & Gates (3 sheets)
26	Parity and Address Compare
27	IBM Interface (6 sheets)
D-IC-DX11-B-28	Unibus Connectors
D-BS-DX11-B-29	Select Bypass & Power Fail
D-CS-7008792-0-1	Misc Controls & EPO

APPENDIX A
REFERENCE MATERIAL

This appendix contains selected data for reference by the reader. Included are a glossary of mnemonic terms and a conversion method for hexadecimal to octal to EBCDIC notation. Further reference material can be found as part of the appendices to the *IBM System/360 Principles of Operation Manual* as listed in the Foreword of this manual.

A.1 GLOSSARY

In Table A-1 the signals are arranged in alphanumeric order as they appear on the block schematics. In instances of complex signals the following notation is used:



The left column contains the signal being defined; the right column contains the definition of the actions involved. In the example, the DXBA (Bus Address Register) receives the contents of (gets) the DXND (NPR Data Register) and the contents of Bus Out.

These mnemonics can be further clarified by reference to Drawing 04, Sheets 1 through 9.

Table A-1
DX11-B Signal Glossary

Signal	Definition
ACLO	PDP-11 AC voltage low
ADR OUT	Address Out
ADRECC	Control Unit Address Compare
ADRECD	Device Address Compare
ADRI(0)/(1)	Address In
ADRIN	Address In
ADRO	Address Out
ADRSxx	Address (Unibus) xx = 02, 04, 06, 10, 12, 14, 16, 24
ADRS SELO	Address Select Out
ADRS TO BUS	Address to Bus

(continued on next page)

Table A-1 (Cont)
DX11-B Signal Glossary

Signal	Definition
ARSO	CU*AR*←BU*SO*
ATTEN(0)/(1)	Attention
BAAR	DX*BA*←CU*AR*
BADB	DX*BA*←*DB*us
BAINxx	Bus Address In xx = 00–15
BALF(0)/(1)	Bus Alternator Flip-Flop
BANDSO	DX*BA*←DX*ND*·BU*SO*
BANX	DX*BA*←TT*ND*X
BAOR	DX*BA*←CU*OR*
BCMO	Byte Count Minue One
BG IN	Bus Grant In
BG OUT	Bus Grant Out
BIAR	*B*us *I*n←CU*AR*
BISR	*B*us *I*n←CU*SR*
BIT x COMP	Bit x Compare x = 0–3
BLAST	A "Get Off" signal resulting from IBMRST
BLLF(0)/(1)	Bus Load Limiting Flip-Flop
BLLM	Unibus Load Limiting Mono
BR OUT	Break Request Out
BSY(0)/(1)	Busy
BSYEN(0)/(1)	CU Busy Enable
BSYS	Busy Sent
BUDATA	Bus Data
BUS Axx	Bus Address xx = 00–17
BUS AC LO	Bus AC Low
BUS BBSY	Bus Busy
BUS BGxIN	Bus Grant In x = 4–7
BUS BR x	Bus Request x = 4–7
BUS Cx	Bus Control Lines x = 0, 1
BUS Dxx	Bus Data xx = 00–15
BUS DC LO	Bus DC Low
BUS IN x	Bus In x = 0–7

(continued on next page)

Table A-1 (Cont)
DX11-B Signal Glossary

Signal	Definition
BUS IN P	Bus In Parity
BUS INIT	Bus Initialize
BUS INTR	Bus Interrupt
BUS MSYN	Bus Master Sync
BUS NPG IN/OUT	Bus Non-Processor Grant
BUS NPR	Bus Non-Processor Request
BUS OUT x	Bus Out x = 0–7
BUS OUT P	Bus Out Parity
BUS PA	Bus Parity Available
BUS PB	Bus Parity Bit
BUS SACK	Bus Selection Acknowledge
BUS SSYN	Bus Slave Sync
BUSI	IBM Bus In Register
BUSI xx (0)/(1)	Bus In xx = 00–07
BUSO	IBM Bus Out Register
BUSO xx	Bus Out xx = 00–07
BUSOB xx (0)/(1)	Bus Out Buffered xx = 00–07
BUSOB CLKA/B	Bus Out Buffered Clock
BYPAS(0)/(1)	Bypass
CAW	Channel Address Word
CCW	Channel Command Word
CE	Channel End
CH	Channel
CHDEND	Channel Data End
CHEND(0)/(1)	Channel End
CHENDS	Channel End Sent
CHIS(0)/(1)	Channel Initiated Sequence
CLINTR	Clear Interrupt
CLK GATE	Clock Gate
CLK OUT	Clock Out
CLKN1/2/3	Negative Clock
CLKO	Clock Out

(continued on next page)

Table A-1 (Cont)
DX11-B Signal Glossary

Signal	Definition
CLKP1/2	Positive Clock
CLR CUAR	Clear the CUAR
CLR CUCR	Clear the CUCR
CLR CUSR	Clear the CUSR
CLR DXBC	Clear the DXBC
CLR DXCB	Clear the DXCB
CLR DXCS	Clear the DXCS
CLR DXDS/A	Clear the DXDS
CLR DXMI	Clear the DXMI
CLR FCTN	Clear the Function
CLR IRC	Clear IBM Reset Conditions
CMD OUT	Command Out
CMDCHN	Command Chain
CMDO(0)/(1)	Command Out
CMDREJ	Command Reject
CMDSRV	Command Service
CONI	Control Lines In Register
CONIZ	Control Lines In Zero
CONO	Control Lines Out Register
CONO CLKA/B	Control Lines Out Clock
CONOB xx(0)/(1)	Control Lines Out Buffered xx = 00–07
CONOB CLKA/B	Control Lines Out Buffered Clock
COUNT BA	Count Bus Address
COUNT BC	Count Byte Counter
COUNT TTNDX	Count Tumble Table Index
CRSO	CU*CR*←BU*SO*
CRSRND	CU*CR*←CU*SR*·DX*ND*
CSW	Channel Status Word
CU	Control Unit
CUA xxX	Control Unit Address xx = 00, 02, 04, 06, 10, 12, 14, 16, 20, 22, 24, 26, 30, 32, 34, or 36 X = Device Address

(continued on next page)

Table A-1 (Cont)
DX11-B Signal Glossary

Signal	Definition
CUAR xx(0)/(1)	Control Unit Address Register xx = 00–07
CUAR CLKA/B	Control Unit Address Register Clock
CUAR DATA LOAD	Control Unit Address Register Data Load
CUBSY(0)/(1)	Control Unit Busy
CUCR xx(0)/(1)	Control Unit Command Register xx = 00–07
CUCR CLK EN	Control Unit Command Register Clock Enable
CUCR CLKA/B	Control Unit Command Register Clock
CUD	Control Unit Done
CUDEND	Control Unit Data End
CUDX(0)/(1)	Control Unit Data Control
CUEND(0)/(1)	Control Unit End
CUFBM(0)/(1)	Control Unit Forced Burst Mode
CUI5	Control Unit Initiated (Input)
CUI6	Control Unit Initiated (Output)
CUOR xx	Control Unit Offset Register xx = 10–15
CUSR	Control Unit Status Register
CUSR CLK EN/A	Control Unit Status Register Clock Enable
CUSRNZ	Control Unit Status Register Non-Zero
DATA IN	Data In
DATA OUT	Data Out
DATA STROBE	Data Strobe
DATA TO BUS	Data to Bus
DATA WAIT	Data Wait
DATO EN HIGH/LOW	Data Out Enable
DATOK	Data OK
DBUS xx	Data Bus xx = 00–15
DCLO	DC Low
DE	Device End
DEV	Device
DEVCOM	Device Compare
DEVEND(0)/(1)	Device End

(continued on next page)

Table A-1 (Cont)
DX11-B Signal Glossary

Signal	Definition
DIS DXTO(0)/(1)	Disable DX Timeout
DISC IN	Disconnect In
DONE	Done
DS	Data Sequence
DSCRSP(0)/(1)	Disconnect Response
DT	Data Transfer
DX	DX11-B
DXBA xx	NPR Address Register xx = 00–15
DXBA CRY	NPR Address Register Carry
DXBC xx	DX Byte Count Register xx = 00–15
DXCA	DX Command & Address Register
DXCB	DX Control Bits Register
DXCLRP1/2/3/4	DX Clear Pulse
DXCS	DX Control Unit Status Register
DXCS CLKA	DXCS Clock
DXCS CLKEN/A	DXCS Clock Enable
DXDS	DX Device Status Register
DXES1/2	DX Extra Signals Register
DXMI	DX Maintenance In Register
DXMO	DX Maintenance Out Register
DXMOB	DX Maintenance Out Buffered Register
DXND xx(0)/(1)	DX NPR Data Register xx = 00–15
DXND CLK A/B	DXND Clock
DXND STB A/B	DXND Strobe
DXOS	DX Offset and Status Register
DXRESET	DX Reset
DXRESET OR INIT	DX Reset or Initialize
DXTO	DX Timeout
EN DATA TO BUS/A	Enable Data to Bus
EN FCTN	Enable Function
ENABLE	Enable
END CYCLE	End Cycle

(continued on next page)

Table A-1 (Cont)
DX11-B Signal Glossary

Signal	Definition
ENDEN	End Enable
EPO	Emergency Power Off
ERRSTA	Error Status
ES	Ending Sequence
ESEND	Ending Sequence End
ETAP	Emulator Terminal Applications Program
EVEN	Even
FASTCU	Fast Control Unit Response to an ISS while busy
FASTSRV	Fast Service-In Response to Service-Out while in Test Mode
FASTX	Fast Control Unit Response to INFDSC
FCTN CLK	Function Clock
FCTN1/2(0)/(1)	Function
FTCU	Fast Control Unit Response While Busy
FUNEQ0/1/2/3	Function Equals
GO	Go
HIO	Halt I/O
HLD OUT	Hold Out
HLDO/(1)	Hold Out
IBMRST	IBM Reset
IN	In
INBA 08	In Bus Address Bit 08
INFDSC(0)/(1)	Interface Disconnect
INIT	Initialize
INMSTR	Interrupt Master
INTEN	Interrupt Enable
INTEN CLKEN	Interrupt Enable Clock Enable
IOD	I/O Done
IRCCLR	IBM Reset Conditions Clear
IRS(0)/(1)	IBM Reset Stored
ISS	Initial Selection Sequence
ISSREJ(0)/(1)	ISS Reject

(continued on next page)

Table A-1 (Cont)
DX11-B Signal Glossary

Signal	Definition
LDADRL 24	Load Address Location 24
LOAD BUS DXMI	Load Bus DX Maintenance In
LOAD DXBA	Load DXBA
LOAD DXBC	Load DXBC
LOCKO(0)/(1)	Lockout
m	Modifier Bit
MARK IN 1/0	Mark In
METER IN	Meter In
METER OUT	Meter Out
MNCLEN(0)/(1)	Maintenance Clock Enable
MNCLKF(0)/(1)	Maintenance Clock Flip-Flop
MPX	Multiplexer Channel
MUXD xx	Multiplexed Data xx = 00–15
NCLK1/2	Negative Clock
NDCRSR	DX*ND*←CU*CR*·CU*SR*
NDDB	DX*ND*←*D*ata *B*us
NODST	No Offset to Device Status Table
NPMSTR	NPR Master
NPRDLY	NPR Delay
NPRT(0)/(1)	NPR Transfer Direction
NPRTO(0)/(1)	NPR Timeout
NPRX(0)/(1)	NPR Transfer
NPTDON(0)/(1)	NPR Transfer Done
NXM(0)/(1)	Non-Existent Memory
ODD	Odd
OFFLINE	Off Line
OFFRCVREN	Off Line Receivers Enable
ON	On
ONLINA	On Line A
ONLINB(0)/(1)	On Line B
ONLINE	On Line
ONLINE EN SW	On Line Enable Switch

(continued on next page)

Table A-1 (Cont)
DX11-B Signal Glossary

Signal	Definition
ONLINE1	On Line 1
ONOFFT	On/Off Transition
OPL IN	Operational-In
OPL OUT	Operational-Out
OPLI(0)/(1)	Operational-In
OPLO(0)/(1)	Operational-Out
OUT HIGH	Out High
OUT LOW	Out Low
PARER	Parity Error
PARI	Parity In
PARO	Parity Out
PAROK	Parity OK
PARSTP	Parity Stop
PCLK1	Positive Clock
PDPTO(0)/(1)	PDP Timeout
PHx(0)/(1)	Phase x = 0, 1, 2
$PHSx_1x_2$	Phase State x_1 = 0–7, x_2 = 1, 2
PSW	Program Status Word
PWR OK RLY	Power OK Relay
RADRI	Received Address-In
RBUSI xx	Received Bus-In xx = 00–07
RBUSI P	Received Bus-In Parity
REQ IN	Request-In
REQI(0)/(1)	Request-In
REQIN	Request-In
ROPLI	Received Operational-In
RREQI	Received Request-In
RSELI	Received Select-In
RSRVI	Received Service-In
RSTAI	Received Status-In
SEL	Selector Channel
SEL IN/A/B	Select-In

(continued on next page)

Table A-1 (Cont)
DX11-B Signal Glossary

Signal	Definition
SEL OUT/A/B	Select-Out
SEL SYNC	Select Sync
SELECT	Select
SELI(0)/(1)	Select-In
SELO(0)/(1)	Select-Out
SELRST(0)/(1)	Selective Reset
SENSE	Sense
SETPHx	Set Phase x = 0–4
SICR	BU*SI*←CU*CR*
SIO	Start I/O
SISR	Bu*S In*←CU*SR*
SISREN	BU*SI*←CU*SR EN*able
SOSIEN	*S*ervice-*O*ut/*S*ervice-*I*n *EN*able
SPW	Status Pointer Word
SRDB	CU*SR*←*D*ata *B*us
SRNDH/L	CU*SR*←DX*ND*
SRNDEN	CU*SR*←DX*ND EN*able
SRSO	CU*SR*←BU*SO*
SRV IN	Service-In
SRV OUT	Service-Out
SRVI(0)/(1)	Service-In
SRVO(0)/(1)	Service-Out
SSP	Single Step Pulse
SST	Single Step Transition
SSTF(0)/(1)	Single Step Transition Flip-Flop
SSYN	Slave Sync
STA IN	Status-In
STAI EN	Status-In Enable
STAI(0)/(1)	Status-In
STAMOD(0)/(1)	Status Modifier
STB/A CUSR	Strobe Control Unit Status Register
STKSTA(0)/(1)	Stack Status

(continued on next page)

Table A-1 (Cont)
DX11-B Signal Glossary

Signal	Definition
STOP	Stop
STR SYNC	Strobe Sync
SUP OUT	Suppress-Out
SUPO(0)/(1)	Suppress-Out
SYNC(0)/(1)	Synchronization
SYSRST(0)/(1)	System-Reset
TCH	Test-Channel
TESTIO	Test-I/O
TIC	Transfer In Channel
TIO	Test I/O
TPx	Time Pulse x = 1, 2
TSFF(0)/(1)	Time State Flip-Flop
TSTADR IN	Test Address-In
TSTADR OUT	Test Address-Out
TSTBUS IN x	Test Bus-In x = 0–7
TSTBUS IN P	Test Bus-In Parity
TSTBUS OUT x	Test Bus-Out x = 0–7
TSTBUS OUT P	Test Bus-Out Parity
TSTCMD OUT	Test Command-Out
TSTHLD OUT	Test Hold-Out
TSTOPL IN	Test Operational-In
TSTOPL OUT	Test Operational-Out
TSTREQ IN	Test Request-In
TSTSEL IN	Test Select-In
TSTSEL OUT	Test Select-Out
TSTSRV IN	Test Service-In
TSTSRV OUT	Test Service-Out
TSTSTA IN	Test Status-In
TSTSUP OUT	Test Suppress-Out
TT	Tumble Table
TTNDX xx	Tumble Table Index xx = 00–07
TTNDX CLR	Tumble Table Index Clear

(continued on next page)

Table A-1 (Cont)
DX11-B Signal Glossary

Signal	Definition
UCHECK(0)/(1)	Unit Check
UCHKS	Unit Check Sent
UEXCEP(0)/(1)	Unit Exception
XBAxx(0)/(1)	Extended Bus Address xx = 16, 17

A.2 HEX/OCTAL/EBCDIC CONVERSION CHART

Table A-2 provides a means of converting the hexadecimal representation for the keyboard functions to their octal equivalent (Note, ASCII reference implied) and to the EBCDIC representations.

Table A-2
Hex/Octal/EBCDIC Conversion Chart

Hex	Octal	EBCDIC	Hex	Octal	EBCDIC
00	000	NUL	20	040	DS
01	001	SOH	21	041	SOS
02	002	STX	22	042	FS
03	003	ETX	24	044	BYP
04	004	PF	25	045	LF
05	005	HT	26	046	ETB
06	006	LC	27	047	ESC
07	007	DEL	2A	052	SM
0A	012	SMM	2B	053	CU2
0B	013	VT	2D	055	ENQ
0C	014	FF	2E	056	ACK
0D	015	CR	2F	057	BEL
0E	016	SO	32	062	SYN
0F	017	SI	34	064	PN
10	020	DLE	35	065	RS
11	021	DC1	36	066	UC
12	022	DC2	37	067	EOT
13	023	TM	3B	073	CU3
14	024	RES	3C	074	DC4
15	025	NL	3D	075	NAK
16	026	BS	3F	077	SUB
17	027	IL	40	100	SP
18	030	CAN	4A	112	¢
19	031	EM	4C	114	<
1A	032	CC	4D	115	(
1B	033	CUI	4E	116	+
1C	034	IFS	4F	117	|
1D	035	IGS	50	120	&
1E	036	IRS	5A	132	!
1F	037	IUS	5B	133	$

(continued on next page)

Table A-2 (Cont)
Hex/Octal/EBCDIC Conversion Chart

Hex	Octal	EBCDIC	Hex	Octal	EBCDIC
5C	134	*	A6	246	w
5D	135	)	A7	247	x
5E	136	;	A8	250	y
5F	137	¬	A9	251	z
60	140	-	C1	301	A
61	141	/	C2	302	B
6B	153	,	C3	303	C
6C	154	%	C4	304	D
6D	155	_	C5	305	E
6E	156	>	C6	306	F
6F	157	?	C7	307	G
7A	172	:	C8	310	H
7B	173	#	C9	311	I
7C	174	@	D1	321	J
7D	175	'	D2	322	K
7E	176	=	D3	323	L
7F	177	"	D4	324	M
81	201	a	D5	325	N
82	202	b	D6	326	O
83	203	c	D7	327	P
84	204	d	D8	330	Q
85	205	e	D9	331	R
86	206	f	E2	342	S
87	207	g	E3	343	T
88	210	h	E4	344	U
89	211	i	E5	345	V
91	221	j	E6	346	W
92	222	k	E7	347	X
93	223	l	E8	350	Y
94	224	m	E9	351	Z
95	225	n	F0	360	0
96	226	o	F1	361	1
97	227	p	F2	362	2
98	230	q	F3	363	3
99	231	r	F4	364	4
A2	242	s	F5	365	5
A3	243	t	F6	366	6
A4	244	u	F7	367	7
A5	245	v	F8	370	8
			F9	371	9

A.3 HEXADECIMAL/OCTAL CONVERSION

To convert a hexadecimal number to octal, proceed as follows:

1. Convert the hex number to binary.
2. Partition the binary number in 3-bit groups.
3. Write the octal equivalent for each 3-bit group.

To convert an octal number to hexadecimal, proceed as follows:

1. Convert the octal number to binary.
2. Partition the binary number in 4-bit groups.
3. Write the hex equivalent for each 4-bit group.

INDEX

DX11-B System 360/370
Channel to PDP-11 Unibus
Interface Maintenance Manual

DEC-11-HDXBA-B-D

READER'S COMMENTS

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
146 Main Street
Maynard, Massachusetts 01754

DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASSACHUSETTS 01754