Digital Equipment Corporation
Maynard, Massachusetts

digital

decsystem10

Maintenance Manual

DX10 DATA CHANNEL

EK-DX10-MM-001

decsystem10

DX10 DATA CHANNEL MAINTENANCE MANUAL

digital equipment corporation • marlborough, massachusetts

1st Edition September 1975

The material in this manual is for informational purposes and is subject to change without notice.

Digital Equipment Corporation assumes no responsibility for any errors which may appear in this manual.

Printed in U.S.A.

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

CONTENTS

Page

CHAPTER 1 DESCRIPTION

CHAPTER 2 INSTALLATION

CHAPTER 3 OPERATION AND PROGRAMMING

CONTENTS (Cont)

CONTENTS (Cont)

Page

ILLUSTRATIONS

ILLUSTRATIONS (Cont)

TABLES

CHAPTER 1
DESCRIPTION

1.1 GENERAL

The DX10 Data Channel (Figure 1-1) is a high data transfer rate, I/O processor used in the DECsystem-10 to transfer data in any of four different formats between the system memory and a TU70 Magnetic Tape Subsystem. Data transfer is carried out over the system Memory Bus; transfer is completely independent of the DECsystem-10 CPU (KA10, KI10 or KL10) and is controlled by a hex-board-mounted PDP-8/A Microprocessor located within the DX10. The TU70 Magnetic Tape Subsystem consists of a DX10 Data Channel and a TX01 Tape Control Unit operating in conjunction with up to eight high-speed (200 in./sec) dual- or treble-density tape drives. The tape subsystem is capable of transferring up to 320K characters/sec between any 7- or 9-track tape drive and DECsystem-10 memory. Once the DX10 is initialized by a CONO instruction from the DECsystem-10 CPU, the PDP-8/A takes over the CPU function; that is, it operates the DX10 as an I/O processor in accordance with the instructions contained in a magnetic tape Channel Program that is stored in DECsystem-10 memory. Once the Data Transfer operation (or Tape Control instruction) has been executed, an interrupt from the DX10 control logic to the PDP-8/A causes the PDP-8/A to store ending status in a specified DECsystem-10 memory location, send a priority interrupt request to the DECsystem-10 CPU, and place the DX10 Data Channel in an idle state to wait for the next CONO instruction from the CPU. [If an error occurred during the operation, the PDP-8/A can be directed to store extended status (i.e., tape unit and control unit information) prior to interrupting the DECsystem-10 CPU.]

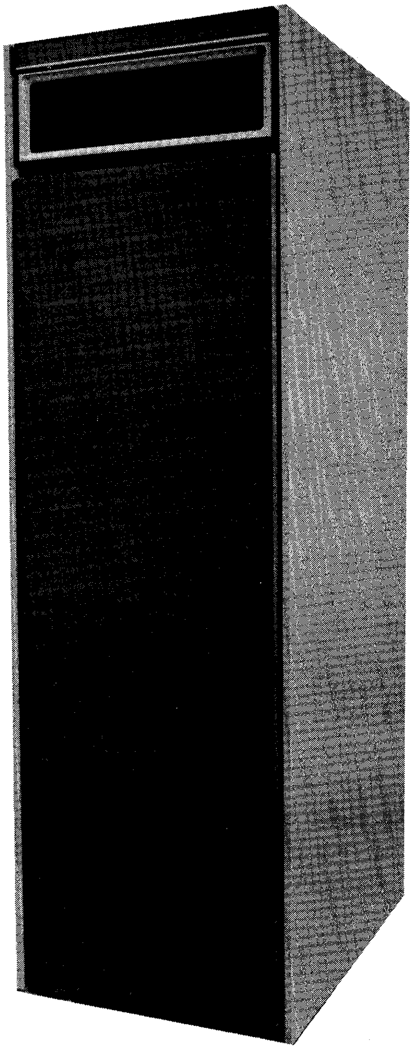
1.2 SYSTEM CONFIGURATION

Two TU70 Magnetic Tape System configurations are available: 1 by 8 or 2 by 16. These represent either a single TU70 Subsystem that drives up to 8 tape drives using a single DX10 Data Channel, or two TU70 Subsystems that drive up to 16 tape drives (8 drives each) using two data channels. In the latter configuration,* which uses tape unit switching arrangements within the tape control units, any of the 16 tape drives may be selected by either DX10 Data Channel. Also, since there are two data channels, the DECsystem-10 can read or write on one tape drive at the maximum data transfer rate (320KB/sec) while simultaneously reading or writing at the same data rate on another drive. Figure 1-2 illustrates the 1 by 8 system configuration. The optional 2 by 16 configuration (Figure 1-3) can also be shared by another CPU† as shown in Figure 1-4.

Two tape drive models are available for use with either system configuration. The model TU70 Tape Unit is a high-density, 9-track drive which offers dual-density, program-selectable operation at either 800 bpi, NRZI (Non-Return to Zero, Inverted) or 1600 bpi, PE (Phase-Encoded). The TU71 is a lower density tape unit that is capable of NRZI reading or recording at any of three program-selectable bit densities: ~200, 556, or 800 bpi. Both tape drives operate at 200 in./sec and may be intermixed in any combination of units. Both of the described system configurations are available for either 50 Hz or 60 Hz operation.

*Available on quotation.

†Not initially available; planned for future use.



7513 7

Figure 1-1 DX10 Data Channel

1.3 OPERATING MODES

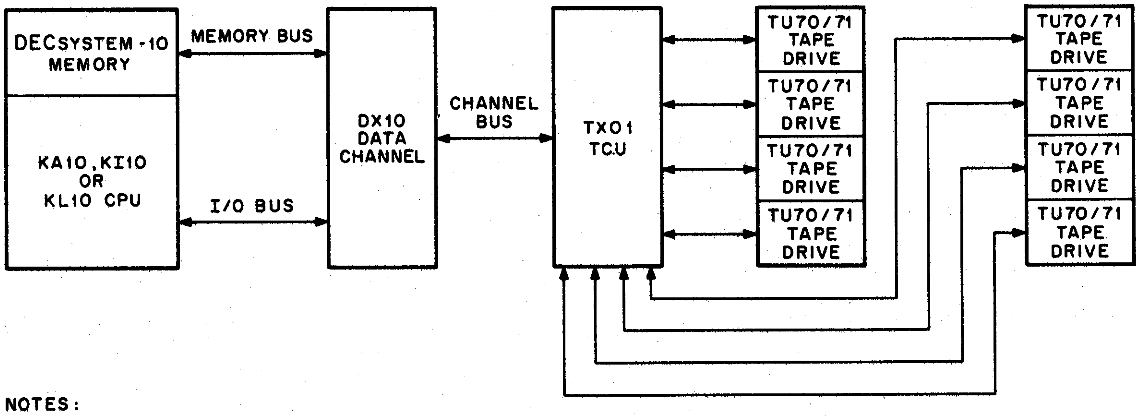
The DX10 I/O processor operates in any of three different modes:

1. Channel Control
2. Device Control
3. Data Transfer

Each operating mode is controlled by a corresponding type of channel instruction:

1. Channel Control
2. Device Command
3. Data Transfer

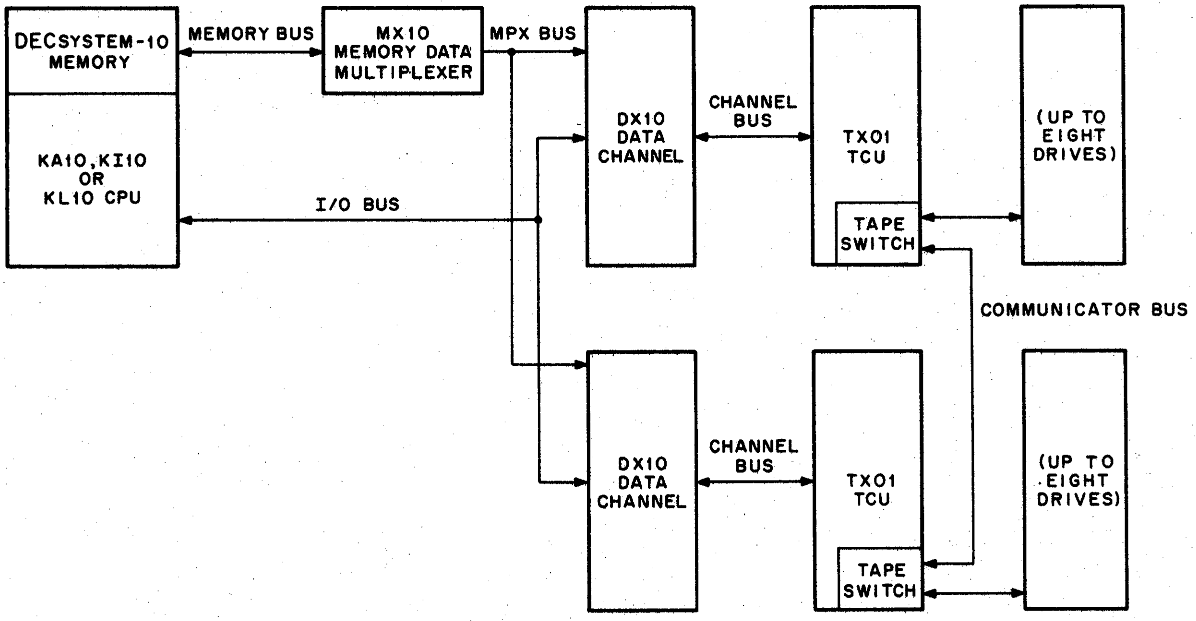
Each of these instructions is discussed in detail in Chapter 3. In addition, any one of four separate data modes (or formats) may be used, depending on the type of drive selected and the operation(s) being performed.



NOTES:
1. Drives are shown attached to radial bus
2. Drives can be 9-track or 7-track head configuration
3. 7-track drives are 200/556/800 BPI, treble-density; 9-track drives are 800/1600 BPI, dual-density
4. All drives operate at 200 IPS, with 625 IPS rewind speed

10-1372

Figure 1-2 Standard 1 × 8 Configuration



10-1373

Figure 1-3 Optional 2 × 16 Configuration

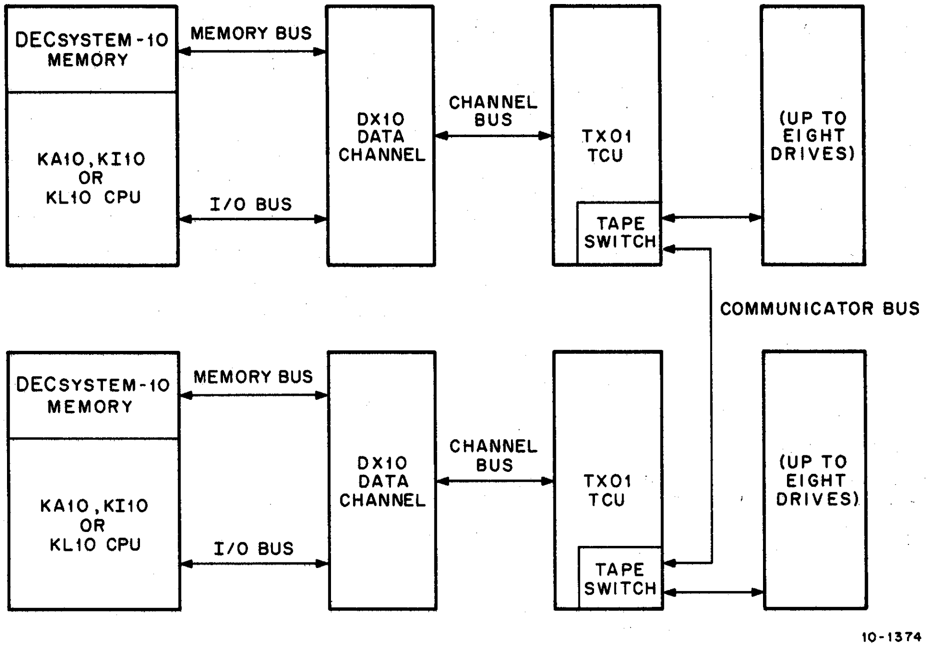


Figure 1-4 Future CPU-Shared, 2 × 16 Configuration

1.4 CHARACTERISTICS

The DX10 Data Channel (Figure 1-5) consists of large, M-series printed circuit boards plugged into two DECsystem-10 unit logic panels mounted in a standard, H950-A 19-in. equipment rack. Two each, input and output I/O and Memory Bus quick-latch connectors, a Channel Bus quick-latch connector, and associated I/O Bus Driver and Receiver (M664 and M564) modules are also included. Figure 1-6 depicts the locations of the operating modules. In addition to pluggable modules, the DX10 contains a front-mounted maintenance panel (Figure 1-7), an 861-C Power Control, and three power supplies: two H740-D (+5 V, -15 V) supplies and an H716-B (-5 V) supply. The 861-C Power Control allows the DX10 to be powered up or down by remote control from the DECsystem-10 Power Control Bus via the power control bus cable.

The physical and operating characteristics of the DX10 follow.

Physical

Dimensions	
Width	22 in. (0.56 m)
Height	72 in. (1.83 m)
Depth	30 in. (0.76 m)
Weight	350 lb (160 kg)
Required Access Clearance	
Front	See Figure 2-1
Rear	See Figure 2-1

Power Requirements

Input Voltage	110/220 V, 50/60 Hz
Current Required	5 A at 110 V 2.5 A at 220 V

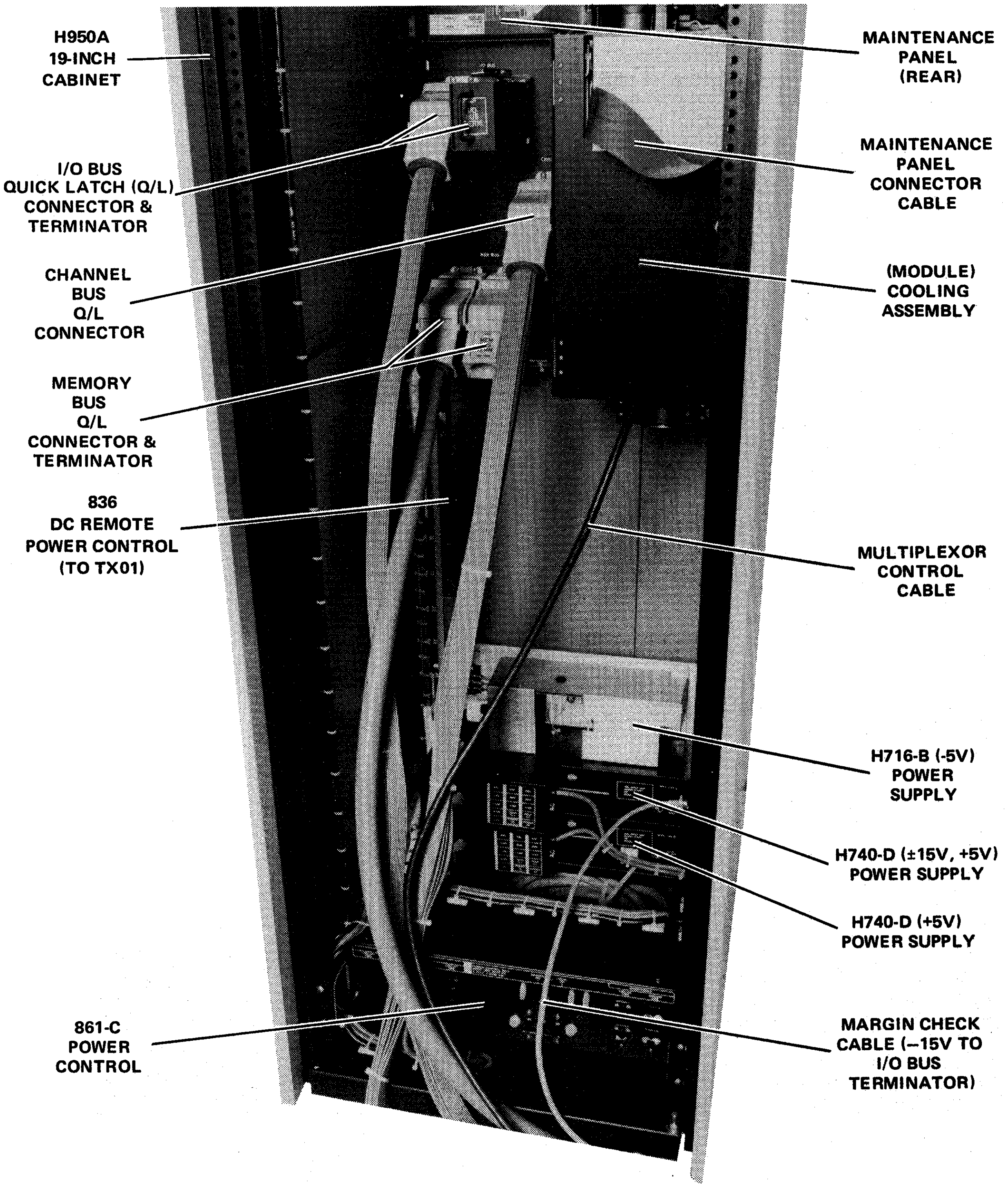

H950A 19-INCH CABINET
I/O BUS QUICK LATCH (Q/L) CONNECTOR & TERMINATOR
CHANNEL BUS Q/L CONNECTOR
MEMORY BUS Q/L CONNECTOR & TERMINATOR
836 DC REMOTE POWER CONTROL (TO TX01)
861-C POWER CONTROL
MAINTENANCE PANEL (REAR)
MAINTENANCE PANEL CONNECTOR CABLE
(MODULE) COOLING ASSEMBLY
MULTIPLEXOR CONTROL CABLE
H716-B (-5V) POWER SUPPLY
H740-D (±15V, +5V) POWER SUPPLY
H740-D (+5V) POWER SUPPLY
MARGIN CHECK CABLE (–15V TO I/O BUS TERMINATOR)
7513-1

Figure 1-5 DX10 Physical Layout

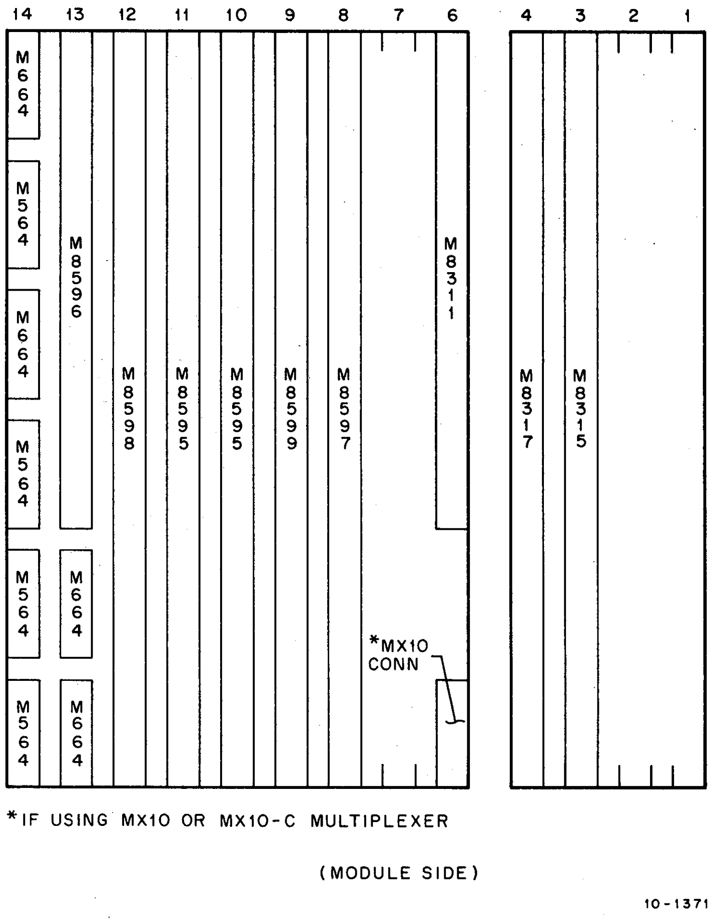


Figure 1-6 DX10 Module Locations

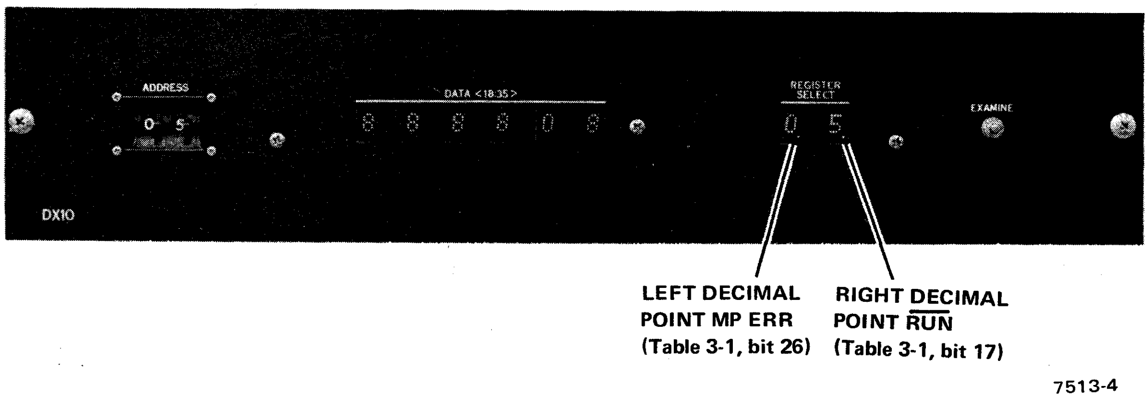

ADDRESS
DATA <18:35>
REGISTER SELECT
EXAMINE
DX10
LEFT DECIMAL POINT MP ERR (Table 3-1, bit 26)
RIGHT DECIMAL POINT $\overline{\text{RUN}}$ (Table 3-1, bit 17)
7513-4

Figure 1-7 DX10 Maintenance Panel

Environmental Requirements (Class A device*)

Ambient Temperature	15° to 32°C(59° to 90°F)
Relative humidity	20 to 80%

Operation

Word Length	36-bit data word

Data Modes

Mode	Use	Data Format
Byte (IBM-Compatible)	9-track only	Four 8-bit bytes transferred per 36-bit word. Four remaining bits ignored.
Sixbit	7-track only	Six 6-bit bytes transferred per 36-bit data word. Transfers 6 characters at a time (minus parity) from 7-track tape into memory.
Core Dump	9-track only	Four 8-bit bytes and one 4-bit byte transferred per 36-bit word. Four-bit byte represents four least significant channels on tape.
ASCII	9-track only	Five 7-bit bytes and one additional bit transferred per 36-bit word. Each byte stored in seven least significant tracks, with most significant track left blank (except for 5th byte). Additional bit (bit 35 of 36-bit memory data word) stored (recorded) in MSB track of 5th byte.

Track Format (Tape Units)

7-Track (TU71)	Industry standard
9-Track (TU70)	Industry standard

Recording Density (Tape Units)

TU70	800 bpi NRZI; 1600 bpi PE
TU71	200, 556, 800 bpi NRZI

Data Transfer Rate (Per System)

W/TU70	160/320 KB/sec
W/TU71	40/111/160 KB/sec

Interblock Gap (Tape Units)

TU70	0.6 in.
TU71	0.75 in.

Data Transfer Control	Remotely controlled by PDP-8/A Microprocessor located within DX10 Data Channel.

*Environmental requirements for the magnetic tape used with TU70/TU71 Tape Drives are more stringent than those required for DX10 operation. These are on the order of 60° to 80° F ambient temperature and 40 to 60% humidity.

1.5 REFERENCE DOCUMENTS

The following documents contain information supplementary to that contained in this manual:

PDP-8/A Miniprocessor Handbook (90P230)

PDP-8/E, PDP-8/M, and PDP-8/F Small Computer Handbook, 1973 Edition (90P245)

DECsystem-10 System Reference Manual (DEC-10-HGAE-D)

DX10 Customer Print Set

KL10 Power Distributon System Functional Description (EK-KLPWR-TM-001)

STC Field Engineering Theory of Operation Manual – Magnetic Tape Units (CTM 2400-01-1)

STC Illustrated Parts Catalog – 3400 Tape Units (9014)

STC Field Engineering Installation Manual – 3800-II/III Tape Subsystems (CSI-3800-II/III-01)

STC Subsystem Program for Analysis and Repair (SPAR) ASM (9035)

STC SPAR Diagnostic Document (27646)

STC Tape Unit* and Tape Control Schematics

MAINDEC-10-DCDXA-( )-DX10 Basic DX10 Diagnostic Program (Part 1)

MAINDEC-10-DCDXB-( )-DX10 Basic DX10 Diagnostic Program (Part 2)

MAINDEC-10-DCTUA-( )-TU70 Basic TU70 Magnetic Tape Subsystem Diagnostic Program

MAINDEC-10-DCTUB-( )-TU70 TU70 Magnetic Tape Subsystem Reliability Diagnostic Program

MAINDEC-10-DXMPA-(A)-DX10 Microcode Document

MAINDEC-08-DJKKA-( )-PDP8 PDP-8/A CP Test

MAINDEC-08-DJMSA-MS8-A MOS Memory Test

MAINDEC-08-DJEXB-2-32K RAM Reference Instruction Exerciser

STC Control Unit 3800 III – Theory of Operation Manual (9127)

*Each set of schematics is serialized with unit.

CHAPTER 2
INSTALLATION

2.1 GENERAL

The DX10 Data Channel is a functional subunit of the TU70 Magnetic Tape Subsystem; therefore, the installation of the DX10 comprises only a part of the entire subsystem installation. This chapter describes the installation procedures related to the positioning, powering up, and initial checkout of the DX10 itself. TX01 Control Unit and/or TU70 and TU71 Tape Unit installation procedures are described in the appropriate service manuals for those units.

The diagnostic programs specified in this chapter should be run only after all the units of the TU70 Magnetic Tape Subsystem have been properly installed and have been mechanically and electrically checked for proper operation as prescribed in the applicable checkout procedures.

2.2 UNPACKING AND INSPECTION

The DX10 is bolted to a pallet when it is delivered to a DECsystem-10 site. For initial positioning, unbolt the DX10 from the pallet, wheel it into place adjacent to the TX01 Control Unit, and level it using the leveling pads provided at the base of the cabinet.

Before installation and leveling, however, conduct a thorough visual inspection of the DX10 to ensure that no physical damage was incurred during shipment; cabinet doors and panels, connectors, the cooling assembly, pluggable module connections, and cable connections should be thoroughly checked for dents, broken fasteners, looseness, broken connectors, etc. Any damage or excessive looseness should be carefully recorded and reported on the appropriate forms to Digital Equipment Corporation, Marlboro, Mass. (Attn: Field Service Dept.) and to the delivering commercial carrier.

After inspecting the DX10 (cabinet, wiring, connectors, etc.), ensure that the additional equipment listed in Table 2-1 has been received and is also in good working condition. Any equipment damage should be similarly recorded and reported.

2.3 INSTALLATION

If no damage or only minor damage (not serious enough to prevent installation) is found during unpacking and inspection, proceed as follows:

NOTE

The following procedure assumes that the DX10 is being installed in an operating, powered-up DECsystem-10 computer installation.

1. Roll the DX10 cabinet (on its casters) into position.

2. Using the leveling legs provided at the bottom of the unit, adjust the DX10 cabinet for proper height and alignment with the adjacent cabinet. Figure 2-1 illustrates the DX10 cabinet dimensions and indicates the clearances required after the cabinet has been positioned and leveled.

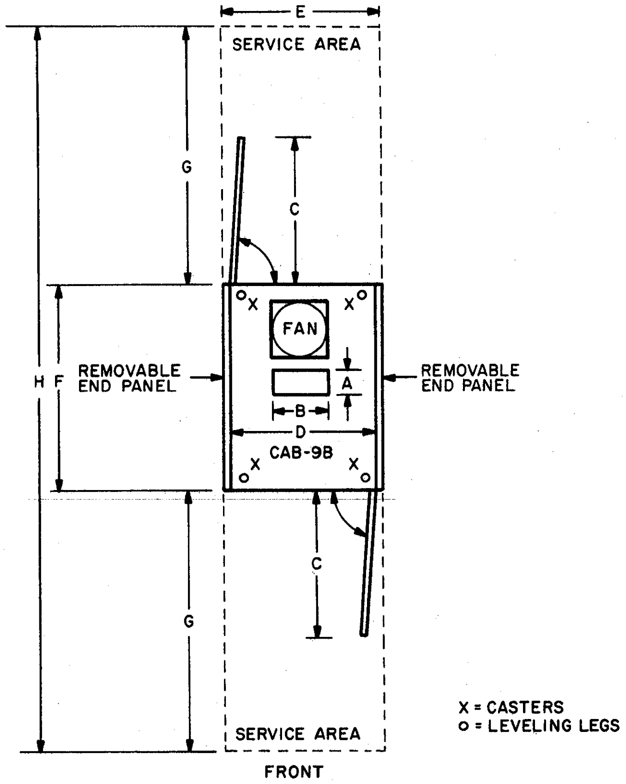
Table 2-1
Equipment Supplied

Quantity	Item	Document, Type or Part No.
Software		
1	DX10 Diagnostic Kit and Write-Ups: MAINDEC-10-DDDXA-DX10 MAINDEC-10-DDDXB-DX10 MAINDEC-10-DDTUA-TU70 MAINDEC-10-DDTUB-TU70 MAINDEC-10-DXMPA-DX10 MAINDEC-08-DJKKA-PDP8-A MAINDEC-08-DJMSA-MS8-A MAINDEC-08-DJEXB-2-32K	
1*	SPAR Diagnostic Magtape†	9035
1*	SPAR Diagnostic Write-Up†	27646
Prints		
1	DX10 Print Set	Per Drawing B-DD-DX10-0
1	TX01 Logic Print Set (Vol. 1,2,3)†	
1	TU70/71 Logic Print Set† (serialized)	
Manuals		
1	DX10 Maintenance Manual	EK-DX10-MM-001
1	PDP-8/A Miniprocessor Handbook	90P230
1	TX01 Maintenance Manual†	9127
1*	Tape Drive Compatibility List†	16000
1*	TU70 Maintenance Manual†	4361
Hardware		
1	Channel Bus Cable (prespecified length)	70-10078-XX
1	EPO Plug Cable (prespecified length)	70-104083-XX
1	I/O Bus Cable (prespecified length for KA10 or KI10/KL10)	BC10K-XX or BC10-JXX (continued)

Table 2-1 (Cont)
Equipment Supplied

Quantity	Item	Document, Type or Part No.
1	Memory Bus Cable [prespecified length for old (pre-MF style) or new memories]	BC10K-XX or BC10JXX
1*	Memory Bus Terminator	H688
1*	TU70 Signal Cable (prespecified length)	17-00041-XX
1*	TU70 Power Cable (prespecified length)	17-00040-XX
1	Tag Terminator	12-12172
1	Bus Terminator	12-12171
1	TX01 Kickplate Assembly Kit†	(TBS)
1*	TU70 Kickplate Assembly†	(TBS)
1	Margin Check Cable (prespecified length)	70-10375-XX‡
1	Remote Turn-On Cable (prespecified length)	70-10483-XX‡
1	MX10 Cable (prespecified length if system has MX10-C or MX10)	70-05541-XX‡
Miscellaneous Items		
1 can	Tape Cleaner	29-15199
1 pkg	Swabs with Sponge Rubber Tips (Tex-Wipe)	N/A
1*	Blank Magtape	18-9543
1	Cabinet Filter (spare)	H950-S

*Per subsystem.
†Supplied by vendor (STC)
‡XX indicates length in feet.



DIMENSIONS	A	B	C	D	E	F	G	H
METERS	0.09	0.18	0.49	0.52	0.54	0.72	0.90	2.50
INCHES	3.5	7.0	19.3	20.3	22.0	29.0	36.0	100

Figure 2-1 DX10 Installation: Dimensions and Access Clearances

3. Connect the 861-C Power Control input power cable to a 120 Vac, 50–60 Hz power source. Energize the DX10 cabinet by placing the 861-C Power Control circuit breaker in the ON position.

4. Set the 861-C REMOTE ON/OFF/LOCAL ON toggle switch to the LOCAL ON position.

5. Determine that the cabinet blowers are operating properly and that the +5 V, -5 V, and -15 V operating voltages are present on the backplane power connection pins.

6. Perform the maintenance panel lamp test as follows:

   a. Using the ADDRESS thumbwheel switches located on the maintenance panel, select any register address between 40 and 77.

   b. Observe that each of the six DATA (18:35) and two REGISTER SELECT LED-type indicators display the number 8.

7. Perform the IBUS display test as follows:

   a. Select register address 17 on the ADDRESS switches.

   b. Press the EXAMINE pushbutton on the maintenance panel.

   c. Observe that all 0s are displayed on the DATA (18:35) LEDs and that 17 is displayed on the REGISTER SELECT LEDs.

CAUTION

Repair any faults encountered in steps 5–7 before proceeding to step 8.

8. Install Memory Bus and I/O Bus terminator quick-latch connectors, if required.

NOTE

Ensure that the TX01 Control Unit and the TU70 and/or TU71 Tape Drives have been properly installed and are operating before proceeding with the next step.

9. Connect the I/O Bus, Memory Bus, and Channel Bus cables between the DX10 and the DECsystem-10 CPU, memory (or multiplexer), and the TX01 Control Unit.

10. Connect the DECsystem-10 Multiplexer control cable, if required.

NOTE

Figure 2-2 depicts a typical DIP switch of the type used on the M8597, M8311, M8315, and M8317 modules. In steps 11 and 12, individual switch numbers and positions are as shown in the diagram.

Setting a switch actuator to the OFF position (the switches have rocker-type actuators) is equivalent to setting a 0 into that bit position. A 1 is set into a bit by placing the switch actuator in the ON position.

11. Ensure that all of the switches listed in Table 2-2 are set to the indicated positions.

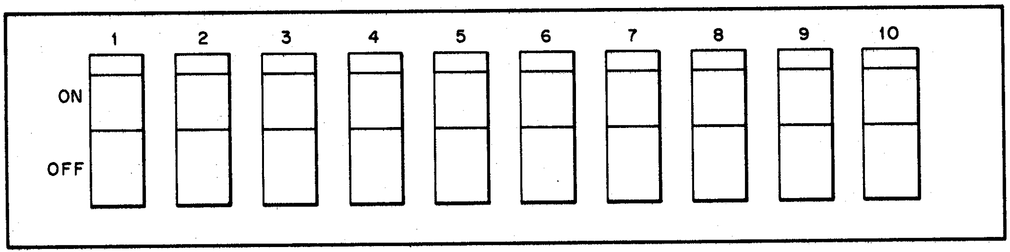


Figure 2-2 DIP Switch Layout

Table 2-2
Required Initial DIP (Dual, In-Line Package)
Switch Settings (0 = OFF, 1 = ON)

Module	Bit No.	Function	DIP Loc.	Switch No.	Required Setting at Installation
M8597	IBUS (IB)	Feature Register			
	18	Not Used	E5	4	OFF
	19	Memory Bus Configuration	E5	3	Para. 2.2, Step 12
	20	1st (High Order) 2nd (Low Order) ECO level (00–79) in BCD format	E5	1	X
	21		E5	2	X (Each ECO
	22		E4	4	X specifies a
	23		E4	9	X new setting)
	24		E4	9	X
	25		E4	8	X
	26		E5	8	0 = M8599 Rev D or below 1 = M8599 Rev D or above.
	27	1st digit 2nd digit 3rd digit Unit serial no. (000–199) in BCD Format	E5	9	
	28		E5	10	Enter serial no. of unit in ascending (1st, 2nd, 3rd, digit) order. For example (unit 39) = 0 0 0 1 1 1 0 0 1 1st DIGIT 2nd DIGIT 3rd DIGIT
	29		E4	10	
	30		E4	5	
	31		E5	6	
	32		E5	5	
	33		E4	7	
	34		E4	6	
	35		E5	7	
	SKIP SW	Not used	E4	2	OFF
	Spare	Not connected	E4	1	OFF
M8311		Switches 1–6 and 10 ON. Switches 7–9 OFF.			
M8315		Switch 7 ON. Switches 1–6 and 8 OFF.			
M8317			E79	S1-1	ON
				S1-2	ON
				S1-3	ON
				S1-4	OFF
				S1-5	OFF
				S1-6	OFF
				S1-7	OFF
				S1-8	ON
			E87	S2-1	OFF
				S2-2	OFF
				S2-3	OFF
				S2-4	OFF
				S2-5	ON
				S2-6	ON
				S2-7	ON
				S2-8	ON

12. Figures 2-3 through 2-7 illustrate five normal installation configurations for the DX10 using both "old" (18-bit address) and "new" (22-bit address) Memory Buses. If the configuration shown in Figure 2-7 is being used, ensure that E5, switch 3 on the M8597 module is ON* (logical 1) before proceeding with step 13.

13. Mount the DX10 Diagnostic Distribution tape on a TU70 9-track tape drive and ready the drive.

14. Set device code 220_8 (if one DX10 is being used) or 224_8 (if a second DX10 is used) into the READ-IN DEVICE switches on the DECsystem-10 console.

15. Press the READ-IN switch.

16. Observe that the DECsystem-10 prints out diagnostic loading and run procedures on the console teletypewriter.

17. Load and run (in sequence) the diagnostic programs listed below. (Refer to the documentation supplied with the individual diagnostic programs for procedures to be followed in connection with program operation.)

    MAINDEC-10-DDDXA
    MAINDEC-10-DDDXB
    MAINDEC-10-DDTUA
    MAINDEC-10-DDTUB

NOTE

This completes the DX10 installation and functional checkout procedure. Additional testing may be performed, as desired, using the Diagnostic Distribution tape and the associated procedures.

*This switch must be in the OFF position for all other configurations.

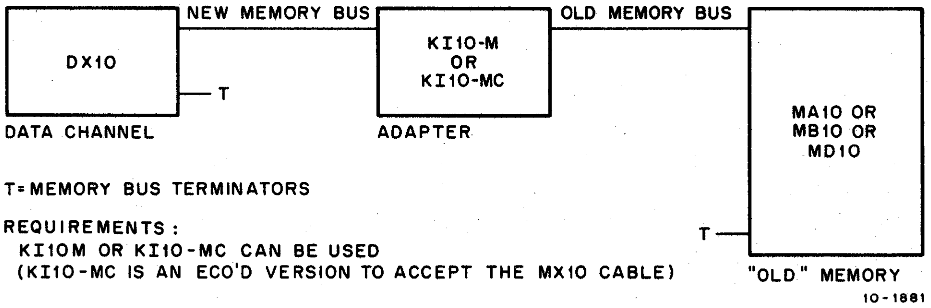


Figure 2-3 Normal DX10 Configuration for Old Memories Not Being Multiplexed

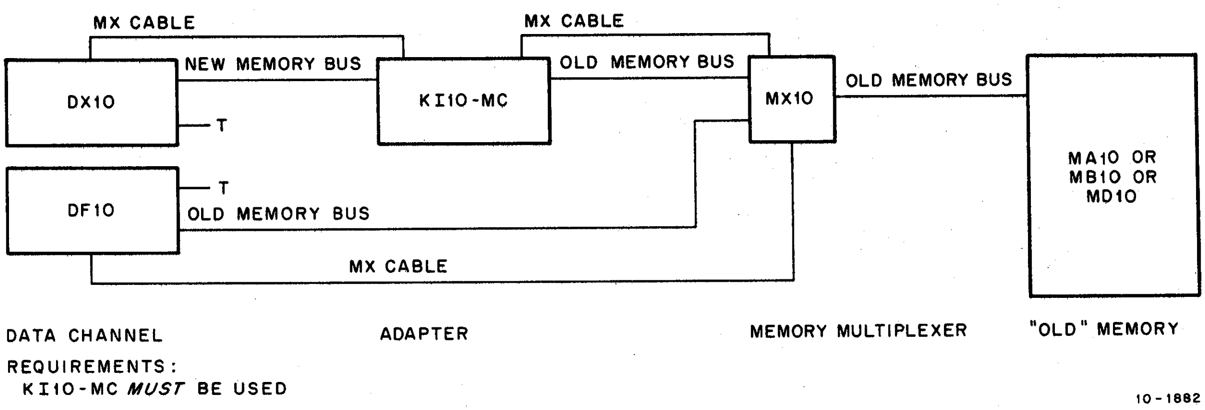


Figure 2-4 Normal DX10 Configuration for Old Memories Being Multiplexed

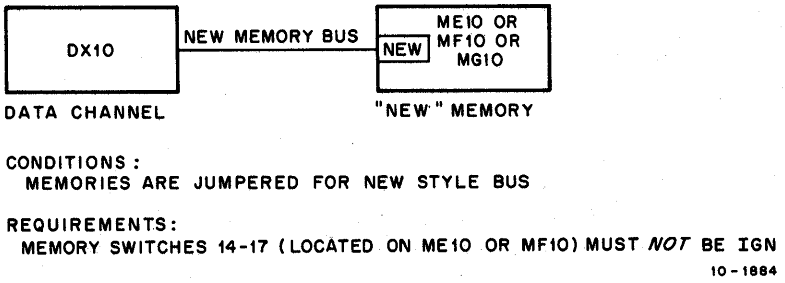

DX10
NEW MEMORY BUS
NEW
ME10 OR
MF10 OR
MG10
DATA CHANNEL
"NEW" MEMORY
CONDITIONS:
MEMORIES ARE JUMPERED FOR NEW STYLE BUS
REQUIREMENTS:
MEMORY SWITCHES 14-17 (LOCATED ON ME10 OR MF10) MUST NOT BE IGN
10-1884

Figure 2-5 Normal DX10 Configuration with New Style Memory Bus (Not Multiplexed)

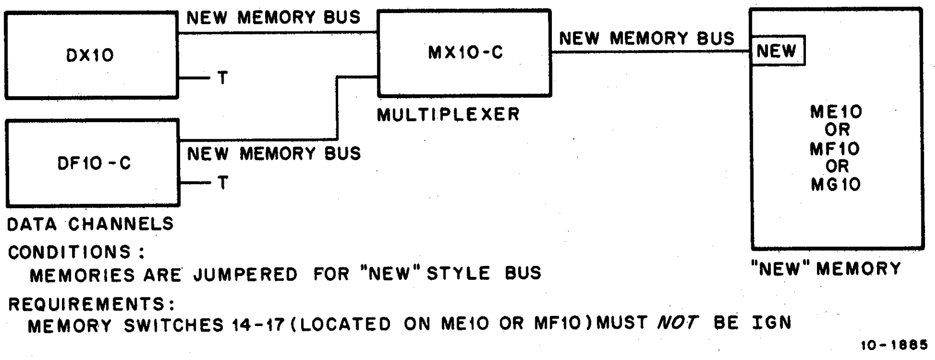
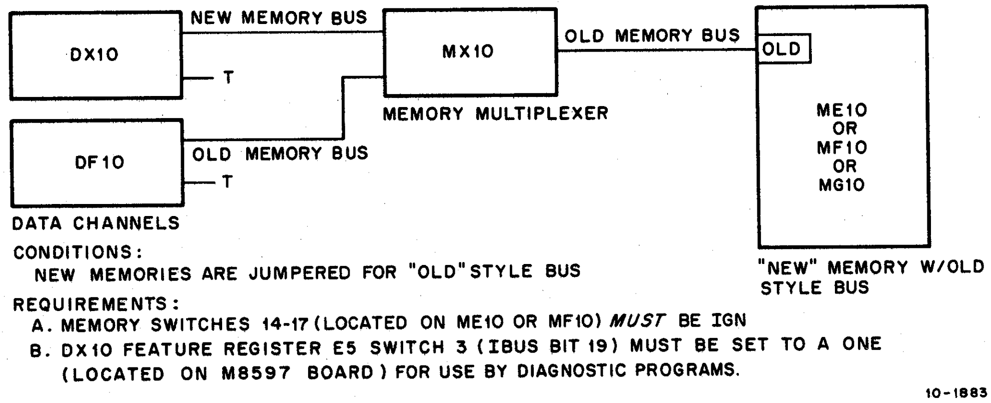


Figure 2-6 Normal DX10 Configuration with New Style Memory Bus (Multiplexed)



CONDITIONS:
NEW MEMORIES ARE JUMPERED FOR "OLD" STYLE BUS

REQUIREMENTS:
A. MEMORY SWITCHES 14-17 (LOCATED ON ME10 OR MF10) *MUST* BE IGN
B. DX10 FEATURE REGISTER E5 SWITCH 3 (IBUS BIT 19) MUST BE SET TO A ONE (LOCATED ON M8597 BOARD) FOR USE BY DIAGNOSTIC PROGRAMS.

Figure 2-7 DX10 Configuration for New Memories Being Multiplexed with Old Style Memory Bus

CHAPTER 3
OPERATION AND PROGRAMMING

3.1 GENERAL

The DX10 is a programmable data channel (PDC) that operates automatically under program control (once it is initialized by a CONO instruction) from a Channel Program located in DECsystem-10 memory. The Channel Program is fetched and executed by a PDP-8/A Microprocessor located within the DX10. The DX10 is an automatic device; therefore, no operating controls or indicators exist. A special maintenance panel is provided inside a hinged door at the front of the cabinet to allow examination of internal registers, etc. by maintenance personnel.

3.1.1 Operating Procedures

The DX10 Data Channel may be manually started from the DECsystem-10 operator's console using the standard read-in procedures outlined in the *DECsystem-10 System Reference Manual*. Procedures associated with tape reel mounting, manually loading or unloading tape drives, and the use of off-line diagnostic procedures using the TX01 Tape Control Unit and the SPAR diagnostic program are fully covered in the appropriate STC Control Unit or Magnetic Tape Unit Theory and Operation Manuals.

The DX10 Data Channel, under program control from the DECsystem-10 memory, transfers data between DECsystem-10 memory and the TU70 Magnetic Tape Subsystem independently of currently operating DECsystem-10 programs.

Data transfer is accomplished over the DECsystem-10 Memory Bus under control of a Channel Program assembled in DECsystem-10 memory by the system monitor. Because control of the Channel Program is exercised by the DX10, the data channel operates as an I/O processor that only interrupts the DECsystem-10 CPU at the end of the Channel Program to indicate the completion of the data transfer (control, etc.) operation and to inform the CPU that unit and channel status have been stored at a preassigned location in memory.

The DX10 channel instruction repertoire includes Write, Read, and Read Backward data transfer functions and Device Control, Test I/O, and Sense (Read Status) control functions. Each of these data transfer and control functions is further subdivided into specific motion, nonmotion, and mode control commands which are sent as 8-bit command bytes over the Channel Bus to the TX01 Tape Control Unit during execution of the Channel Program by the DX10 Data Channel.

3.1.2 Device Availability

The TU70 Subsystem presents (via a Device Status word stored in the program area) various levels of availability to the Channel Program as follows:

1. If only a tape drive is busy, any other tape drive is available to the Channel Program.
2. If the control unit is busy, only the data channel is available (for channel control instructions).
3. If the data channel is busy, the entire magnetic tape subsystem is unavailable for other tasks.

Channel and tape unit switches are available (by special quotation) to provide multiple data paths to any given tape drive, thus making the tape drive more accessible for high data transfer rate applications (Paragraph 1.2).

3.1.3 Data Transfer

Data transfer between DECsystem-10 memory and the TU70 Magnetic Tape Subsystem is accomplished in the following manner:

1. During system initialization or when requested to do so by a Device Driver Program, the DECsystem-10 monitor assembles a DX10 data transfer management (channel) program in memory. This Channel Program is designated by an operating program (by loading its starting address into the data channel's Initial Channel Program Counter) whenever the operating program requires the assembled sequence of magnetic tape operations to be performed.

   NOTE

   Before assembling the Channel Program, the monitor (immediately after power on) loads the PDP-8/A microcode into PDP-8/A control storage (4K MOS memory) and initializes the channel. This starts the PDP-8/A Microprocessor, which enters an idle loop, awaiting the first operating program I/O control instruction (CONO).

   Because the DX10 uses a volatile (MOS) memory for control storage, the monitor must also reload the microcode during system recovery operations following a power loss or a microprocessor (PDP-8/A) HALT.

2. When the current program in DECsystem-10 memory wants to perform a magnetic tape operation, it issues a CONO instruction with the DX10 device select code (220 or 224) over the system I/O Bus. Using either the same CONO instruction, or a subsequent one, the Device Driver Program loads the Initial Channel Program Counter (ICPC) in the DX10 Channel Command Register with the starting address of the Channel Program that the monitor assembled in DECsystem-10 memory.

3. The PDP-8/A Microprocessor then transfers the contents of the ICPC Register to the Channel Program Counter (CPC) and fetches the first instruction in the Channel Program from DECsystem-10 memory over the system Memory Bus and places it in the PDP-8/A register (8R) file. (During this sequence, any fault that occurs causes the PDP-8/A to send a program interrupt to the CPU via the I/O Bus.)

   NOTE

   Ordinarily, the first instruction in a Channel Program is a JMP instruction to another address. This is because a JMP instruction can specify a 22-bit address (full memory); a CONO only loads a 9-bit address into the ICPC.

4. Three types of Channel Program instructions are used in the DX10: Device Command, Data Transfer, and Channel Control. The PDP-8/A checks to determine that the first (non-JMP) Channel Program instruction is a Device Command. If it is not, a sequence error is indicated. If the first instruction is a Device Command, the PDP-8/A issues a tape drive address and a Device Command to the tape control unit (TCU). This causes the TCU to select the tape drive and initialize it in preparation for the operation specified by the Channel Program. (For details regarding execution of the Channel Program, refer to Paragraph 3.4.)

5. Following initialization of the selected tape drive (start tape motion), a Data Transfer instruction is issued by the Channel Program to initiate the movement of data between the selected drive and DECsystem-10 memory. This may include one or several Data Transfer instructions. If an error occurs during instruction execution, the PDP-8/A stores status and interrupts the CPU. If the ICPC+3 location is negative, it is used to provide byte count and address information and causes the PDP-8/A to perform an extended status store operation before the interrupt (with an error flag set) is sent to the CPU.

6. Following a normal data transfer (one without an error), normal status information is transferred to DECsystem-10 memory. This status transfer is accomplished via a Channel Control STORE instruction. When status has been stored in memory, the PDP-8/A notifies the CPU of status availability by means of a STAT AVAIL interrupt.

NOTE

Channel Program instructions are discussed in detail in Paragraph 3.4. The PDP-8/A microcode is discussed in Chapter 4.

3.1.4 Channel Termination

If there are no further channel instructions and the PDP-8/A has stored device and channel status in DECsystem-10 memory, the PDP-8/A notifies the CPU, disconnects the TX01 Control Unit, places the DX10 in an idle state, and enters an idle loop, waiting for the next CONO instruction over the I/O Bus from the operating program. To terminate the channel, the operating program must issue a CONO instruction with a 1 in bit 29 to clear the STAT AVAIL flag, thereby removing the CPU interrupt from the assigned DX10 I/O Bus Priority Interrupt (PI) line. Optionally, the program could reset the priority assignment to 0 (using a CONO) and the interrupt would be removed (because PI0 in the DX10 is not connected to the I/O Bus).

3.2 GENERAL PROGRAMMING

The DX10 utilizes three discrete programming levels in processing and transferring data between the DECsystem-10 memory and individual TU70/TU71 Tape Units:

1. DECsystem-10 Operating System or Executive Mode Program – This is the program currently being executed by the DECsystem-10. The operating system or an executive mode program may require the use of the TU70 Magnetic Tape Subsystem. To accomplish this, a special executive mode Device Driver Program issues the appropriate I/O instruction to the DX10. This includes CONO, CONI, DATAO, and DATAI instructions which are sent by either the KA10, KI10, or KL10 CPUs to the DX10 over the system I/O Bus. These instructions are used to control the DX10 registers and/or the PDP-8/A Microprocessor.

2. TU70 Channel Program – The TU70 Channel Program is unique to the TU70 Subsystem. It is assembled by the DECsystem-10 monitor and is resident in DECsystem-10 memory. The Channel Program is used by the PDP-8/A Microprocessor to control data transfers between DECsystem-10 memory and the TU70 Magnetic Tape Subsystem via the TU70 Channel and DECsystem-10 Memory Buses. Three types of instructions are contained in the Channel Program: Channel Control, Device Command, and Data Transfer.

These Channel Program instructions are sequentially fetched from DECsystem-10 memory by the PDP-8/A Microprocessor (once it has been initialized by a CONO from the operating program). The necessary microinstructions are then issued by the PDP-8/A for device selection, status or data transfer, and initialization or termination of the Channel Bus.

3. PDP-8/A Microcode – This is a PDP-8/A program which is loaded into the microprocessor memory (also referred to as control storage) either by a bootstrap loader in response to an RDI console command or by the monitor during system initialization or system recovery. It is used by the PDP-8/A to control execution of the Channel Program. In addition to the PDP-8/A instruction set, which it uses in performing internal calculations and logical operations associated with channel instruction decoding and execution, the microprocessor issues a set of IOT instructions to the DX10 decoder logic to control the data transfer and control functions associated with Channel Program commands.

3.3 EXECUTIVE MODE PROGRAMMING

Once the DX10 has been initialized, it operates automatically under control of the Channel Program and the PDP-8/A Microprocessor. Before it can be started, however, the Initial Channel Program Counter (ICPC), located in the DX10 Channel Command Register (CCR), must be loaded by the operating program. The ICPC is a 7-bit register used to designate the DX10 Channel Program starting address. Because it is a part of the Channel Command Register, it can be loaded with the same CONO instruction used to start the DX10.

3.3.1 CONO/CONI Instruction

The DECsystem-10 ordinarily controls the DX10 by means of a single CONO instruction issued over the I/O Bus. The initial conditions E specified by the CONO instruction (refer to CONO description in DECsystem-10 System Reference Manual) are transferred into the Channel Command Register (CCR) located within the DX10. This 18-bit "register" actually consists of a 7-bit address register (ICPC), a 3-bit (Priority Interrupt Assignment) register, and several discrete flip-flops, each of which performs a separate control function. The PDP-8/A Microprocessor uses the CCR to communicate with the DECsystem-10 CPU via the system I/O Bus. Some of its bits can be set only from the DECsystem-10 CPU by a CONO instruction and can be cleared only by the PDP-8/A Microprocessor. Others can be set only by the PDP-8/A (or DX10 control logic) and cleared by the CPU. The CCR can be read by either device; it is set (or cleared) from the DECsystem-10 by a CONO and is read by a CONI. The bit assignments for the TU70 CONO/CONI instructions are shown in Figure 3-1.

NOTE

Bits marked by asterisks in Figure 3-1 (if set in the CCR) are cleared by a CONO with the corresponding bit(s) set to 1.

Each of the bit assignments corresponds, one-for-one, with the individual bits in the CCR.* The Microprocessor Error (MP ERR), Memory Parity Error (MEM PAR ERR), Nonexistent Memory (NXM), and Status Available (STAT AVAIL) bits in the CCR, when set by either the PDP-8/A or DX10 error detection logic, generate an interrupt to the CPU via one of the seven I/O Bus Priority Interrupt (PI1-PI7) lines.

The Channel Command Register is located on the M8598 module and comprises part of the IOC (Input/Output Control) logic. Figure 3-2 shows the output signal assigned to each register flip-flop and its relationship to the corresponding CCR bit assignment. Table 3-1 lists the function of each CONO/CONI bit in loading, reading or clearing the CCR.

*Except bits 17 and 25. Bit 17 is used to read the microprocessor RUN bit via a CONI. DATAO/DATAI instructions are illegal when this bit is set to 1. Bit 25 does not actually set a bit in the CCR. It is used to specify whether or not the ICPC bits in the CCR are to be loaded with IOB bits 18–24.

NOTE

Bits marked by asterisks in Figure 3-2 generate an interrupt on a PIA line. These bits are set in the CCR by either the DX10 fault detection logic or the PDP-8/A and are cleared by the DECsystem-10 program issuing a CONO with a 1 in the corresponding bit(s).

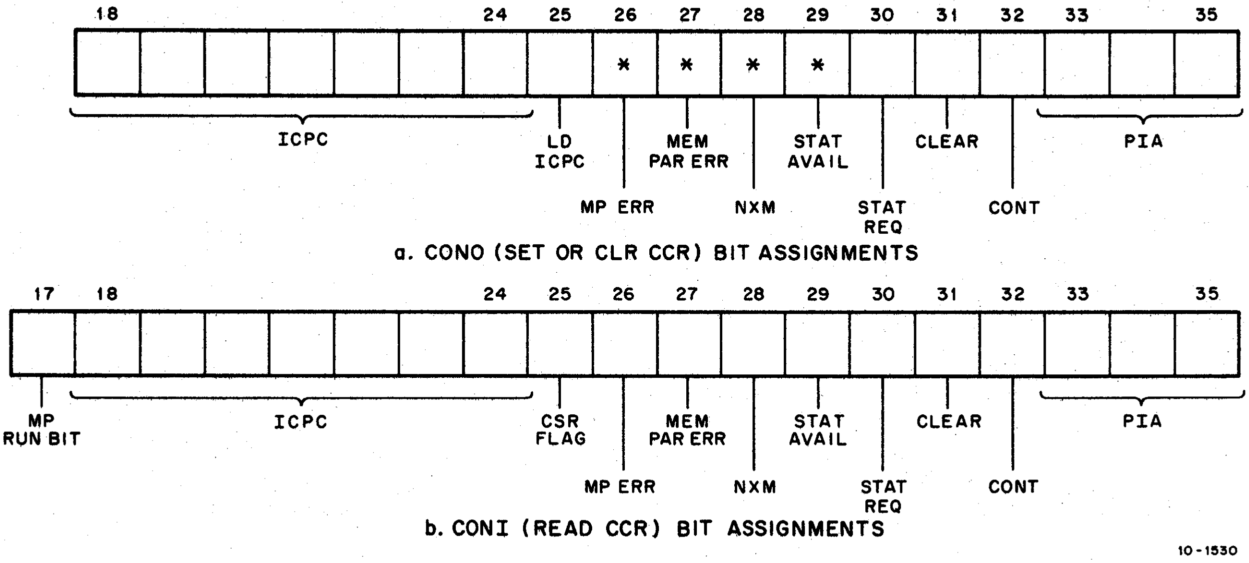


Figure 3-1 TU70 CONO/CONI Bit Assignments

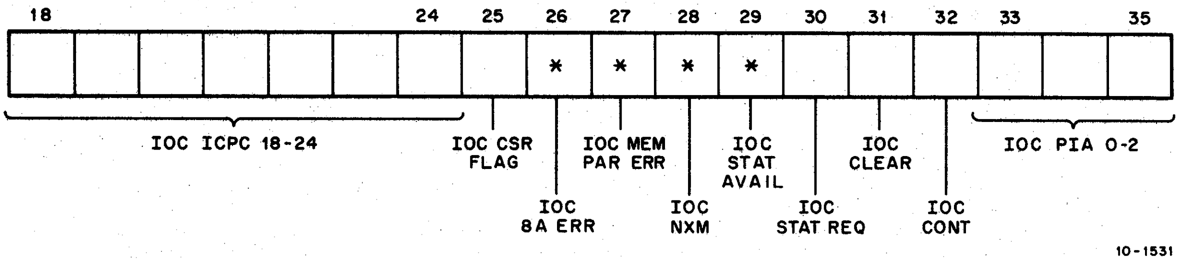


Figure 3-2 Channel Command Register

Table 3-1
Channel Command Register Bit Assignments

I/O Bus No.	I/O Instruction (CPU)	Associated CCR Bits	Bit Description/Action Indicated
18–24	CONO/CONI (Load/Read)	ICPC18–24	Initial Channel Program Counter. Specifies the memory address of the first channel instruction and status storage locations. As shown in Figure 3-3, any address of 0 through 774_8 that is a multiple of 4 can be used. ICPC will be loaded by a CONO only if bit 25 is not set.
25	CONO (Inhibit)	None	No corresponding CCR bit set. Inhibits bits 18–24 of the CONO word from being loaded into the CCR ICPC field. If I/O Bus bit 25 is not set, a CONO will cause the ICPC to be loaded with CONO bits 18–24.
25	CONI (Read)	CSR Flag	A bit of interest to the operating program has been set in the Channel Status Register (CSR). The CSR flag is set by the PDP-8/A and can be cleared by either a system Reset pulse or by a DATAO Clear command from the CPU. It is normally cleared by a 6512 IOT instruction from the microprocessor at the beginning of the next magtape startup routine (refer to Chapter 4). The CSR flag sets Status Available, which generates an interrupt to the CPU. Once STAT AVAIL is set, the microprocessor waits for the CPU to reset it before resuming data processing operations.
26	CONO/CONI (Clear/Read)	MP ERR	Microprocessor error. Occurs if some malfunction prevents the microprocessor from resetting an internal timer. Starting the microprocessor enables the bit. It is automatically set if the MP Timer is allowed to time-out due to failure of the PDP-8/A to issue a Set Timer instruction within a specified period. This indicates that the PDP-8/A may be hung in some type of processing loop. Setting MP ERR causes an interrupt to the CPU. It is cleared by issuing a 1 in CONO bit position 26. (A 0 in this CONO bit position will not affect MP ERR.)
27	CONO/CONI (Clear/Read)	MEM PAR ERR	Memory Parity Error. Sets when a parity error occurs on the Memory Bus. MEM PAR ERR causes an interrupt and stops the channel. A CONO with bit 27 set to 1 clears MEM PAR ERR. If bit 27 is 0, the CONO has no effect.

Table 3-1 (Cont)
Channel Command Register Bit Assignments

I/O Bus No.	I/O Instruction (CPU)	Associated CCR Bits	Bit Description/Action Indicated
28	(CONO/CONI (Clear/Read)	NXM	Nonexistent Memory. The DECsystem-10 memory addressed by the PDP-8/A did not respond with Address Acknowledge within 100 μs. NXM is set automatically if the Memory Timer is allowed to time-out due to ADDR ACK failing to reset MEM CYCLE. An interrupt is generated and the channel becomes idle. A CONO with bit 28 set to 1 clears NXM. If bit 28 is a 0, the CONO has no effect.
29	CONO/CONI (Clear/Read)	STAT AVAIL	Status Available. The PDP-8/A has stored status in the DECsystem-10 memory, and the CPU is to be notified. The CPU is expected to read the memory status locations, then clear this bit. STAT AVAIL is set by the PDP-8/A and is cleared by a CONO with bit 29 set to a 1. If bit 29 is a 0, the CONO has no effect.
30	CONO/CONI (Set/Read)	STAT REQ	Status Request. Set by a CONO with bit 30 set to a 1. STAT REQ causes the PDP-8/A to store status in the DECsystem-10 memory locations shown in Figure 3-4. STAT REQ can be serviced only when the DX10 is idle, so the stored status will be identified as Idle Status. After the status is in memory, STAT REQ is cleared by a 6512 IOT instruction from the microprocessor, and STAT AVAIL sets. The STAT REQ bits can also be cleared as the result of a CPU CLEAR (generated by a DATAO with bit 16 set to 1) or by issuing an I/O Bus Reset pulse. **NOTE** **Generating a CPU CLEAR initializes the DX10.**
31	CONO/CONI (Set/Read)	CLEAR	Clear. Set by a CONO with bit 31 set to 1. When set, all error flags are cleared, and a system reset is issued to the Channel Bus. If Continue is set at the same time, however, the channel is started and the above sequence does not occur. The PDP-8/A clears this flip-flop before returning the DX10 to the idle loop. It is also cleared by a DX10 Initialize command or I/O Bus Reset.

Table 3-1 (Cont)
Channel Command Register Bit Assignments

I/O Bus No.	I/O Instruction (CPU)	Associated CCR Bits	Bit Description/Action Indicated
32	CONO/CONI (Set/Read)	CONT	Continue. Set by a CONO with bit 32 set to 1. This bit is used to start the DX10 or continue after a Halt. It remains set until the channel is idle. The Continue bit is used in conjunction with the Clear bit to specify the channel starting address. Changing CONT from 1 to 0 causes the channel to stop after the present channel instruction. Cleared by the PDP-8/A along with the Clear flip-flop before returning the DX10 to the idle loop. Like the Clear flip-flop, it may be cleared by a CONO (bit 32 = 0), DX10 Initialize or I/O Bus Reset.
33–35	CONO/CONI (Set/Read)	PIA	Priority Interrupt Assignment. Octal number assigned by the monitor (CONO with appropriate bits in bit positions 33–35). Specifies the number of the interrupt line to be used by the DX10 when sending an interrupt to the CPU. Three-bit binary number is decoded and gated by one of four ORed interrupt conditions through a BCD decoder to select the corresponding interrupt line (PI1–PI7) when any of the four interrupt-type fault conditions occurs.

NOTE

To facilitate direct channel to CPU communication, the DX10 generates an interrupt to the CPU when it halts or stores status in memory. Any of four bits (MP ERR, MEM PAR ERR, NXM and STAT AVAIL) in the Channel Command Register (CCR) may be set at the time of the interrupt. The interrupt remains pending until the CPU either:

1. **Sets all of the bits to 0 with a CONO**
2. **Sets the priority interrupt assignment (PIA) to 0.**

The DX10 interrupts to location 40 + 2N (N = PIA) in the DECsystem-10 memory. PIA 0–2 can also be reset to 0 by DX10 Initialize or I/O Bus Reset.

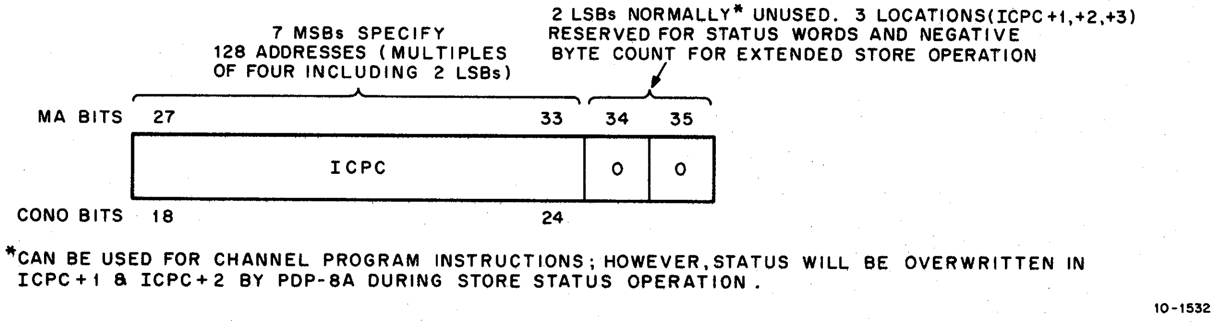

2 LSBs NORMALLY* UNUSED. 3 LOCATIONS(ICPC+1,+2,+3) RESERVED FOR STATUS WORDS AND NEGATIVE BYTE COUNT FOR EXTENDED STORE OPERATION
7 MSBs SPECIFY 128 ADDRESSES (MULTIPLES OF FOUR INCLUDING 2 LSBs)
MA BITS
27
33
34
35
ICPC
0
0
CONO BITS
18
24
*CAN BE USED FOR CHANNEL PROGRAM INSTRUCTIONS; HOWEVER,STATUS WILL BE OVERWRITTEN IN ICPC+1 & ICPC+2 BY PDP-8A DURING STORE STATUS OPERATION.
10-1532

Figure 3-3 Initial Channel Program Counter

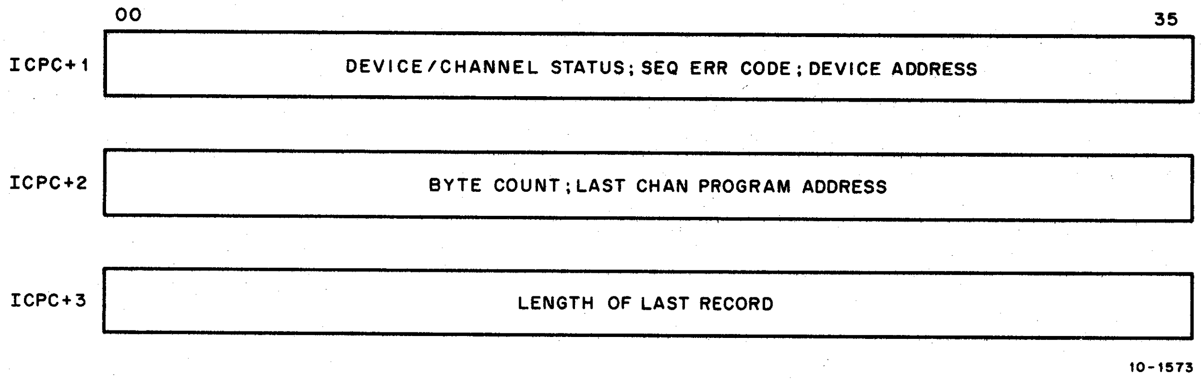

00
35
ICPC+1
DEVICE/CHANNEL STATUS; SEQ ERR CODE; DEVICE ADDRESS
ICPC+2
BYTE COUNT; LAST CHAN PROGRAM ADDRESS
ICPC+3
LENGTH OF LAST RECORD
10-1573

Figure 3-4 Channel Status Format

3.3.2 DATAO/DATAI Instruction

In addition to the capability of initializing and restarting the data channel using CONO instructions or reading status and control information from the Channel Command Register using CONI instructions, the DECsystem-10 Device Driver Program can also communicate directly with any of the DX10 registers, counters, etc. using DATAO and DATAI instructions. (BLKO and BLKI instructions affect the DX10 in the same way as DATAOs and DATAIs.)

Internally, the DX10 utilizes a common interface bus, called an Internal Bus (IBUS), for intercommunication between the operating registers, counters, and storage flip-flops. The PDP-8/A Microprocessor also utilizes the IBUS to communicate with these storage devices and counters. DATAO and DATAI I/O instructions are used by the operating program to communicate directly with the DX10 internal registers, the PDP-8/A Microprocessor, and/or the 4K RAM control storage via the IBUS. These instructions can be used only if the PDP-8/A is halted (RUN = 0) or in single-step mode.

DATAO/DATAI instructions perform only four functions in the DX10:

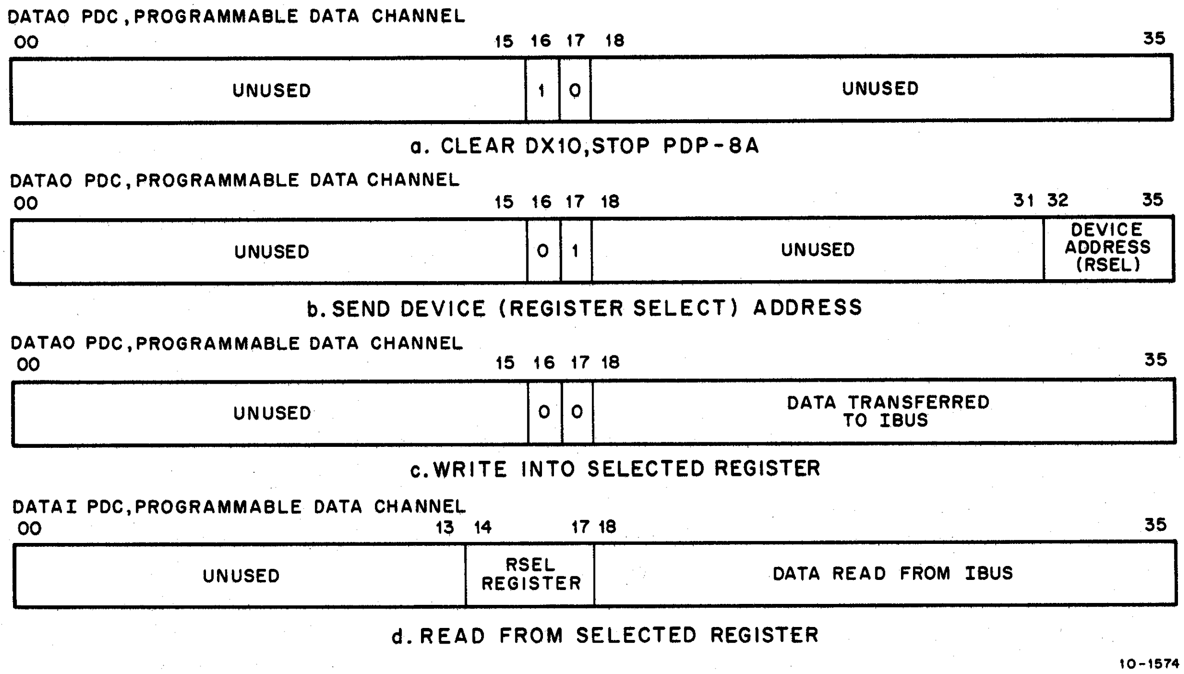
1. Initialize the DX10 and stop the microprocessor.
2. Select a register (counter, etc.).
3. Read from the selected register.
4. Load (write into) a selected register.

No other operations are defined for these instructions in the DX10. Since diagnostic programs for the DX10 must utilize DATAO/DATAI instructions to logically test the IBUS and its associated registers, these programs must be written in either Executive or User IOT mode.

Executive mode programs using the DX10 must reload the PDP-8/A microcode into control storage (RAM) following a power loss because the RAM is a volatile (nonpermanent) memory. This is accomplished through the use of series of DATAO instructions with the address of the first control storage location to be loaded contained in bits 24–35 of an RSEL5 DATAO, followed by a series of RSEL4 DATAOs with the data word in bits 24–35 and bit 23 (the Deposit bit) set to 1. The address is automatically incremented for each successive DATAO transfer. After loading of the microcode is complete, the PDP-8/A microprogram must be started before a CONO can be issued to start the channel.

To accomplish this, a DATAO is issued with the address of the control storage starting location contained in bits 24–35 and bit 20 (Continue) set to 1.

3.3.2.1 Selecting A Register, Counter, Etc. (Figure 3-5) - Before it can read a particular register, counter, etc. connected to the IBUS, the DECsystem-10 CPU must address the register (counter, etc.) by sending a 4-bit address over the I/O bus to a special Register Select (RSEL) Register. This is accomplished by sending a DATAO PDC data word with bit 17 set to 1 and the 4-bit address in bits 32–35. Before issuing this command, the operating program must first clear the DX10 control logic and stop the PDP-8/A. This is accomplished by a DATAO PDC data word with bit 16 set to 1.



a. CLEAR DX10,STOP PDP-8A

b. SEND DEVICE (REGISTER SELECT) ADDRESS

c. WRITE INTO SELECTED REGISTER

d. READ FROM SELECTED REGISTER

Figure 3-5 IBUS Read/Write Sequence

Table 3-2 lists the information that will be placed on the IBUS as a result of selecting any of the 17 possible RSEL addresses (which can be sent to the RSEL register using a DATAO PDC with bit 17 set to 1). The right-hand column indicates whether the information is to be supplied from the corresponding bits of a subsequent data word (DATAO PDC) [this would occur for a programmed data transfer (write) into the selected register, memory address, etc.] or from the register to be read by the operating program using another DATAI instruction (read). A Write operation (DATAO PDC) is indicated by a W; a Read operation (DATAI PDC) is indicated by an R. Each DATAO or DATAI reference to the IBUS uses the right half (bits 18–35) of the 36-bit PDC data word. A PDC data word with bit 17 = 1 affects only the RSEL Register.

CAUTION

Before the operating program attempts to address a register (counter, etc.), it should perform a CONT to determine the state of the microprocessor RUN bit (bit 17). The microprocessor should be stopped as described above before the operating program attempts to address any device connected to the IBUS. When the PDP-8/A is stopped, RUN = 0.

3.3.2.2 Information Transfer – Once a device has been selected for information transfer over the IBUS, the operating program must issue a DATAO or DATAI instruction to write into or read from* the selected device. The individual registers, data bits, etc. addressed through the RSEL register are listed in Table 3-2. The resultant information transferred over the IBUS is described in Table 3-3.

3.4 CHANNEL PROGRAM

The Channel Program is assembled and stored in DECsystem-10 memory by the system monitor whenever a Device Driver Program calls for a TU70 channel operation to be performed. It consists of a series of instructions necessary to carry out the operation called for by the Device Driver Program. These instructions are unique to the operation of the DX10/TU70 Subsystem and fall into three specific categories: Channel Control, Device Command, and Data Transfer.

The DX10 is initially started by a CONO command from the CPU. This causes the microprocessor to fetch an instruction from the Channel Program in DECsystem-10 memory. Ordinarily, when the channel is started for the first time, the location specified by the CONO contains a JUMP to another address. This is because the JUMP can specify a 22-bit address; the ICPC (loaded by the CONO) can specify only a 9-bit address.

Before data can be transferred, however, a Device Command instruction must be executed. First, the device specified by the instruction is selected and a Device Command byte is sent to the device. The microprocessor then reads initial selection status from the device. If the device status is error-free, the next instruction is executed. The Device Command instruction also loads the IGN LEN ERR bit. IGN LEN ERR instructs the DX10 to ignore data transfer termination discrepancies between the DX10 and the TX01.

After starting the device, a Data Transfer instruction is executed to move information between the selected device and a predesignated data buffer area in DECsystem-10 memory. If the Device Command instruction contains a device command which requires further arguments, the Data Transfer instruction moves these and any additional arguments that may be necessary (using additional Data Transfer instructions) from memory to the device. Otherwise, a Data Transfer instruction will start the data moving between memory and the device.

NOTE

Any number of Data Transfer instructions may be executed sequentially to facilitate scatter-write or gather read-operations. (Data chaining is implicit.) The IGN LEN ERR bit may be set or cleared as desired using the Device Command instruction.

*The RSEL address selected is transferred by a DATAI PDC onto the I/O Bus in bit positions 14–17 along with the data in bits 18–35.

Table 3-2
IBUS Communications

Selected RSEL Address	Register, Counter, Data, etc. On IBUS (By Bit No.)																		Read or Write
	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	
0 0 0 0	Byte Counter (Bits 0–13)														DAC/CPC (Bits 14–17)				R/W
0 0 0 1	Data Register (Bits 0–17 or 18–35)																		R/W
0 0 1 0	Adr In	Sta In	Srv In	Par In	Dis In	Opl In	Sel In	Req In			Par Out	Adr Out	Cmd Out	Srv Out	Opl Out	Hld Out	Sel Out	Sup Out	R
										Set Srv In					Set/ Clr	Bit Select			W
0 0 1 1	Bus In (Bits 0–7)										But Out 0–7								R
										En Loop	Bus Out Reg 0–7								W
0 1 0 0	SS	Hlt	Cont	Ev Par	En Exam*	En Dep*	8A MD (Bits 0–11)												R/W
0 1 0 1	MU†	8RSEL†		RUN†			8A CPMA (Bits 0–11)												R/W
0 1 1 0	8R (Bits 18–35)																		R/W
0 1 1 1	8R (Bits 0–17)																		R/W
1 0 0 0	MR (Bits 0–17)																		R/W
1 0 0 1	MR (Bits 18–35)																		R/W
1 0 1 0	DAC (Bits 18–35)																		R/W
1 0 1 1	CPC (Bits 18–35)																		
1 1 0 0	FR (Bits 18–35)																		R/W
1 1 0 1					State Sel*		8/A States (Bits 0–11)†												R/W
1 1 1 0									Mem Cycle	ICPC (Bits 27–33)									R
1 1 1 1	Microprocessor Display Register																		

*Write (DATAO) only.
†Read (DATAI) only.

Table 3-3
IBUS Information Transfer

Address Sent	Device Selected/Information Transferred
RSEL = 00	Byte Counter and the high-order four bits (bits 14–17) of the Data Address Counter (DAC) or the Channel Program Counter (CPC), depending on which low-order bits were written or read last by a DATAO or DATAI.
RSEL = 01	Data Register containing the most recent 36-bit word to be transferred to or from the data byte assembly logic. Reading the DR yields bits 0–17 on the first reference, bits 18–35 on the second reference. Writing the DR loads bits 18–35 during the first reference; the second reference transfers 18–35 to 0–17, and loads the new data into 18–35.
RSEL = 02	*Read:* The Tag In lines and the Tag Out Register are read (multiplexed via the IBUS) onto the I/O Bus, together with the Channel Bus Parity In and Parity Out bits, by a DATAI as follows: **Bit** **Tag** 18 ADR IN 19 STA IN 20 SRV IN 21 PARITY IN 22 DIS IN 23 OPL IN 24 SEL IN 25 REQ IN 28 PARITY OUT 29 ADR OUT 30 CMD OUT 31 SRV OUT 32 OPL OUT 33 HLD OUT 34 SEL OUT 35 SUP OUT **NOTE** **The Tag In lines are multiplexed after the Channel Bus receivers; the Tag Out lines are multiplexed directly from the Channel Bus.** *Write:* The Tag Out Register (and a single Tag In line) may be modified one bit at a time using a DATAO as follows: Bit 27* Simulate a Service In (SRV IN) condition (Set tag = 1; Clear tag = 0) Bit 32 Set/clear the Tag Out Register bits selected by bits 33–35. Bits 33–35 Select Tag Out Register bit as follows:

*Do not set bit 27 unless bits 33–35 are non-zero.

Table 3-3 (Cont)
IBUS Information Transfer

Address Sent	Device Selected/Information Transferred	
	33 34 35	
	0 0 0	SRV OUT B (set only; cleared by SRV IN)
	0 0 1	ADR OUT
	0 1 0	CMD OUT
	0 1 1	SRV OUT A
	1 0 0	OPL OUT
	1 0 1	HLD OUT
	1 1 0	SEL OUT
	1 1 1	SUP OUT
RSEL = 03	*Read:* The received Channel Bus In lines are read onto I/O Bus bits 18–25.	
	The Channel Bus Out lines are read directly from the bus onto I/O Bus bits 28–35.	
	Bit 27	Load Loop Enable. Allows Bus Out (0–7, P) to be transferred to Bus In lines.
	Write: The Bus Out Register is loaded from bits 28–35. It is gated to the Bus Out lines only when a block transfer Write is not in progress.	
RSEL = 04 *	Bits 24–35 represent the microprocessor (PDP-8/A) Memory Data lines (MD0–11). Additional bits are provided as indicated for microprocessor control. These are:	
	18 SS	Single Step. When set, one PDP-8/A cycle is executed each time the CONT bit is set.
	19 HALT	When set, one PDP-8/A instruction is executed each time the CONT bit is set. Setting this bit while the microprocessor is running will stop the PDP-8/A.
	20 CONT	On a 0 to 1 transition, execution begins at the control storage address in the CPMA register. CONT also causes one additional cycle if SS is set or one additional instruction to be executed if HALT is set.
	21 EV PAR	Set to cause parity errors to occur for diagnostic purposes. (See RSEL 11.)
	22 EN EXAM	Allows a DATAO to access a PDP-8/A memory location by initiating a memory cycle via the Omnibus. This transfers the contents of the word addressed by the CPMA into the memory buffer. The word can then be read over the MD0–11 lines by a DATAI, RSEL 04.
	23 EN DEP	Enables a deposit of bits 24–35 into the control storage location addressed by the microprocessor CPMA Register.

*See Paragraph 3.5 for use of RSEL04 and RSEL05 in performing Deposit and Examine operations in the microprocessor's control storage memory.

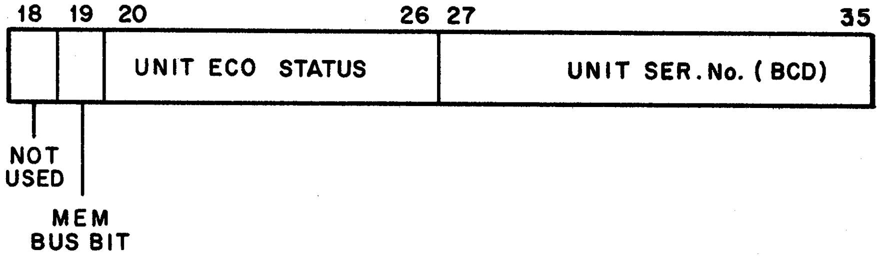
Table 3-3 (Cont)
IBUS Information Transfer

Address Sent	Device Selected/Information Transferred
	24–35 8A MD0–11 DATAO causes a deposit of these bits in control storage memory at the location specified by the CPMA Register if EN DEP = 1. DATAI examines these bits to determine the current data on the Omnibus MD lines. A DATAO with either bit 22 or 23 set to 1 causes the CPMA to increment for the next reference. CONT (bit 20) must = 0 for Examine and Deposit functions. This is the mechanism by which control storage is loaded after initial power-on.
RSEL = 05	Bits 24–35 access the microprocessor CPU's current Memory Address Register (CPMA). This register contains the address of the next word in the control storage to be referenced by the PDP-8/A CPU, whether instruction or operand. Bits 19 and 20 access the microprocessor-programmable 8RSEL Register, which indicates which 8R the microprocessor has currently selected. Bits 19 and 20 are read-only bits. (See Paragraph 5.6.2.7.) Bit 21 (RUN) is a read-only bit which indicates that all 8A CPU circuits are active and are currently executing instructions. Bit 18 (MU – Memory User) indicates which memory address register (DAC or CPC) was used for the address of the last DECsystem-10 memory cycle. It is a read-only bit. 0 = DAC 1 = CPC
RSEL = 06	The microprocessor utilizes a 4-word X 36-bit register "file" (primarily to transfer data and instructions between its own 4K MOS memory and DECsystem-10 memory). This 4-word file is located on the M8597 module and is referred to as the 8R Register file, with each of its 36-bit, addressable registers designated 8R0–8R3. RSEL 06 selects the low-order 18 bits (bits 18–35) of the file register last selected by the microprocessor for transfer of these bits to or from the IBUS. (Register 8R0 is automatically selected immediately following a system Reset or initial power turn-on.)
RSEL = 07	Same as RSEL 06 for 8RX bits 0–17.
RSEL = 10	Memory Register bits 0–17. This is the word last transferred to/from DECsystem-10 memory.
RSEL = 11	Same as above, except for MR bits 18–35. In addition, reading MR bits 18–35 causes the parity of the full 36 bits to be tested against the memory parity bit. Bad (even) parity causes the CCR MEM PAR ERR bit to set unless EV PAR is set. The memory parity bit is loaded from DECsystem-10 memory.
RSEL = 12	The low-order 18 bits of the Data Address Counter (DAC), which points to the next core memory location to be referenced for a channel data transfer.
RSEL = 13	The low-order 18 bits of the Channel Program Counter (CPC), which points to the next core memory location to be referenced directly by the microprocessor for a Channel Program command transfer.

Table 3-3 (Cont)
IBUS Information Transfer

<table>
<tr><th>Address Sent</th><th>Device Selected/Information Transferred</th></tr>
<tr><td>RSEL = 14</td><td>

The Feature Register (FR) allows the software to check installation-dependent variables in the hardware. It is read with a DATAI, and written by setting individual switches on the Feature Register DIP switches (located on the M8597 PDP-8/A Interface Module). The format is:



</td></tr>
<tr><td>RSEL = 15</td><td>

Bits 24–35 read any of four microprocessor (PDP-8/A) 12-bit functions; bits 22 and 23 select which function. Bits 24–35 are read-only bits.

00 Status
01 MQ
10 Data Bus (All Zeros)
11 AC

(Refer to PDP-8/A Miniprocessor Handbook)

</td></tr>
<tr><td>RSEL = 16</td><td>Bit 26 reads the MEM CYCLE bit. Bits 27–33 read the ICPC register, which is loaded by a CONO (part of CCR).</td></tr>
<tr><td>RSEL = 17</td><td>Used to test the IBUS for diagnostic purposes. When RSEL 17 is selected and the microprocessor is halted, the diagnostic program looks for all 0s on the IBUS (nothing selected). If any bit is set, a malfunction is indicated. When the microprocessor is running, it uses the maintenance panel DATA indicators as a display register by selecting RSEL 17.</td></tr>
</table>

The DX10 microprocessor has Channel Program interrupt capability; that is, a trap address in DECsystem-10 memory is transferred to when a device error occurs. If the Unit Check or Unit Exception bits (Paragraph 3.4.8) are on during initial selection or ending status read, the microprocessor will examine location ICPC+3. If that location is negative, bits 14–35 are used as a pointer to the error status storage area. See Figure 3-6 for status storage format. (This relieves the CPU of the task of having to read device sense information after an error.)

At the beginning of a Device Command instruction, the microprocessor reads initial selection status from the TX01 and stores it in the Device Status Register location in control storage memory. If an error occurs during the selection sequence, the microprocessor performs a Status Store operation in which the Device Status Register is stored in ICPC+1 together with the Channel Status Register, a code (SEQ ERR CODE) that indicates the type of error that occurred, and the device address (from the Device Command instruction). Additional (normal) status information is stored in ICPC+2 and ICPC+3 as shown in Figure 3-6.

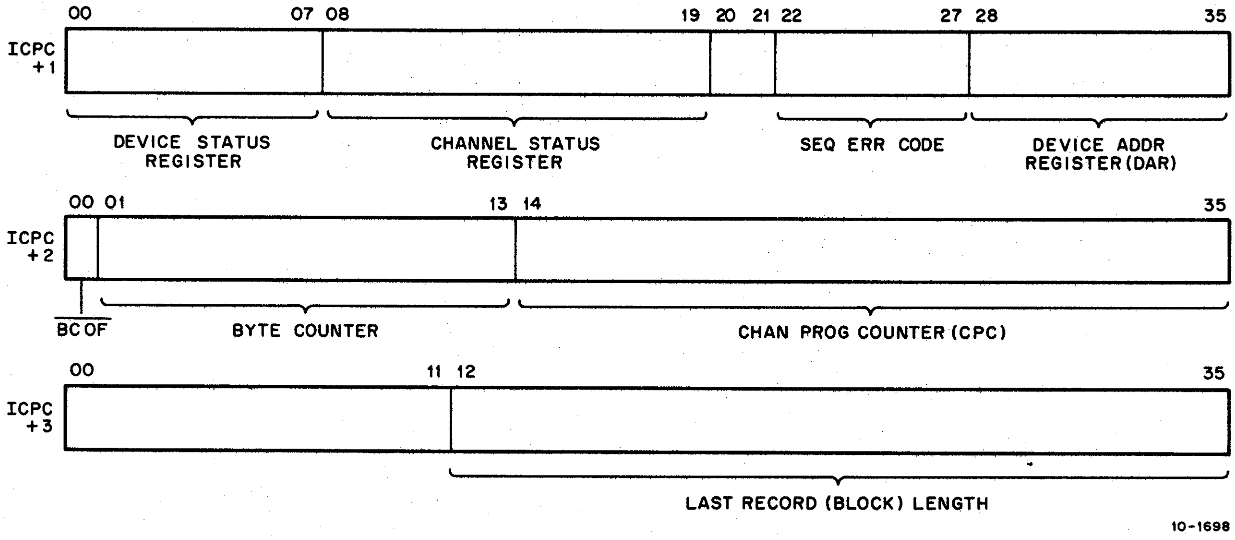


Figure 3-6 Normal Status Format

Before storing status, however, the microprocessor examines location ICPC+3. If a negative byte count is stored in this location, the microprocessor stores the specified number of device status Sense bytes (specified in bits 0–13) into a corresponding number of byte locations (4 bytes per word) starting at the location specified in bits 14–35 of the word in ICPC+3. Storing of the additional (Sense) bytes is referred to as an Extended Status Store operation and may also be forced by a Channel Control instruction with bit 5 set to 1.

If no error occurred during initial selection, the above status is not stored in DECsystem-10 memory. At the end of the operation, the microprocessor reads ending status from the TX01 and stores this status in the CSR and DSR before performing a normal status store operation, again with the restriction that ICPC+3 is examined before the store operation to determine whether a Normal Status or an Extended Status Store operation is to be performed.

The Channel Program controls device selection, data transfer, and status storage of status by means of three discrete types of Channel Program instructions.

3.4.1 Channel Control Instructions (00)

Channel Control instructions are used to:

1. Synchronize the CPU with the DX10.

2. Terminate a Channel Program.

3. Cause the DX10 to store status (normal or extended) in DECsystem-10 memory.

A Channel Control instruction has an operating code of 00, which is decoded by the PDP-8/A Microprocessor when the instruction is fetched from DECsystem-10 memory and placed into a PDP-8/A (8R) register. The individual bit definitions in the Channel Control instruction (Figure 3-7) and their uses in directing the microprocessor to perform channel-related operations are as follows:

Bits 0,1 — Op Code 00. Instruct the PDP-8/A Microprocessor to set up to perform channel-only functions.

Bit 2 — Go. If set, fetch the next instruction and continue. If 0, clear the CONT bit and halt. (The PDP-8/A then enters the idle loop.)

NOTE

A Channel Control instruction with the Go bit = 0 is also referred to as a Channel Halt.

Bit 3 — Jump. Load bits 14–35 of this instruction into Channel Program Counter (CPC) bits 14–35. With the GO bit set, the PDP-8/A will continue at (jump to) this new address. If Go = 0, the PDP-8/A will Halt the DX10 (after loading CPC) and return to the idle loop.

Bit 4 — Store. Perform a normal status store subroutine. This subroutine stores status in memory at locations ICPC+1 through ICPC+3 in the format shown in Figure 3-6, and interrupts the CPU (by setting STAT AVAIL in the Channel Command Register) when status is stored. The microprocessor labels the stored status as Program Status by setting the PRG STAT bit in the Channel Status Register (CSR).

Bit 5 — Forced Extended Status. Instruct the microprocessor to store normal status in ICPC+1 and ICPC+2 and then perform an Extended Store subroutine (see microcode description, Chapter 4) using ICPC+3 as a Data Transfer instruction. (The EXT STAT STORE subroutine causes the microprocessor to examine ICPC+3. If it finds the contents indicated in Figure 3-8, it stores the specified number of bytes in successive memory locations starting at ADDR. If ICPC+3 is positive, the microprocessor exits to the idle loop.)

Bits 6–13 — Unused

Bits 14–35 — New Channel Program Counter (Jump) Address. This value will be loaded into the CPC by the microprocessor before continuing, if the Jump bit (bit 3) is set.

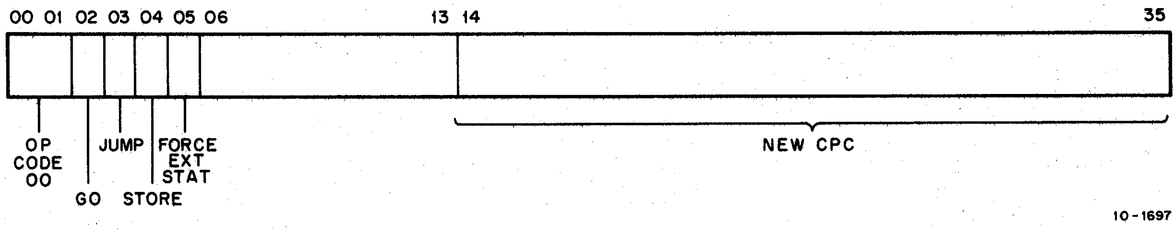


Figure 3-7 Channel Control Instruction Bit Assignments

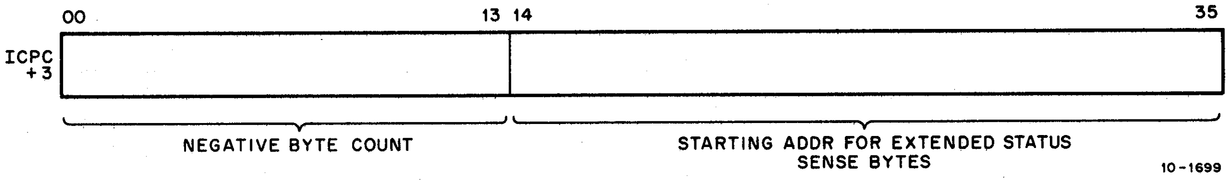


Figure 3-8 Extended Status Store Control Word Format

3.4.2 Device Command Instructions (01)

The Device Command instruction (Op Code 01) shown in Figure 3-9 is used by the Channel Program to select a specific tape unit via an 8-bit Device Address byte (bits 28--35), and to inform the TX01 of the operation to be performed when the addressed drive is selected. This is accomplished by sending the controller, in addition to the tape unit address, an 8-bit Device Command byte (bits 16–23 of the Device Command instruction). In addition to the Device Command and Address bits, the Device Command instruction specifies the mode of operation (for data transfer instructions) during data transfers, as well as during status control information transfers, to instruct the microprocessor in proper handling of error conditions and byte packing. The data modes as specified by bits 5 and 6 (and defined in Chapter 1) are:

00	Core Dump
01	Byte (Industry Compatible)
10	ASCII
11	SIXBIT

These mode control bits are decoded and used in the DX10 to control special byte formatting logic during disassembly of data bytes from the SILO and formatting into a 36-bit data word in the Channel Byte Data Register.

The individual bits are used by the Channel Program to instruct the microprocessor as follows:

Bits 0, 1 Op Code 01. Instruct the PDP-8/A to set up to perform device functions.

Bit 2 Ignore Length Error (IGN LEN ERR). This bit is stored by the microprocessor in the Channel Status Register (CSR) located in the 4K MOS control storage memory. It instructs the microprocessor not to store status upon occurrence of a length error.

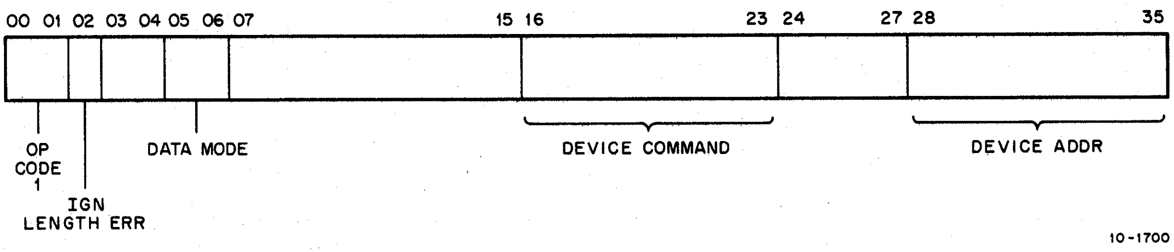


Figure 3-9 Device Command Instruction Word

3.4.3 Data Transfer Instructions (1X)

A Data Transfer instruction (Figure 3-10) is used in the Channel Program following a Device Command instruction to move information between a selected tape drive and DECsystem-10 memory. It is recognized by a 1 in bit 0 (the Op Code field). Bit definitions and usage follow:

Bit 0	Op Code 01. Instruct the PDP-8/A Microprocessor to set up the DX10 logic to transfer the specified number of bytes between the selected tape drive and DECsystem-10 memory starting at the data address indicated.
Bits 1–13	Byte Count. Specify up to 8192 bytes (characters) of data to be transferred.
Bits 14–35	Data Address. Use of 22-bit address allows direct addressing of any starting location in memory. During Forward Read or Write operation, specify location of first word of data block in memory. Address is incremented in the Data Address Counter (DAC) for each 36-bit word transferred. During a Read Backward operation, address specified represents last word in data block. The PDP-8/A Microprocessor uses the byte count to compute the position of the first byte read into the last word location. The DAC is decremented for each word transferred into memory. If the data address field = 0 during a Read operation (specified by preceding Device Command instruction), the PDP-8/A is instructed to skip (read but do not transfer) the specified number of bytes from the tape drive.

NOTE

Data Transfer instructions must be preceded by a Device Command instruction or another Data Transfer instruction.

3.4.4 Commands and Status

Three distinct levels of commands are generated within the DX10. These are:

1. Microprocessor commands from the CPU (control bits in a CONO instruction)
2. Microprocessor and channel commands (control bits in the Device Command and Channel Control instructions.)
3. Device Command bytes (from the Device Command instruction).

Commands are used to control the microprocessor, the data channel or a selected device.

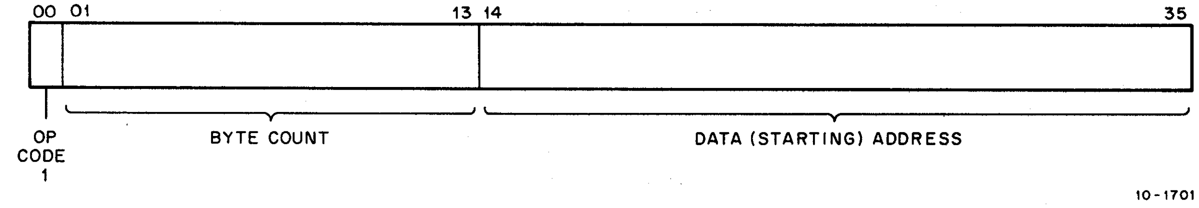


Figure 3-10 Data Transfer Instruction Word

For each level of command there is a corresponding status level. The particular level of status called for is read and stored by the microprocessor in response to various commands and/or lower level status conditions (Paragraph 3.4.8). When status has been stored, the type of status and condition(s) under which it occurred are signaled in one of three different registers as follows:

1. Channel status stored in the Channel Command Register (CCR).
2. Channel status stored in the Channel Status Register (CSR) in RAM control storage.
3. Device status stored in the Device Status Register (DSR) (also in RAM control storage) and Sense bytes from the device.

The CCR (Figure 3-2) is used for direct CPU/DX10 communication, and was discussed at the beginning of the chapter. (Refer to PDP-8/A Microcode, Chapter 4, for commands generated in the CCR.) Catastrophic failures, which interfere with memory communication (parity error or nonexistent memory), are flagged as status in the CCR and cause an interrupt to the CPU. The Status Available flag is also located in the CCR to inform the CPU of availability of new (lower level) status (stored in the CSR and DSR). Both of these registers (CSR and DSR) are stored in memory location ICPC+1 whenever status is stored.

The DSR and CSR are located in PDP-8/A control storage and are shown in Figures 3-15 and 3-13. Figure 3-16 illustrates the various levels of status and the relationships between each register. It can be seen how CSR FLAG (in the CCR) is the logical OR of all error bits and bits of interest. Bits which are set in the CCR only do not cause the PDP-8/A to store status.

The various commands used in the DX10 operate as described in the following paragraphs.

3.4.5 Device Commands

Of the three types of Channel Program instructions used to control the TU70 Magnetic Tape Subsystem, two instructions (Channel Control and Data Transfer) control the DX10 through individual command bits within the instruction itself. The Device Command instruction, however, causes the PDP-8/A to transfer Device Command "bytes" directly from DECsystem-10 memory to the TX01 Control Unit. Device Command bytes are issued over the Channel Bus Bus Out lines with the CMD tag line active. A command byte is sent to the control unit following the address byte during the initial selection sequence of the Device Command instruction; the Device Command bytes specify the operation to be performed by the selected tape drive. Figure 3-11 depicts a typical Channel Command sequence. Once the DX10 has been initialized via the Channel Command Register by a CONO instruction from an ExecutiveMode Program, the PDP-8/A Microprocessor begins fetching instructions over the system Memory Bus from the Channel Program in DEC system-10 memory. Upon decoding a Device Com mand instruction, the microprocessor stores the 8-bit Device Address byte in the Device Address Register Out (DARO) location and the 8-bit Device Command byte into the Device Command Register (DCR) location in control storage memory. The microprocessor then sets up the channel byte control logic and executes the initial selection sequence, sending first the Tape Drive (Device) Address byte, then the Command byte to the TX01 via the Channel Bus Output Register. (Refer to Paragraph 5.7.2 and the accompanying initial selection sequence timing diagram for sequence of operations.)

Figure 3-12 depicts the six types of device commands that can be issued by the Channel Program. Each of the commands is read into the DCR and interpreted by the microprocessor before being sent over the Channel Bus to the TX01. The specific bits "looked at" by the microprocessor are partitioned in the illustration. The shaded portion of the diagram denotes those bits which are decoded by the TX01 Control Unit (CU) only. The manner in which the microprocessor sets up the Channel Bus Control Register (and thus the byte formatting and transfer control logic) is determined by the microprocessor after it determines the type of command to be executed. The command byte (if it is a legal code) is then transferred from the DCR to the TX01 over the Channel Bus. The function of each type of command byte is listed in Table 3-4.

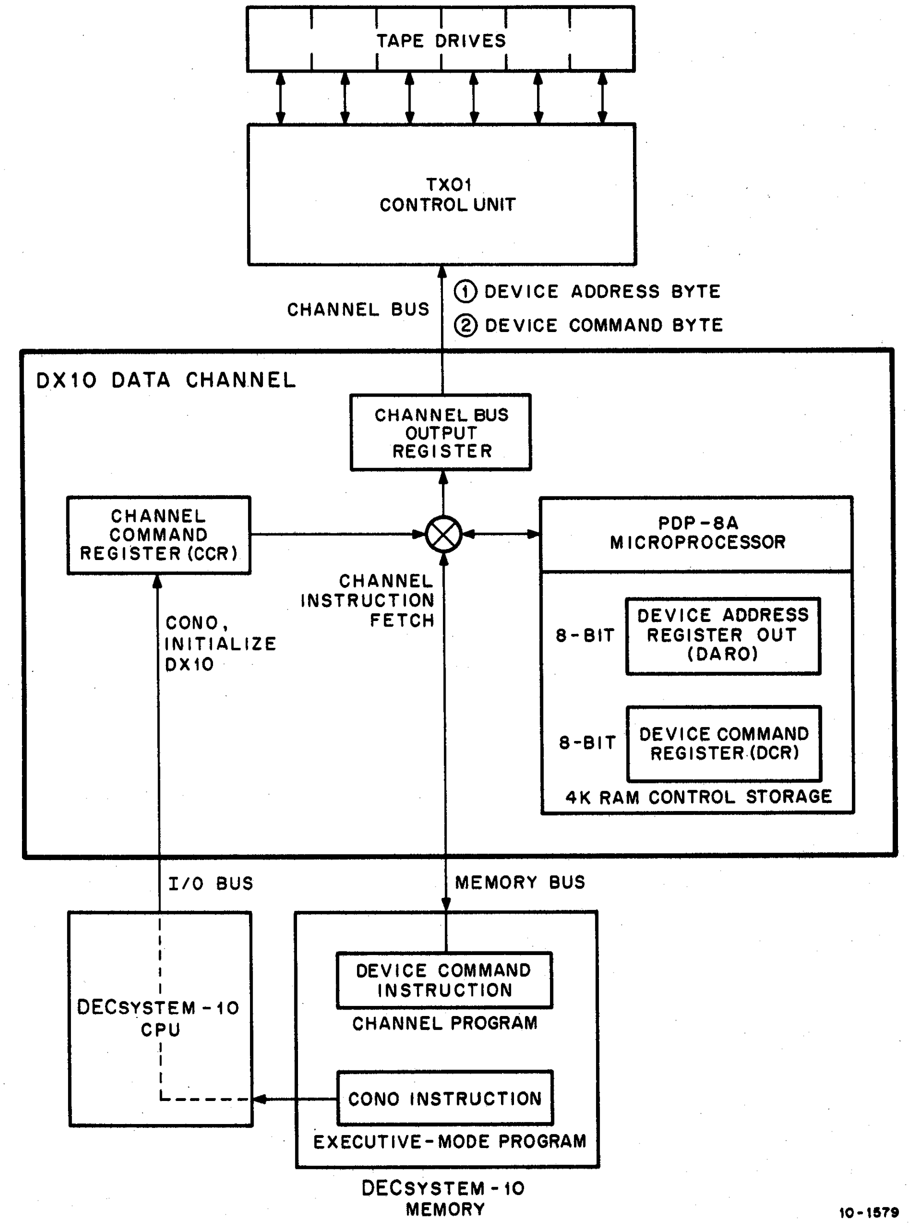
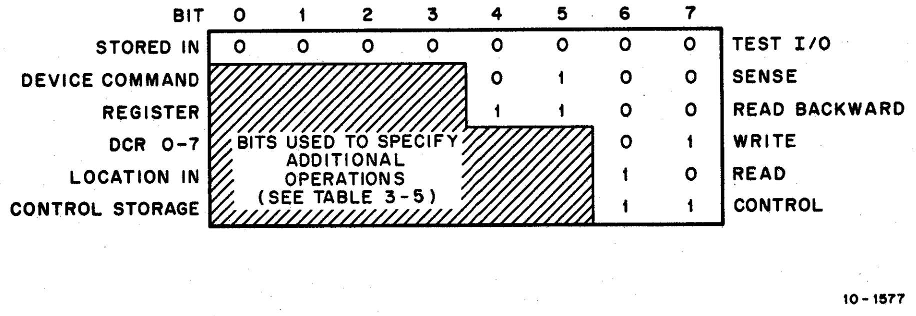


Figure 3-11 Channel Command Sequence



BIT	0	1	2	3	4	5	6	7	
STORED IN	0	0	0	0	0	0	0	0	TEST I/O
DEVICE COMMAND	▨	▨	▨	▨	0	1	0	0	SENSE
REGISTER	▨	▨	▨	▨	1	1	0	0	READ BACKWARD
DCR 0-7	BITS USED TO SPECIFY ADDITIONAL OPERATIONS (SEE TABLE 3-5)	▨	▨	▨	▨	▨	0	1	WRITE
LOCATION IN	▨	▨	▨	▨	▨	▨	1	0	READ
CONTROL STORAGE	▨	▨	▨	▨	▨	▨	1	1	CONTROL

Figure 3-12 Device Command Register

Table 3-4
DCR Command Functions

Command Type	Function
Test I/O	Causes the device status byte of the selected unit to be sent to the channel.
Sense	Causes the device sense bytes to be transferred to memory under control of a DATA XFER instruction.
Read Backward	Causes the selected tape unit to read the tape backward. The words are filled backward and stored in memory backward. (The memory address counter decrements.)
Write	Write forward on the storage medium.
Read	Read forward from the storage medium.
Control	Control commands are divided into Motion Control, Nonmotion Control, and Mode Set functions. Control commands may not be followed by a DATA XFER instruction.

Commands that effect a data transfer (Burst commands) cause the tape drive, the control unit, and the data channel to remain busy until command completion. Immediate (Nondata Transfer) commands (with the exception of Motion Control commands) also exhibit the same characteristics. A Rewind command causes the tape drive to remain busy after the control unit and the data channel have become available. If the Channel Program attempts to use the drive while it is busy, the PDP-8/A sets the DSR flag in the CSR, stores status, and interrupts the CPU. During motion control commands, both the drive and the control unit remain busy.

3.4.5.1 Command Byte – When Command Out is up during the initial selection sequence, the information on Bus Out is the command byte. The microprocessor issues Command Out to initiate, allow, or terminate an operation in an addressed tape drive. The Command byte requires decoding by the control unit only during a DX10-initiated selection sequence (when the DX10 addresses a tape drive).

At all other times, the byte is 0 (parity is unimportant). The low-order bit positions indicate the type of operation; the high-order bit positions indicate a modification code which expands the basic operation at the control unit or tape unit.

The control unit responds to the following commands from the DX10 (Table 3-5):

Burst Commands
Motion Control Commands
Nonmotion Control Commands

3.4.5.2 Burst Commands – All Burst commands transfer information across the DX10 Channel/control unit interface.

Table 3-5
Device Command Codes

Command Code (Hexadecimal)	Operation
Burst Commands (Data Transfer)	
01	Write
02	Read Forward
04	Sense
1B	Request Track in Error
0C	Read Backward
8B	Loop Write to Read
Motion Control Commands	
07	Rewind
0F	Rewind Unload
17	Erase Gap
1F	Write Tape Mark
27	Backspace Block
2F	Backspace File
37	Forwardspace Block
3F	Forwardspace File
97*	Data Security Erase
Nonmotion Control Commands	
03	No Operation

Mode Set 1 Commands (7-track)

	Density			Parity		Translator		Data Converter	
	200	556	800	Odd	Even	On	Off	On	Off
63		X			X		X		X
73		X		X			X		X
A3			X		X		X		X
B3			X	X			X		X
23	X				X		X		X
33	X			X			X		X

NOTE

Seven-track Mode Set 1 commands are treated as "No-Op Reset Sense" when issued to a tape control without the 7-track NRZI compatibility feature.

The 200 bpi Mode Set 1 command (hexadecimal command codes 13, 23, 2B, and 33), when issued to a subsystem with a 7-track feature, sets 556 bpi if the 200 bpi feature is not installed.

*This command is valid only when preceded by an ERG (17) command.

Table 3-5 (Cont)
Device Command Codes

Command Code (Hexadecimal)	Operation
Mode Set 2 Commands (9-track)	
C3	PE, 1600 bpi, 9-track
CB	NRZI, 800 bpi, 9-track

NOTE
Nine-track Mode Set 2 commands are treated as "No-Op Reset Sense" when issued to a tape control without the 9-track NRZI compatibility feature.

Command Code (Hexadecimal)	Operation
I/O Commands	
0B	Diagnostic Mode Set
00	Test I/O

The "ending status" is indicated by Channel End and Device End when the operation is completed. Burst commands and their corresponding mnemonics are as follows:

Write (WRT) – This command records data on tape and creates an interblock gap (IBG) at the end of each record. The control unit verifies each byte of data received from the DX10.

Read Forward (RDF) – The tape unit is set to a read forward status. Data is read until an IBG is detected. The control unit verifies parity for each individual byte of data and corrects it if necessary. Thus, the control unit can correct* more than one "track in error" provided they do not occur simultaneously.

Read Backward (RDB) – The tape unit is set to a read backward status. Data flow is the same as in a Read Forward operation, except that the NRZI Data Converter mode cannot be used.

Sense (SNS) – The Sense command causes the Sense bytes to be transferred to the channel. The control unit can transfer up to 24 bytes of sense data. The Channel Program Data Transfer instruction specifies the number of Sense bytes to be transferred and the starting storage address. The information transferred includes error conditions associated with the last operation, and provides information about the current conditions present in the control unit and tape unit.

Request TIE – A Request TIE command is a request for a channel to return to the control unit a data byte containing track-in-error information. The DECsystem-10 program must gather this information from the sense information following a PE read or write error or a 9-track NRZI read error.

A TIE request following a 9-track NRZI read operation enables the control unit to correct a single bit error (read correction).

NOTE
An uncorrectable error is indicated if bits 6 and 7 are set in the data byte.

*In PE mode only.

Loop Write-To-Read (LWR) - This is a maintenance aid and does not move tape. It allows for checking of the read/write circuitry of the control unit.

LWR uses the same data path as a Write operation. The tape unit must be "ready" but need not be in write status.

3.4.5.3 Motion Control Commands - Motion Control commands are those commands which move tape, but do not transfer data between TU and CU. The sequence of operation for these commands is as follows: Channel End is signaled when the command is accepted (Initial Status); CH END and Device End are signaled when the operation (except for Rewind Unload) is complete (Ending Status); the tape control responds with TU Busy or CU Busy if the tape unit is addressed while executing a Motion Control command.

Motion Control commands and their mnemonics are as follows:

Rewind (REW) - This command causes a tape unit to rewind to load point.

Rewind Unload (RUN) - For Rewind Unload, the command is executed by the tape unit. If, while the tape unit is unloaded, the channel attempts to address the unloaded tape unit, Device End is sent when the tape unit is reloaded and becomes ready.

This command causes the tape unit to unload tape after rewinding.

Erase Gap (ERG) - A tape unit erases a length of tape while moving forward. The sequence in which the command is issued creates the following slight variations:

1. Successive ERG commands add approximately 3.6 in. to the IBG.

2. At load point, a single ERG command erases approximately 4.2 in. of tape after the identification burst.

3. A single ERG command at any other time erases approximately 4.2 in. of tape.

Write Tape Mark (WTM) - A tape unit moves tape forward and writes a tape mark (end of file block). Upon issuing a WTM command, an ERG is performed first, which erases 3.6 in. of tape after the last record written, or 4.2 in. if at load point. Following this gap, the tape mark is written on the tape. A PE tape mark consists of 64 discrete tape mark characters. NRZI 9-track and 7-track tape marks consist of a single character and an LRCC. At load point, a PE tape unit writes an identification burst with a WTM command.

PROGRAMMER'S NOTE

The control unit responds with a control unit busy sequence while performing a WTM operation. A WTM command on a file protect tape is not valid and results in Command Reject.

Backspace Block (BSB) - A tape unit moves backward to the next interblock gap and no data bytes are transferred. BSB does not move tape beyond the load point.

Backspace File (BSF) - A tape unit moves backward to the interblock gap beyond the next tape mark, or until load point is detected. No data bytes are transferred.

Forward Space Block (FAB) - A tape unit moves forward to the next interblock gap. If a tape mark is read, Unit Exception and Device End are set in the status byte.

Forward Space File (FSF) – A tape unit moves forward to the interblock gap beyond the next tape mark. No data bytes are transferred. Unit exception is not set when the tape mark is read.

Data Security Erase (DSE) – A tape unit moves forward, performing successive ERG type operations, until EOT is reached. The TI* indication sets Device End in the Status byte.

A TI indication terminates DSE.

NOTE

DSE is valid only when following immediately after an ERG command. Command Reject is set under any other conditions, or if the tape is file protected.

PROGRAMMER'S NOTE

If the tape unit drops "ready" or logically fails during DSE, the ending status contains Device End, Unit Check, and Sense bit 4 of byte 7. Device End is signaled at the detection of End of Tape (EOT) during a normal DSE completion. The operating program must issue enough ERG commands to erase any data which may be beyond TI. Fourteen ERG commands are usually sufficient (approximately 50 in. of tape).

3.4.5.4 Nonmotion Control Commands – These commands do not cause tape motion, and information is not transferred to the channel. The nonmotion controls and their mnemonics are as follows:

No Operation (NOP) – Performs no function on any unit and does not transmit data. NOP does not disturb sense data in the control unit.

Mode Set 1 (MS1) – MS1 commands are for 7-track operation and are valid regardless of the position of the tape. Bits 0 and 1 determine the density (200/556/800 bpi) and bits 2, 3, and 4 are modifiers which determine parity (odd or even), the status of the data converter (on or off), and the status of the translator (on or off). An MS1 command controls all 7-track units on a control unit, and remains set until another set is given.

Mode Set 2 (MS2) – MS2 commands are used for 9-track dual density (800 bpi NRZI or 1600 bpi PE) operation. MS2 commands are valid only at load point for succeeding write operations. The default case is at 1600 bpi each time the tape returns to the load point.

MS2 commands sent to control units without the dual density feature are treated as NOP commands, except that Sense data bytes are reset (NOP Reset Sense).

Diagnostic Mode Set Maintenance Aid (DMS) – DMS sets a transparent signal loss condition that checks read and write error detection circuits, as follows:

1. In Phase-Encoded mode, whenever write data contains all 1s in any track, writing in that track is inhibited until the last 1 bit is reached.

2. In 9-track NRZI mode, no bits are written in track P.

*Tape Indicate: Indicates that the physical end of tape (EOT) has been reached.

3. In 7-track NRZI mode, no bits are written in track C.

The DMS command is valid only for the command immediately following it.

Test I/O (TIO) - This Nonmotion Control command causes the status byte for a selected tape unit to be sent to the DX10 for analysis. No other function is performed.

3.4.6 Command Sequences

All the Device Command codes for the TU70 Subsystem are shown in Table 3-5. The following information is of special interest to the programmer. The TU70 Subsystem has no interlock to prevent issuing improper sequences of Read/Write operations. It is a program responsibility to avoid sequences which may result in partial records or extraneous bits on tape.

The following two sequences should be avoided:

1. A write-type operation after a forward read-type operation, except as follows:

   a. A block or tape mark read is known to be followed by a tape mark.
   b. A block or TM read is known to have been followed by ERG when written, or known to have been the last block written before a backward operation.

   Example: *RRW O avoid

   *WBRW allowed

2. A read forward-type operation following write-type operations.

   Example: *RBWR O avoid

   *WBRR O avoid

where:

W = A write-type operation: Write, Write TM, or Erase Gap.

R = A read forward-type operation: Forward Space Block or Forward Space File.

B = A read backward-type operation: Read Backward, Backspace Block, or Backspace File.

O = The logical record on which problems may occur.

Because of the difficulty in ensuring the above safe situations, in the general programming sense, a write after read forward sequence should be confined to situations where format and command sequences are strictly controlled.

3.4.7 Status Usage and Definition

DX10 and TU70 status information are stored by the microprocessor in two status registers (12 bits for DX10 status and 8 for TU70 status) located in control storage memory. These register locations are referred to as the Channel Status Register (CSR) and the Device Status Register (DSR). Status information in both registers is rewritten upon initiation of each tape operation (initial status), except for a TIO operation, and is updated during the operation to provide channel status and device ending status for subsequent storage in specified Channel Program locations following termination of the oper ation. After termination of Channel Bus transfer operations, the microprocessor stores the CSR and DSR contents in Channel Program Status Word 1 (SW1) at location ICPC+1.

3.4.7.1 Channel Status – The Channel Status Register contains information bits related to operation of the Channel Bus; the bits are set or cleared by the microprocessor during execution of a Device Command or Data Transfer instruction. Any time the microprocessor sets a bit in the CSR, it also sets the CSR flag in the Channel Command Register.

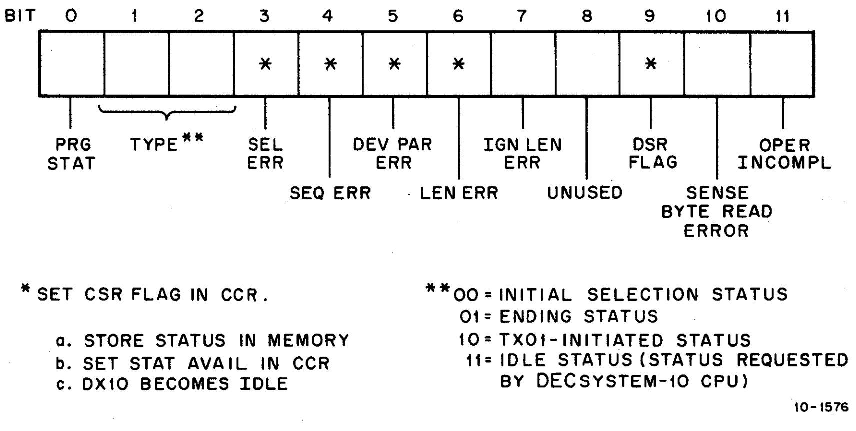
This indicates to the CPU, at the time of the next interrupt (which occurs after the next Store Status operation), that a status bit of interest in the CSR has been set or that an error has occurred during execution of the most recent Channel Program instruction. In addition to eight status bits and a command bit (IGN LEN ERR), the CSR also contains a 2-bit type code to indicate to the operating program the type of status contained in the DSR; Figure 3-13 identifies each of the CSR bits. Table 3-6 describes the status condition represented by each bit.

3.4.7.2 Device Status – The microprocessor stores device status in DECsystem-10 memory under any of the following conditions:

1. Completion of a Rewind operation.
2. As a result of an error or busy condition during execution of a device command.
3. Under control of Channel Control (Normal or Extended) Store Status instruction.
4. When requested to do so by the DECsystem-10 CPU by means of a CONO instruction with bit 30 set to 1.

Figure 3-14 illustrates a typical status storage sequence in response to condition 3. The operation is performed as follows: Status information from the TX01 Control Unit is sent to the DX10 as a 9-bit (8 bits plus parity) status byte over the system Channel Bus with the Status In tag line active. The microprocessor reads and interprets the byte of information as a status byte.

3.4.7.3 Status Byte – The status byte is read in through the Channel Bus In logic by the microprocessor and stored directly (unmodified) into the Device Status Register (DSR) in control storage memory.



* SET CSR FLAG IN CCR.

a. STORE STATUS IN MEMORY
b. SET STAT AVAIL IN CCR
c. DX10 BECOMES IDLE

**00 = INITIAL SELECTION STATUS
01 = ENDING STATUS
10 = TX01-INITIATED STATUS
11 = IDLE STATUS (STATUS REQUESTED BY DECSYSTEM-10 CPU)

Figure 3-13 Channel Status Register

Table 3-6
Channel Status Conditions

Bit	Designation	Description
0	PRG STAT	The Channel Program executed a Channel Control instruction with Store = 1. This bit clears after status has been stored. The status should be ending status with no errors.
1,2	TYPE	Indicates the cause and/or type of the last status stored in the DSR.
3	SEL ERR	The device specified by the Device Address Register was not addressable (did not respond). DSR is thus invalid.
4	SEQ ERR	A sequence of control signals was received that the microprocessor was not programmed to handle, or a Data Transfer instruction was executed out of sequence, or a software time-out occurred while waiting for a hardware function, etc. (Table 4-2).
5	DEV PAR ERR	If reading, the channel received a byte with bad parity from a device. If writing, an internal SILO parity error has occurred.
6	LEN ERR	The device did not agree with the channel regarding the number of bytes transferred. LEN ERR is invalidated when UNIT CHK (DSR) is set.
7	IGN LEN ERR	Prevent microprocessor from storing status on LEN ERR.
8	ILL CMD	The Device Command Register contained an illegal code.
9	DSR FLAG	A bit of interest has set in the DSR (Figure 3-10).
10	SENSE BYTE RD ERR	A read error occurred during an attempted Sense (04) operation, or a Channel Control Force instruction.
11	OPER INCOMPL	A Data Transfer operation failed to complete in 10 seconds.

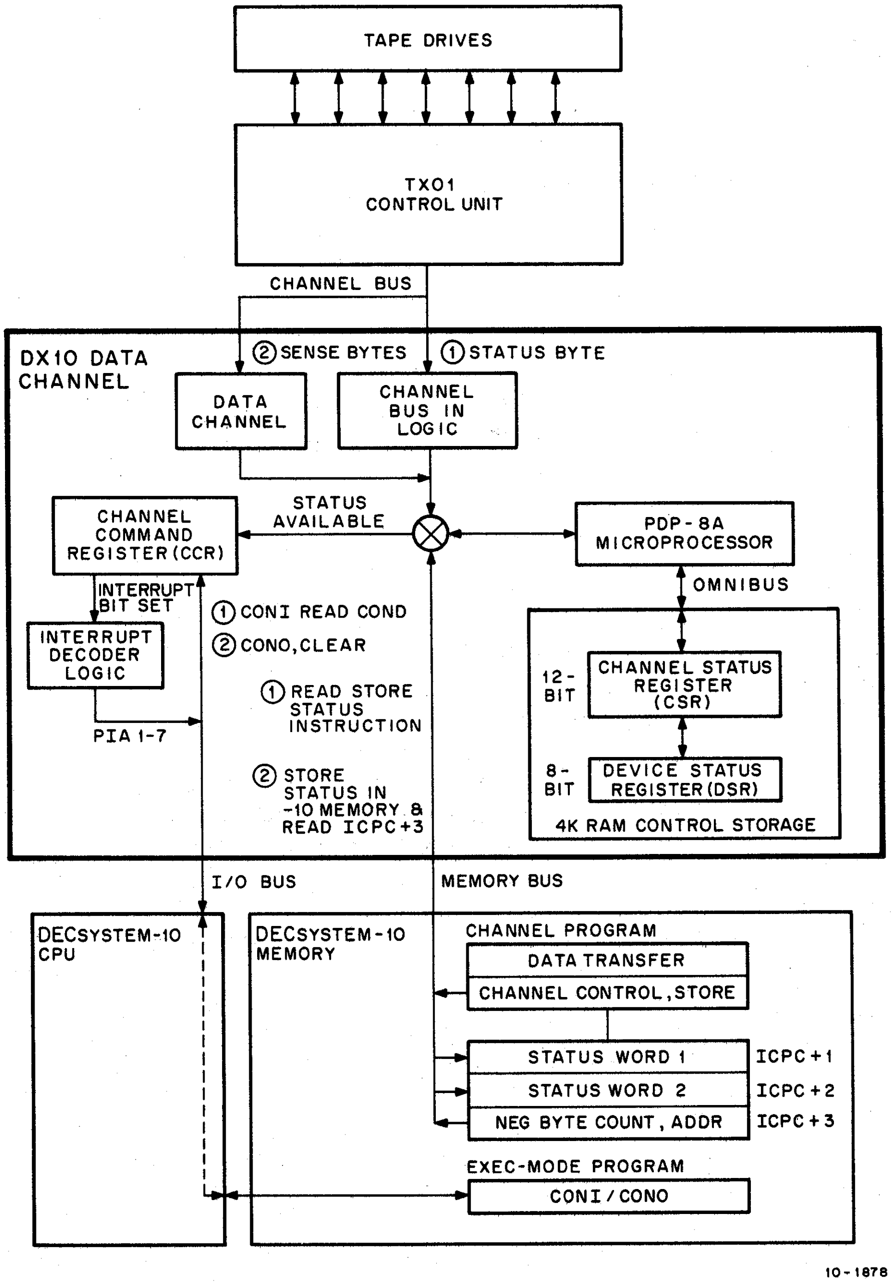


Figure 3-14 Channel/Device Status Sequence

The Status byte has the following format:

Bit Position	Designation
P	Parity
0	Attention
1	Status Modifier
2	Control Unit End
3	Busy
4	Channel End
5	Device End
6	Unit Check
7	Unit Exception

NOTE

The Status byte is transmitted to the DX10 over the Channel Bus in six separate situations:

1. **During the initial selection sequence.**
2. **To present channel-end status at the termination of a data transfer.**
3. **To present the Device End signal and any associated conditions to the DX10. The TX01 remains busy during an operation until the DX10 accepts the Device End status.**
4. **To present Control Unit End or Device End status, which signals that the control unit (or device) that was previously busy and then interrogated is now free.**
5. **To present any previously stacked status when allowed to do so.**
6. **To present any externally initiated status (Attention and Device End because of not-ready-to-ready transition).**

Once accepted by the DX10, any given Status byte is reset and is not presented again.

The DSR is stored in DECsystem-10 memory whenever one of the above described conditions occurs.

Figure 3-15 identifies each of the Device Status byte bits. Table 3-7 describes the conditions represented by each bit, each of which is set automatically by the TX01 before being sent over the Channel Bus.

Table 3-7
Device Status Conditions

Bit	Designation	Description
0	Attention	Not used
1	Status Modifier	Used in conjunction with bit 3. When bit 1 is ON, bit 3 indicates that the TX01 Control Unit (CU) is busy, or an interrupt is pending. When bit 1 is OFF, bit 3 indicates that the tape unit (TU) is busy, or an interrupt is pending.
2	Control Unit End	Control Unit End indicates that the control unit is available for another operation; this bit is rarely used. Control Unit End is set: 1. After completion of every operation during which a CU Busy was signaled. 2. After completion of a control operation which had Channel End in the initial status and during which a Unit Check or Unit Exception was detected while the tape unit was selected. 3. When working with one interface and the other attempts selection.*
3	Busy	Busy indicates the TU or CU (as indicated by Bit 1) cannot execute a command or instruction because of a pending interrupt, or because the unit is currently occupied with a previously initiated operation.
4	Channel End	Channel End indicates that the channel interface is no longer required for the operation. It is set when a Read, Read Backward, Write, Sense, TIE, or Set Diagnostic command has been completed, or when a control command has been accepted.
5	Device End	Device End is set on completion of an I/O operation at the tape unit, or when the tape unit is manually made ready. Device End also is set when: 1. The tape unit becomes ready after selection was attempted when the unit was not ready. 2. When a Rewind-Unload operation is completed at the tape control level. 3. When a control command other than DSE, REW or RUN is completed at the tape unit level. 4. Along with Channel End, at the completion of other commands.

*Not possible with presently available configurations.

Table 3-7 (Cont)
Device Status Conditions

Bit	Designation	Description
5 (cont)		5. If a tape unit performing an operation becomes not ready (for example, power off, manual reset). 6. When a tape unit becomes ready after selection was attempted while it was busy. 7. On the first initial selection after the tape unit becomes ready.
6	Unit Check	Unit Check indicates the subsystem has encountered an unusual condition. The cause of a Unit Check is stored as sense data, which is available to the program in response to a Sense command. Unit Check is set when any of the following occurs: 1. Any sense byte 0 error indicator is set. 2. A Read Backward, BSB, or BSF operation is initiated into or at load point. 3. A Run operation is completed at the CU level. 4. Bit 7 of sense byte 1 (not capable) is set. 5. Tape unit is not ready, or drops 'ready' during a Data Security Erase, Rewind, or if Device End is armed and a load failure occurs. 6. Bit 3 of sense byte 5 (PE ID Burst Check) is set.
7	Unit Exception	Unit Exception is set when the tape control detects a condition that usually does not occur, and does not necessarily indicate an error. Unit Exception is set: 1. If Tape Indicate is on during a Write, Write Tape Mark, or Erase Gap operation. 2. If a tape mark is detected during a Read, Read Backward, Forward Space Block, or Backward Space Block operation.

NOTES:

1. The tape unit sets Tape Indicate when it senses the trailing edge of the end-of-tape (EOT) reflective marker while tape is moving forward.

2. A Service Out response to Status In clears Unit Exception from the status byte.

3. A subsequent Write, Write Tape Mark, or Erase Gap command causes Unit Exception to appear again along with Device End, if Tape Indicate is not reset.

4. A command which moves tape backward, so that the tape unit again senses the trailing edge of the EOT marker, resets Tape Indicate; hence, Unit Exception may not occur again.

Table 3-7 (Cont)
Device Status Conditions

NOTES:
(cont)

5. Rewinding or unloading tape also resets Tape Indicate.

6. Data Security Erase sets Unit Exception. The operation is complete when the tape unit reaches Tape Indicate.

7. A Service Out response to Status In clears Unit Exception from the status byte. In a read and space block operation, Unit Exception is not set again; therefore, it is important to handle a Unit Exception when it is recognized.

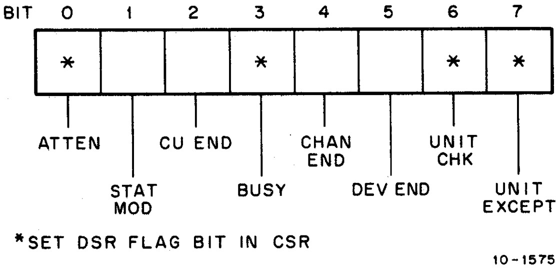
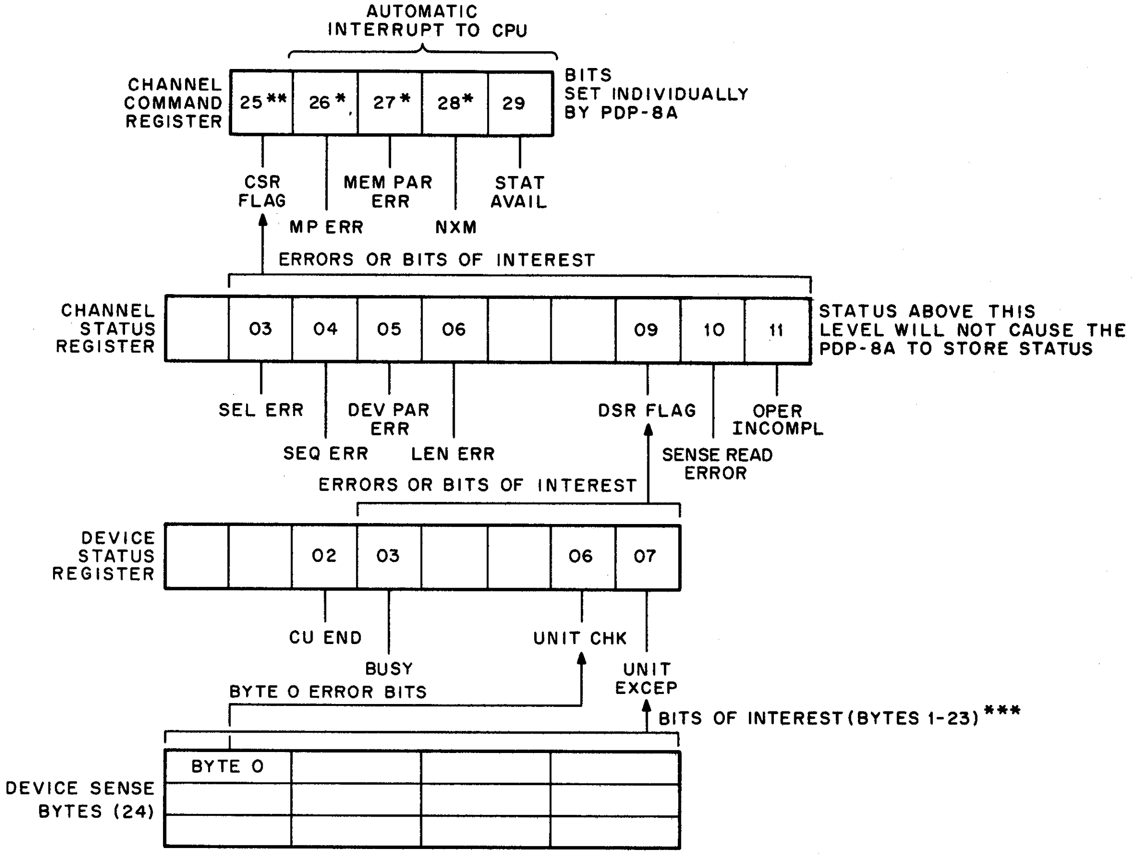


Figure 3-15 Device Status Byte

3.4.8 Status Levels/Relationships

Referring to Figure 3-15, if the microprocessor finds, after reading a device status byte from the TX01, that any of four special Status bits is set (Attention, Busy, Unit Check, or Unit Exception), the microprocessor sets the DSR flag in the CSR before storing status in DECsystem-10 memory. This, in turn, causes the PDP-8/A to set the CSR flag in the Channel Command Register before status is stored. If, during the Status Store operation, the microprocessor finds a negative byte count and an extended status store address in ICPC+3, it performs an Extended Store operation during which it reads the specified number of 8-bit Device Sense bytes from the TX01 Control Unit over the Channel Bus and stores them (4 bytes per word) in DECsystem-10 memory starting at the location specified in ICPC+3. These Sense bytes are read into DECsystem-10 memory through the normal data channel data path. They indicate specific conditions within the addressed tape unit which caused a Unit Check (error condition) or Unit Exception (condition of interest) bit to be set in the Device Status byte (detected when it was read into the DSR by the microprocessor). Figure 3-16 shows the conditions (errors or bits of interest) under which the Unit Check or Unit Exception bits in the DSR are set. As indicated in the illustration, each level of status causes the microprocessor to set a flag in the next higher level of status and store status in DECsystem-10 memory (with the exception of the CSR).

When the CSR flag is set in the CCR by the microprocessor, the MP continues to process instructions in the Channel Program. When the PDP-8/A again stores status, or when an interrupt condition is encountered, either the microprocessor (by setting STAT AVAIL) or the interrupt condition will interrupt the CPU. When the operating program reads the CCR with a CONI, the CSR flag informs the operating program of the existence of channel status in memory.



*INDICATES AN ERROR TO THE MONITOR CAUSING IT TO READ STATUS.

**AFTER SETTING THE CSR FLAG, THE PDP-8A SETS STAT AVAIL.

***OR'ED TO SET A SPECIFIC BYTE 0 ERROR BIT

10-1572

Figure 3-16 Status Levels and Relationships

Figure 3-17 summarizes each of the 24 (maximum) Sense bytes read by the microprocessor into DECsystem-10 memory during the Extended Store operation. Table 3-8 explains the conditions represented by the individual bits in each of the bytes that contain status information.

3.5 PROGRAMMING CONSIDERATIONS AND EXAMPLES

3.5.1 Channel Programming

Figure 3-18 illustrates a sample Channel Program that will write one record on a pre-positioned tape. Sense information is read after the transfer, although this is not necessary, because an error condition causes the PDP-8/A to store status at ICPC+1-ICPC+3 and halt (unless ICPC+3 contains a negative Data Transfer instruction, in which case the microprocessor reads the number of Sense bytes specified in the Data Transfer instruction and stores them in the specified memory locations before halting).

Figure 3-19 shows how a record may be read on one drive while another drive is rewinding. This is also an example of the Channel Program instruction sequence starting at the ICPC location. Notice that the Store Status, Halt instruction is overwritten by Status Word 1 after Device 1 is finished rewinding.

Byte	Bit 0 (8)	Bit 1 (4)	Bit 2 (2)	Bit 3 (1)	Bit 4 (8)	Bit 5 (4)	Bit 6 (2)	Bit 7 (1)
0	COMMAND REJECT	INTERVEN. REQ'D	BUS OUT CHECK	EQUIPMENT CHECK	DATA CHECK	OVERRUN	WORD COUNT ZERO	DATA CONVERTER CHECK
1	NOISE	TU STATUS A	TU STATUS B	7-TRK	LOAD POINT	SELECTED & WR STATUS	FILE PROTECTED	NOT CAPABLE
2				TRACK IN ERROR BYTE				
3	R/W VRC	MTE/LRC	SKEW ERROR	END DATA CK/CRC	ENV CK/ SKEW REG VRC	1600 BPI SET IN TU	BACKWARD	C COMPARE
4		REJECT TU	TI	WRITE TGR VRC	START RD CHECK	LWR		
5	NEW SUB-SYSTEM	NEW SUB-SYSTEM			START RD CHECK		DIAGNOSTIC MODE	RPQ
6					TAPE UNIT MODEL IDENTIFICATION			
7					DATA SECURITY ERASE			
8								
9		VELOCITY CHECK						TCU RESERVED
10				WTM NOT DETECT BLOCK				
11								
12								
13	CU FEATURES		CONTROL UNIT UNIQUE IDENTIFICATION (HI-ORDER PART OF SERIAL NO.)					
14			CONTROL UNIT UNIQUE IDENTIFICATION (LO-ORDER PART OF SERIAL NUMBER)					
15								
16								
17	2CS FEATURE	SW FEAT. IDENT. (0–7) LO-ORDER			REFLECTS DIAGNOSTIC RELEASE LEVEL OF TCU			
18								
19	BUSY STATUS, LO-ORDER TAPE UNITS TU 7	TU 6	TU 5	TU 4	TU 3	TU 2	TU 1	TU 0
20	BUSY STATUS, HI-ORDER TAPE UNITS TU F	TU E	TU D	TU C	TU B	TU A	TU 9	TU 8
21								
22								
23								

10-1888

Figure 3-17 TU70 Tape Drive Sense Byte Bit Summary

Table 3-8
TX01/TU70 Sense Byte Bit Definitions

Bit	Designation	Description
Sense Byte 0 (Unit Check)		
0	Command Reject	Command Reject is set: 1. When a Write, Write Tape Mark, Erase, or Loop-Write-To-Read (LWR) command is issued to a file-protected tape unit. 2. When an unidentified command code is received by the CU. 3. If a DSE command is issued and does not immediately follow an Erase Gap operation. 4. If Reserve Sense or Release Sense is issued* to a tape control unit that does not have the programmed, two-channel switch feature, or if Reserve Sense or Release Sense is issued other than as the first command in a chain sequence.
1	Intervention Required	Intervention Required is set whenever the address tape unit is "not ready" or nonexistent. **NOTE** **Dropping "ready" while performing a command causes Unit Check, along with any other ending status.**
2	Bus Out Check	Bus Out Check is set whenever Bus Out has incorrect (even) parity during command or data byte transfer.
3	Equipment Check	Equipment Check is set: 1. When sense byte 4, bit 1 (Reject TU) is set. 2. When sense byte 4, bit 6 is set if the tape unit is performing an operation (TU Check).
4	Data Check	Data Check is set: 1. When sense byte 1, bit 0 is set (Noise). 2. When sense byte 3, bit 0, 1, 2, 3, 4, or 7 is set during a read or write-type operation. 3. When end-of-block is sensed before any data bytes are detected during a PE (1600 bpi) Read or Read Backward operation. (This condition also sets sense byte 1, bit 0.) 4. With bit 3 in sense byte 4; if 2803* mode, with bit 4 in sense byte 4. 5. If 3803 mode, with bit 4 in sense byte 5.

*Not used with currently available configurations.

Table 3-8 (Cont)
TX01/TU70 Sense Byte Bit Definitions

Bit	Designation	Description
Sense Byte 0 (Unit Check) (Cont)		
5	Overrun	Overrun is set when service is requested but data cannot be transferred during a Read, Write, or Read Backward operation. Data transfer stops as soon as an overrun is detected. **NOTE** **Data Check during overrun suppresses the overrun indication.**
6	Word Count Zero	Word Count Zero is set: 1. If data transfer is stopped before the tape control unit received the first byte of data during a Write operation. (Channel responded to tape control's first Service In with Command Out.) 2. When tape control receives a Halt I/O command after receipt of a Write command but before tape motion commences.
7	Data Converter Check*	When operating in Data Converter mode for a Read operation, Data Converter Check (DCC) is set to indicate that the last byte (or only byte) sent to the channel was padded with 0s. The following conditions will cause a DCC error to occur on records which are not an even multiple of four characters. 1. If one character is read from tape, and the byte sent to the channel had bits 6 and 7 with 0s. 2. If two characters are read from tape, and two bytes are sent to the channel with the second byte padded with 0s in bits 4, 5, 6, and 7. 3. If three characters are read from tape, and three bytes are sent to the channel with the third byte padded with 0s in bits 2, 3, 4, 5, 6, and 7. **NOTE** **Data Converter Check cannot occur in a Read Backward operation.**
Sense Byte 1		
0	Noise	Noise is set: 1. If a Data Check occurs during a 1600 bpi Read or Read Backward operation. 2. If no data is transferred on an 800 or 1600 bpi Read or Read Backward operation. 3. If data is detected during an NRZI read stop delay.

*Not used with currently available configurations.

Table 3-8 (Cont)
TX01/TU70 Sense Byte Bit Definitions

Bit	Designation	Description
Sense Byte 1 (Cont)		
1	TU Status A	TU Status A is set when an addressed tape unit is selected, ready, and not busy.
2	TU Status B	TU Status B is set when an addressed tape unit is not ready, is rewinding, or is under control of another tape control. Assuming no outstanding device end status, bits 1 and 2 determine response to initial selection as follows:

TU Status A	TU Status B	TU Status	Response to Initial Selection
Off	Off	Nonexistent	Unit Check
Off	On	Not Ready	Unit Check, set for Device End
On	Off	Ready and not rewinding	Clean status

		Unit Check is not signaled for a sense operation. Following a Unit Check or Busy indication, Device End is signaled when the tape unit becomes ready and is not either rewinding or switched.
3	Seven Track	Seven Track is set when the selected tape unit has the 7-track feature.
4	Load Point	Load Point is set when the selected tape unit is at the beginning of a tape.
5	Selected and Write Status	Selected and Write Status is set when the selected tape unit is in write mode.
6	File Protected	File Protected is set when the selected tape unit is in read (file-protected) mode. A tape unit in file-protected mode (no write enable ring) cannot perform write-type commands.
7	Not Capable	Not Capable is set: 1. When a 3800-III subsystem without NRZI capability attempts to read an NRZI tape that was written without a PE identification burst at load point. 2. When an attempt is made to read or write on a 7-track tape unit and the tape control does not have the 7-track NRZI feature. 3. When an attempt is made to read or write NRZI on a 9-track tape unit and the tape control does not have the 9-track NRZI feature.

Table 3-8 (Cont)
TX01/TU70 Sense Byte Bit Definitions

Bit	Designation	Description
Sense Byte 2		
		This sense byte contains the track-in-error (TIE) indicator bits set at the end of a Read, Read Backward, Write, or Loop-Write-to-Read (LWR) command. For PE operations, sense byte 2: 1. Indicates the tracks that have amplitude loss or phase errors. 2. On Read or Read Backward operation without a Data Check, a single bit indicates a track in error. The data is corrected, however, during the Read Backward operation. For NRZI 9-track Write or LWR operations, bits 6 and 7 are ON. In Read or Read Backward operations: 1. A single bit and Data Check indicate the track in error. 2. Bits 6 and 7 with Data Check indicate an uncorrectable error pattern. 3. Bits 6 and 7 without Data Check indicate normal operation. During NRZI 7-track Read or Read Backward operations, the Track-in-Error byte is always 03 (tracks 6 and 7).
Sense Byte 3 (Data and Equipment Checks)		
0	Read/Write Vertical Redundancy Check (R/W VRC)	R/W VRC is set when a Vertical Redundancy Check that cannot be corrected occurs during a Read or Read Backward operation, or during a PE Write if a VRC occurs without an Envelope Check.
1	Multiple Track Error/Longitudinal Redundancy Check Error (MTE/LRC)	MTE/LRC error is set: 1. During a PE Read, Read Backward, or PE Write operation when there is a weak signal in more than one track. Data is incorrect. 2. In NRZI, when a Longitudinal Redundancy Check occurs during a Read, Read Backward, Write, or Write Tape Mark operation.
2	Skew Error	Skew Error is set: 1. In PE mode, when excessive skew is detected during a Read or Read Backward operation. 2. In NRZI mode, when excessive skew is detected during a Write, Write Tape Mark, or Erase operation.

Table 3-8 (Cont)
TX01/TU70 Sense Byte Bit Definitions

Bit	Designation	Description
Sense Byte 3 (Data and Equipment Checks) (Cont)		
3	End Data Check/ Cyclic Redundancy Check (CRC)	End Data Check/CRC is set: 1. During a PE Read or Read Backward, when Sync Burst following a data block is not properly recognized, or is erroneously recognized before the actual end of data. 2. During a PE Write, when Sync Burst following a data block is not properly recognized. 3. During an NRZI Read or Read Backward, when a CRC register error occurs.
4	Envelope Check/ Skew Register VRC	Envelope Check/Skew Register VRC is set: 1. During a PE Write, when at least one track had low amplitude while writing. 2. During an NRZI Write, Write Tape Mark, or Erase, when a byte in the auxiliary register had incorrect parity.
5	1600 bpi set in TU	Set when the selected tape unit is in phase-encoded mode.
6	Backward	Backward is set when the selected tape unit is in backward mode.
7	C Compare	C Compare checks that correct parity (odd or even) is maintained by the tape control unit while processing data.
Sense Byte 4		
0		Not used
1	Reject Tape Unit	Reject Tape Unit is set if the selected tape unit dropped "ready" during performance of a Tape Motion command, or if a change in read status occurs.
2	Tape Indicate	Tape Indicate is set whenever the end-of-tape reflective marker is sensed during a Forward Tape operation (3803 mode only).
3	Write Trigger VRC	Write Trigger VRC is set if the byte written by the Write Triggers has incorrect parity.
4	Start Read Check	For 2803 mode* only (see description for Sense Byte 5, bit 4).
5	LWR	Present during Loop-Write-to-Read operations (3803 mode only).
6,7		Not used.

*Not used with currently available configurations.

Table 3-8 (Cont)
TX01/TU70 Sense Byte Bit Definitions

Bit	Designation	Description
Sense Byte 5		
0	New Subsystem	Always 0.
1	New Subsystem	Always present in 3803 mode. Always 0 in 2803 mode.*
2		Not used
3		Not used
4	Start Read Check	For 3803 mode only. Present when IBG becomes active before the Beginning Ones marker but after BOB on a Read operation.
5		Not used
6	Diagnostic Mode	Always 0 if in 3803 mode; always present in 2803* mode.
7	Not Used or RPQ	Set when operating in 9-track, multidensity mode. **NOTE** **Sense Bytes 6–23 are used only in 3803 mode.**

Sense Byte 6

Bit	Designation	Description				
3		Not used				
		75 in./sec	**100 in./sec**	**125 in./sec**	**200 in./sec**	**250 in./sec**
4		0	0	0	0	0
5		0	0	1	1	1
6		1	1	0	0	0
7		1	1	0	1	1

Sense Byte 7

Bit	Designation	Description
0–3		Not used
4	Data Security Erase	Does not cause Ready to drop. Will not be on at normal completion of DSE (TU reaches TI).
5–7		Not used

*Not used with currently available configurations.

Table 3-8 (Cont)
TX01/TU70 Sense Byte Bit Definitions

Bit	Designation	Description
Sense Byte 9		
0		Not used
1	Velocity Check	This bit is on when the velocity variations of the capstan exceed the specified limits (too fast and too slow).
2–6		Not used
7	TCU Reserve	Present when the tape control is in reserved status only if the Two-Channel Switch* feature is installed.
Sense Byte 10		
0–2		Not used
3	WTM Not Detect Block	Present when block is not detected for sufficient length of time on a WTM operation.
4–7		Not used
Sense Byte 13		
0	CU Features	00: Basic CU; 01: 7-track NRZI
1	CU Features	10: 9-track NRZI; 11: 7- and 9-track NRZI
2	CU ID High	Control Unit serial number, high order bits
3	CU ID High	
3	CU ID High	
4	CU ID High	
5	CU ID High	
6	CU ID High	
7	CU ID High	

*Not used with currently available configurations.

Table 3-8 (Cont)
TX01/TU70 Sense Byte Bit Definitions

Bit	Designation	Description
Sense Byte 14		
0	CU ID Low	Control Unit serial number, low order bits
1	CU ID Low	
2	CU ID Low	
3	CU ID Low	
4	CU ID Low	
5	CU ID Low	
6	CU ID Low	
7	CU ID Low	
Sense Byte 17		
0	2CS Feature	Programmed 2-channel switch feature present.
1–3	SW Features	000: 1 X 8 Device Switch Lo (addresses 0–7)* 001: 2 X 8 Device Switch Lo (addresses 0–7)* 010: 3 X 8 Device Switch Lo (addresses 0–7)* 011: 4 X 8 Device Switch Lo (address 0–7) 100: Remote Control Unit 101: 2 X 8 Device Switch Hi (addresses 8–F)* 110: 3X 8 Device Switch Hi (addresses 8–F)* 111: 4 X 8 Device Switch Hi (addresses 8–F)* **NOTE** **All units ship as 1 X 8 (000), without 2CS feature.**
4		Not used
5	CU EC Level	Reflects diagnostic release level of control unit.

*Device address for tape units physically attached to the control unit.

Table 3-8 (Cont)
TX01/TU70 Sense Byte Bit Definitions

Bit	Designation	Description
Sense Byte 17 (Cont)		
6,7		Not used
Sense Byte 19		
0	Busy Status Lo Order TUs	Primed for Device End Tape Unit 7.
1		Primed for Device End Tape Unit 6.
2		Primed for Device End Tape Unit 5.
3		Primed for Device End Tape Unit 4.
4		Primed for Device End Tape Unit 3.
5		Primed for Device End Tape Unit 2.
6		Primed for Device End Tape Unit 1.
7		Primed for Device End Tape Unit 0.
Sense Byte 20		
0	Busy Status High Order TUs	Primed for Device End Tape Unit F.
1		Primed for Device End Tape Unit E.
2		Primed for Device End Tape Unit D.
3		Primed for Device End Tape Unit C.
4		Primed for Device End Tape Unit B.
5		Primed for Device End Tape Unit A.
6		Primed for Device End Tape Unit 9.
7		Primed for Device End Tape Unit 8.

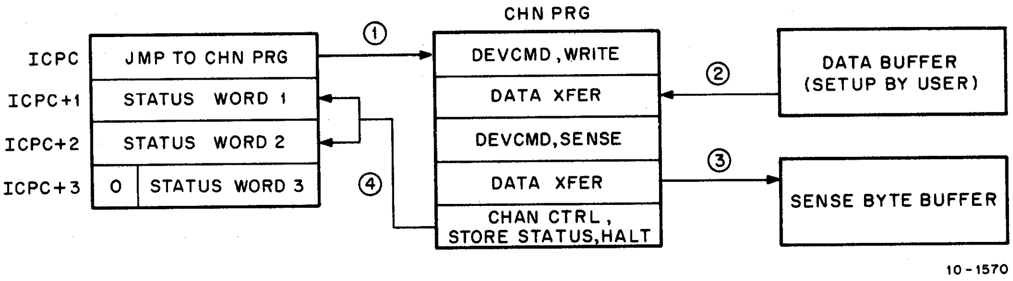
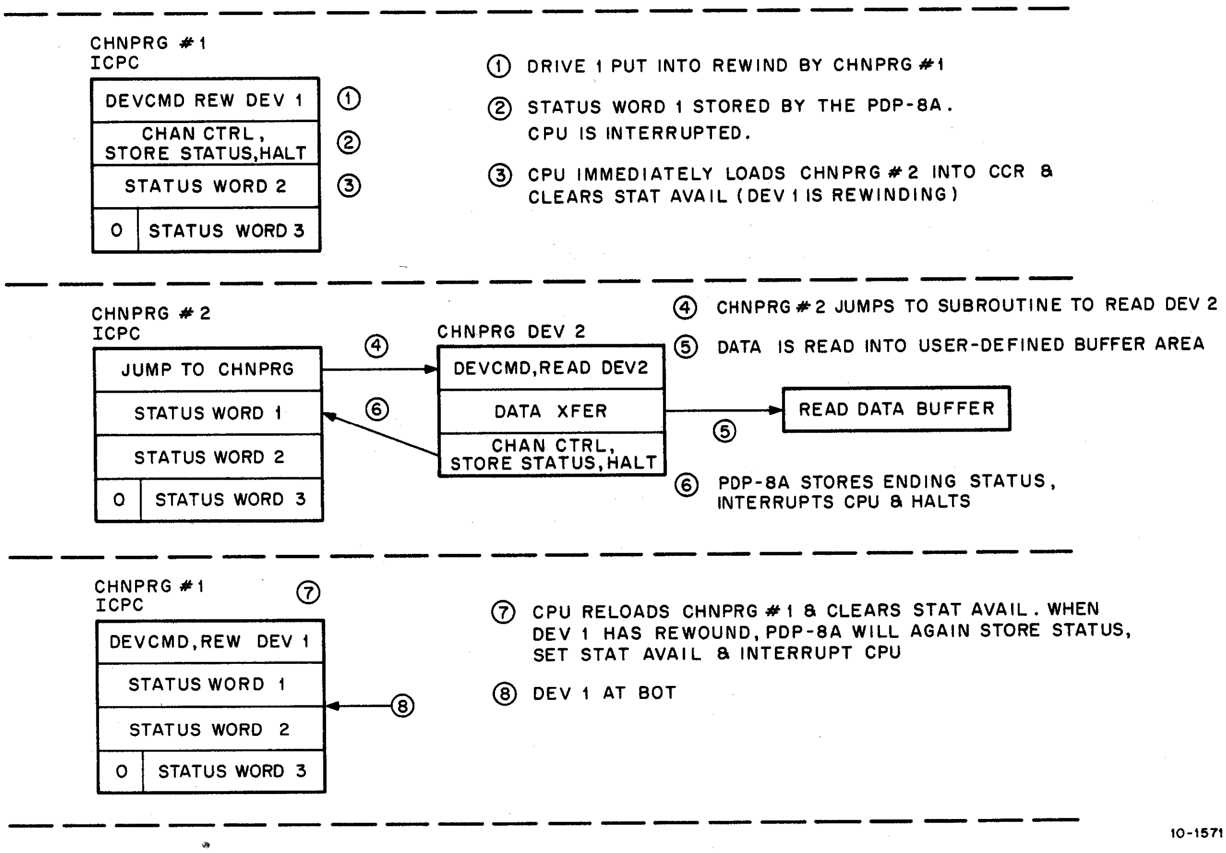


Figure 3-18 Write Record/Read Device Sense



(1) DRIVE 1 PUT INTO REWIND BY CHNPRG #1

(2) STATUS WORD 1 STORED BY THE PDP-8A.
CPU IS INTERRUPTED.

(3) CPU IMMEDIATELY LOADS CHNPRG #2 INTO CCR &
CLEARS STAT AVAIL (DEV 1 IS REWINDING)

(4) CHNPRG #2 JUMPS TO SUBROUTINE TO READ DEV 2

(5) DATA IS READ INTO USER-DEFINED BUFFER AREA

(6) PDP-8A STORES ENDING STATUS,
INTERRUPTS CPU & HALTS

(7) CPU RELOADS CHNPRG #1 & CLEARS STAT AVAIL. WHEN
DEV 1 HAS REWOUND, PDP-8A WILL AGAIN STORE STATUS,
SET STAT AVAIL & INTERRUPT CPU

(8) DEV 1 AT BOT

Figure 3-19 Simultaneous Operations

In Figure 3-19, between events 2 and 3, the CPU should test the DEV END bit in status word 1. If set, the drive was at BOT when the Rewind command was given, and will not interrupt again.

As shown in the illustration, the different Channel Programs residing in DECsystem-10 memory can be pointed to at the time the channel is started. This allows status to be stored in memory at a point related to the Channel Program just executed.

Figure 3-20 shows how the DX10 can retrieve Device Sense bytes after an error occurs, but before the CPU is informed of the error. CHNPRG (①) reads a record and stores the data in a memory data buffer that is split into core blocks ② and ③ . If there is no error during the Read operation, the microprocessor stores normal status and halts with the CSR flag (Paragraph 3.4.8) not set. If UNIT CHK or UNIT EXCEP occurred during the read transfer, an Extended Status Store is executed [if ICPC+3 (④) contains a negative byte count and address] after the error status has been stored. During the Extended Store operation, the PDP-8/A requests the specified number of Sense bytes from the control unit and stores them (⑤) in the area specified by the address in ICPC+3. If the Sense Byte Retrieval and Storage operation is successful, the PDP-8/A sets STAT AVAIL and halts without attempting to store further status. If the Sense operation is unsuccessful, the failing status for the Sense operation is stored in the usual manner (Paragraphs 3.4.7.1 and 3.4.7.2). Another store status routine cannot occur until another CONO is issued by the CPU. Before doing this, however, the CPU would probably enter the sense information into the system error log.

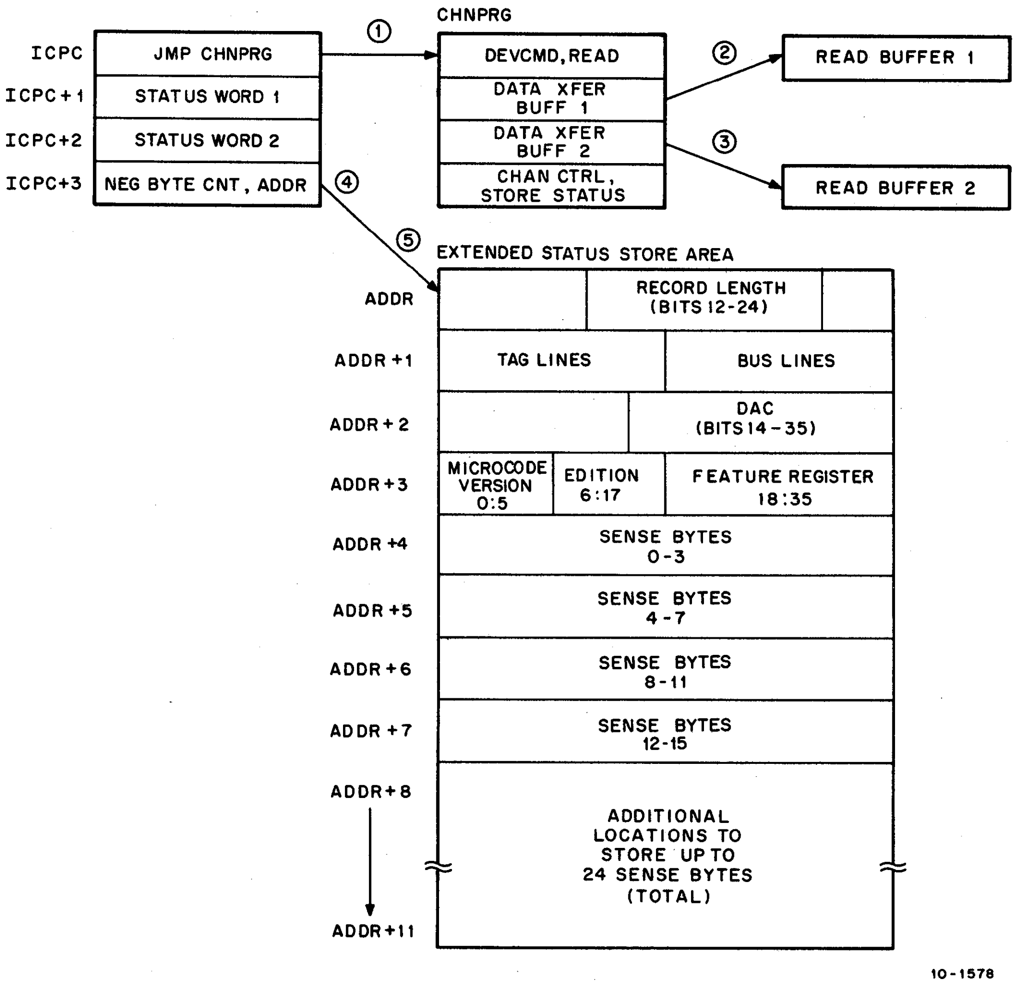


Figure 3-20 Extended Status Sense Byte Store on Error

3.5.2 Executive Mode Programming Using the IBUS

NOTE
These examples assume the PDP-8/A is halted.

To read the contents of the Feature Register:

DATAO PDC,[1,,14] — Load the RSEL Register with a 14 using a DATAO sending a data word with bit 17 set to specify loading the RSEL, and a 14 in bits 32 to 35.

DATAI PDC, — Read the contents of the Feature Register with a DATAI. The data word returned will contain a 14 in bits 14–17 and the register contents in bits 18–35.

To write 7070 into PDP-8/A memory location 3720:

DATAO PDC,[1,,4] — Load RSEL with a 4.

DATAO PDC,[0,,0] — Write 0s into the register. This is to clear the CONT bit which must be cleared before any memory references can be attempted.

DATAO PDC,[1,,5] — Load RSEL with a 5.

DATAO PDC,[0,,3720] — Write a 3720 into the CPMA.

DATAO PDC,[0,,4] — Load RSEL with a 4.

DATAO PDC,[0,,17070] — Set the Deposit bit and write data 7070 into the memory buffer. This DATAO will cause a PDP-8/A memory write cycle into location 3720 and will cause the CPMA Register to increment by 1.

To read the data from PDP-8/A memory locations 3000 and 3001:

DATAO PDC,[1,,4] — Load RSEL with a 4.

DATAO PDC,[0,,0] — Clear CONT.

DATAO PDC,[1,,5] — Load RSEL with a 5.

DATAO PDC,[0,,3000] — Write 3000 into CPMA.

DATAO PDC,[1,,4] — Load RSEL with a 4.

DATAO PDC,[0,,20000] — Set EXAM. This DATAO will cause a PDP-8/A memory read cycle of location 3000 and put the data from that memory location into its memory buffer. The CPMA is also incremented to 3001.

DATAI PDC, — Read the data from the PDP-8/A memory buffer. The data word is returned in bits 24–35.

DATAO PDC,[0,20000] — Set EXAM again. This DATAO will cause another read memory cycle, this time of location 3001.

DATAI PDC, — Read the data from the PDP-8/A memory buffer.

To start the PDP-8/A at location 200:

```
DATAO PDC,[1,,4]          Load RSEL with a 4.

DATAO PDC,[0,,0]          Clear SS, HLT, and CONT.

DATAO PDC,[1,,5]          Load RSEL with a 5.

DATAO PDC,[0,,200]        Write 200 into CPMA.

DATAO PDC,[0,,4]          Load RSEL with a 4.

DATAO PDC,[0,,100000]     Set CONT. The PDP-8/A is now running.
```

3.5.3 TU70 Program Assembly

The following procedures relate to assembling a program to cause a tape unit to write a record and store status in memory.

1. Set up four separate blocks of memory as shown in Figure 3-21.

2. Send a CONO to the DX10 containing the ICPC address, CLEAR, CONT, and a PIA assignment.

3. The Channel Program will run and perform the Write operation. The DX10 will cause an interrupt to the PDP-10 with STAT AVAIL set. At that time, a CONI should be used to determine if the Write operation completed without error. With no error, ICPC+1, +2, and +3 will contain information about the operation. If an error did occur, ICPC+1 and +2 will contain information about the error and more information, including the Sense Bytes, will be in the 24-byte buffer for status.

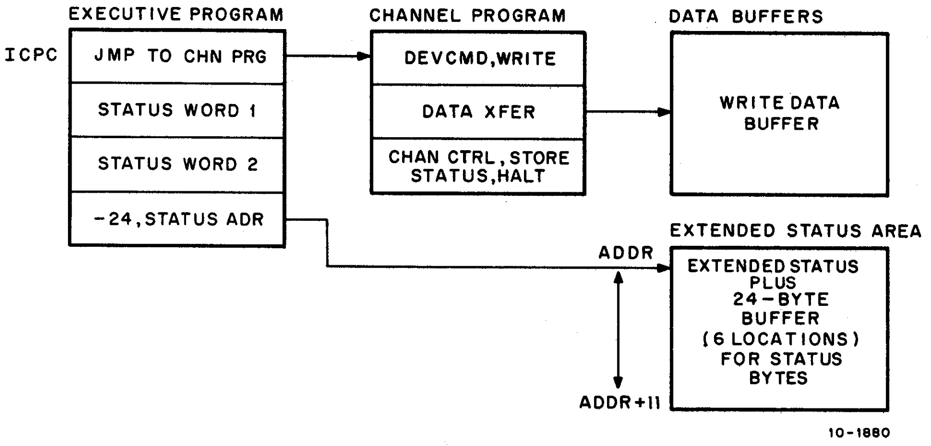


Figure 3-21 Tape Write Sequence

CHAPTER 4
PDP-8/A MICROCODE

4.1 GENERAL

The DX10 microcode is a PDP-8 program which controls the DX10 to allow it to function as an I/O processor.

The logic within the DX10 is controlled by a series of special IOTs, which allow setting and clearing of flip-flops, selection of registers, and loading or reading of these registers.

The microcode will cause the microprocessor to perform these sequences in a manner that will allow it to fetch and process commands from DECsystem-10 memory. Typical commands direct the microprocessor to read and write magtape.

The microcode is loaded into the 4K control storage memory by a DECsystem-10 program using special DX10 hardware that allows loading and reading the memory of the PDP-8/A using DATAO/DATAI commands.

It is then started at PDP-8/A memory location 200. The microprocessor first initializes the DX10 and then starts the MP error timer. After starting the timer, the microprocessor enters an idle loop where it waits for further direction from the DECsystem-10 monitor. This direction comes from the Channel Command Register. The monitor can put four basic commands into this register:

1. If the monitor sets both Clear and Continue in the Channel Command Register, this directs the microprocessor to start the Channel Program at the address specified by the ICPC Register.

2. If the monitor sets only Continue, this directs the microprocessor to continue the Channel Program at the address specified by the CPC Register.

3. If the monitor sets only Clear, this directs the microprocessor to perform a System Reset.

4. If the monitor sets the Status Request bit, this directs the microprocessor to perform a Store Status operation. The address of the status buffer comes from the address specified by the ICPC Register.

Once a Channel Program is started, it continues running until the tape system encounters an error condition or a channel Store Status command is fetched with the Go bit off.

If the microprocessor stops the Channel Program, it does so by first clearing the Continue bit, then storing status, generating a system interrupt, and finally returning to the idle loop to await further instructions from the system monitor.

4.2 MAJOR SUBROUTINES

The microcode contains various subroutines which perform the following operations:

1. A loop, during which the microprocessor waits for a command from the monitor.

2. A Channel Program section where it fetches and decodes Channel Program commands.

3. A magtape initializing section which performs the required coding for starting a magtape unit.

4. A magtape terminating section which performs the required coding for stopping a magtape unit.

5. Special coding to handle both fatal errors and operational errors.

Each of the operating PDP-8/A subroutines is listed by mnemonic and function performed in Table 4-1. The following major subroutines are summarized in this chapter.

Idle Loop

STPRG

DOCMD

BLKDN

CMDST

Device Command Termination Routines
SRVDEV
GETSTS

Error Handling Routines and Error Codes
STORE
STREXT

Bootstrap Operation

Read-in

4.2.1 The Idle Loop

In the idle loop, the PDP-8/A waits for a command from the system monitor by continuously testing three bits in the Channel Command Register (Clear, Continue, and Status Register). If one of these bits is set, the microprocessor performs the specified command. The idle loop flow is as follows:

1. Reenable the MP ERR timer to prevent time-out.

2. Check to see if Status In is present (used in spacing tape). If it is set, stop the device.

3. Check to see if Request In is present (used only in rewind). If it is set, stop the device and store rewind status.

4. Check Continue; if set, start the Channel Program.

5. Check Clear; if set, perform system reset.

6. Check Status Request; if set, store status.

7. Display an incrementing count in Register 17.

8. Loop back to step 1.

Table 4-1
DX10/PDP-8/A Microcode Subroutine

Subroutine Mnemonic	Subroutine Function Performed
IDLE, IDLE1, IDLE2	Idle loop.
RESET, RESETA, RESETB	System, tape drive resets.
GETRWS	Get rewind or unload status.
SETBIT	Set bit in CSR.
GETSTS, GET1	Service TX01 STA IN or REQ IN.
GETBC	Get the byte count from the register into BCL and BCH.
TENSEC, TENEX	Time-out 10 seconds.
DIV, DIV1–DIV3	Perform double-precision divide on byte count in BCH and BCL. Dividend is negative and is stored in the AC (BCH) and MQ (BCL).
STPRG, STPRG1, IFETCH, DECODE, STEXT	Start up Channel Program.
DOCMD, DOCMDA–B, DSRERR	Do Device command.
SRVDEV, DOCMD1–2	Handle Device command termination.
HALTNG	Halt Channel Program.
PRGER0–PRGER5	Process microcode errors.
CHNCHK, CHNCHK1	Check for command chaining requirement.
ILGCHK, ILG1	Check for illegal commands.
CMDTBL	Legal command table.
CALCBC, CLCBC1	Calculate CBC Register contents.
CALCSR	Calculate CSR contents.
PUTLNK	Put bit to be tested into link.
DBLSUB	Perform double-precision subtraction of divisor from byte count.
ADDBYT	Accumulate the byte count.

Table 4-1 (Cont)
DX10/PDP-8/A Microcode Subroutines

Subroutine Mnemonic	Subroutine Function Performed
IMMCHK, IMMCK1	Check for rewind or unload.
PRESET, PRESET1, PTABL, PTABLS, PTABL4, PTABL6, FMTBL, MDVSR, DVSR, DATFMT	Calculate byte preset value.
PRGJMP	Perform Channel Program jump.
XFRPC	Transfer ICPC to CPC.
STRREQ	Store program requested Status Type 3.
STORE, STORE1, STREXT	Store status in ICPC+1, +2, and +3 or EXTENDED status area.
MEMRD	Perform DECsystem-10 memory Read operation.
MEMWR	Perform DECsystem-10 memory Write operation.
STRSTA	Perform Status Store operation as specified by channel control instruction.
STPDEV	Stop device upon receipt of STA IN.
PFETCH, PFA, PF1–2	Prefetch Channel Program to optimize Data Transfer instruction storage.
SETUP, SET1	Set up CBC register and Byte Counter.
CHKCMD, CKEXT	Check for Device Command or Channel Control Store instructions.
STRCMD	Get the instruction in 8R0 and store it in control storage memory.
STRLEN	Store record length in control storage memory.
LBCDA	Load the Byte Counter and the Data Address Counter.
BLKDNI, BLKDN, BLKDN1, BLKEX, BLKDB, SETSTP, RDALL	Service a Block Done interrupt.
WAITI, WAIT, WAITA–B	Wait for Block Done or STA IN.

Table 4-1 (Cont)
DX10/PDP-8/A Microcode Subroutines

Subroutine Mnemonic	Subroutine Function Performed
GETCMD	Get next Channel Control instruction from 8R0.
SHRTRC, RSTDA	Service short record.
CMDST, CMDSTA, CMDST1–3, CMDEX, CMDERR, CMDBSY, CHKSPR, CKSPR1–2	
CONO, CONSO1–2	Test tag lines; skip if one is set.
CONSZ, CONSZ1–2	Test tag lines; skip if all are reset.
CHKPAR	Test for Channel Bus parity error.
ERRCHK, ERCHK1–2	Check for device and channel errors.
GETSEN, GETEXT, TEMP2	Store IBUS Registers and 24 sense bytes.
RDSEN, RDSEN1, RDEXT, RDERR	Read all 24 sense bytes into C(ICPC+3) + 8.
TRMCHK, TMCHK1	Check for correct read termination.
RDBUS	Read Bus In lines.
SKRDCK	Check Skip Read; if set, set CBC Register bit 8.
SQERR 00–41	Process sequence errors.
INC 17	Display incrementing count on maintenance panel LEDs if address 17 is selected.
STPTST	Inhibit diagnostics' entry point.
DIAG	Continuous diagnostic execution entry point.
TSTBEG	Normal diagnostic entry point.
TST01–TST10	Check to see that L8A, L8B, L8C, LIS, INT, LCB, SLB, STM, I8S, and LBO all clear the AC.
TST11	Check loading an 8R register.

Table 4-1 (Cont)
DX10/PDP-8/A Microcode Subroutines

Subroutine Mnemonic	Subroutine Function Performed
TST12	Test 8R selection instructions.
TST13	Test Dump mode Skip Read through SILO to Data Registers.
TST14	Same as above using Slow Clock.
TST15	Test byte mode Skip Read through SILO to Data Register.
TST16	Same as above using Slow Clock.
TST17	Test ASCII mode Skip Read through SILO to Data Register.
TST20	Same as above using Slow Clock.
TST21	Test SIXBIT mode Skip Read through SILO to Data Register.
TST22	Same as above using Slow Clock.
TST23	Test Byte Counter overflow logic.
TST24	Test CPC for holding data.
TST25	TEST DAC for holding data.
TSTEND, TSTEXT	End of self-testing.
DER100–DER131, LPADR	Process diagnostic errors.
STLST	Set up pattern list in auto-index 10 and 11.
LD8RS	Load 8R register with data pattern.
LDDRG	Load Data Register.
GETDR	Get Data Register.
LCPC	Load CPC Register.
GETCPC	Get CPC Register.
GETDAC	Get Data Address Counter.

Table 4-1 (Cont)
DX1-/PDP-8/A Microcode Subroutines

Subroutine Mnemonic	Subroutine Function Performed
LDAC	Load Data Address Counter.
CP8R, CPEX	Compare 8R register with RGA, RGB, RGC.
SAVRG	Save CPC and 8R0.

4.2.2 Command Fetching and Decoding Routine (STPRG)

This routine fetches the channel command and decodes or processes that command as follows:

1. Test Clear; if set, transfer the ICPC to the CPC, thus performing a program start (as opposed to continuing).
2. Initialize various internal error handling flags.
3. Fetch the command from DECsystem-10 memory into a selected 8R Register.
4. Read the left-most 12 bits into the AC Register.
5. If the first command is a control command, generate a sequence error.
6. If the command is a device command, go to DOCMD to start the device.

NOTE
Because of the screening sequence, if the PDP-8/A gets to step 7, it must have fetched a control command in the correct sequence (after having first seen a device command).

7. If the Go bit is off, set a flag to halt the channel.
8. If the Jump bit is set, perform a program jump.
9. If the Store bit is set, store status.
10. If the Halt flag was set in step 7, return to the idle loop. If it was not set, continue the Channel Program by going to step 2.

4.2.3 Magtape Starting Routines (DOCMD)

A magtape startup sequence is initiated in response to a device command and is processed in the following manner:

1. The device command is saved in DEVCMD.
2. The device address is saved in DEVADR.
3. The DX10 is initialized and the DSR, CSR, and CBC are cleared.

4. If the command is a Rewind, Unload, Test I/O, or Request Track in Error, set the immediate flag and start the command by going to CMDST. Upon returning from CMDST, set Ending Status type code in the CSR, and store the Ending Status if the command was in error. Finally, proceed to the next Channel Program instruction, if no errors occurred.

5. If the next instruction is not an IMM command, but is a Motion command, set the Ending Status type code in the CSR and return to the idle loop to wait for Status Available.

6. If either a write or read data:

   a. Clear the accumulated byte count field

   b. Prefetch all of the channel XFER words into PDP-8/A control storage memory; load byte counter with the byte count value from the first XFER instruction; load the DAC with the first data address.

   c. Calculate the value for the CBC.

   d. Start the device writing or reading.

   e. Load the CBC, thus setting Run to start the data channel.

   f. Go to the Block Done service routine (BLKDN) and wait.

4.2.4 Block Done Service Routine (BLKDN)

This routine handles both Block Done (byte count overflow) and Command Termination sequences as follows:

1. Wait for Block Done or Status In.

2. Check for byte count overflow. If not set, handle short record condition.

3. Check for another XFER word. If none (FULL8R = 1), stop device. When writing, this means set the last block and wait for Status In. When reading, it means wait for Status In to propagate through SILO; then turn off RUN and wait for Status In.

4. If there are more XFER words, reload CBC and the byte counter.

5. Get next channel command.

6. If the channel command is an XFER word, perform a skip read check and return to step 1.

7. If it is a control command, set FULL8R = 1 and go to step 1.

4.2.5 Command Start Routine (CMDST)

This subroutine manipulates the tag lines on the Channel Bus as follows to cause the TX01/TU70 to execute the device command.

1. Get device address.

2. Place on Bus Out lines.

3. Set Address Out.

4. Set Hold Out.

5. Set Select Out.

6 Wait for OPL IN to set; then go to step 8.

7. If Select In occurs before OPL IN, generate sequence error 14, which specifies unit was not addressable.

8. Clear Address Out.

9. Wait for ADR IN to set.

10. If performing an IMM command, clear both Hold Out and Select OUT at this time. This ensures that the TX01 will disconnect at the end of the sequence. If not, go to step 11.

11. Read the device address being transmitted on Bus In lines.

12. Check to see if correct; if not, generate a sequence error.

13. Get device command. Place on Bus Out lines.

14. Set CMD OUT.

15. Wait for ADR IN to reset.

16. Clear CMD OUT.

17. Wait for STA IN to set.

18. Get status from Bus In lines.

19. Place status into DSR.

20. If status shows that the controller is busy (short busy), disconnect and repeat the command by going to step 2.

21. Set SRV OUT.

22. Wait for STA IN to reset.

23. Clear SRV OUT.

24. If status shows an error (CHK or device busy), take error return.

25. If no errors, take skip return.

4.2.6 Device Command Termination Routines

When a device command is to be terminated, the microcode goes to the SRVDEV routine. This routine uses several other routines for the device termination as described in the following paragraph.

4.2.7 Service Device Routine (SRVDEV)

The Service Device routine obtains the ending status of the TX01 and disconnects the TX01 from the DX10, checks for operational errors, and then checks to see if there are more commands to process. This is accomplished in the following manner:

1. Go to CHKPAR and check for a Channel Bus parity error. If a parity error is detected, set the DEVPAR bit in CSR.
2. Go to GETSTS; get ending status and disconnect from TX01.
3. Check continue; INT the Channel Bus Control Register, and set the Halt flag if Continue is clear.
4. Go to ERRCHK and check for operational-type errors. If any errors are detected, ERRCHK will clear Continue in the Channel Bus control Register, set the Halt flag, and take a nonskip return, which will cause the microcode to go to STORE and store the error status.
5. Clear Device Busy flag.
6. Check the Halt flag; if set, go to the idle loop. If not set, process the next command. If FULL8R = 1, then the next command is in 8R0. If FULL8R = 0, then fetch the next command (CPC).

4.2.8 Get Status Routine (GETSTS)

The GETSTS routine is used to get ending status for all commands except IMM commands. After status is read in, the routine ensures that the TX01 disconnects from the DX10. It operates in the following manner:

1. Check REQ IN and go to step 9 if not set. (REQ IN comes up only for Rewind or Unload.)
2. Set HOLD OUT.
3. Set SEL OUT
4. Wait for ADR IN to set.
5. Read the address into DARI.
6. Set CMD OUT.
7. Wait for ADR IN to reset.
8. Clear CMD OUT.
9. Wait for STA IN to set.
10. Read TX01/TU70 status into DSR.
11. Set SRV OUT.
12. Wait for STA IN to reset.
13. Clear SEL OUT.
14. Clear HOLD OUT.

15. Wait for OPL IN to reset.

16. Clear SRV OUT.

17. Clear Run flip-flop in CBC.

18. Return to calling code.

4.2.9 Error Handling Routines

The microcode is responsible for detecting various errors. When it detects an error it sets the correct bit in the CSR and stores the error status. These errors are:

Selection Error — This error occurs if the Channel Program sends the address of a device that is not present.

Sequence Error — This is a *fatal* error; the program must be reloaded if it occurs. It is considered a hardware failure. The microcode encodes the type of sequence error that occurred into the code field of the status stored in ICPC+1 (Table 4-2).

Device Parity Error — This error indicates that a Channel Bus parity error occurred while reading or writing data.

Length Error — This error specifies that the record just read was either shorter or longer than expected.

OPI Error — Operation Incomplete. This error will occur if a transfer fails to complete within 10 seconds. (The microprocessor times each transfer command word.)

The following routines are entered upon detection of the indicated error conditions by the microprocessor.

1. If the microcode detects an operational error, it sets the appropriate Status bits and stores the error status. Failures such as Device Parity Error, Length Error, and Unit Check are considered operational failurs.

2. If the DX10 encounters an MP error, NXM error, or a Memory Parity error, the PDP-8/A is halted and the DECsystem-10 is interrupted.

3. If the microcode encounters either a selection error or a sequence error, it stores the error status, halts the Channel Program, performs a system reset, and returns to the idle loop.

NOTE

Selection error indicates that the device was not addressable. Sequence error typically specifies that the PDP-8/A got a software time-out while waiting for a hardware function to take place.

When status is stored, it is normally stored into just ICPC+1 and ICPC+2. However, if ICPC+3 is nonzero, this condition serves as a command to perform an Extended Store. This causes additional information to be stored into the area designated by the pointer in ICPC+3.

Table 4-2
Sequence Error Codes

Code No.	Type of Error
00	Channel Program Command List is structured incorrectly.
01	Auto-Index Register ran out (STRCMD or GETCMD).
02	Bus In parity error.
03	Address In did not set (GETSTS).
04	Address In did not reset (GETSTS).
05	Status In did not set (GETSTS).
06	Status In did not reset (GETSTS).
07	Operational In did not reset (GETSTS).
10	8R did not go ready (MEMRD).
11	8R did not go ready (MEMWR).
12	Status In did not set (STPDEV).
13	Status In did not set (SETSTP).
14	Short Busy Error – Could not select the TX01.
15	Operational In or Select In not seen (CMDST).
16	Address In did not set (CMDST).
17	Address In did not reset (CMDST).
20	Status In did not set (CMDST).
21	Status In did not reset (CMDST).
22	Operational In did not reset (CMDST-IMM CMDS).
23	Operational In did not reset (CMDERR).
24	Read Sense Operation did not complete (BLOCK DONE did not set).
25	Silo did not go ready.
26	A Read or Write command was not followed by Data Transfer instructions (XFER Word).
27	Did not find XFER word within 10 words of Device command (pre-fetch).
30	Status In did not drop in CMDBSY (CU End not in Status Byte).
31	Status In did not propagate thru SILO on Read Data operation.
32	Byte Available did not have GO = 0 (TMCHK2).
33	TX01 raised Disconnect In (TX01 Error-CMDST).
34	TX01 raised Disconnect In (TX01 Error-WAIT).
35	SILO did not go ready while reading the sense bytes (Extended Store).
36	Not all of the desired bytes were XFERed in RDSEN.
37	Address returned from TX01 did not compare with address sent.
40	Spurious interrupt during BLKDNI (Block Done Interrupt Service Routine).
41	Status In asserted when it should not have been.

4.2.10 Extended Store

An Extended Store subroutine is executed following a Store Status operation in which the PDP-8/A detects a negative byte count and starting address for extended status information in ICPC+3 (Paragraph 4.2.7). Status is stored either in response to a channel command or as the result of an operational error detected during the Device Command End subroutine. The PDP-8/A stores the following status information in the designated DECsystem-10 memory locations and sets the STAT AVAIL bit in the Channel Command Register, which interrupts the operating program to indicate that status has been stored.

If the ICPC+3 contains -24B13+300 (777200000300), then the following information will be stored, starting at location 300.

NOTE

The number of Sense bytes is controlled by the byte count in ICPC+3. It can be any number in the range 1 through 24.

Location	Data Stored
300	Record Length (12:24)
301	Tag lines, Bus Lines
302	DAC (14:35)
303	Microcode version (0:5); Edition (6:17); Feature Register (18:35)
304	Sense Bytes 0–3
305	Sense Bytes 4–7
306	Sense Bytes 8–11
307	Sense Bytes 12–15
310	Sense Bytes 16–19
311	Sense Bytes 20–23

4.3 BOOTSTRAP OPERATION

When RDI is activated by the operator:

1. The M8317 ROM Bootstrap loader program (Read-In Microcode) is read into the PDP-8/A 4K RAM memory, and initializes the PDP-8/A CPU and DX10 logic.

2. The PDP-8/A begins running and reads the first available tape drive.

3. The PDP-8/A microcode loader (on tape) and the PDP-8/A microcode are loaded into DECsystem-10 memory.

4. Starting at location 100, the PDP-8/A sends RDI DATA (together with minus 1 word count/address 0 and JRST to location 100) over the I/O Bus to the CPU.

5. The microcode loader loads the PDP-8/A microcode into the 4K RAM from DECsystem-10 memory and instructs the PDP-8/A to jump to the first location in the microcode.

6. The PDP-8/A enters the idle loop.

4.4 HARDWARE READ-IN MICROCODE

The hardware Read-In microcode is stored in a 128-word ROM. This routine is block transferred into the last page of control storage memory when the READ-IN key is actuated on the processor console. This page of code reads the first record on the first ready drive into DECsystem-10 memory and starts the loaded program at location 100. The code read in from the ROM operates as follows:

1. Initialize DX10.
2. Set up Rewind command in CMD.
3. Clear device address = 0.
4. Perform Rewind.
5. If error (drive unavailable), increment device address and go to step 4.
6. When a ready unit is found, go to step 7.
7. Change command to read data.
8. Set up Byte Counter and Data Address Registers.
9. Start Read command.
10. Load Channel Bus Control Register; set Run bit.
11. Wait for Block Done.
12. Start DECsystem-10 Channel Program at location 100.

CHAPTER 5
THEORY OF OPERATION

5.1 DX10 FUNCTIONAL ORGANIZATION

The DX10 Data Channel is designed to function as an I/O processor which allows the DECsystem-10 to perform Data Transfer operations using the TU70 Magnetic Tape Subsystem. As an I/O processor, it is fully capable of fetching, interpreting, and executing DECsystem-10 magnetic tape Channel Program instructions. A DECsystem-10 Executive Mode Program starts the DX10 by issuing a CONO instruction that specifies the initial operation to be performed (by setting bits 31 and 32 of the CONO) and by setting the Initial Channel Program Counter (ICPC) Register to the starting address of the Channel Program loaded by the executive program into DECsystem-10 Memory. Once the channel has been initialized by a CONO instruction, operation of the channel is under independent control of a PDP-8/A Miniprocessor (used in the DX10 Data Channel as a microprocessor) which then fetches and executes the Channel Program from DECsystem-10 memory. Figure 5-1 illustrates the general functional organization of the DX10 Data Channel. The DX10 contains the following functional elements:

PDP-8/A Microprocessor (including Omnibus)
4K MOS Memory
Bootstrap Loader
Internal Bus (IBUS)
Memory Bus Register and Control
Channel Command Register (CCR)
Data Channel
DECsystem-10 I/O Bus–IBUS interface
PDP-8/A–IBUS/Data Channel Interface
Channel Bus Interface

5.2 SYSTEM OPERATION

The DX10 operates via three separate buses. The system Memory Bus and I/O Bus provide the operating interface to the DECsystem-10. Interface with the TU70 Subsystem is provided through a dedicated cable designated the Channel Bus.

5.2.1 I/O Bus

The DX10 Data Channel connects to the DECsystem-10 CPU (KA10, KI10 or KL10) via the system I/O Bus. The I/O Bus communicates directly with the Channel Command Register (CCR) and with a group of buffer-multiplexers in the data channel. The buffers allow direct communication between the I/O Bus and the DX10's Internal Bus (IBUS). The CCR is controlled by CONO/CONI instructions and provides communication between the DX10 microprocessor and the CPU for control and interrupt purposes. Communication between either the DX10 microprocessor (the PDP-8/A Miniprocessor) or the IBUS and the system I/O Bus is accomplished by means of DATAO/DATAI instructions issued by the DECsystem-10 monitor. Use of CONO/CONI and DATAO/DATAI instructions in initializing the DX10 and effecting I/O Bus communications (between either the DX10 Data Channel or the microprocessor and the DECsystem-10 CPU) is discussed fully in Chapter 3.

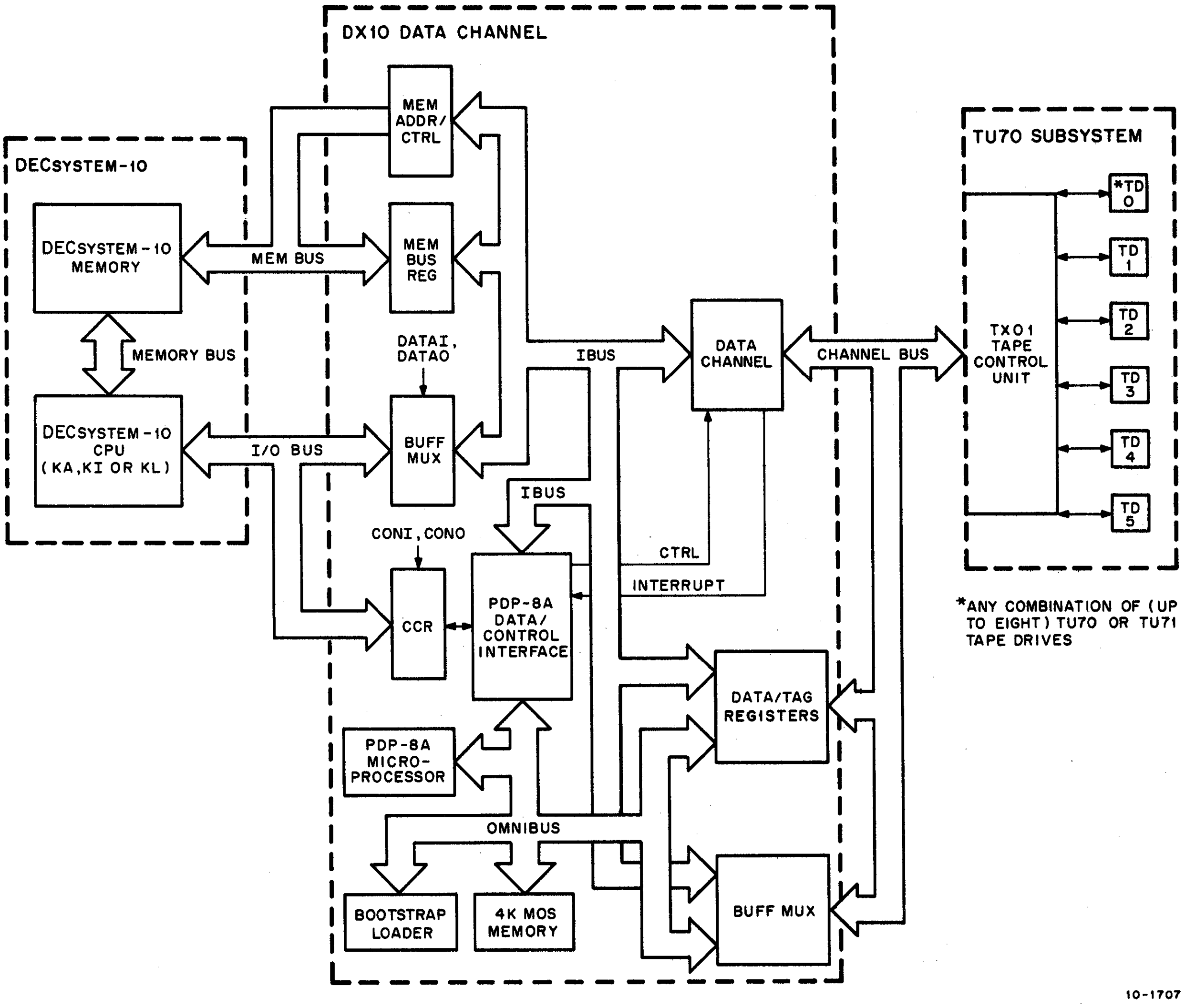


Figure 5-1 DX10 System Operation

5.2.2 Memory Bus

Communication and Data Transfer operations between the DX10 and DECsystem-10 memory are carried out over the system Memory Bus. Data interface between the IBUS and the Memory Bus is accomplished through a 36-bit bidirectional Memory Bus Register, with memory address and control lines being routed through separate addressing and control logic circuits. Memory addressing is accomplished through the use of two 22-bit Memory Address Registers - one for data addressing, the other for Channel Program addressing.

5.2.3 Channel Bus

Data and control information between the DX10 and the TU70 Magnetic Tape Subsystem are transferred (in 8-bit bytes) over the TU70 Subsystem Channel Bus. The required data and control interface between the DX10 IBUS and the Channel Bus is carried out through special input/output data formatting, transfer, and control logic, which function together as a data channel. Input/output processing through the data channel operates automatically (once it has been initialized and set up by the PDP-8/A Microprocessor) to process data between DECsystem-10 memory and any one of up to eight individually addressable tape drives in the

TU70 Subsystem selected for a Read or Write operation. Data/control interfacing between the Channel Bus and the selected tape drive is accomplished through a TX01 Tape Control Unit, which contains all the necessary logic to control unit selection and disconnection as well as data and status formatting/transfer operations.

5.3 OPERATING INTERFACES

Interface between the DX10 and the three operating buses is provided through the use of standard DEC interface modules and/or industry-compatible and product-compatible line drivers and receivers. Figure 5-2 depicts the DX10/bus interface scheme.

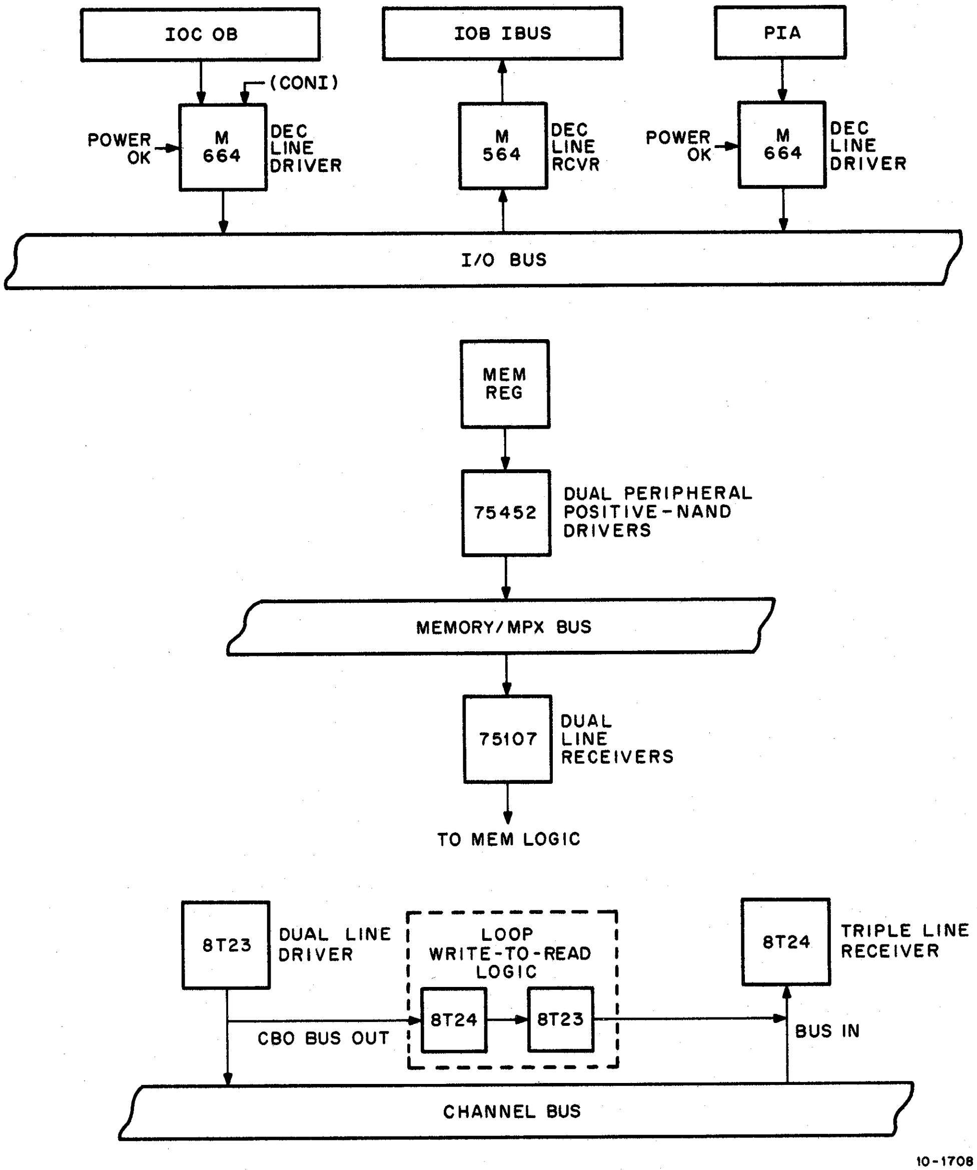


Figure 5-2 DX10/Bus Interface

5.3.1 I/O Bus

Interface between the DX10 and the I/O Bus is accomplished through M564 and M664 single-height Line Receiver and Line Driver modules. When power is removed from the DX10, the line drivers are removed from the I/O Bus by means of a POWER OK signal (deactivated during normal operation by a relay that operates from the DX10 power supply).

Information on the I/O Bus is inverted through the line receivers and distributed via an I/O Bus Input Bus (IOB IBUS) to the Channel Command Register, the RSEL Register, and the I/O Bus/IBUS control and transfer logic. I/O Bus control signals (CONO, DATAO, DATAI, etc.) are routed separately to gating logic circuits where they are used to control CPU to DX10 data and control information transfers. Information from either the DX10 IBUS or the Channel Command Register is gated onto the I/O Bus (through the M664 Bus Drivers) by the IOC IN signal, which is activated when either a DATAI or a CONI is sent over the I/O Bus to the DX10 control logic from the CPU.

Normally, the CCR is available for transfer to the I/O Bus through a 22-bit multiplexer using a CONI. However, when a DATAI instruction is sent by the monitor to read a register, counter, etc., the IBUS is selected to transfer the contents of the selected device through the multiplexer onto the I/O Bus. (The IBUS is 18 bits; the remaining four MUX bits read the contents of the RSEL Register used by the DATAI to select the device to be read into the CPU.) When a CONI is issued to read the CCR, the PDP-8/A Run bit is also read out to the I/O Bus (bit 17) to inform the DECsystem-10 CPU of the condition of the microprocessor. This allows the CPU to wait until the microprocessor is stopped before attempting to read from or write into any of the devices connected to the IBUS. Transfers to the system I/O Bus are made using the low-order bit positions on the bus (bits 17–35).

5.3.2 Memory Bus

Interface with the DECsystem-10 Memory Bus is accomplished through an M8595 Memory Bus Interface module. This module uses SN75452 Dual, Positive-NAND, Peripheral Drivers to transfer information from the Memory Bus Register, either of the Memory Address Counters, or the memory control logic to the Memory Bus and/or the multiplexer control cable. These peripheral drivers are switched off the bus (or control cable) by a POWER OK signal whenever power is removed from the DX10. The M8595 module uses SN75107 Dual Line receivers to transfer information from the Memory Bus or multiplexer cable into the Memory Register and memory interface logic circuits. The SN75107 Receivers incorporate level converters which convert the 0 to -3 V Memory Bus signal levels to the TTL-compatible, 0 to +5 V logic levels used in the DX10.

5.3.3 Channel Bus

The Channel Bus interface is implemented in the M8596 Channel Bus Interface module using 8T23 industry-compatible, Dual Line Drivers. Data inputs to these drivers are from an 8-bit multiplexer that selects the 8-bit byte outputs from either the SILO (intermediate storage) memory or directly from the high-order eight bits of either the IBUS or the PDP-8/A Microprocessor (through the 8R Register input multiplexer). Tag information to the TX01 Tape Control Unit over the Channel Bus is also placed on the bus through 8T23 Line Drivers. Data and tag information from the TX01 TCU is received at the DX10 interface from the Channel Bus through 8T24 high-noise-immune, hysteresis-type, Triple Line Receivers, also located on the M8596 Interface module. These receivers present TTL-compatible, 0 to +5 V logical inputs to the 8234 Input Multiplexers which are used to distribute incoming data and tag information to the SILO, the IBUS, and the microprocessor's Omnibus.

5.4 DATA FORMATS

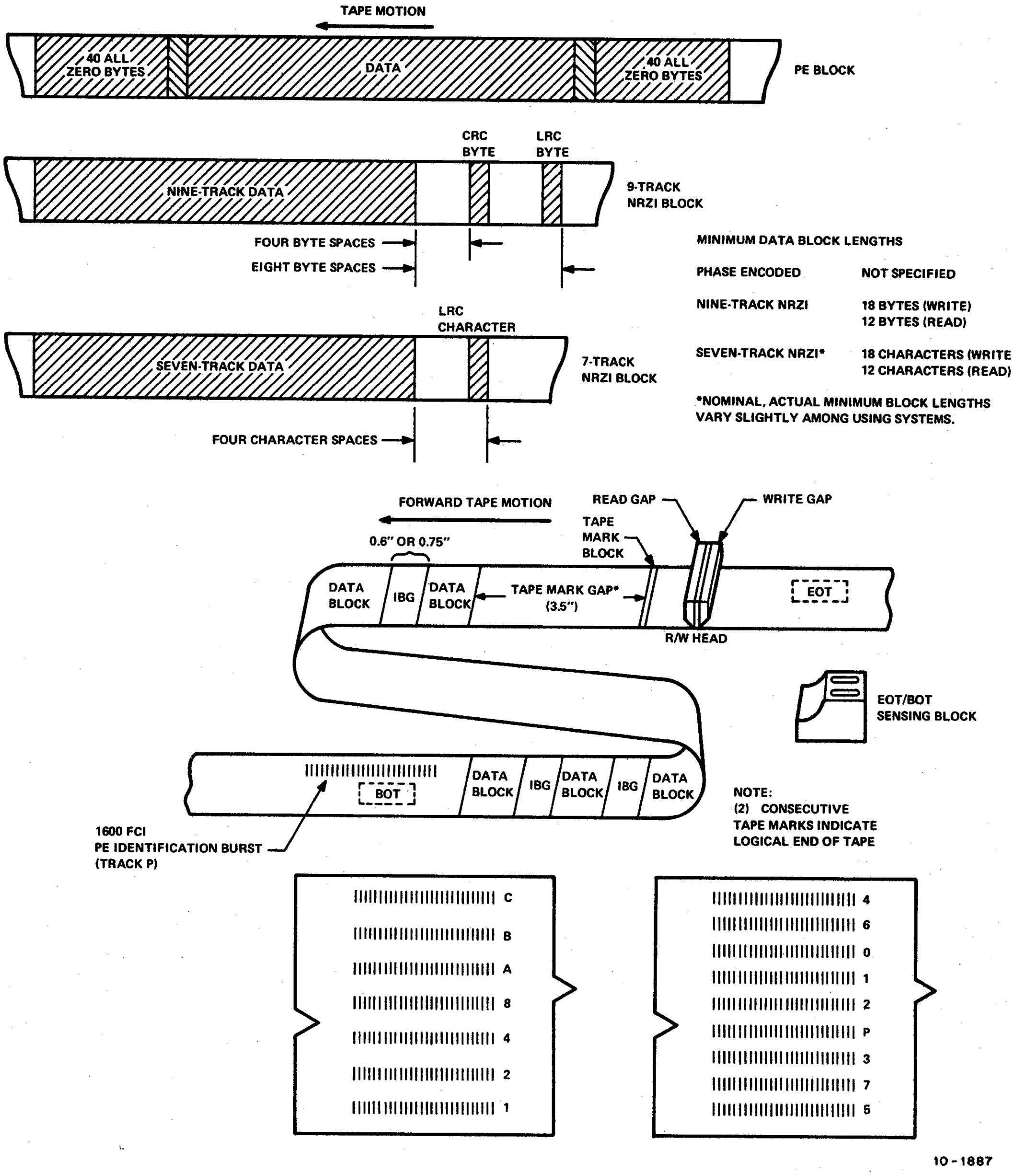
This section describes the 7- and 9-track data formats and the different DX10 operational modes (byte/word formats).

5.4.1 Magnetic Tape

The tape units used with the TU70 Subsystem are TU70A (dual, high-density) and TU71A (treble, low-density) Drives.

The TU70 is a 9-track, 200 in./sec unit that reads and writes variable-length records in 800 bpi (NRZI or PE) or 1600 bpi PE format.

The TU71 Drive reads or writes on 7-track tape in 200, 556, or 800 bpi NRZI (only) format at 200 in./sec. The arrangement of data blocks on tape and the minimum block lengths for 7- and 9-track tapes are shown in Figure 5-3. The track formats for 7- and 9-track tapes are shown in Figure 5-4, together with the methods of writing information in both NRZI and PE modes of operation. Each block of data is separated from the preceding or succeeding block by an interblock gap (IBG) of either 0.6 in. (on 9-track tape) or 0.75 in. (on 7-track tape).



MINIMUM DATA BLOCK LENGTHS

PHASE ENCODED	NOT SPECIFIED
NINE-TRACK NRZI	18 BYTES (WRITE) 12 BYTES (READ)
SEVEN-TRACK NRZI*	18 CHARACTERS (WRITE) 12 CHARACTERS (READ)

*NOMINAL, ACTUAL MINIMUM BLOCK LENGTHS VARY SLIGHTLY AMONG USING SYSTEMS.

Figure 5-3 Magnetic Tape NRZI and PE Data Formats

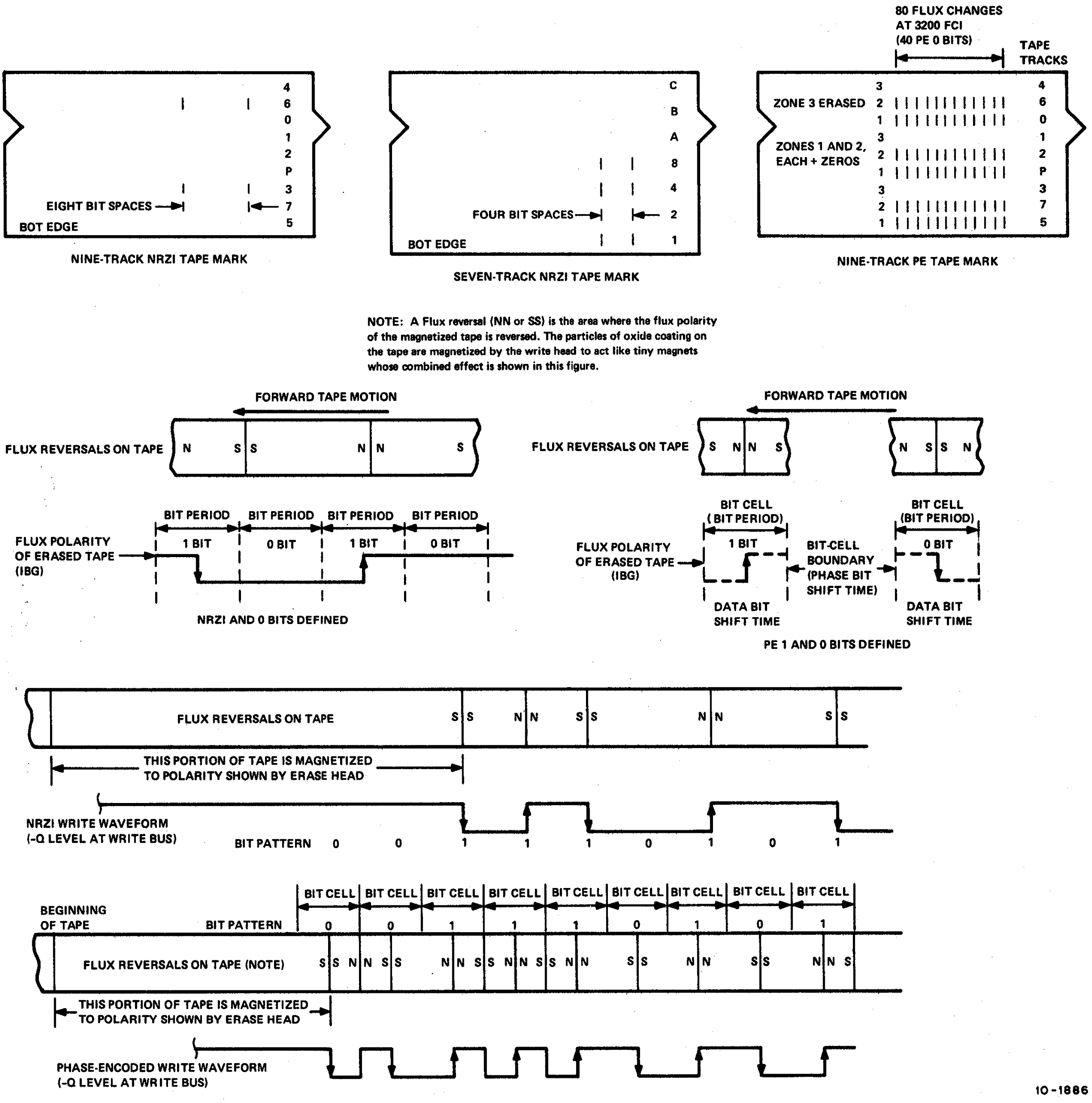

4
6
0
1
2
P
3
7
5
EIGHT BIT SPACES
BOT EDGE
NINE-TRACK NRZI TAPE MARK
C
B
A
8
4
2
1
FOUR BIT SPACES
BOT EDGE
SEVEN-TRACK NRZI TAPE MARK
80 FLUX CHANGES AT 3200 FCI (40 PE 0 BITS)
TAPE TRACKS
ZONE 3 ERASED
ZONES 1 AND 2, EACH + ZEROS
NINE-TRACK PE TAPE MARK
NOTE: A Flux reversal (NN or SS) is the area where the flux polarity of the magnetized tape is reversed. The particles of oxide coating on the tape are magnetized by the write head to act like tiny magnets whose combined effect is shown in this figure.
FORWARD TAPE MOTION
FLUX REVERSALS ON TAPE
N S S N N S
BIT PERIOD
1 BIT
0 BIT
FLUX POLARITY OF ERASED TAPE (IBG)
NRZI AND 0 BITS DEFINED
FORWARD TAPE MOTION
FLUX REVERSALS ON TAPE
S N N S
N S S N
BIT CELL (BIT PERIOD)
1 BIT
0 BIT
FLUX POLARITY OF ERASED TAPE (IBG)
BIT-CELL BOUNDARY (PHASE BIT SHIFT TIME)
DATA BIT SHIFT TIME
PE 1 AND 0 BITS DEFINED
FLUX REVERSALS ON TAPE
S S N N S S N N S S
THIS PORTION OF TAPE IS MAGNETIZED TO POLARITY SHOWN BY ERASE HEAD
NRZI WRITE WAVEFORM (-Q LEVEL AT WRITE BUS)
BIT PATTERN 0 0 1 1 1 0 1 0 1
BIT CELL
BEGINNING OF TAPE
BIT PATTERN 0 0 1 1 1 0 1 0 1
FLUX REVERSALS ON TAPE (NOTE)
S S N N S S N N S S N N S S N N S S N N S S N N S
THIS PORTION OF TAPE IS MAGNETIZED TO POLARITY SHOWN BY ERASE HEAD
PHASE-ENCODED WRITE WAVEFORM (-Q LEVEL AT WRITE BUS)
10-1886

Figure 5-4 7- and 9-Track Tape Information Storage

Data blocks are made up of n bytes of information (characters) assembled as 7-or 9-bit bytes, with each bit of every byte written in a separate track. Placement of bits in specific tracks is controlled by the character mode (Core Dump, ASCII, Sixbit, etc.) selected for writing by the monitor before the record is written. A group of data blocks on tape is referred to as a tape file. Each tape file is separated from other files by a tape mark written in either NRZI or PE format depending on the mode of operation. Two tape marks appearing together in succession on a 7- or 9-track tape indicate to the monitor the end of the usable portion of tape.

5.4.2 Operating Modes

The DX10 transfers data between the TX01 Tape Control and DECsystem-10 memory in 36-bit words. Data transfers over the Channel Bus are in 8-bit bytes. The Channel Bus Control Register (CBC) mode bits (shown after the mode name in the following paragraph headings), sent by the microprocessor to the CBC register, determine how the DX10 performs the 8/36 or 36/8 bit conversion. The modes of operation described in the following paragraphs are program-selectable for use in conjunction with the TU70 Subsystem.

5.4.2.1 Byte Mode (01) (IBM-Compatible) - Byte mode is so named because it packs and unpacks (formats) a 36-bit word in the same manner as a DECsystem-10 byte instruction. Four 8-bit bytes are transferred per word; the remaining four bits are ignored, i.e., not written on tape (during a write), or are read as zeros i.e., zero filled (during a read), as shown in Figure 5-5.

5.4.2.2 SIXBIT Mode (11) - This mode causes six 6-bit bytes to be packed into a 36-bit word. It is useful when using 7-track tape. The format is shown in Figure 5-6.

5.4.2.3 Core Dump Mode (00) - Core Dump mode utilizes a full 36-bit word. Four 8-bit bytes and one 4-bit byte per word are transferred. This mode is useful because of compatibility with other DEC magtape subsystems. The format is shown in Figure 5-7.

5.4.2.4 ASCII Mode (10) - This mode transfers four 7-bit bytes and one 8-bit byte per word. One bit (the MSB) per 8-bit channel byte is not utilized for the first four bytes. The LSB of the fifth byte (bit 35 of the data word) is transferred to the MSB of the Channel Bus (bit 0) during a Write operation. During a Read operation, the bit is repositioned into the LSB (bit 35). The format is shown in Figure 5-8.

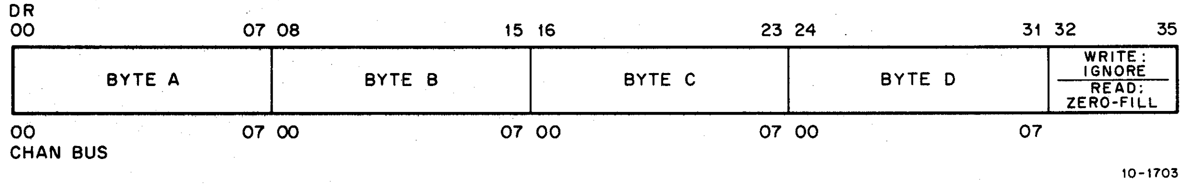


Figure 5-5 Byte Mode Format

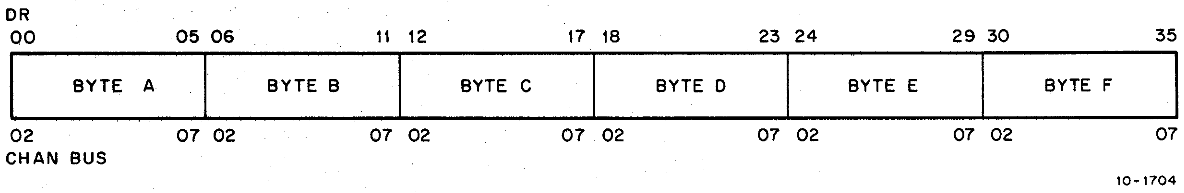


Figure 5-6 SIXBIT Mode Format

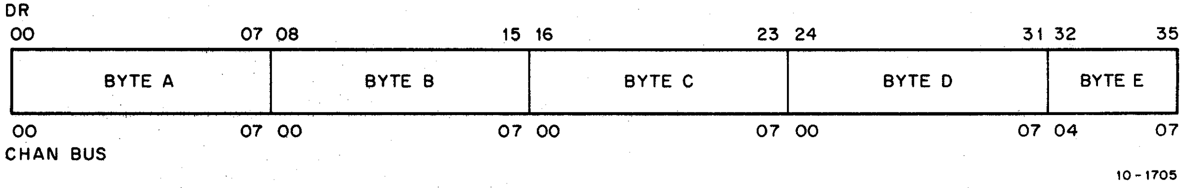


Figure 5-7 Core Dump Mode Format

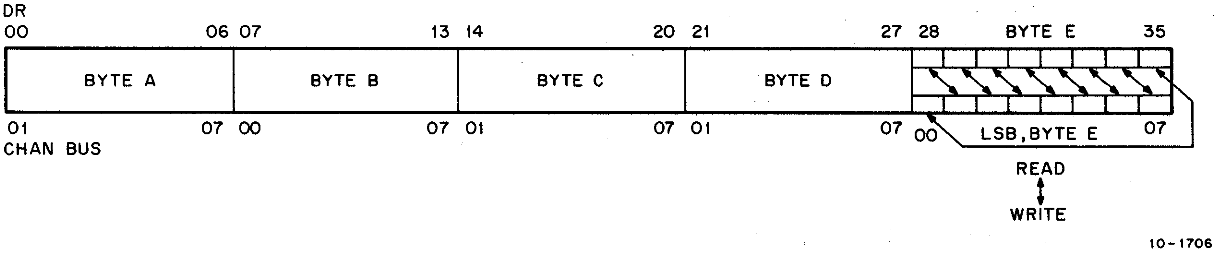


Figure 5-8 ASCII Mode Format

5.4.3 Internal Data Buffering

A SILO buffer is provided to alleviate data overrun problems. It will hold as many as 256 bytes of data. In terms of DECsystem-10 words, this represents the following:

Byte Mode	=	64 words
SIXBIT Mode	=	42-2/3 words
Core Dump Mode	=	51-1/5 words
ASCII Mode	=	51-1/5 words

5.5 FUNCTIONAL OPERATION

The DX10 Data Channel operates as an independent I/O processor, transferring data between any of several tape units in the TU70 Magnetic Tape Subsystem and a preassigned data storage area in DECsystem-10 memory. Chapter 3 presents the programming and data transfer sequences carried out by the DX10 in performing the Channel Program instructions associated with these Data Transfer operations. The following paragraphs present a functional description of the various logical elements within the DX10 as they are used in controlling the various operations related to execution of the Executive Mode Program, Channel Program, and microprocessor Microcode instructions.

5.5.1 Logical Elements

Engineering Drawing No. D-BD-DX10-0-BD is an overall functional block diagram that depicts the logical elements which make up the DX10 Data Channel. The basic processing/control element in the DX10 is a PDP-8/A Microprocessor, which consists of a central processor (M8315 PDP-8/A CPU module), a 4K control storage memory (M8311 4K MOS Memory module), a programmable "console" or transfer control logic section, and an IOT instruction decoder logic section, all interconnected via an Omnibus.

NOTE

For automatic self-start, the microprocessor also contains a ROM (M8317 Bootstrap Loader) module which is used in conjunction with the DECsystem-10 console hardware read-in feature. This allows the console operator to automatically load the microcode program from a TU70 Tape Unit into the 4K control storage microprocessor memory. (Refer to Chapter 4 for operational details.)

The microprocessor [by means of the DX10 Internal Bus (IBUS) and its own Omnibus and IOT decoder logic] communicates with and controls (through special tristate buffers over the IBUS) a data channel section. The data channel (under control of a Channel Program) transfers data (at rates up to 320K bytes/second) between DECsystem-10 memory and the TU70 Subsystem (which includes a TX01 Tape Control Unit and up to eight TU70 or TU71 Tape Units). The data channel section contains the following functional elements:

1. A 36-bit Data Register, which interfaces with the system Memory Bus via the Internal Bus (IBUS) and the 36-bit Memory Register (maximum transfer rate = 3.5 μs/word).

2. A 256-byte, "first-in, first-out," buffer storage memory (SILO), which receives individual data bytes from either the Byte Assembly Register (during a Write operation) or the Channel Bus In line receivers (during a Tape Read) and stores each byte, as it is received, at its output end. (The SILO has its own Input and Output Registers and is used to store data parity and control information along with each 8-bit data byte.

3. An 8-bit, serial shift Byte Assembly Register (BAR) to assemble individual bytes of information from the Data Register and transfer them into the SILO during a Write operation.

4. An 8-bit Byte Disassembly Register (BDR) to disassemble individual data bytes (one byte at a time) from the SILO during a (forward or backward) Tape Read operation and shift them serially (right to left or left to right) into the 36-bit Data Register, positioning each one within the word in a format determined by the operating character mode (Core Dump, ASCII, SIXBIT, etc.)

5. A 14-bit Byte Counter used by the microprocessor to control the number of bytes transferred and to determine when the end of a data block is reached. The size of the data block (byte count) is specified by a Channel Program Data Transfer instruction and is loaded into the Byte Counter at the beginning of a Data Transfer operation by the microprocessor. A Byte Counter Overflow (BCOF) signifies the end of a data block to the microprocessor.

6. A Channel Bus Control (CBC) Register used by the microprocessor to control the data channel, permitting it to automatically transfer information between DECsystem-10 memory and the TU70 Subsystem. Data channel control information is loaded into the CBC Register by the microprocessor at the beginning of every Data Transfer operation. The CBC Register also contains a Run bit, which is set by the microprocessor to activate the byte assembly logic and start the operation. Data is then clocked, one bit at a time, between the Data Register and either the BAR or the BDR by a special (20 MHz) channel clock.

7. A Bit Counter to automatically count the number of bits transferred into or out of the Data Register. When a specified number of bits (6, 7, or 8 as determined by the operating character mode) has been shifted to or from the Data Register, the Bit Counter generates a Carry, which increments a Byte Assembly Counter. This indicates that a data byte has been transferred in or out of the Data Register.

8. A Byte Assembly Counter (BAC) to count up the number of bytes transferred into or out of the Data Register. When the required number of bytes (4, 5, or 6, depending on the character mode) has been shifted into or out of the Data Register, the BAC generates a Word Done signal and, if a Read operation is in progress, the entire contents of the Data Register are transferred into the DECsystem-10 memory data buffer over the IBUS and system Memory Bus. If a Write operation is being performed, another 36-bit word is read into the Data Register from the memory data buffer.

9. A Channel Bus Interface which includes:

   a. Tag In* and Bus In line receivers and associated multiplexers to route Channel Bus data directly into the SILO or onto either the microprocessor's Omnibus for reading into control storage or the IBUS for reading by the DECsystem-10 CPU using DATAI instructions.

   b. A Tag Out Register (TOR)* used by either the microprocessor (to identify information on the Channel Bus Out lines during Channel Bus initializing and terminating sequences) or the data channel (to identify data during data transfers). The Tag Out Register can also be set from the IBUS by the DECsystem-10 CPU for diagnostic purposes using DATAO instructions.

   c. A Channel Bus Output Register that is used by either the microprocessor (to place individual bytes of information on the Channel Bus Out Lines) or the DECsystem-10 CPU for diagnostic purposes.

In addition to the microprocessor and the data channel, the DX10 includes a microprocessor/IBUS data and control interface, an RSEL Register, and a series of RSEL decoders to permit selection of various registers and counters either for operational control purposes or for transfer of the contents of a selected register, counter, etc. over the IBUS to the maintenance panel LED display or over the system I/O Bus to the DECsystem-10 CPU. The decoded RSEL outputs generate the necessary control signals (through bus transfer control logic, PDP-8/A transfer control logic, etc.) to allow the desired data from the selected counter (register, 8/A memory, etc.) to be placed onto the IBUS. The contents may then be transferred through a multiplexer to the system I/O Bus by a DATAI instruction.

A memory data, addressing, and control interface is also included in the DX10 to provide access to DECsystem-10 memory either directly via the system Memory Bus or through an MX10 Memory Multiplexer (using a multiplexer control cable in conjunction with the Memory Bus). The memory interface logic contains a 36-bit, direct set, Memory Register and two 22-bit Counting Address Registers [the Channel Program Counter (CPC) and the Data Address Counter (DAC)]. The CPC provides address sequencing for the Channel Program, while the DAC provides data address sequencing for the memory data buffer area. Both the DAC and the CPC are loaded from the 18-bit MA IBUS in two transfers (one 4-bit, one 18-bit) when selected for loading by an RSEL 0 and either an RSEL 12 or an RSEL 13 selection. Whether or not either or both the DAC and the CPC are incremented or decremented after each word transfer is determined by Channel Bus Control Register bits 0 and 1.

The DX10 memory interface also contains the necessary memory control and gating logic to accomplish the normal Memory Bus (read or write) request and disconnect dialogue with either a DECsystem-10 memory or an MX10 or MX10-C Multiplexer. (Refer to DECsystem-10 Interface Manual for details of the Memory Bus dialogue.)

CPU control of the DX10 is exercised over the system I/O Bus using CONO/CONI and DATAO/DATAI instructions via the DX10 I/O Bus data transfer and control (IOC) logic section. As previously stated in the I/O Bus interface description, the DX10 utilizes an intermediate I/O Bus (designated the IOB-IBUS) to provide TTL-compatible, positive logic signals to the Channel Command Register and the DATAO-selectable IBUS multiplexer switch. Use of the Channel Command Register and the IBUS switch in controlling DX10 operations is discussed extensively in Chapter 3.

The DX10 also includes a device number decoder which is used to enable the input/output control logic section for acceptance of DATAO, CONO, etc. instructions and to enable setting of the Read-in flip-flop to initiate a hardware Read-In operation when device code 220_8 is selected on the console READ-IN DEVICE switches and the READ-IN key is pressed.

*Both the Tag Out Register and Tag In lines can be read over the IBUS by the DECsystem-10 CPU using DATAI instructions.

The input/output control section processes all CONO, CONI, DATAO, and DATAI commands and provides a control interface to the PDP-8/A Microprocessor transfer control logic and the Omnibus for direct control of microprocessor operation.

Direct reading (using CONI or DATAI instructions) of the Channel Command Register, microprocessor, or IBUS-connected registers, counters, etc. is accomplished through individual multiplexer switches within the I/O Bus control (IOC) logic section.

Hardware interrupts are generated in the DX10 IOC logic and are gated onto the I/O Bus through a program-interrupt assignment (PIA) decoder whenever any of four automatic interrupt conditions (Microprocessor error, Memory Parity error, Nonexistent Memory, or Status Available) occur.

A Feature Register is also included in the DX10 (located on the M8597 PDP-8/A Interface module) to permit the serial number and other hardware related data (ECO level Memory Bus configuration, etc.) to be recorded for subsequent readout by the CPU or by maintenance personnel using selector switches located on the DX10 maintenance panel.

An indicator panel is incorporated as part of the front-mounted maintenance panel to permit viewing of any of the registers, counters, etc. connected to the IBUS on a 6-octal-digit LED display. The panel also includes two register select decimal point indicators: one (on the right-hand RSEL indicator) to indicate the state of the microprocessor Run bit; the other (on the left-hand RSEL indicator) to indicate a microprocessor timeout error (MP ERR). The entire 18-bit IBUS is displayed on the 6-digit DATA display anytime the EXAMINE pushbutton.on the panel is pressed. Selection of a given register counter, etc. for display via the IBUS is accomplished using two ADDRESS thumbwheel switches (also located on the maintenance panel).

5.5.2 PDP-8/A Microprocessor

The PDP-8/A Microprocessor is used in the DX10 to initialize and control Data Transfer operations which essentially move data between DECsystem-10 memory and any addressed tape unit in the TU70 Subsystem.

As previously stated, the microprocessor includes an M8315 CPU module, an M8311 4K MOS Memory module, and a set of IOT decoders, all interconnected through an Omnibus to perform the various microprocessor functional operations. These operations are carried out as described in the following paragraphs.

5.5.2.1 Channel Program Instruction Decoding – Channel Program instructions stored in DECsystem-10 memory are read into the DX10 Memory Register as a 36-bit word from the system Memory Bus. The instruction is then transferred (in two serial 18-bit transfers) over the 18-bit IBUS. First the left half, then the right half of the word are transferred into one of four locations (normally 8R0) in the 4-word, 8R Register file. This file is made up of four addressable, 36-bit flip-flop type registers, any one of which can be selected for loading or reading by means of a 2-bit counter referred to as the 8RSEL Register. This "register" (actually a binary counter) is loaded by the microprocessor to select the location into which the Channel Program instruction is to be read.

When the instruction has been read into the 8R0 location, the microprocessor reads it into the PDP-8/A CPU Accumulator (AC) and examines it to determine the operation to be performed. Depending on the type of instruction, the microprocessor goes to any of several different subroutines in the microcode to initiate the required actions to perform the operation. (Refer to Paragraph 3.4.3.1.) In the following sequence of operations (as well as in the instruction fetch sequence itself), control of the various transfer operations is exercised by the microprocessor. To accomplish the necessary transfer control, the microprocessor issues a series of Input/Output Transfer (IOT) instructions. These IOTs are decoded through IOT decoder logic, and the necessary control and data transfer signals are activated to perform such control/transfer functions as requesting a DECsystem-10 memory cycle to read a Channel Program instruction into the 8R Register, transferring the contents of the 8R Register into the Accumulator, skipping on parity error, etc. Table 5-1 lists each of the IOT instructions used by the microprocessor in controlling data transfers within the DX10. The control or data transfer function performed by each IOT is listed, together with the IOT mnemonic and a description of the function itself, opposite each IOT Operation Code. The decoded signals for each of the IOTs listed are shown and discussed in the IOT Decoder functional description (Paragraph 5.6.2.10). Details of the Memory Bus to IBUS transfers and Memory and/or Multiplexer control and gating are similarly discussed within the Memory Data Transfer/Addressing and Interface Control functional descriptions.

5.5.2.2 PDP-8/A Transfer Control – The PDP-8/A transfer control logic section functions as a programmable microprocessor "operator's console" for the DECsystem-10 CPU. By utilizing DATAO/RSEL instructions, the DECsystem-10 can perform direct control storage memory loading or microprocessor Single Step, Halt, Continue, Deposit, and other functions for diagnostic purposes. (Refer to Paragraph 3.4.1.2 for RSEL address associated with operation of the microprocessor. Omnibus transfer control signals associated with microprocessor operation are shown in Figure 5-18 and are discussed in the associated functional circuit description.)

5.5.2.3 Omnibus-IBUS Data/Address Transfers – To implement the IBUS to Omnibus data and address transfers associated with DECsystem-10 CPU control of microprocessor console functions, the DX10 utilizes a 12-bit, tristate flip-flop storage register. These tristate devices present a high impedance to the Omnibus lines when not enabled for transfer operations. For transfers to the IBUS from the Omnibus during DECsystem-10 diagnostic Read operations, gated tristate buffers are utilized; these buffers provide high impedance Omnibus-IBUS isolation when not in use. (Omnibus-IBUS data and address gating control signals are shown in Figure 5-17.

5.5.3 Data Channel

The data channel section of the DX10, once initialized by the microprocessor, transfers data between DECsystem-10 memory and a selected tape drive automatically, without intervention by the microprocessor, until completion of the data transfer.

5.5.3.1 Channel Initialization – At the beginning of a Data Transfer operation (following a tape Read or Write command), the microprocessor loads the program-specified byte count into the Byte Counter and the address of the DECsystem-10 memory data buffer into the Data Address Counter (DAC). After the Byte Counter and DAC are loaded with the correct byte count and data address, the microprocessor loads the Channel Bus Control (CBC) Register with operation-related control information (mode, read or write, backward or forward, byte assembly count, etc.) and sets the CBC Run bit. This activates the byte assembly logic and enables the Channel Bus Out SILO output gate (to allow information transfers from the SILO to the Channel Bus Out lines for a tape Write operation and to allow initialization of a memory cycle to transfer the first data word into the Data Register.)

The Channel Bus Control Register (CBC) is the mechanism by which block transfers are controlled and executed. The microprocessor loads the CBC after the initial selection sequence has been performed to allow Channel Bus data transactions to occur automatically. These transactions continue until the transfer is terminated by the DX10 or the TX01. The PDP-8/A loads the CBC from the AC via an IOT (6510) instruction. The bits are defined in Table 5-2.

Table 5-1
IOT Instructions

Op Code	Transfer/Control	Mnemonic	Description

6513	AC → 8RA	L8A	Transfer the 8R (PDP-8/A register) to/from the PDP-8/A Accumulator. The 8R is 36 bits X 4 words, each word broken into three 12-bit segments.
			A = 0–11
			B = 12–23
			C = 24–35
			The 8R is connected to the IBUS for access to Memory and other registers. The 8R SEL Register further specifies which of the four 36-bit words is accessed.

6514	AC → 8RB	L8B	
6515	AC → 8RC	L8C	
6503	8RA → AC	G8A	
6504	8RB → AC	G8B	
6505	8RC → AC	G8C	
6506	BI → AC (4:11)	GBI	Loads the Channel Bus In lines into the Accumulator.
6507	TI → AC (4:11)	GTI	Loads the Channel Tag In lines into the Accumulator.
6510	AC (2:11) → CBC	LCB	Loads the Accumulator into the Channel Bus Control Register. This is used to control data block transfers after unit selection. See Table 4-2 for bit descriptions.
6511	AC (10:11) → 8RSEL	L8S	The Accumulator is loaded into the 8R Select (8RSEL) Register, selecting one of four 36-bit words for further operations.
6512	AC (11) → CSRF, CLR SR	LSF	Performs two distinct operations: AC bit 11 is loaded into the CSR FLAG bit in the CCR. The STAT REQ bit in the CCR is cleared.
6500	CLR CLR/CONT	CCC	Clears the CLEAR and CONT CCR bits.
6501	SET LAST BLK	SLB	Sets the Last Block flip-flop. This should be set during a block transfer to indicate that no further data will be transferred by the CU. After the last byte has been transferred and Last Block is set, the DX10 responds to the SVC IN bus signal with CMD OUT.
6502	SET TIMER	STM	Restarts the watchdog timer to indicate that the PDP-8/A is running properly. The MP ERR bit sets in CCR after a time-out, interrupting the DECsystem-10 CPU.
6557	INC 8RSEL	I8S	Adds one to the 8R Select Register, modulo 4.

Table 5-1 (Cont)
IOT Instructions

Op Code	Transfer/Control	Mnemonic	Description
6517	AC (0:7)→BOR	LBO	Loads the Channel Bus Out Register from the AC. This will be transferred to the Bus Out lines if the Channel Bus Control Register Run bit = 0.
6520	SKIP IF CBPE	TCP	Skip the next PDP-8/A instruction in sequence if a parity error was detected on the Channel Bus data path. (See Note 1.) CBPE is cleared with a 6510 IOT.
6521	SKIP IF BLK DONE	TBD	Skip if, during a block transfer, one of the following occurs: Write: the Byte Counter has overflowed, indicating that all data in the current block has been loaded into the SILO. Read: the Byte Counter has overflowed and the last word has been stored in memory.
6522	SKIP IF CSRF	TSF	Skip if the CCR CSR FLAG = 1.
6523	SKIP IF STAT REQ	TSR	Skip if the CCR STAT REQ bit = 1.
6524	SKIP IF CLEAR	TCL	Skip if the CCR CLEAR bit = 1.
6525	SKIP IF CONT	TCT	Skip if the CCR CONT bit = 1.
6526	SKIP IF STAT AVAIL	TSA	Skip if the CCR STAT AVAIL bit = 1.
6527	SKIP IF RDI	TRI	Skip if the DX10 has been requested to do a read-in. Cleared only by the Initialize IOT (6007).
6531	CLR ADR OUT	CAD	Clear the flip-flop connected to the channel Tag Out line of the same name. Initialize (6007) clears all the Tag Out flip-flops.
6532	CLR CMD OUT	CCM	
6533	CLR SVC OUT A	CSV	
6534	CLR OPL OUT	COP	
6535	CLR HLD OUT	CHL	
6536	CLR SEL OUT	CSE	
6537	CLR SUP OUT	CSU	
6570	SET SVC OUT B	SVB	Set the SVC OUT B flip-flop, which is ORed to the SVC OUT bus line with SVC OUT A. SVC OUT B clears automatically with the assertion of SVC IN; SVC OUT A does not.

Table 5-1 (Cont)
IOT Instructions

Op Code	Transfer/Control	Mnemonic	Description

6571
6572
6573
6574
6575
6576
6577
	SET ADR OUT
	SET CMD OUT
	SET SVC OUT A
	SET OPL OUT
	SET HLD OUT
	SET SEL OUT
	SET SUP OUT
		SAD
		SCM
		SVA
		SOP
		SHL
		SSE
		SSU
			Set the flip-flop connected to the channel Tag Out line of the same name.

6540	SKIP IF 8R RDY	T8R	Skip if the 8R to/from memory transfer requested has completed.
6550 6551	8R ← MEM (READ) 8R → MEM (WRITE)	MRD MWR	Perform a DECsystem-10 memory cycle to transfer a word to or from the 8R Register selected by the 8RSEL Register. One 36-bit word is transferred. The transfer may be performed during a data block transfer. The CPC is used to specify the address of the word transferred into or out of DECsystem-10 memory.
6552	DEC 8RSEL	D8S	Subtracts one from the 8R Select Register, modulo 4.
6553 6554 6555 6556	8RL → IB 8RR → IB 8RL ← IB 8RR ← IB	LIL LIR GIL GIR	Transfers the 8R right or left 18 bits to or from the IBUS. The IBUS provides a path to or from the register selected by the RSEL Register. The register select codes are the same as those described in Paragraph 3.3.2 for DATAO/DATAIs. The 8R used is specified by the 8R SEL Register. IBUS transfers may not be executed during a block transfer.
6516	AC (8:11) → RSEL	LIS	Loads the RSEL (Register Select) Register from the Accumulator. This allows the PDP-8/A to access any register connected to the IBUS.
6541	SKIP IF BUS IN PARITY OK	TBP	Skip if parity in a 9-bit data byte coming from the Channel Bus In lines is odd (OK).
6542	SKIP IF DEVICE NOT DONE	TDN	Skip if the STA IN pulse from the Channel Bus Tag In lines has not yet propagated through the parity and control SILO.
6543	SKIP IF BYTE COUNT OVERFLOW	TBC	Skip if the 14-bit Byte Counter has reached a count of 8192 (Byte Count Overflow).
6544	SKIP IF SILO OUTPUT READY	TOR	Skip if a data byte is ready for clocking out of the SILO into the SILO Output Register.

Table 5-1 (Cont)
IOT Instructions

Op Code	Transfer/Control	Mnemonic	Description
6545	SKIP IF BYTE UNAVAILABLE	TBA	Skip if a byte is waiting to enter the SILO.
6546	SKIP IF SILO NOT EMPTY	TSE	Skip during a tape Read operation if the STA IN signal has not yet been clocked into the Parity and Control Output Register.
6565	SET EVEN PARITY FOR SILO INPUT	SEP	Forces an odd number of data bits at the SILO parity generator input. This causes the odd parity bit (CBO ODD PARITY) to reset, generating even parity into the SILO.
6566	ENABLE READIN	ERI	Sets RDI FIRST WORD FF in the RDI logic to initiate a JRST to location 100 in DECsystem-10 memory following the loading of the Bootstrap Loader into that memory area.
6567	SET STAT AVAIL	SSA	Set the STAT AVAIL bit in CCR, interrupting the DECsystem-10 CPU.
6007	INITIALIZE	INT	Initializes the entire DX10. Clears the AC, link, and most of the control flip-flops. Also known as CAF (Clear All Flags). Issuing an I/O Reset pulse from the DECsystem-10 CPU has a similar effect on the DX10.
6562	SET READ-IN FLOP	SRI	Sets Read-In flip-flop to initiate a programmed Bootstrap operation. (Refer to Paragraphs 4.3 and 4.4 for details of bootstrap loader and microcode.)
6563	SET FAST CLOCK	SFC	Sets a Fast Clock flip-flop on the M8599 module to allow a data clocking rate of 50 ns. This allows a memory cycle to be requested on the system Memory Bus every 3.5 μs (approx.).
6564	SET SLOW CLOCK	SSC	Sets a Slow Clock flip-flop on the M8599 module to allow a data clocking rate of 200 ns. This allows a memory cycle request every 10 μs (approx.) over the system Memory Bus.

NOTE

1. **The parity bit accompanying each data byte from the Channel Bus is stored in the Parity and Control SILO at the same time the data byte is being stored in the SILO. The parity bit is checked when the byte leaves the SILO and enters the BDR. Channel Bus Parity Error (CBPE) sets if parity is in error. The byte having bad parity will be stored in the DECsystem-10 memory data buffer area. CBPE can be tested by the microprocessor through the use of a 6520 IOT instruction.**

2. **Skip IOTs (if the required conditions are met) cause the Omnibus SKIP line to be asserted.**

Table 5-2
Channel Bus Control (CBC) Register

AC Bit No.	Function	Description
0	CPC DEC	If 1, a memory reference decrements the CPC; otherwise, it increments.
1	DAC DEC	If 1, a memory reference decrements the DAC; otherwise, it increments. **NOTE** **Initialize clears both bits.**
2	SPECIAL BYTE	Set by the PDP-8/A if the first byte transferred during a Read Backward operation requires special treatment by the byte assembly logic. SPECIAL BYTE is generated automatically in all other situations, as in Core Dump mode, byte E, and ASCII mode, byte E.
3	RUN	Activates the byte assembly logic. Words are transferred over the Memory Bus and bytes are transferred over the Channel Bus until the Byte Counter overflows or the TX01 signals termination by asserting STA IN. RUN should be cleared after the transfer to prevent memory requests and to allow the PDP-8/A access to the Bus Out lines.
4, 5	MODE 0 MODE 1	Specify operating data mode for the byte assembly logic. See Paragraph 4.4.2 for definitions.
6	READ DEVICE	Specifies direction of transfer.
7	BACKWARD	If READ DEVICE = 1, BACKWARD may be set to assemble data in the reverse order.
8	MEM INHIBIT	Prevents generation of the signal that gates the Data Register onto the IBUS for transfer to the Memory Register, thus inhibiting any unwanted data from being transferred to DECsystem-10 memory during a tape Read operation.

Table 5-2 (Cont)
Channel Bus Control (CBC) Register

AC Bit No.	Function	Description
9–11	BAC PRESET	These three bits load the Byte Assembly Counter (BAC) to specify the number of bytes per 36-bit word. The BAC must be preset only for the first word of a Read Backward operation, to specify the position of the first byte in the word. Referring to the data mode formats of Paragraph 5.4.2, the values to be loaded are:

First Byte Position	BAC Preset
A	0 0 1
B	1 1 1
C	1 1 0
D	1 0 1
E	1 0 0
F	0 1 1

5.5.3.2 Data Transfer – If a tape Read operation is being performed, the Channel Bus Out SILO output gate is inhibited, and data is transferred from the Channel Bus In lines through the input line receivers and a gating multiplexer (enabled by READ DEVICE) into the SILO Input Register. Thus, data bytes being transferred from the SILO to the BDR will not appear on the Channel Bus Out lines.

Data bytes are clocked into the SILO Input Register by the SILO as soon as they arrive from the Channel Bus In data lines. (During a tape Write operation, READ DEVICE is deactivated, and data bytes are clocked in from the Byte Assembly Register.)

As soon as they are read into the Input Register, data bytes are "rippled" through the SILO (alternately through its odd and even sides) and "stacked up" at the output end, where they are alternately clocked into the SILO Output Register. As each 8-bit data byte arrives in the SILO Output Register by a control signal from the byte control logic, a corresponding 4-bit byte of control information is clocked into the 4-bit Parity and Control Output Register.

The four bits in the Parity and Control Output Register consist of one parity bit (CBO CU ODD PAR) for each incoming data byte in the SILO during a Read operation, another parity bit (CBO LOCAL ODD PAR) for each outgoing byte during a Write operation, and two operational control bits. During a Read operation, CBO STATUS IN, as it emerges from the SILO, terminates any further data transfer from the SILO to the byte assembly logic. Also, following transfer of the Data Register contents over the IBUS to the Memory Bus Register, BLOCK DONE is set to inform the microprocessor that the last word in the data block has been transferred. During a Write operation, CBO LAST BYTE is sent through the SILO following the last character in the data block.

5.5.4 Memory Bus Data Transfer

Data words or instructions (36 bits plus parity) are direct-set into the DX10 Memory Register as soon as they are placed on the Memory Bus data lines (MBD0–MBD35) by the DECsystem-10 memory unit. Parity from the Memory Bus is checked whenever the last half of the word is transferred out of the Memory Register onto the IBUS. If parity is incorrect (even), the MEM PAR ERR bit in the Channel Command Register is set, causing an immediate interrupt to the DECsystem-10 CPU, and the transfer is halted. Data is transferred between the Memory Register and the 18-bit IBUS in two half-word transfers under control of an RSEL10 and an RSEL11 selection and special memory control and gating logic. This same logic controls readout of the Memory Register onto the Memory Bus as well as interface control of the Memory Bus and multiplexer control cable logic.

5.5.5 I/O Bus Control

Interface with the system I/O Bus (utilizing the CONI/ CONO, DATAI/DATAO, and priority interrupt facilities) is accomplished through the use of multiplexing switches, an intermediate I/O Bus (IOB IBUS), an 18-bit Channel Command Register, and the necessary I/O Bus drivers and receivers to provide conversion between the negative (0 V to -3 V) I/O Bus logic signals and the positive (0 V to +5 V), TTL-compatible logic signals used in the DX10. Communication between the IOB IBUS and the 18-bit IBUS is accomplished through an 18-bit multiplexer switch which is enabled by a DATAO or can alternatively be used (when selected by an RSEL16) to transfer the Initial Channel Program Counter (ICPC) and the Memory Cycle bit onto the IBUS. (The Memory Cycle bit should be OFF whenever it is read by a DATAI. If it is a 1, either an illegal DATAI has been executed or a memory control logic failure has occurred.) Data from the IBUS is inverted onto the MA IBUS before being read out to the I/O Bus through a multiplexer switch by a DATAI. (A CONI is normally used to read the contents of the Channel Command Register through this switch.) The PIA number from the I/O Bus is decoded and gated by any of four interrupt conditions to select the corresponding interrupt line (P1–P7) to the DECsystem-10 CPU. Setting and clearing of individual CCR bits are discussed in Chapter 3. An additional multiplexer switch allows either a CONI to examine the microprocessor Run bit from the Omnibus or a DATAI to read out the contents of the RSEL Register in I/O Bus bit positions 14–17 whenever the IBUS is being examined by the CPU.

Data interface between the microprocessor and DECsystem-10 memory via the Memory Bus interface logic is implemented through the 8R Register file (four 36-bit registers individually addressable from the micro processor, by means of IOT instructions, through the use of an incrementing/decrementing counter used as an 8R Select Register).

Communication between the microprocessor and other data elements (registers, counters, multiplexers, etc.) in the DX10 is implemented over the IBUS utilizing tristate registers and buffers individually selectable through the RSEL Register.

5.5.6 Feature Register

The Feature Register allows hardware-dependent variables, such as unit serial number, ECO level, etc. to be entered manually through a switch register (located on the M8597 module) for subsequent program readout by an operating program or for display on the maintenance panel indicators by maintenance personnel. Functional use of the Feature Register is discussed in Paragraph 5.6.2.1.

5.5.7 Maintenance Panel

A maintenance panel is provided to permit service personnel to examine various registers, counters, etc. connected to the IBUS under certain specified conditions. Functional operation and use of the maintenance panel is discussed in Paragraph 5.6.2.1.

5.6 FUNCTIONAL CIRCUIT DESCRIPTION

5.6.1 Introduction

As explained in the previous section, operation of the DX10 Data Channel in transferring information between DECsystem-10 memory and the TU70 Magnetic Tape Subsystem (once the transfer has been initiated by the PDP-8/A Microprocessor) is fully automatic under control of the DX10 Data Channel byte assembly and transfer control logic. The DX10 design incorporates the necessary features to allow the system monitor or other Executive Mode Programs to read or write directly into or out of various DX10 registers, counters, etc. via the IBUS and the system I/O Bus using DATAI or DATAO instructions. Using these same instructions, information can be read out of or stored in individual control storage memory locations by means of the PDP-8/A transfer control logic.

In addition to being able to perform the above mentioned data transfer and operational monitoring functions, the DX10 design also incorporates the logic necessary to execute a variety of diagnostic functions under either DECsystem-10 diagnostic program or PDP-8/A control utilizing the IBUS and/or the Omnibus for control information and diagnostic data transfers. For example, the DX10 includes a Loop Write-to-Read feature which permits diagnostic testing of the data channel by looping bytes of data transferred to the Channel Bus Out line drivers back to the Channel Bus In line receivers so that they can be read by the diagnostic program using DATAI instructions. In addition to programmable diagnostic features built into the DX10, the I/O processor also includes a diagnostic maintenance panel. This feature allows manual selection of any of the DATAI-selectable registers, counters, etc. for viewing purposes through the use of two thumbwheel switches and two sets of light-emitting diodes (LEDs): one to verify the register selected, the other for display of the register contents.

The contents of the selected register (counter, etc.) are transferred onto the IBUS (and thus to the LED drivers) as a result of activation of the transfer logic by a small, EXAMINE pushbutton (also located on the panel).

The circuits associated with the various operations performed by the DX10 are shown in Figures 5-9 through 5-26. These circuits generally correspond to the organization and data flow shown in the DX10 block diagram (in the engineering drawing set) and are functionally described in the following sequence:

Figure 5-9	I/O Bus to IBUS Transfers
Figure 5-10	I/O Bus and IBUS to Channel Command Register (CCR) Transfers
Figure 5-11	Memory Address Interface, RSEL0, CPC/DAC to IBUS Transfers
Figure 5-12	Memory Bus and IBUS to Memory Register Transfers, Memory Transfer Control
Figure 5-13	Multiplexer/Memory Control Interface, RSEL10–13
Figure 5-14	Memory Data Transfer Control
Figure 5-16	Omnibus and IBUS to 8R Register Transfers
Figure 5-17	RSEL4–7, 15, Omnibus to IBUS Transfers
Figure 5-18	PDP-8/A Transfer Control
Figure 5-19	PDP-8/A IOT Decoders
Figure 5-20	Channel Bus Control Register, Data Transfer Control Logic
Figure 5-21	Bit Counter, Byte Assembly Counter
Figure 5-22	Byte Counter, Byte Disassembly Register, Data Register
Figure 5-23	Channel Bus In, SILO, Byte Assembly Register
Figure 5-24	Byte Transfer Control Logic
Figure 5-25	Channel Bus Output Register, Tag Out Register, Channel Bus Out, Tag Out
Figure 5-26	Tag In, Tag/Channel Bus to Omnibus/IBUS Transfers

5.6.2 Circuit Functions

The DX10 Data Channel consists of a number of hex- and quad-height PC boards inserted into individual slots in two jumpered backplanes, both of which are mounted, adjacent to each other, in a standard, 19-inch, DEC H950-A equipment rack, with the maintenance panel, the I/O Bus, Memory Bus and Channel Bus quick-latch connectors, the H716 and H740-D Power Supplies, the 861-C Power Control, the 836 Remote Power Control (margin) Panel, and the power connector bracket assembly.

In addition to the M564 and M664 single-height Bus Driver and Receiver Interfacing modules, the following modules are used in the DX10 to carry out the various operating functions associated with data channel I/O processing.

Two M8595 Memory Bus Interface Modules
One M8596 Channel Bus Interface Module
One M8597 PDP-8/A Interface Module
One M8598 I/O and Memory Control Module
One M8599 Byte Assembly Logic Module

Each of these modules performs a unique function within the DX10 as indicated by the module name. Within each module, however, a number of operational functions are performed in carrying out the microprocessor and data channel operations described in the previous section.

Each of the functions depicted in Figures 5-9 – 5-26 follows an overall operational sequence rather than a strict hardware module orientation. However, in order to aid in locating individual circuits and their related input/output data and control signals, each operating module has been generally subdivided into discrete functional areas. Correspondingly, each signal generated in the DX10 carries a prefix which uniquely identifies the functional area where that signal originates.* Figure 5-27 illustrates each of the functional areas (and prefixes) defined for the DX10 and the individual modules on which each is located. Table 5-3 cross-references each of the major registers, counters, etc. by function and module.

The major control flip-flops and register signals used to carry out individual DX10 circuit functions are listed in Table 5-4 and are explained in the following functional circuit descriptions.

5.6.2.1 I/O Bus to IBUS Transfers (Figure 5-9) – Communication between the DECsystem-10 CPU and the DX10 is implemented utilizing DATAI and DATAO instructions over the system I/O Bus and the DX10 IBUS. Monitor program control of the channel is also exercised over the system I/O Bus through the use of CONI and CONO instructions operating in conjunction with the PDP-8/A Microprocessor through the DX10's Channel Command Register. (Refer to Paragraph 3.4.1 for details related to program operation.)

Both I/O Bus to IBUS data transfers and monitor/microprocessor control/interface operations (through the Channel Command Register) are implemented in the M8598 I/O and Memory Control module. As shown in Table 5-3, the M8598 module contains the Channel Command Register, the RSEL Register, and the logic inverters necessary to transfer 18 bits of information from the IBUS to the (inverted) MA IBUS. The M8598 also contains the Data Address Counter (DAC) and the Channel Program Counter (CPC), in addition to the maintenance panel indicator drivers, the I/O Bus device selection and gating logic, the Memory Register gating controls, and the logic necessary to gate the Memory Register, the CPC, and the DAC onto the IBUS for transfer or display.

*The exception to this convention is encountered in the case of bus signals (Memory Bus, Omnibus, I/O Bus, etc.) which carry no identifying prefixes.

Table 5-3
Locations of Registers and Counters,
by Module and Function

Module	Function	Register, Counter, Etc.
M8598	IOC	CCR (Channel Command Register)
		Output Bus
		RSEL Register
	MA	MA IBUS
		DAC
		CPC
	MC	RSEL 10XX Decoder
		MEM REG Gating Controls
		IOB Select (IOS)
	IND	Indicator Drivers
M8597	8A	IOT Decoders
	8C	Control Functions
	8R	Four 8R Registers
		8R Gating Controls
	CT	RSEL 01XX Decoder
		8RSEL Register
		MD/CPMA to Bus
		8/A Memory-8R Transfer Control
	FR	Feature Register
	IB	MA to IBUS Transfer
		MD to IBUS Transfer
		EA Data Bus to IBUS Transfer
		IBUS to 8/A Data Bus Transfer
		8/A IND1, 2 SELECT
M8596	CBI	Channel (Data) Bus In
		Channel Out to IBUS Transfer
		Tag Out to IBUS Transfer
		CBI Parity Error
	CBO	Channel Bus Output Register
		Channel (Data) Bus Out
		SILO Parity Check
	CBT	Tag Out Register
		Channel Bus Tag Out
		Channel Bus Tag In
		Tag In/Channel In to IBUS Transfer
		Tag In/Channel In to Omnibus Transfer
	CBL	Channel Bus Out to Channel Bus In Loop
	RDI	RDI First Word (Neg. Word Count)
		RDI Second Word (JRST to 100)

Table 5-4
Control Flip-Flops by Function

Module	Function	Flip-Flop/Signal
M8595	MBI	No Control Flip-Flops
M8596	RDI	RDI First Word
		RDI Second Word
M8597	8C	8C LA
		SING STEP
		8C CONT
		INITIALIZE
		STOP
		EV PAR
	CT	RUN
		SEL
		8R RDY
		WRITE
		8R RT
		8A MEM REQ
		MPX CLR
M8598	IOC	DATAO
		CONO SET
		DATAI
		IOC IN
		PI1–7
		REG SEL 0–3
		HALT 8A
		CLR 8A ERR
		CLR MEM PAR ERR
		CLR NXM
		CLR STAT AVAIL
		IOC RDI
		IOC INIT
		MEM PAR ERR
		IOC PAR ERR
		IOC FULL PAR
		IOC NXM
		IOC STAT AVAIL
		IOC CPU CLR
		IOC STAT REQ
		IOC CSR FLAG
		IOC CLEAR
		IOC CONT
		IOC SKIP (MUX)
		8A ERROR

Table 5-4 (Cont)
Control Flip-Flops by Function

Module	Function	Flip-Flop/Signal	
M8598 (Cont)	MA	MA USER	
		CPC DN	
		DAC DN	
	IND	SW LOAD RSEL	
M8599	CBC	DATA REQ	
		SET DR RDY	
		DR RDY	
		START SHIFT	
		BYTE RDY	
		SET MODE (INV)	
		WCOF	
		SPECIAL BYTE (GATE)	
		FIRST BYTE	
		DR TO MEM (GATE)	
		ZERO FILL	
		CLR DR RDY	
		CBC RUN	
		MODE 0	
		MODE 1	
		READ DEVICE	
		BACKWARD	
		MEM INHIBIT	
		RUN B	
		CBC SET MODE (INV)	
		BIT CTR OF	
	CB ASSY CTRL	BYTE AVAIL	
		NEXT BYTE IN	
		NEXT BYTE OUT	(MUX)
		ODD PAR BIT	
		BLOCK DONE	
		LOAD SILO INPUT REG (INV)	
		SET SVC OUT	
		CLR BYTE RDY	(MUX)
		SI RDY	
		SO RDY	
		NEXT BYTE	
		EVEN SO	
		ODD SO	
		DUMP	
		BYTE	(DECODER)
		ASCII	
		DUMPE	
		BYTEE	
		ASCIIE	
		SIXBITF	
		PRESET 1–3 (MUX)	

Table 5-4 (Cont)
Control Flip-Flops by Function

Module	Function	Flip-Flop/Signal	
M8599 (Cont)	CB DATA REG	CB DR0–35	
	BYTE ASSY REG	CB DRA0–7	
	BYTE DISASSY REG	CB DRD0–7	
		DR SM0, 1 (GATES)	
	BYTE CTR CBO (SILO)	CB BC1–13, CB BCOF	
		ODD BYTE IN	
		ODD BYTE OUT	
		SEL 0–7 (EVEN/ODD SILO OUT)	
		EVEN, ODD IR	
		EVEN, ODD OR	
		ODD, EVEN SHIFT IN	
	CBYTES IN/OUT (SILO CTRL)	PAR IN	
		CU ODD PAR	
		LOCAL ODD PAR	
		STATUS IN	
		LAST BYTE	
		LAST BLOCK	
		STOP	
		DEVICE DONE (GATE)	
	CBS	CBS SCAN CLOCK	
		CLOCK	
		BC TO BUS	(DECODER)
		DR TO BUS	
		TAG TO BUS	
		BUS TO BUS	
		BUS TO BC	
		BUS TO DR	
		BUS TO TAG OUT	
		BUS TO BUS OUT	
		INIT	
		FRM BUS (INV)	
		TO BUS	
		REG SEL (GATE)	

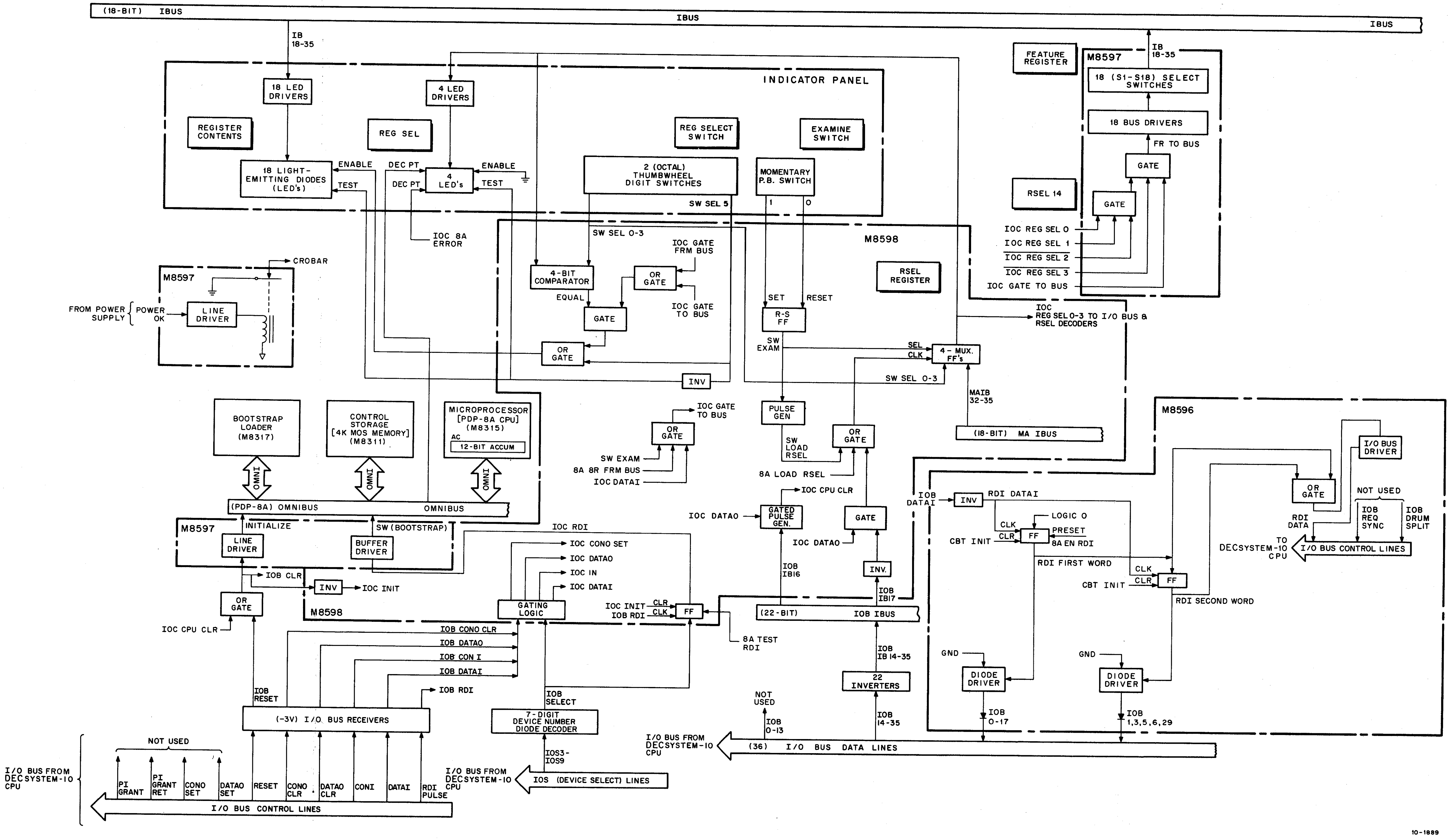


Figure 5-9 I/O Bus to IBUS Transfers

Figure 5-9 illustrates five major DX10 functions as follows:

1. I/O Bus Control and Data Transfers

2. Microprocessor-Omnibus Transfers

3. Maintenance Panel Operation

4. RSEL Register Selection and Loading

5. Feature Register Selection

As shown in Figure 5-9, information on the DECsystem-10 I/O Bus is constantly being transferred from the I/O Bus data lines through 22 M564 Inverting Line Receivers to an intermediate input bus within the DX10 referred to as the IOB IBUS. It is from the electrically isolated IOB IBUS (and its output counterpart, the IOB OBUS, operating in conjunction with 22 M664 Line Drivers) that the DX10 performs all of the data transfers to or from the DECsystem-10 I/O Bus. (The IOB OBUS is illustrated in Figure 5-10.)

Bit 16 from the IOB IBUS is gated by a DATAO instruction to a gated pulse generator (when bit 16 of the DATAO instruction is set and the DX10 is selected with the proper device code) to accomplish a programmed, DX10 Reset by asserting IOC CPU CLR. Bit 17 is gated by a DATAO to set a new value into the RSEL Register for CPU selection of a particular device connected to the IBUS, prior to reading or loading that device. The remaining bits (IOB IB18–35) are used for communication with the Channel Command Register and the IBUS and are discussed in the next paragraph.

Control of the DX10, as previously stated, is accomplished through the use of DATAO, DATAI, CONO, and CONI instructions. As shown in the diagram, these I/O Bus signals are gated into the DX10 input/output control (IOC) logic by the IOB SELECT signal, which is asserted when the device number for the DX10 is placed on the I/O Bus IOS lines by the CPU. IOC CONO SET is used to set and clear individual flip-flops in the Channel Command Register. (The CONO CLR pulse is used to generate IOC CONO SET in the DX10 instead of the CONO SET pulse because of timing considerations.) IOC DATAO is used to gate information onto the IBUS from the IOB IBUS, in addition to gating bits 16 and 17 of the DATAO instruction as described above. IOC IN is asserted by either a CONI or a DATAI and is used to gate either the RSEL register contents and the information on the IBUS (during a DATAI) or the Channel Command Register and the microprocessor Run bit (during a CONI) out to the IOB OBUS for transfer over the I/O Bus data lines to the DECsystem-10 CPU. The IOB RDI signal clocks the IOC RDI flip-flop to assert the SW (BOOTSTRAP) signal to the Omnibus when the DECsystem-10 console READ-IN switch is actuated with the DX10 selected. Asserting the SW signal causes the bootstrap loader module to read in the loader program to the control storage memory over the Omnibus and start the PDP-8/A Microprocessor at the starting location of the loader program. The IOB RESET signal (System Reset) generates Initialize to the Omnibus, which clears the microprocessor AC, Link, and Current interrupts. It also activates IOB CLEAR, which clears the Channel Bus Tag Out Register and the I/O Bus RDI logic. PI GRANT/RET, CONO SET, and DATAO SET are not used in the DX10.

In addition to the I/O Bus Control and transfer functions shown in the diagram, the PDP-8/A Microprocessor interface with the Omnibus is also depicted. Transfers between the control storage module and the PDP-8/A CPU module are carried out over the Omnibus under program control from the PDP-8/A CPU (M8315) module. (Refer to the PDP-8/A Miniprocessor Handbook and/or the PDP-8/E, PDP-8/M and PDP-8/F Small Computer Handbook for details on the PDP-8/A and Omnibus operation.) DX10 to PDP-8/A data and control transfers between the IBUS and/or DX10 control logic and the Omnibus are controlled by logic circuits within the DX10 and are discussed in subsequent paragraphs.

Operation of the maintenance panel is limited to manually selecting any of the program-selectable registers, counters etc. (by means of two RSEL thumbwheel switches) and transferring the contents to an octal character (18 binary bits) LED display on the panel (by means of an EXAMINE pushbutton) instead of to the DECsystem-10 CPU. As in the case of CPU program operations, the maintenance panel EXAMINE switch should not be actuated to view a register while the microprocessor RUN bit is asserted. Before attempting to select any IBUS register, the DECsystem-10 monitor can "look at" the state of the Run bit by issuing a CONI and examining bit 17 on the I/O Bus. This same capability is accomplished on the maintenance panel by observing the right-hand character of the 2-octal-character Register Select (RSEL) indicator. If the decimal point adjacent to the right-hand octal digit is illuminated, the IBUS is not being used by the microprocessor, and any of the 16 "registers" may be selected for viewing on the maintenance panel by authorized maintenance personnel.

CAUTION

If the right-hand decimal point (microprocessor Run bit) is not illuminated (microprocessor is running), the EXAMINE pushbutton should not be actuated. In addition, the EXAMINE pushbutton should never be actuated if the DECsystem-10 monitor is reading or loading an IBUS register. In general, in order to avoid possible memory access conflicts, ensure that the microprocessor is halted and that the DX10 is not selected by the DECsystem-10 before actuating the EXAMINE pushbutton.

RSEL (Register Select) selection and loading can be accomplished from any of three sources:

1. The DECsystem-10 monitor or diagnostic program itself (by means of a DATAO with bit 17 set to 1).

2. Manually, by selecting a register on the two maintenance panel RSEL thumbwheel switches and pressing the EXAMINE pushbutton.

3. Under microprogram control by the microprocessor issuing an address over the IBUS and issuing a 6511 IOT, which asserts 8/A LOAD RSEL.

In the case of a program selection by either the DECsystem-10 CPU or the microprocessor, the RSEL address is transferred to the RSEL register over the IBUS. Manual selection of an RSEL address transfers the address directly from the thumbwheel switches into the RSEL register through an input multiplexer when the EXAMINE pushbutton is actuated.

Figure 5-9 also depicts the Feature Register, which is an 18-bit switch register that is used to store the DX10 unit serial number and the current ECO level, as well as the MX10 version (if used) to which the DX10 is connected.

Bit 18 is unused. Bit 19 is the Memory Bus configuration bit. Bits 20–26 represent the current ECO level (in BCD format) of the DX10. Bits 27–35 contain the unit serial number (also in BCD format). The contents of the Feature Register are transferred onto the IBUS whenever RSEL 14 is selected.

NOTE

The Feature Register should be checked at the time of installation for agreement with the unit serial number and ECO level.

5.6.2.2 I/O Bus/IBUS/CCR Transfers (Figure 5-10) – As discussed in the previous paragraph, data on the I/O Bus is constantly available on the IOB IBUS. Figure 5-10 depicts the interface between the IOB IBUS, the IBUS, and the Channel Command register. As shown in the diagram, individual bits are clocked into the CCR either from the IOB IBUS by the DECsystem-10 CPU (via CONO SET) or under control of individual strobe signals that are activated by means of individual IOT instructions issued by the microprocessor. The exception is the MEM PAR ERR bit, which is set on the trailing edge of MC DATA EN RT (used during a memory Read operation to gate the right-half of the Memory Register onto the IBUS) if the memory word transferred onto the IBUS has even parity. CCR bits are cleared individually by either the microprocessor or the DECsystem-10 CPU as described in Table 3-1. The entire CCR is cleared by a System Reset pulse, asserting IOC INIT (Initialize).

Normally, the Initial Channel Program Counter (ICPC) is available to be read out to the IBUS by selecting RSEL 16. [Selecting RSEL 16 also reads out the state of MC MEM CYCLE (Bit 26) to ensure that the DX10 is not stopped in the middle of a Memory Cycle.] When a DATAO is issued, however, the low-order 18 bits of the IOB IBUS are transferred to the IBUS for loading into a selected register, counter, etc.

The entire CCR is also normally available on the IOB OBUS (output bus) for transfer to the DECsystem-10 CPU when IOC IN is asserted by a CONI. However, issuing a DATAI selects the IBUS (IB 18–35) for transfer to the IOB OBUS so that it can be transferred to the CPU by IOC IN, which is also asserted by a DATAI. (Data transfer is actually from the MA IBUS, which is the logical inversion of the IBUS, since the IBUS is low when true.)

As shown in the diagram, when the DECsystem-10 CPU sends a CONO with I/O Bus bits 33–35 set to 1, the Priority Interrupt Assignment (PIA) Register is set with the CPU-assigned interrupt sequence (priority) number, which is gated out to the I/O Bus when any one of four interrupt conditions (MP ERR, MEM PAR ERR, NXM, STAT AVAIL) occurs.

5.6.2.3 Memory Address Interface, RSEL0, CPC/DAC to IBUS Transfers (Figure 5-11) – The memory address interface logic gates either the 22-bit Channel Program Counter (CPC) or the Data Address Counter (DAC) out to the DECsystem-10 Memory Bus during each memory cycle, depending on which user (microprocessor or data channel) initiates a memory request. The address is transferred to the MADR 14–35 bus address lines.

Register Select (RSEL) 0 asserts the correct logic signals to gate the high-order four bits of either the CPC or the DAC* to or from IBUS bits 32–35.

The CPC contains the 22-bit address of the next Channel Program instruction to be executed from DECsystem-10 memory. The DAC contains the 22-bit address of the next data location in DECsystem-10 memory to be read into or out of during a Data Transfer instruction or a Store Status operation. The low-order 18-bits of either counter are gated onto the IBUS by either an RSEL12 or RSEL13. The CPC/DAC logic also contains the gating controls to up- or down-count either counter according to the memory operation being performed.

5.6.2.4 Memory Bus/IBUS to Memory Register Transfers, Memory Transfer Control (Figure 5-12) – Gating and transfer logic is provided to transfer data between the 18-bit IBUS and both the left and right halves of the 36-bit (flip-flop) Memory Register. Data is preset into this register from the DECsystem-10 Memory Bus data lines as soon as it appears on the Memory Bus. Data is gated to the Memory Bus by the Write pulse 130 ns after WR RQ has been asserted during a memory Write operation. In addition to the 36-bit Memory Register, a parity bit (MBD PAR) is gated onto the Memory Bus from the CCR parity error logic. The parity bit is actually generated within the memory transfer control logic (MEM PARITY GENERATOR) and sent to the CCR parity error logic prior to being returned and gated out onto the Memory Bus. Write clock logic for gating both halves of the Memory Register – each half is contained in a separate M8595 Memory Bus Interface module – is also contained in the memory transfer control logic.

*The particular counter accessed depends on which of the two counters' low-order 18 bits were last accessed ffor IBUS transfer (RSEL12 or 13).

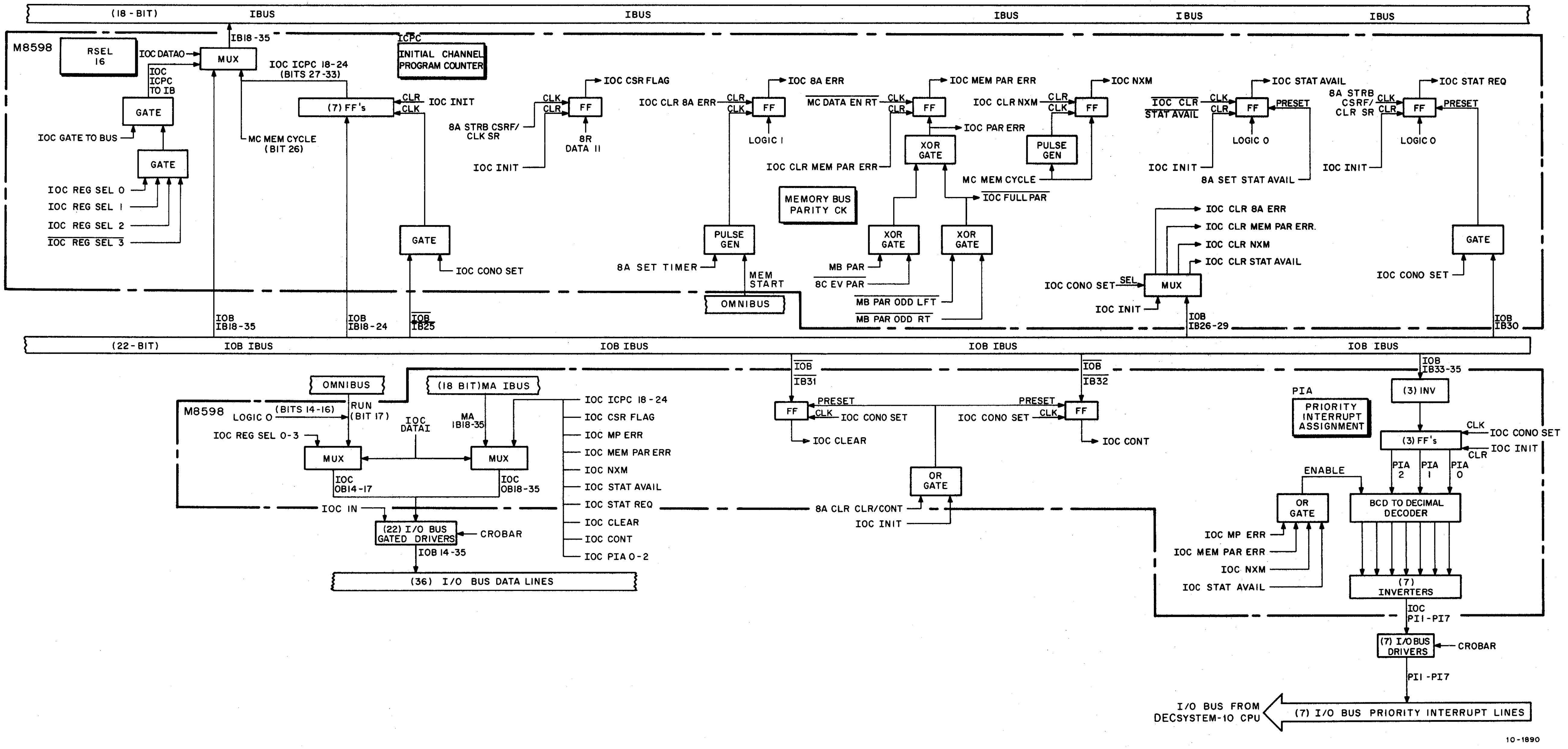


Figure 5-10 I/O Bus and IBUS to Channel Command Register Transfers

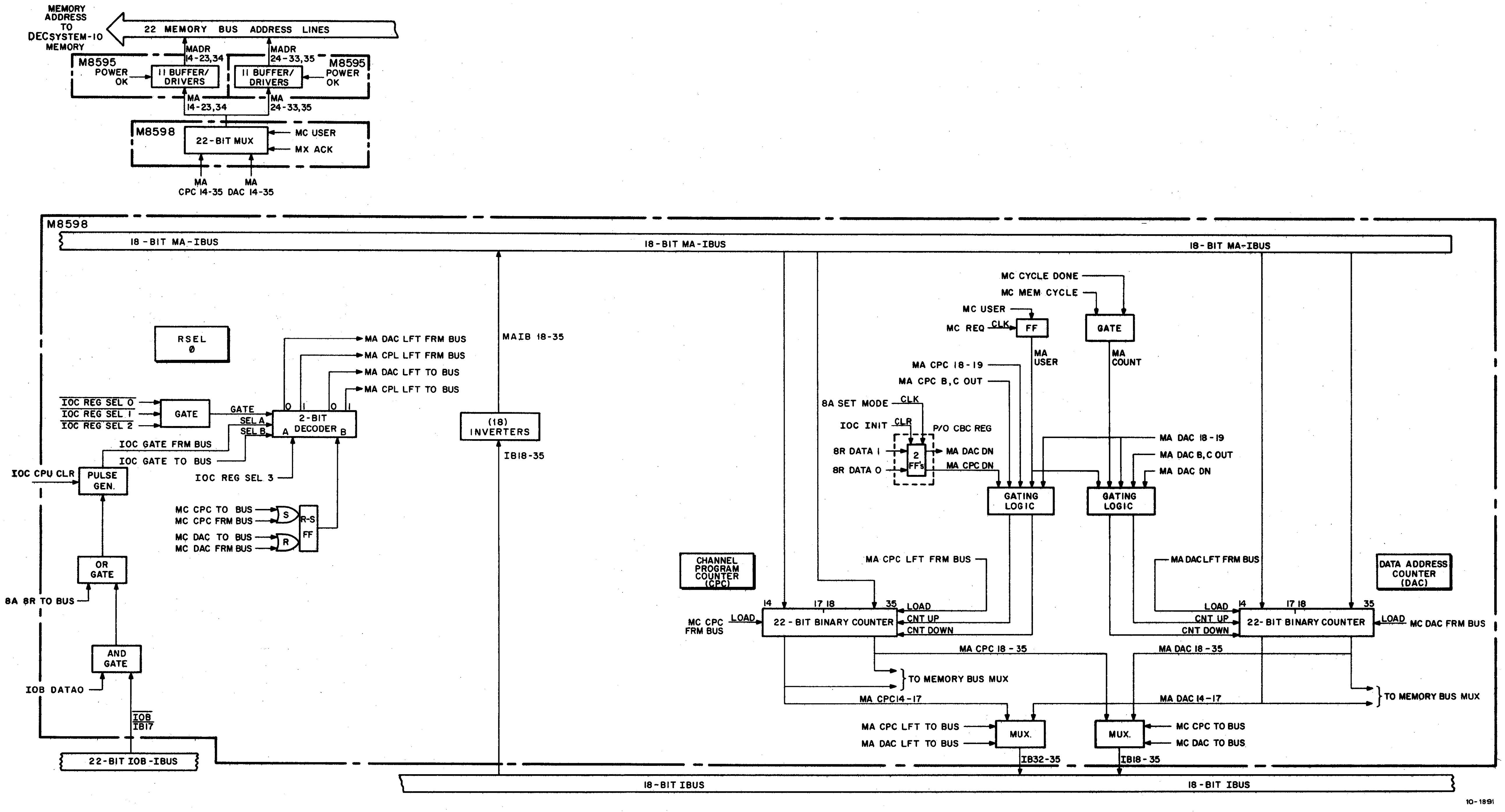


Figure 5-11 Memory Address Interface, RSEL0, CPC/DAC to IBUS Transfers

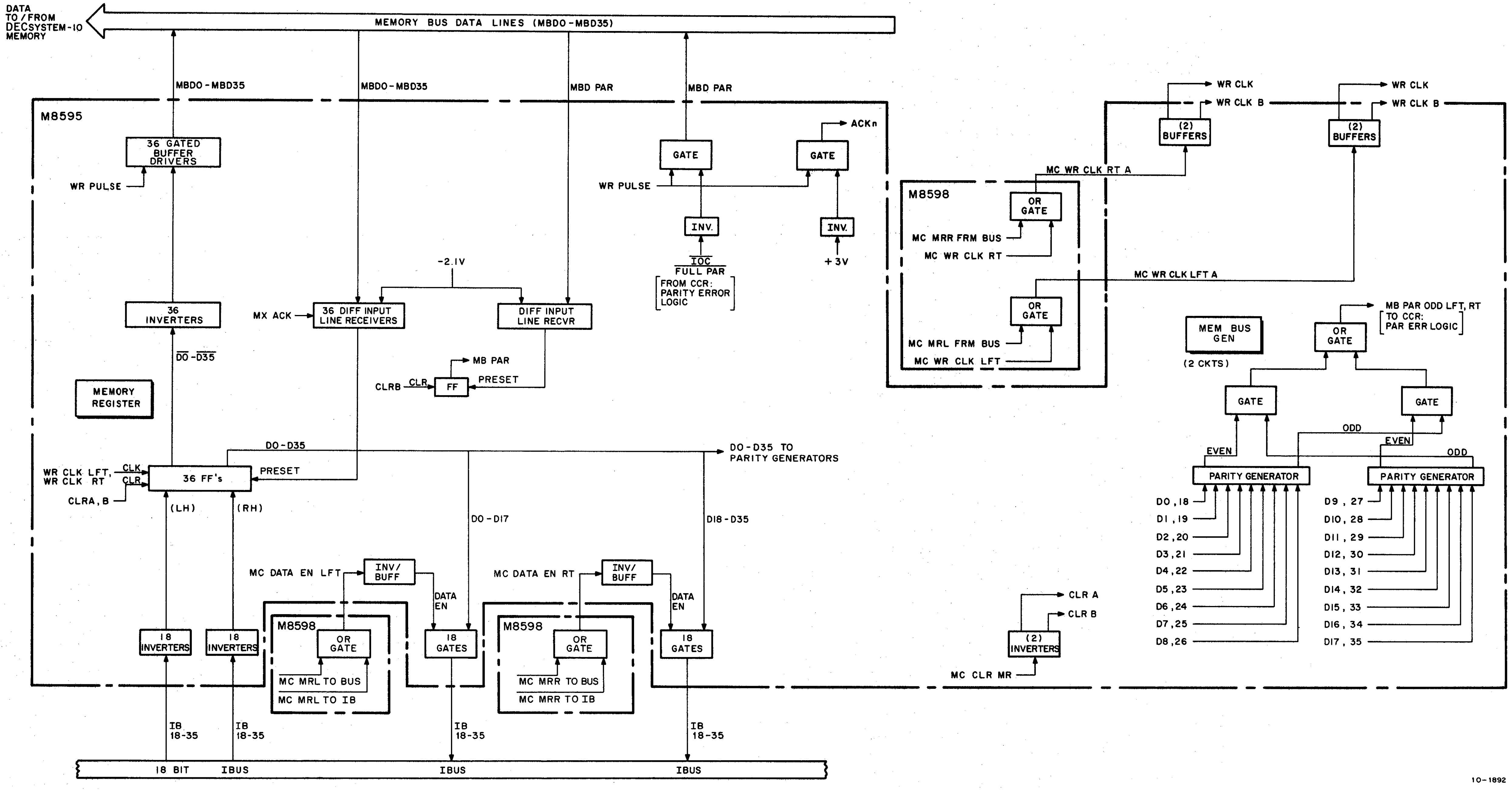


Figure 5-12 Memory Bus and IBUS to Memory Register Transfers; Memory Transfer Control

5.6.2.5 Multiplexer/Memory Control Interface, RSEL10–13 (Figure 5-13) – The multiplexer logic is used when the TU70 Subsystem is connected to a DECsystem-10 memory through an MX10-C (or similar type) Multiplexer and its associated multiplexer cable. ACKN from the MX10-C allows gating of individual read and write control signals out to the memory control lines. A nonexistent memory (NXM) time-out will cause Multiplexer Clear (MPX CLR) to be asserted on the multiplexer cable. MC MEM CYCLE generates a REQ pulse. When a multiplexer is used, the Memory Bus is routed to the multiplexer instead of being directly connected to a DECsystem-10 memory. If the TU70 Subsystem is operated without a multiplexer, the Memory Bus is connected directly to a memory port and MX ACK is continuously asserted within the DX10 multiplexer control logic. When ADR ACK is received from the Memory Bus during a Write operation, WR PULSE is generated to gate the Memory Register contents onto the Memory Bus and to assert the WR RS (Write Restart) signal to DECsystem-10 memory.

The RSEL10–13 decoder is used to generate the control signals to gate the Memory Register left- or right-half or the low-order 18 bits of either the DAC or the CPC onto the IBUS when any of four registers, RSEL10 through RSEL13, is selected for data transfer.

5.6.2.6 Memory Data Transfer Control (Figure 5-14) – The memory data transfer control logic provides the gating and logic signals necessary to gate information between the IBUS, the Memory Register, and the system Memory Bus. MC USER is continuously toggled on and off at a 10 MHz rate and used as a scanner to sample both the microprocessor and the data channel; this provides either one high-speed access to the memory control logic for the purpose of transferring program-related information between the DECsystem-10 memory Channel Program area and the microprocessor or to transfer data between the data channel and the DECsystem-10 memory data buffer area. When either the microprocessor or the data channel has been granted access to the memory data transfer logic, the User flip-flop clock (MC SCAN CLK) is turned off until the termination of the program or data transfer, at which time it resumes toggling the User flip-flop, which scans back and forth until either the data channel or the microprocessor issues another memory cycle request. Access is again granted, the clock stops, and the cycle repeats itself. Figure 5-15 illustrates a microprocessor Read Memory cycle that will cause one 36-bit word to be transferred from a DECsystem-10 memory location (addressed by the CPC) into the 8R Register. As shown in the timing diagram, the MC SCAN CLK is inactive once the MC USER flip-flop has granted access to the microprocessor. The gating and control signals necessary to perform the Memory Bus "handshaking," as well as the Memory Register left- and right-half data transfers onto the IBUS and into the 8R Register, are shown in their proper sequence.

The User flip-flop is prevented from toggling whenever a MEM PAR ERR or NXM condition occurs. This allows the DECsystem-10 monitor to examine the state of the flip-flop prior to clearing the interrupt condition in the CCR. The flip-flop is also prevented from toggling during a system Reset or microprocessor Initialize. The inhibiting signal is MC CLR, which also clears the memory control logic under the above stated conditions.

5.6.2.7 Omnibus/IBUS to 8R Register Transfers (Figure 5-16) – The 8R Register is a 36-bit by 4-word addressable register file and is accessible via the IBUS for access to memory and other registers. Which of the four file locations will be accessed is determined by the 8R SEL Register. Data is transferred directly onto the IBUS in two 18-bit transfers, with first one half, then the other being transferred from the selected 36-bit register through an 18-bit (8R to IB) multiplexer (8R0–17). In addition to the IBUS interface, the 8R can also be transferred in three 12-bit segments, one at a time, to the Data Bus (Data 0–11) under control of three separate select signals (8R RDA, B, and C).

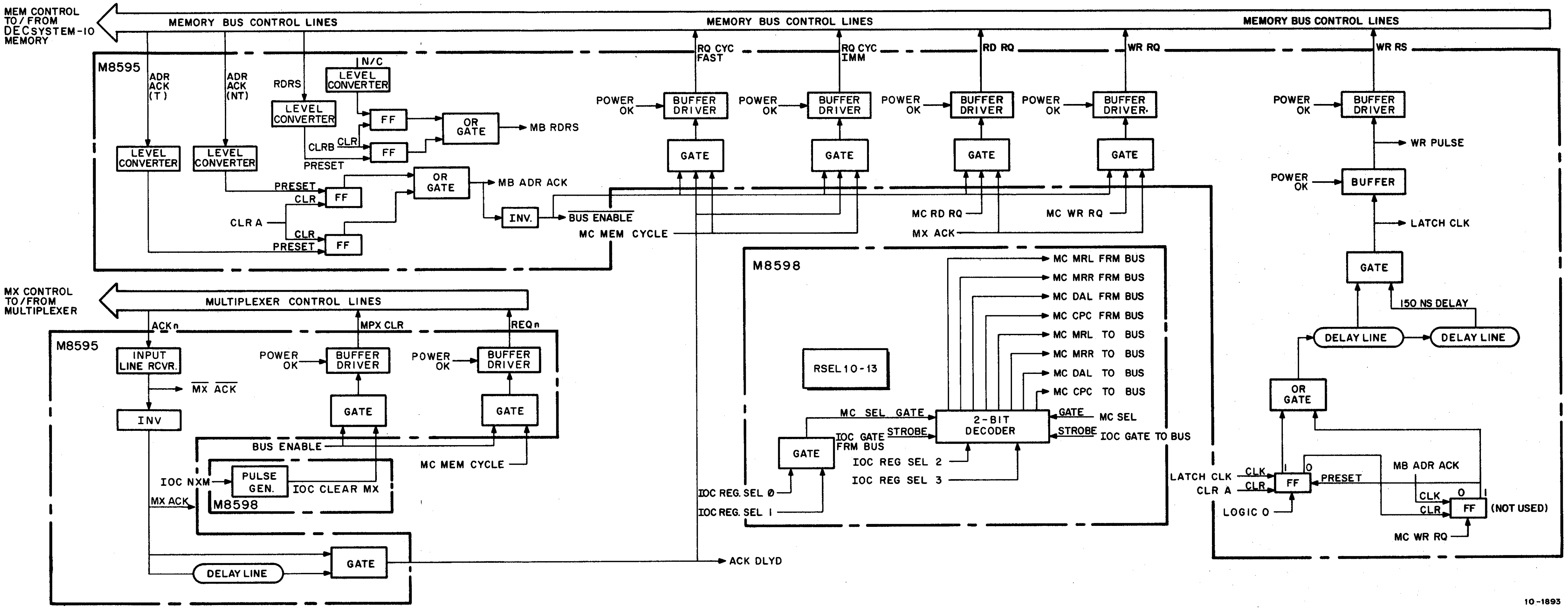


Figure 5-13 Multiplexer/Memory Control Interface;
RSEL10-13

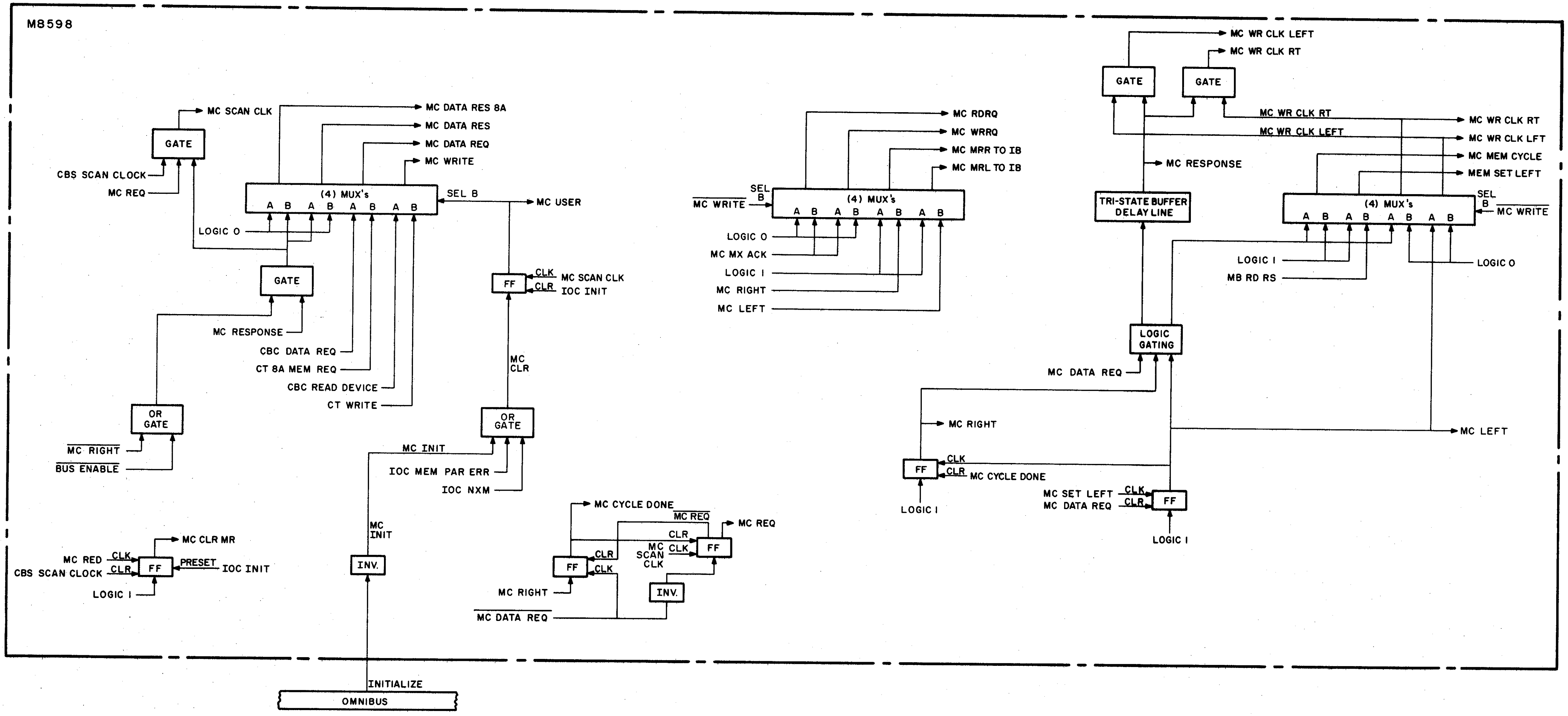


Figure 5-14 Memory Data Transfer Control

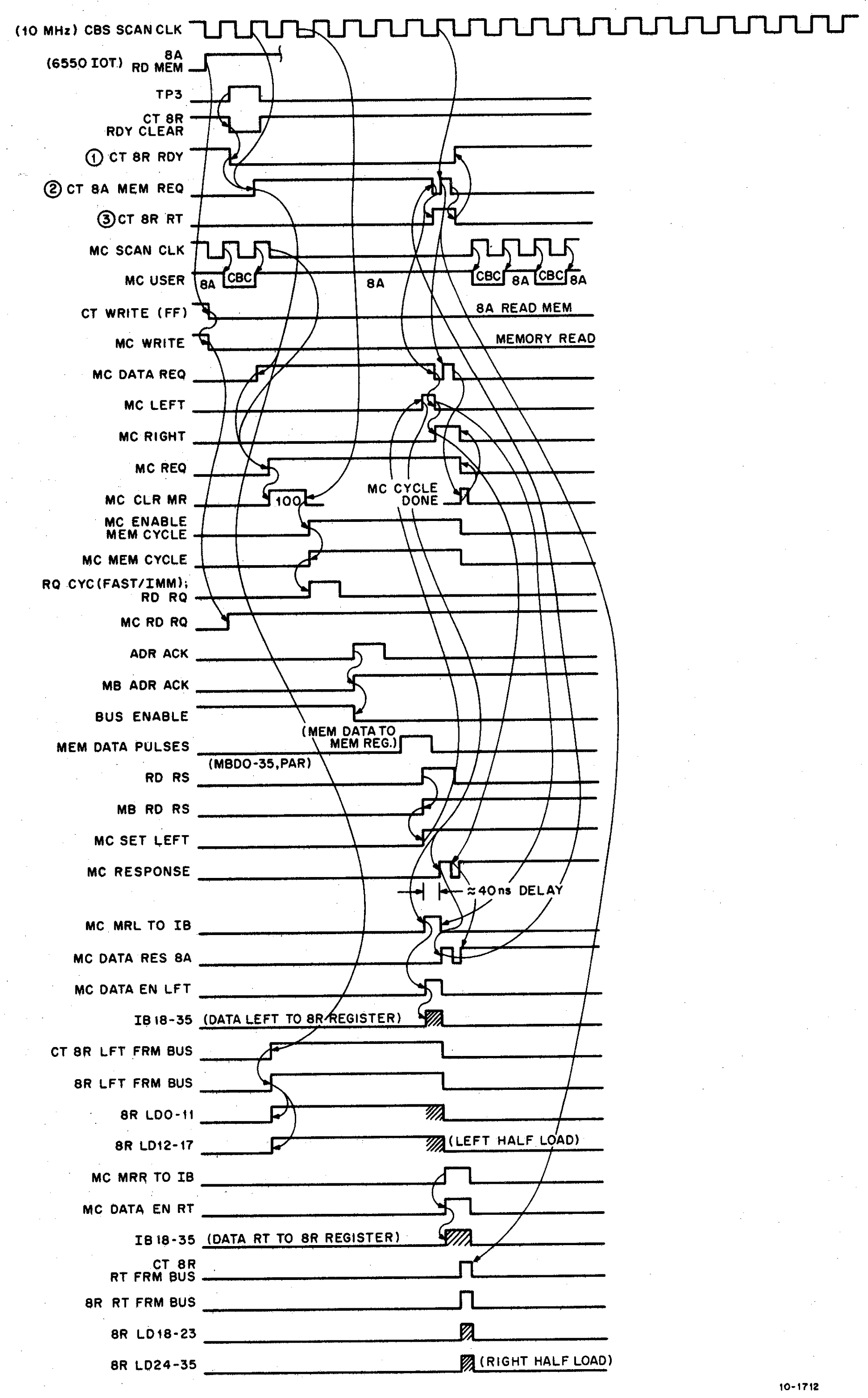


Figure 5-15 Read Memory Cycle

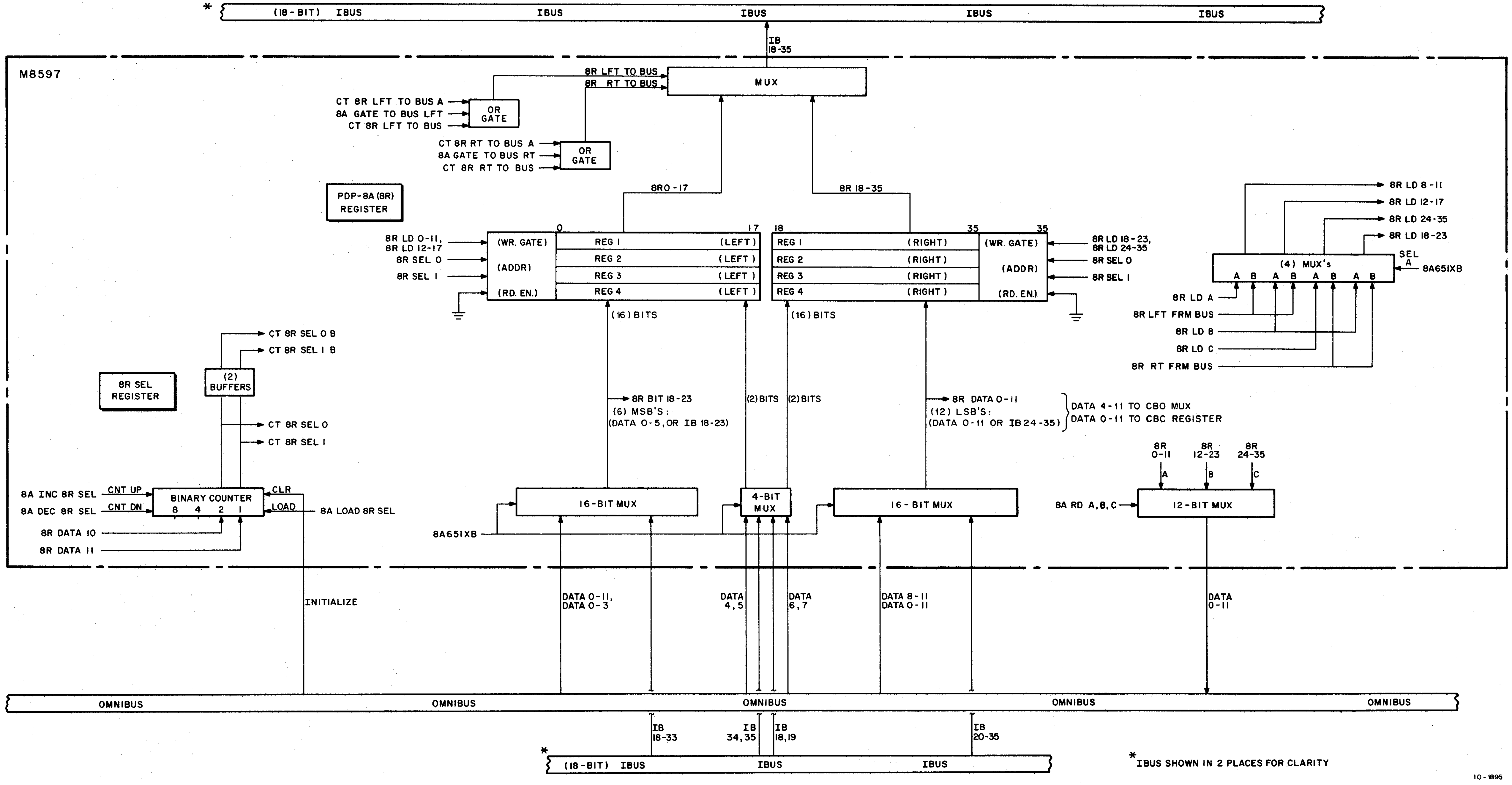


Figure 5-16 Omnibus and IBUS to 8R Register Transfers

The microprocessor initiates memory requests (CT 8A MEM REQ) by issuing either a 6550 (Read Memory) or 6551 (Write Memory) IOT. This causes a 36-bit word to be transferred between the selected 8R Register and a location in DECsystem-10 memory specified by the CPC. Data is transferred between the 8R Register and the microprocesor Accumulator (AC) in three 12-bit transfers over the Omnibus data lines under direct control of three separate Read 8R (6503, 6504, 6505) or Load 8R (6513, 6514, 6515) IOT instructions. The 8R loading signals are derived from the IOT decoders. The IBUS gating signals are generated in the PDP-8/A transfer control logic. Whenever the microprocessor is not performing an 8R Read or Load operation from the Omnibus, the IBUS is available for loading into the 8R REG (left or right) under control of either an LIL or LIR IOT instructlon (6555, 6556), an RSEL selection (RSEL 6 or 7), or the memory transfer control logic (when transferring the MR left- and right-halves into the 8R Register via the IBUS) (Figure 5-15). Data inputs to the 8R Register are also available for loading the CBC Register at the beginning of a channel operation. The low-order eight bits are available for direct transfer of a data byte from the microprocessor to the Channel Bus Out lines. The 8RSEL Register is loaded by bits 10 and 11 of a 6511 (L85) IOT instruction to specify the 8R Register selected for further use by the microprocessor. It is cleared by an Omnibus Initialize pulse whenever a system Reset or a DX10 Initialize sequence is executed, or immediately following a power turn-on sequence.

5.6.2.8 Omnibus to IBUS Transfers, RSEL 4–7, 15 (Figure 5-17) – Communication between the PDP-8/A Microprocessor and the rest of the DX10 is accomplished by either the 18-bit IBUS or the Omnibus. The DX10 transfers data, status, control information, etc. to the IBUS by means of direct 8R Register to IBUS transfers, through the use of multiplexers, or by means of individual gated buffer drivers. Data transfer from the Omnibus to the IBUS is accomplished through gated, tristate buffers that are used to gate control memory data (MD0–11) or CP Memory Addresses (MA0-11) to the IBUS when RSEL 4 or 5 is selected. The 8R Register right- or left-half will be transferred out to the IBUS when RSEL 6 or 7 is selected. When RSEL 15 is selected, the information on the Omnibus Data lines (DATA0–11) is gated to the IBUS. The information on the Omnibus will consist of any of four 12-bit words selected by bits 22 and 23 of the RSEL DATAI instruction. Either the Accumulator (AC) the Multiplier Quotient (MQ) Register from the PDP-8/A CPU or a 12-bit status word* can be placed on the 12 data lines for transfer to the IBUS. The selection codes are as follows:

00	Microprocessor Status Word
01	MQ (CPU)
11	AC (CPU)

If the monitor wants to "look at" the Omnibus, selecting 10_2 in bits 22 and 23 will result in nothing being transferred to the Omnibus data lines, and all zeros should be transferred onto the IBUS. This permits checking the Omnibus for uncleared data bits. In addition to the Omnibus data and address transfers, RSEL 4 and 5 cause PDP-8/A control and status bits to be transferred onto IBUS bit positions 18–23. RSEL 4 reads out the programmable "console switch" states (SINGLE STEP, HALT, CONTINUE, ENABLE, DEPOSIT, and EVEN PARITY) to the IBUS. RSEL 5 reads out the 8RSEL Register contents, the microprocessor Run bit, and the DX10 MC USER flip-flop.

5.6.2.9 PDP-8/A Transfer Control (Figure 5-18) – The PDP-8/A transfer control logic functions as a programmable "console" for the microprocessor. It contains all of the gating and control functions for interfacing the PDP-8/A with the DX10 via the Omnibus. (Details of Omnibus operation are contained in the Small Computer Handbook.) The 8R Register to Memory Register left-half and right-half transfer control signals (Figure 5-16) are generated in the PDP-8/A transfer control logic. STOP is also asserted here to halt the microprocessor whenever NXM, MEM PAR ERR, or INIT occurs. (The microprocessor is also stopped during a programmed Single Step (SS) operation, after each PDP-8/A CP cycle, or through a programmed Halt by an RSEL 4 with bit 19 set to 1.)

*Refer to PDP-8/E, PDP-8/M and PDP-8/F Small Computer Handbook for status word format.

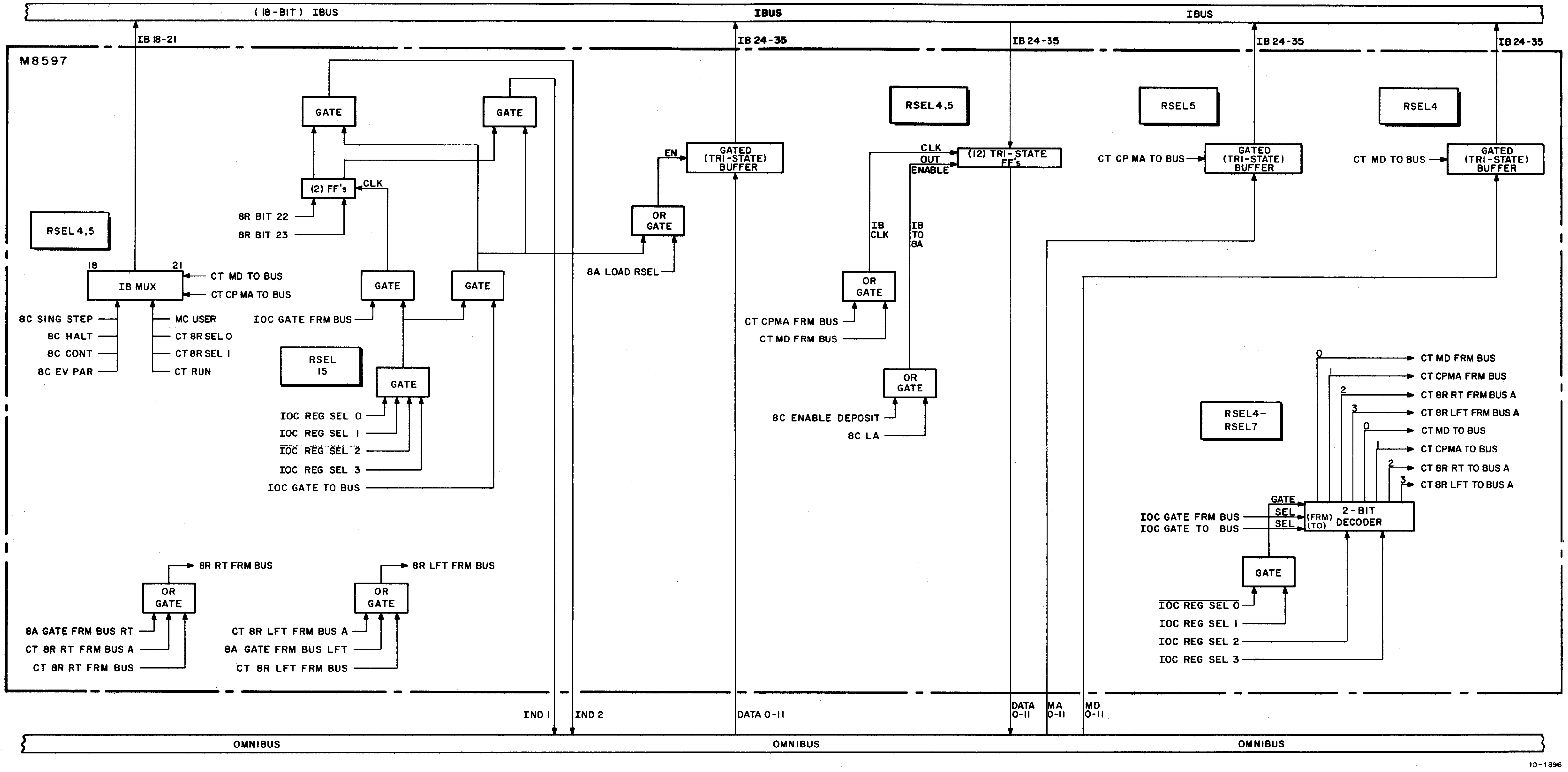


Figure 5-17 RSEL4–7, 15; Omnibus to IBUS Transfers

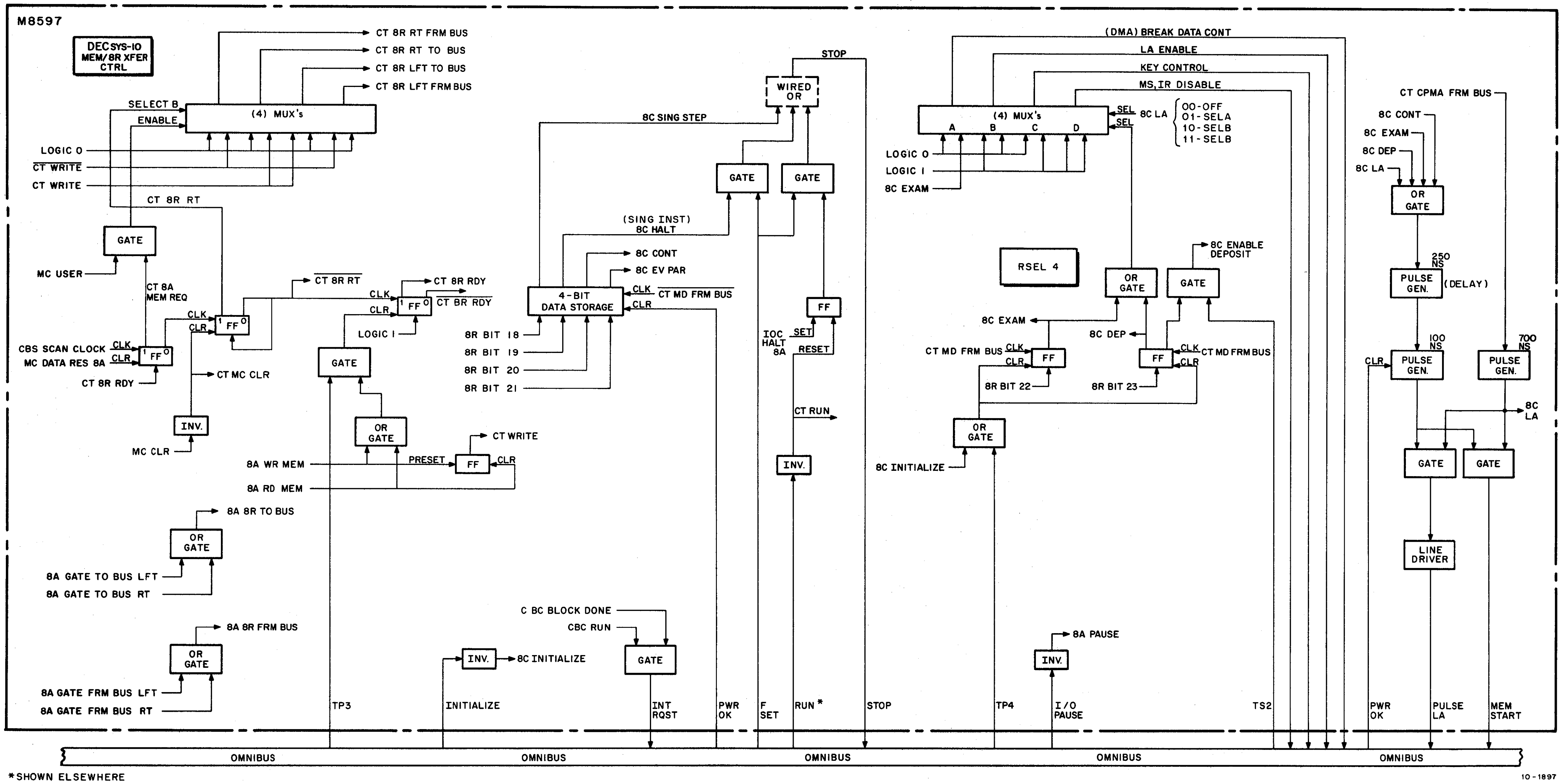


Figure 5-18 PDP-8/A Transfer Control

Initialize performs the same clearing operations in the DX10 that are carried out when a system Reset occurs. INT RQST is asserted at the end of every block transfer operation by CBC BLOCK DONE to inform the microprocessor that the required number of bytes has been transferred as specified by the Data Transfer instruction.

5.6.2.10 PDP-8/A IOT Decoders (Figure 5-19) – The PDP-8/A IOT decoders generate all of the transfer control and gating signals required to accomplish the IOT functions described in Table 5-1. A BCD decoder decodes the high-order six-bits of the IOT instruction on memory lines MD3–MD11 to generate the master gating signals (650X through 657X) for gating each of the low-order three bits (the last octal digit) in the IOT code (issued by the microprocessor) into individual BCD decoders.

The IOT decoder logic also contains two data selectors to implement each of the 652X and 654X Skip IOTs. C0 and C1 control the direction of information transfer between the Omnibus Data lines (DATA0–11) and the microprocessor (PDP-8/A CPU) Accumulator (AC).

5.6.2.11 CBC Register, Data Transfer Control Logic (Figure 5-20) – The CBC Register is loaded from the 8R Register data inputs by a 6510 IOT instruction at the beginning of every Data Transfer operation. The CBC Register is actually comprised of 12 Control flip-flops, all of which are simulatneously set by the PDP-8/A Set Mode signal from the IOT decoder. The function of each of the CBC Register control bits is described in Table 5-2. The control logic contains a decoder, which decodes the CBCR mode bits, and the Special Byte signal to produce the individual control signals for each of the data modes used with the DX10 Data Channel.

The CBC DR RDY (Data Register Ready) flip-flop controls data word transfers between the DR and DECsystem-10 Memory during data transfer sequences. When the CBC Register is initially loaded during a tape Write operation, the trailing edge of the PDP-8/A Set Mode DLYD signal causes the CBC DR RDY flip-flop to be cleared, which initiates a memory cycle and causes the first data word to be transferred from DECsystem-10 memory into the Data Register. As soon as a word has been transferred into the register, CBC DR RDY is set again, and the shift logic is activated to shift the data word out of the DR, one bit at a time, into the Byte Assembly Register. As each byte is assembled, it is transferred into the SILO and then out of the SILO onto the Channel Bus. Each time a word is fully transferred out of the DR, the CBC WORD DONE signal sets the CBC CLR DR RDY flip-flop, which clears the CBC DR RDY flip-flop and initiates another memory cycle to transfer another 36-bit word from DECsystem-10 memory into the DR to be shifted out, a byte at a time, to the Channel Bus. This continues until the Block Done signal interrupts the microprocessor, indicating that the last word in a data block has been transferred into the SILO. The Microprocessor then either reloads the CBC Register for transfer of the next block of data or terminates the Data Transfer operation.

The CBC DATA REQ flip-flop controls transfer of the 36-bit data word in two half-word transfers from the Memory Register to the Data Register.

5.6.2.12 Bit Counter, Byte Assembly Counter (Figure 5-21) – The Bit Counter is used to control the number of bits shifted into or out of the Data Register for each data byte transferred and is loaded every time a byte is ready to be transferred out of the Byte Assembly Register (BAR) during a tape Write operation or into the Byte Disassembly Register (BDR) during a tape Read. When the byte has been transferred, the CBC BYTE RDY flip-flop is cleared, and the Bit Counter is up-counted again by the same 20 MHz clock that is used to shift bits into or out of the DR. The Bit Counter is loaded automatically with the correct bit number for the data mode in which the DX10 is reading or writing tape.

The Byte Assembly Counter controls the number of bytes to be transferred into or out of the 36-bit DR for each data word.

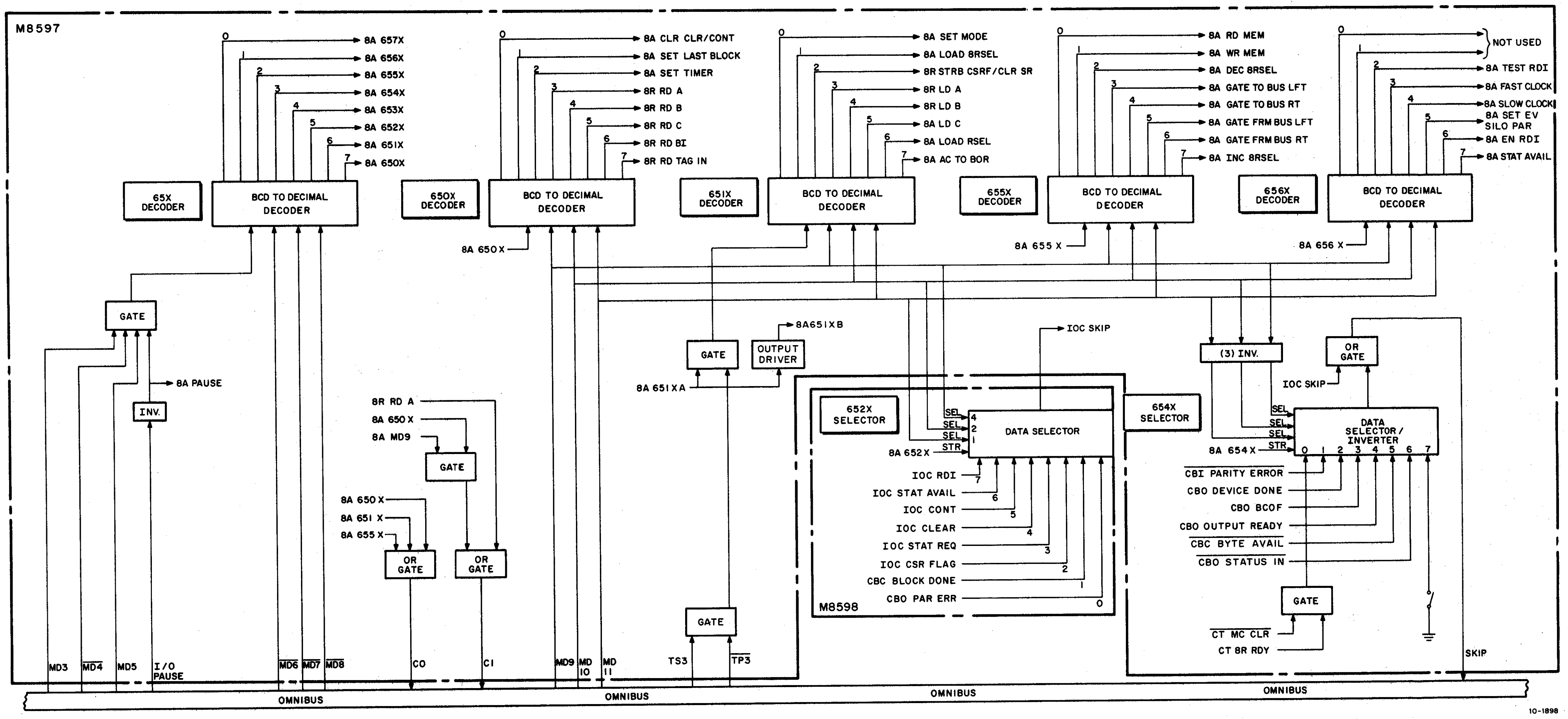


Figure 5-19 PDP-8/A IOT Decoders

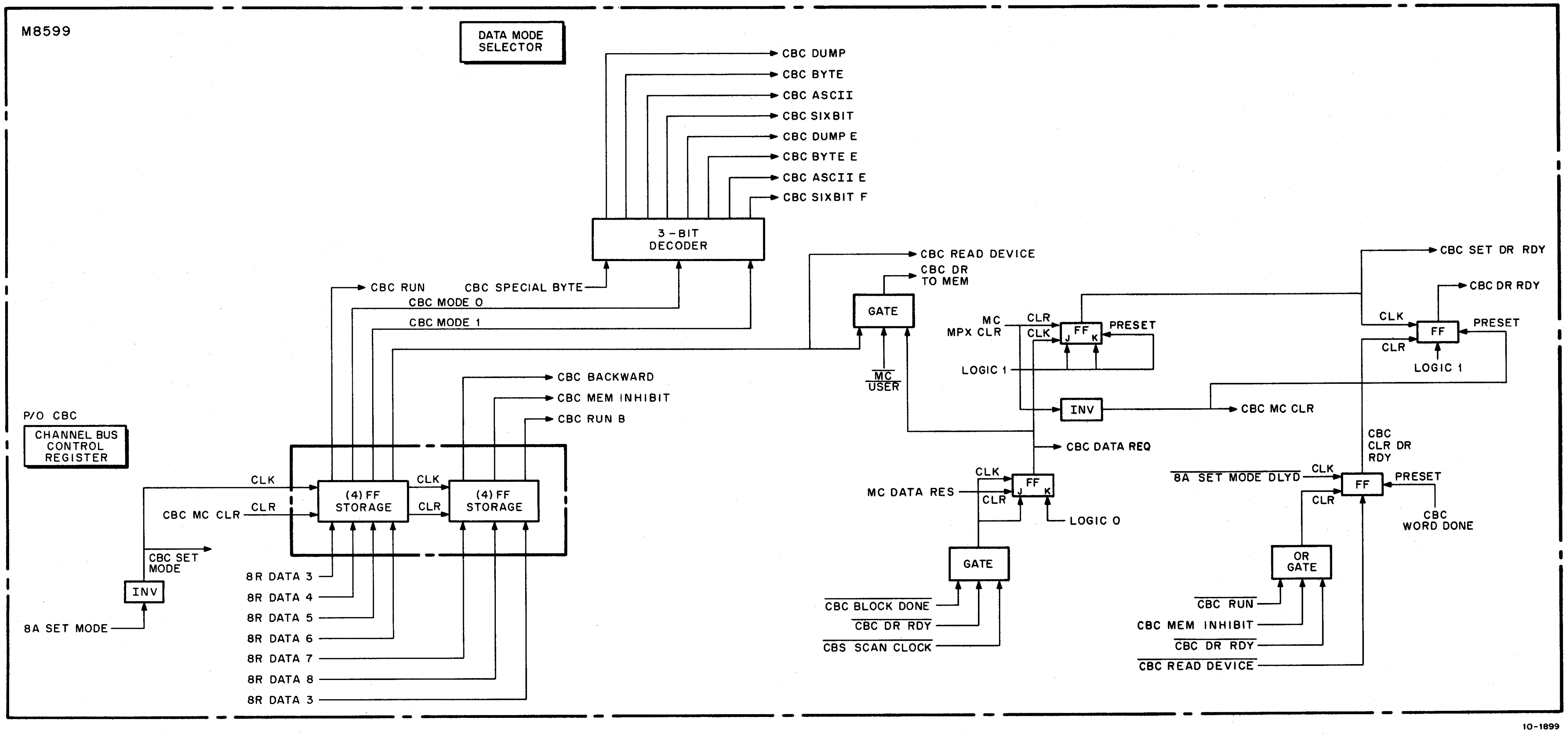


Figure 5-20 Channel Bus Control Register; Data Transfer Control Logic

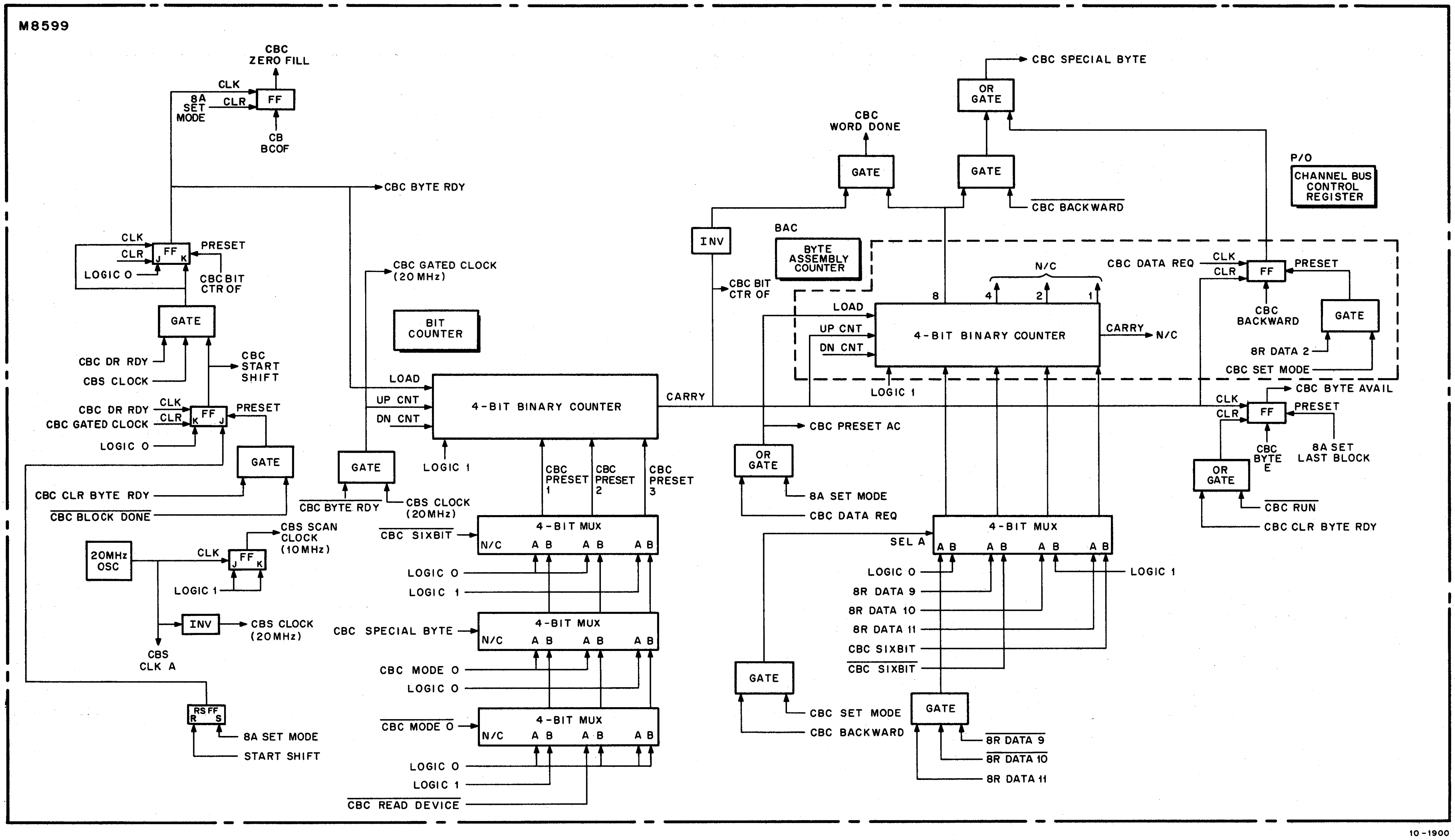


Figure 5-21 Bit Counter; Byte Assembly Counter

The BAC is loaded at the beginning of the Data Transfer operation by 8A Set Mode. Thereafter, as each word is transferred between the Data Register and DECsystem-10 memory, the BAC is loaded by the Transfer Request signal (CBC DATA REQ). The BAC is up-counted every time the Bit Counter overflows, indicating that a byte has been transferred. When both the Bit and Byte Counters overflow, Word Done is asserted, indicating that a word has been shifted into (Read) or out of (Write) the Data Register. Word Done sets the CLR DR RDY flip-flop to cause it to clear the DR RDY flip-flop. This initiates a memory cycle to transfer either another data word from DECsystem-10 Memory into the DR (during a Write operation) or the word in the DR into the memory data buffer area (during a Read).

The Byte Ready flip-flop (CBC BYTE RDY) is used to control shifting of individual bytes into or out of the DR. When CBC BYTE RDY is asserted, a data byte is transferred either from the BAR to the SILO Input Register (Write) or from the SILO Output Register to the BDR (Read). During this time, the 20 MHz clock (CBC GATED CLOCK), which is used to up-count the Bit Counter and shift data bits into or out of the DR, is turned off, and no shifting takes place. At the same time, the Bit Counter is reloaded with a new bit count. When the byte transfer (to or from the SILO) is completed, CBC CLR BYTE RDY clears the Byte Ready flip-flop, and shifting is resumed with a new bit count in the Bit Counter. When the Bit Counter overflows, the Byte Counter is up-counted, and the data transfer process continues.

5.6.2.13 Byte Counter, BDR, Data Register (Figure 5-22) – The Byte Counter controls the number of bytes transferred during each data block operation. It is preset with a program-determined byte count by the microprocessor at the beginning of a Data Transfer instruction. It is up-counted every time a byte is transferred into or out of the SILO (by the CBC BYTE RDY signal negation). When it overflows, it halts the microprocessor and asserts CBC BLOCK DONE, which causes an interrupt to the microprocessor, indicating the end of the current data block.

The Byte Disassembly Register (BDR) receives 8-bit data bytes from the SILO Output Register and shifts them out serially (forward or backward, depending on the operating mode) through a mode control gate to the left- or right-serial shift input of the Data Register. Shifting is controlled by the same 20 MHz clock (CBS CLOCK) that is used to shift the Data Register and up-count the Bit Counter.

The Data Register is a universal, bidirectional shift register which can be parallel loaded (or read out of), serial left-shifted, serial right-shifted, or held in a steady state. Data is transferred in 18-bit half-words from the CB IBUS into the right (low-order) half of the DR during a memory read cycle. On the first clock pulse, the right-half of the DR is loaded with the left-half word from the Memory Register. On the next clock pulse, the MR left-half word is parallel loaded into the left-half of the DR. Simultaneously, the right-half of the Memory Register (now on the CB IBUS) is loaded into the right-half of the DR. Following the loading operation, shifting of data, as previously described, is resumed on the next clock pulse. Left or right shifting is controlled by the CBC Backward signal and is enabled every time the Byte Ready flip-flop is reset or the Bit Counter overflows.

5.6.2.14 Channel Bus In, SILO, BAR (Figure 5-23) – Data bytes from the TX01 Control Unit are received at the DX10 through nine line receivers; they are checked for correct parity and sent to the SILO Input Register and, for diagnostic purposes, to an IBUS Multiplexer for transfers (minus the parity bit) onto the IBUS high-order eight bit positions (IB 18–25) for examination when RSEL 3 is selected, or when the parity bit is examined with the Tag In lines if RSEL2 is selected. During channel initializtion and termination, the microprocessor checks parity on every byte it reads from the Channel Bus. If an error occurs, the Microprocessor records a Sequence (fatal) error and returns to the idle loop (Chapter 4). The parity bit is separately routed through the Parity and Control SILO, and the eight data bits are clocked into the SILO Input Register. (During a Write operation, parity is generated at the input to the parity and control SILO and is sent, along with the eight data bits, through the SILO to the SILO Output Registers for gating onto the Bus Out lines.)

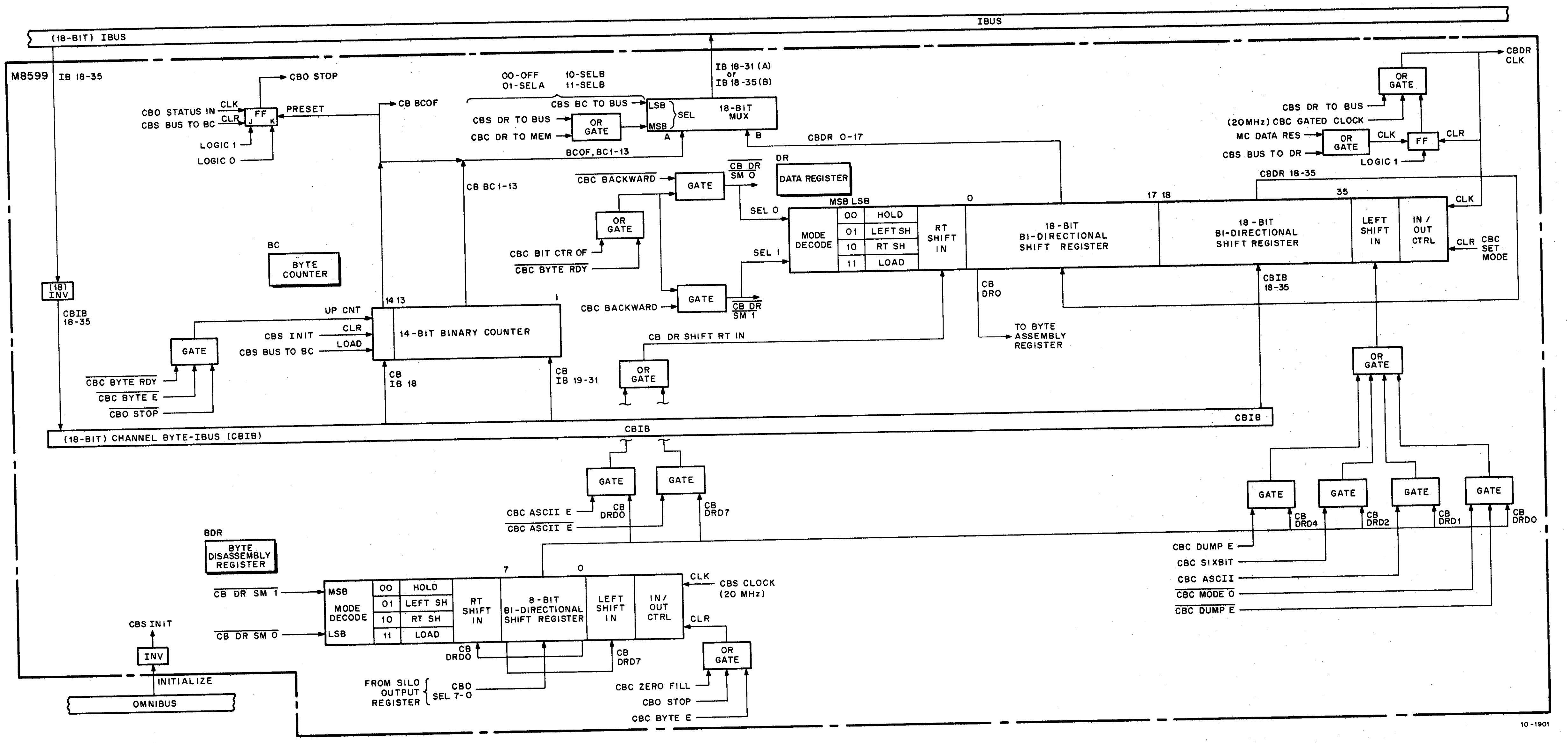


Figure 5-22 Byte Counter; Byte Disassembly Register; Data Register

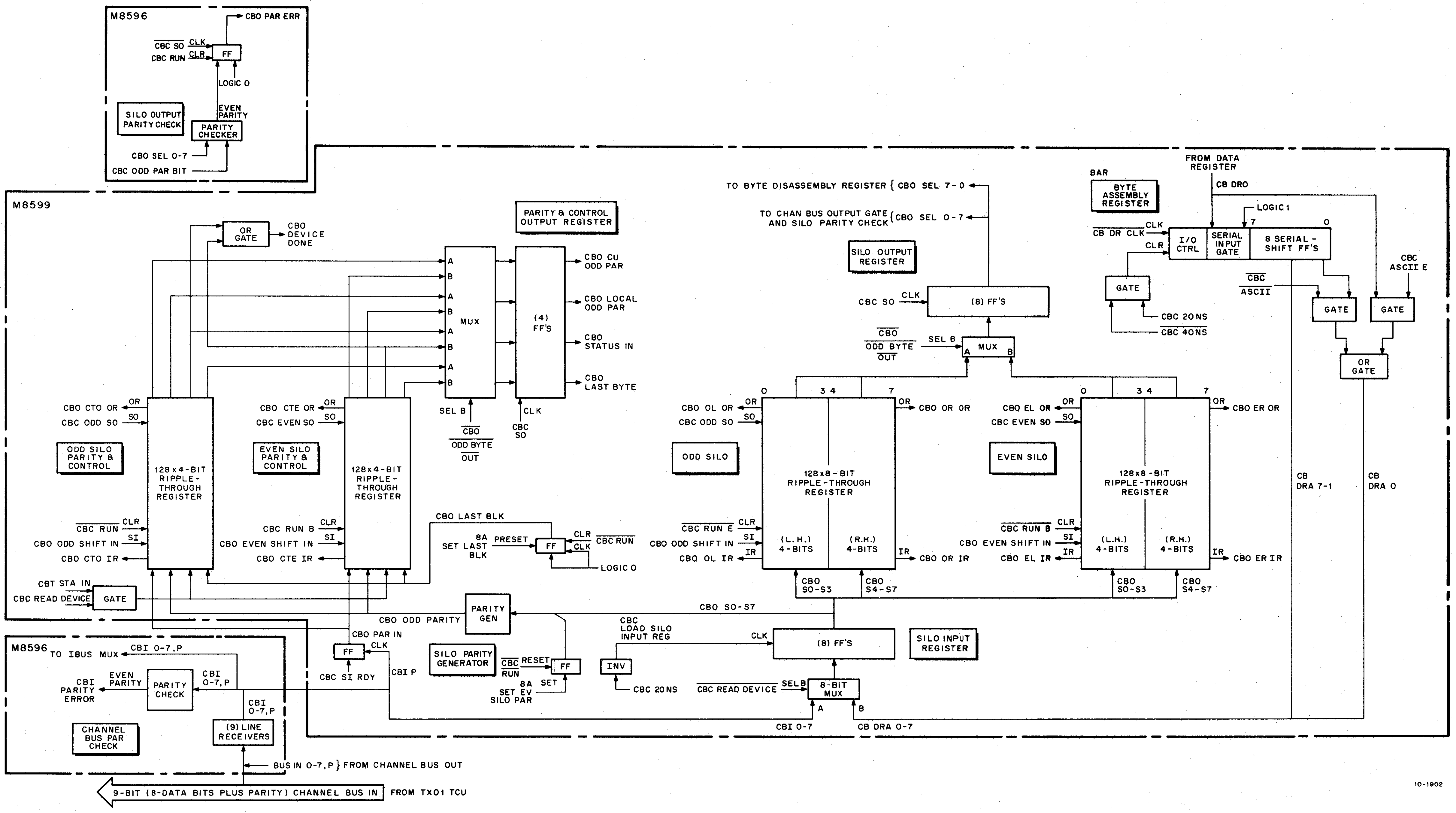


Figure 5-23 Channel Bus In; SILO; Byte Assembly Register

Parity (CU ODD PAR during a Read Device operation, or LOCAL ODD PARITY during a Write operation) is checked on each byte as the byte is transferred into the SILO Output Register. If a parity error occurs, the condition is read by the microprocessor at the end of the Block Data Transfer operation and a bit in the CSR is set by the microprocessor before status is stored in DECsystem-10 memory.

The data bits during a Write operation are clocked in from the 8-bit output of the Byte Assembly Register (BAR) after the data byte has been serially shifted in from the Data Register. Data bytes from the SILO Output Register during a Read operation are parallel-transferred into the BDR and serially shifted into the Data Register.

5.6.2.15 Byte Transfer Control Logic (Figure 5-24) - During Data Transfer operations, control of data bytes in and out of the SILO is exercised by discrete Input Ready (e.g., CBO EL IR) and Output Ready (e.g., CBO ER OR) control signals. The necessary Shift Out (CBC SO) and Shift In (CBO ODD SHIFT IN, CBO EVEN SHIFT IN) signals are generated to control transfers into or out of the SILO as bytes are available at either end and as the Byte Ready flip-flop enables each byte transfer. Other control signals are generated here for various byte transfer logic control operations.

5.6.2.16 Channel Bus Output/Tag Out Registers, Channel Bus/Tag Out Lines (Figure 5-25) - Channel Bus tags are raised (asserted) on the Channel Bus Tag Out lines by setting individual flip-flops in the Tag Out Register. These are selected to be set or cleared by bit 32 of an RSEL2 or by either a 653X or 657X IOT being issued by the microprocessor. Bits 32 through 35 of the RSEL2 (or the individual IOT instruction issued) are decoded to select which flip-flop (tag) in the TOR is set.

The Channel Bus Output Register is used by either the microprocessor or the DECsystem-10 diagnostic program to place individual data bytes on the Channel Bus Out Lines. When not being used by either the microprocessor or the DECsystem-10 diagnostic, the Channel Bus Out lines receive data bytes directly from the SILO Output Register.

A Channel Bus Loop feature is included in the DX10; this feature permits the diagnostic program to place bytes of information on the Channel Bus Out lines and, through a special set of line receivers, read them back onto the IBUS for verification. In addition, through a special set of line drivers each byte is sent to the Bus In line receivers to simulate a byte of information coming from the TX01 on the Bus In lines.

5.6.2.17 Tag Out, Tag/Channel Bus to Omnibus/IBUS Transfers (Figure 5-26) - Tag inputs from the TX01 Control Unit are received in the DX10 through seven line receivers and are routed to the byte transfer and channel control logic to control Data Transfer operations. In addition, they are sent to an IBUS multiplexer where they are available for examination by an RSEL2 selection.

The RSEL0-3 decoder generates the necessary control signals to transfer channel and tag information between the Channel Bus and the IBUS during RSEL2 and RSEL3 selection. The control signals for transferring information between the IBUS and the Data Register, DAC, CPC, and Byte Counter are also decoded here. A tristate multiplexer is also provided to transfer Channel Bus information directly to the microprocessor via the Omnibus.

5.6.3 Circuit Locations (Figure 5-27)

Each of the circuits in the preceding functional circuit diagrams is shown in its operational relationship to other functional circuits rather than in the order in which they appear within the engineering drawing set. However, each signal name is preceded by a functional designator (e.g., CBC, CT, IOC, etc.) which identifies the functional area where that signal originates. Figure 5-27 depicts the individual modules (or boards) where each functional area is located. The schematics in the engineering drawing set will call out (in each functional area) the individual edge connector pins on each module where individual signals are available for viewing (scoping) purposes. Chapter 6 of this manual includes a table which lists the most common signals and their backplane pin numbers.

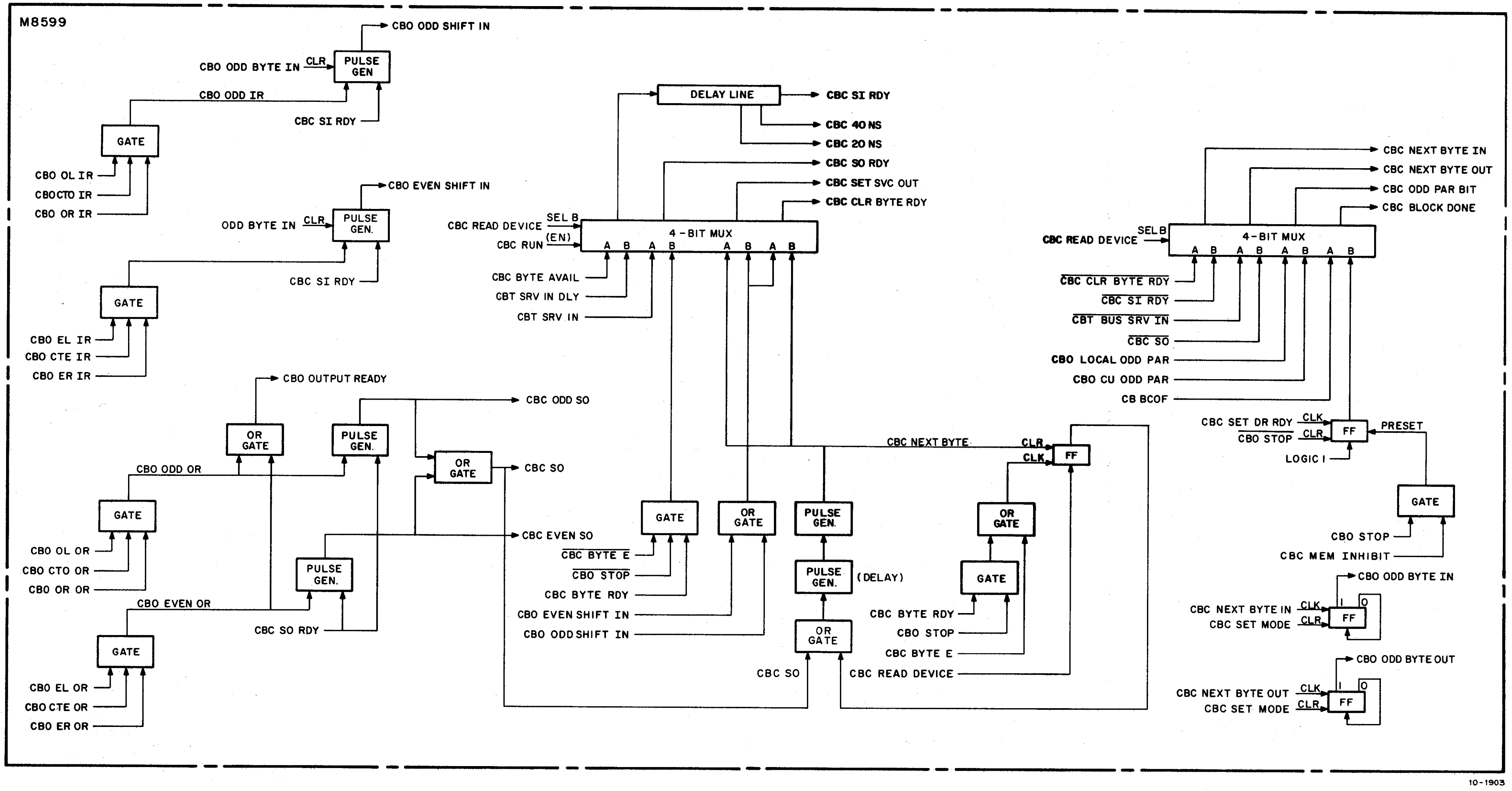


Figure 5-24 Byte Transfer Control Logic

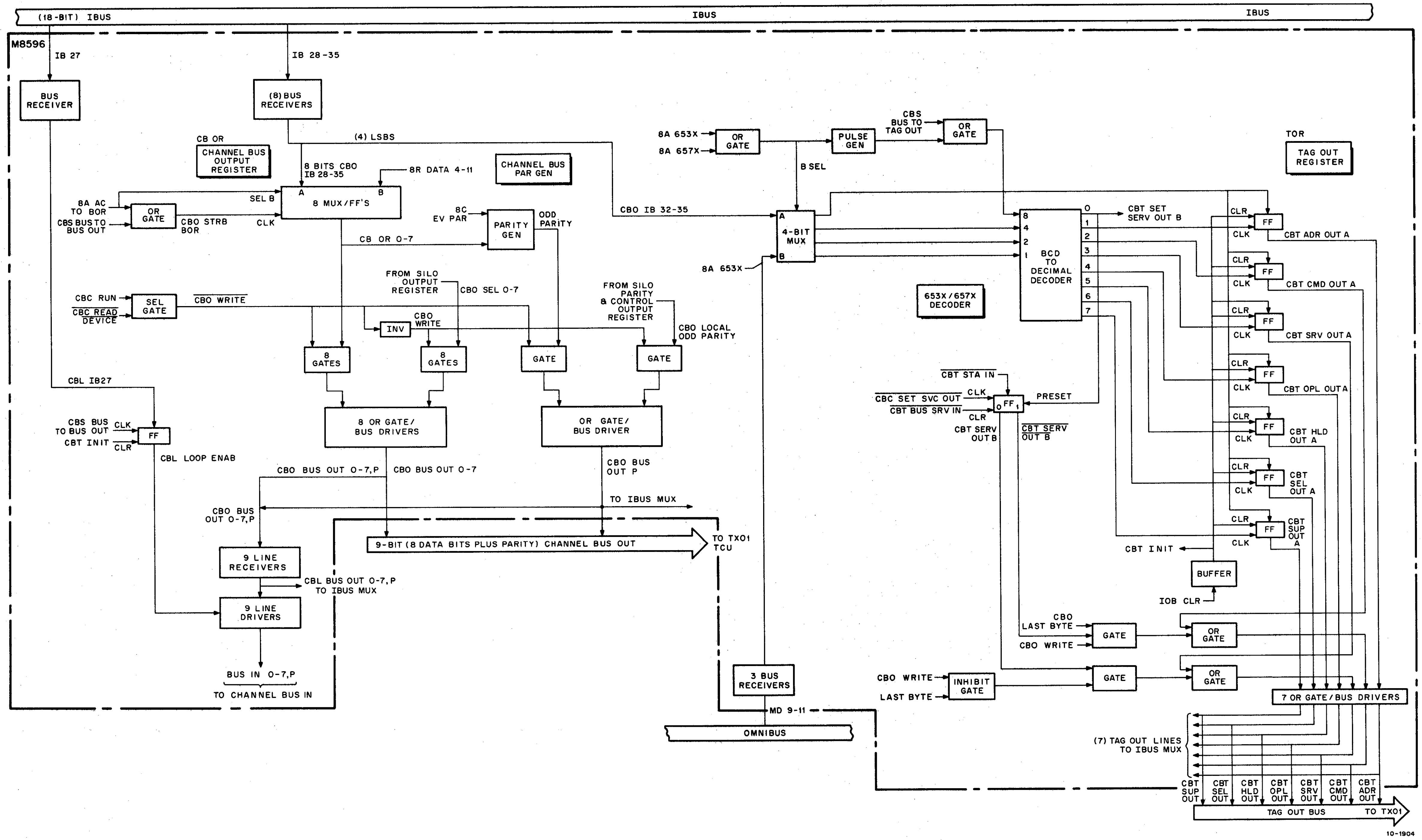


Figure 5-25 Channel Bus Output Register; Tag Out Register; Channel Bus Out; Loop Enable

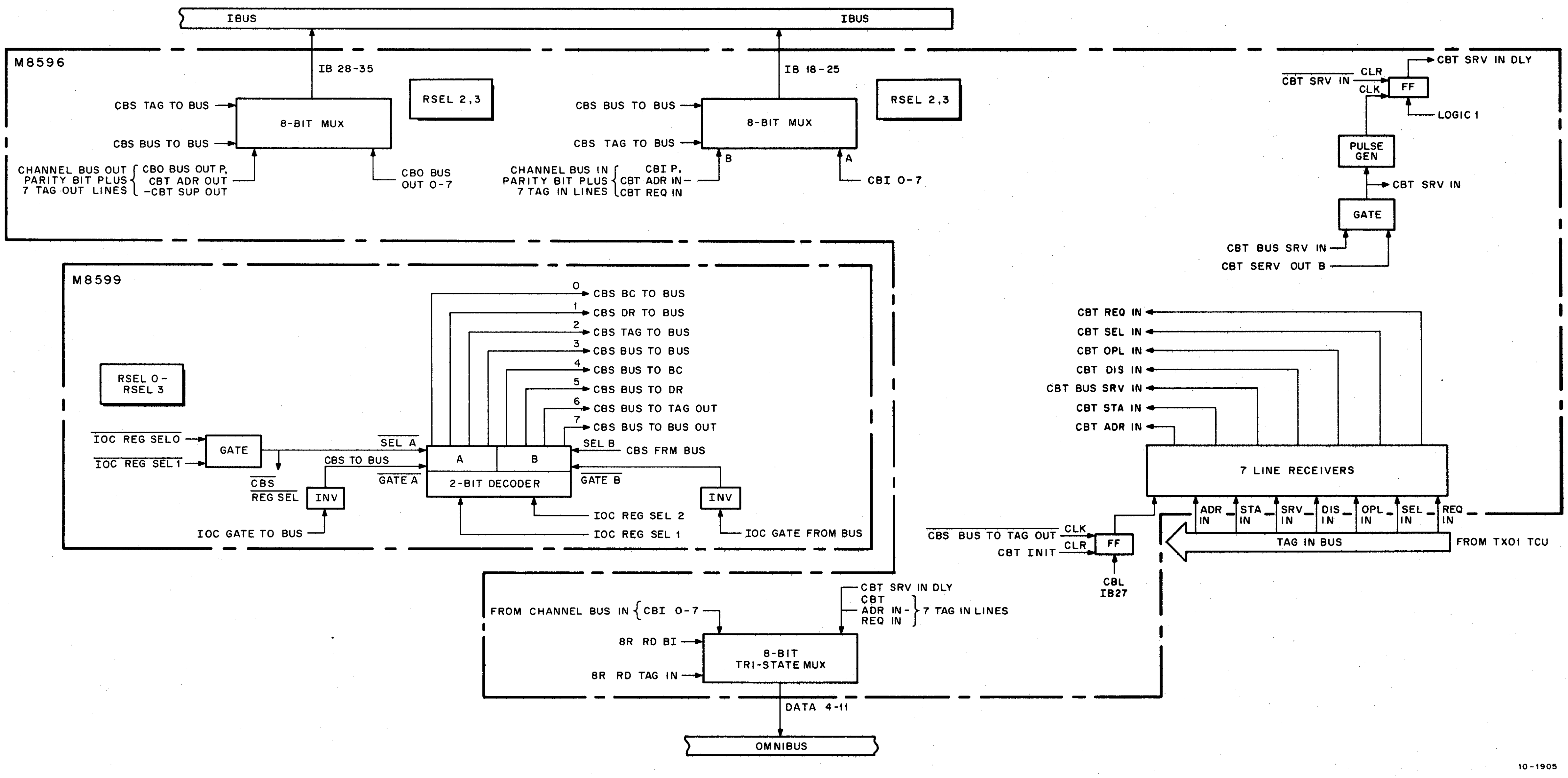


Figure 5-26 RSEL0-3; Tag Out; Tag/Channel Bus to Omnibus/IBUS Transfers

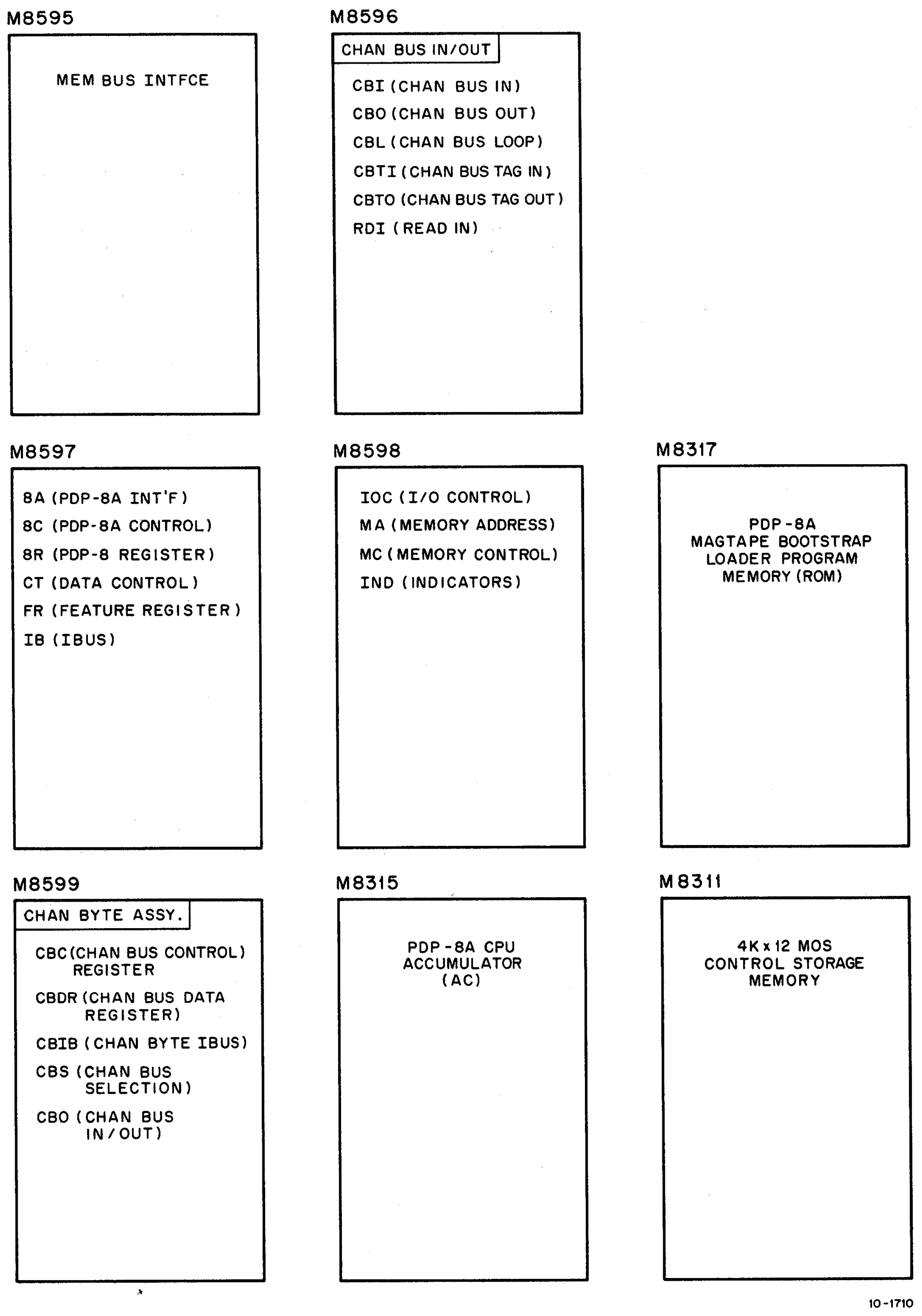


Figure 5-27 DX10 Board Functions

5.7 CHANNEL BUS FUNCTIONAL INTERFACE

5.7.1 Channel Bus Description

The Channel Bus provides the operating interface between the DX10 and the TX01 Control Unit. It consists of two 9-bit (eight data bits plus a parity bit) buses and 14 individual tag (interface control) lines. Informational bytes from the DX10 to the TX01 are sent over the 9-bit Bus Out lines. Bytes coming from the TX01 are received over the nine Bus In lines. Identification of information on either bus and control of DX10/TX01 interface connect/disconnect sequences are accomplished through the use of seven Tag In and seven Tag Out lines. The bus and control lines and their uses are shown in Table 5-5 and are discussed in detail in Paragraphs 5.7.1.1 and 5.7.1.2. Paragraph 5.7.2 presents the sequences associated with Data Transfer and channel interface Control operations. Paragraph 5.7.3 explains channel addressing format and decoding.

5.7.1.1 Information Buses, General (Table 5-6) – Each bus has nine lines (eight information lines and one parity line). On each bus, bit 7 is the low-order bit of the 8-bit byte. Bit 0 is the high-order bit. The remaining bits are in descending order from position 1 to position 6.

If a transmitted byte has less than eight information bits, the bits must occupy the highest numbered contiguous bit positions of the bus. Unused bus lines must include the low-numbered positions (position 0 and adjacent positions).

Logical 0s must be seen by the receiving end on unused lines. The parity position (P) must contain the parity bit in any byte. Odd parity is used as shown in Table 5-6.

The Bus Out line is used to transmit address, commands, control orders, and data to the TX01. The type of information transmitted over Bus Out is indicated by the outbound tag lines.

1. Bus Out specifies the address of the TU70/TU71 with which the DX10 wishes to communicate when Address Out is true during the channel-initiated selection sequence.

2. Bus Out specifies a command when Command Out is true in response to Address In during the initial selection sequence.

3. When Service Out is true in response to Service In (or Data Out is true in response to Data In) during execution of a Write or Control command, the information on Bus Out depends on the type of operation. During a Write operation, for example, it contains data to be written on tape. During a Control operation, it specifies an order code.

The Bus In line is used to transmit addresses, status, sense information, and data from the TX01 to the DX10. The TX01 can place and maintain information on Bus In only when its Operational In line is asserted, except in the case of the Control Unit Busy sequence.

The type of information transmitted over Bus In is indicated by the inbound tag lines:

1. When Address In is true, the address of the currently selected tape drive is specified by Bus Out.

2. When Status In is true, Bus In contains a byte of status information associated with either the tape drive or the control unit.

3. When Service In (or Data In) is asserted during execution of a Read or Sense command, the type of operation specifies the kind of information contained on Bus In. It may contain a byte of data from magnetic tape during a Read operation, or the detailed status of the tape drive and the conditions under which the last operation was terminated during a Sense operation.

Table 5-5
Channel Bus/Interface Control Signals

Signal Name Abbreviations	Explanation/Uses
Bus Out P But Out 0 Bus Out 1 Bus Out 2 Bus Out 3 Bus Out 4 Bus Out 5 Bus Out 6 Bus Out 7–LSB	Used to transmit information (data, I/O device address, commands, control orders) from the DX10 to the TX01.
Bus In P Bus In 0 Bus In 1 Bus In 2 Bus In 3 Bus In 4 Bus In 5 Bus In 6 Bus In 7	Used to transmit information (data, selected device identification, status information, sense data) from the TX01 to the DX10.
Adr Out Adr In Cmd Out Sta In Srv Out Srv In Dis In	Tags – Used for interlocking and controlling information on the buses and for special sequences. (Disconnect In is only used with the I/O error alert feature.)
Opl Out Opl In Hld Out Sel Out Sel In Sup Out Req In	Selection Controls – Used for TX01 selection and data/status suppression.

Table 5-6
Organization of Information Bytes on Channel Bus

Bus Line	BCD (Position Value)	Packed Numeric (Position Value)		Unpacked Numeric (Position Value)	EBCDIC (Bit Positions)	USASCII-8 (Bit Positions)	Binary (Position Value)
P	P	P		P	P	P	P
0	0	8		0	0	7	128
1	0	4	Digit*	0	1	6	64
2	B	2	X	0	2	X	32
3	A	1		0	3	5	16
4	8	8		8	4	4	8
5	4	4	Digit*	4	5	3	4
6	2	2	X + 1	2	6	2	2
7	1	1		1	7	1	1

*X = Higher order digit
X + 1 = Lower order digit
0 = Logical Zero

The tag lines control the period during which information on Bus In is valid. Information on the bus must be valid within 100 ns after assertion of the associated inbound tag and must remain valid until assertion of the responding outbound tag; in a Control Unit Busy sequence, the information must be valid until Select Out drops. The 100-ns delay between assertion of the inbound tag and the signal becoming valid on Bus In requires that the DX10 deskew Bus In. The DX10 also provides a delay in the inbound tag lines to accommodate skew caused by the DX10 internal circuitry (including its receivers). In addition to the deskewing delay, the DX10 provides a delay of at least 100 ns to compensate for skew caused by the cable and for any skew caused by the TX01 bus drivers. This delay provides sufficient time to deskew the information so that the in bound tag can be asserted by the TX01 simultaneously as information is placed on the bus.

The tag lines control the period during which information on Bus Out is valid. When transmitting the device address, information on the bus must be valid from the time Address Out becomes true until Operational In or Select In becomes true. In the case of Control Unit Busy sequence, the information must be valid from the time Address Out becomes true until Status In becomes false. When the DX10 is transmitting any other type of information, information on Bus Out is valid from the time the signal on the associated outbound tag line becomes true until the signal on the responding inbound tag line becomes false.

The DX10 must accommodate skew on Bus Out. (Except as noted under Address Out, the PDP-8/A must delay assertion of the signal on the outbound tag lines by a period sufficient to ensure that at least 100 ns elapse between the information on Bus Out and assertion of the signal on the outbound tag line. This delay is measured at the DX10 cable connectors under worst-case skew conditions; the DX10, therefore, provides a delay that accommodates skew caused by DX10 internal circuitry and, in addition, must provide a delay of at least 100 ns. This delay compensates for skew caused by the cable and, for the TX01, is sufficient to accommodate skew caused by interface receivers. The TX01 can provide additional compensating delay.

5.7.1.2 Selection Controls and Tag Lines – Each selection control and tag line is discussed in the following paragraphs.

Operational Out is a line from the DX10 to the TX01; it is used for interlocking purposes. Except for Suppress Out, all lines from the DX10 are significant only when Operational Out is asserted. Whenever Operational Out is negated, all inbound lines from the TX01 must be negated, and any operation currently in progress over the Channel Bus must be reset. Under these conditions, all TX01-generated interface signals must be negated within 1.5 μs after the negation of Operational Out from the DX10.

Request In is a line from the TX01 to the DX10 and indicates that the control unit is ready to present status information or data and is requesting a selection sequence. Request In should be dropped when Operational In rises, unless additional selection sequences are required, or when the control unit is no longer ready to present the status information or data, or when the selection requirement is satisfied by another path. Request In must never fall later than 250 ns after the fall of Operational In if the sequence satisfies the service requirements of the control unit.

Request In must not remain up when Suppress Out is up if the request for status presentation is suppressible. Under this condition, Request In must fall at the control unit within 1.5 μs after the rise of Suppress Out at the control unit.

Request In can be signaled by more than one control unit at a time.

Address Out is a tag line from the DX10 to the control unit. It provides two functions:

1. Tape Drive Selection. Address Out is used by the PDP-8/A to cause the TX01 to decode the device address on the Bus Out lines. The control unit, when recognizing the address, must respond by raising its Operational In line when its incoming Select Out rises with Address Out still up (except in the case of Control Unit Busy). The rise of Address Out follows the placement of the tape drive address on Bus Out by at least 250 ns. (Address Out must be down for at least 250 ns before it rises for device selection.) If Address Out falls before Select Out rises, the current control unit selection is cancelled.

   Address Out can rise only when Select Out (hold out), Select In, Status In, and Operational In are down at the DX10 (except as otherwise noted in 2). Ultimate use of the address on the Bus Out lines is timed by the next rise of Select Out at the TX01 Control Unit. The rise of Address Out must be delayed at least 250 ns after the address is placed on Bus Out. Once Address Out and Select Out (Hold Out) are up, Address Out must stay up until either Select In or Operational In rises, or until Status In falls, in the case of the Control Unit Busy sequence. During tape drive selection, Address Out can be up concurrently with any other outbound tag line.

2. Disconnect Operation. If Hold Out is down and Address Out rises or Address Out is up and Hold Out falls, the control unit must drop its Operational In, thus disconnecting from the interface. Address Out remains up until Operational In drops. Operational In must drop within 6 μs after receiving the disconnect indication. Mechanical motion in process continues to a normal stopping point. Status information is generated and presented to the DX10 when appropriate. Address Out, in this case, may be up concurrently with another outbound tag line.

Control unit selection is controlled by Select Out, Select In, and Hold Out. Select Out and Select In form a loop from the DX10 through the control unit to a cable terminator block (Select Out) and back again through the control unit to the DX10 (Select In). The control unit selection circuitry may be attached to either Select Out or Select In.

NOTE

In the 1 × 8 configuration, the control unit address (bits 0–3 of the 8-bit address byte) is 0_{16}.

Throughout the following description, the rise of Select Out at the control unit presumes that Hold Out is up, and the fall of Select Out is the result of the fall of Hold Out.

1. Select Out. Select Out is a line from the DX10 to the control unit that has highest priority. This line, together with Select In, provides a loop for scanning the control unit. The control unit can raise its Operational In only at the rise of its incoming Select Out. Once the control unit propagates Select Out, it cannot raise Operational In or respond with a Control Unit Busy sequence until the next rise of Select Out. When an operation is initiated by the DX10, Select Out is raised no sooner than 400 ns after the rise of Address Out, which indicates that the address of the tape unit being selected has been placed on the Bus Out lines.

   The microprocessor must keep Select Out up until either Select In or Address In and Operational In or Status In rise.

   When Select In rises, the microprocessor must drop Select Out and may not raise it again until after Select In falls.

   The control unit is selected only when it raises Operational In. Select Out must then drop so that Operational In may drop. However, after Select Out falls, the control unit must keep Operational In up until the current signal sequence is completed. A rise of the incoming Select Out at the control unit signals that the control unit can become selected by raising Operational In. If the control unit raises Operational In, it must suppress the propagation of Select Out. If the control unit does not require selection, it must propagate Select Out within 1.8 μs.

   When Status In rises in response to Select Out during a selection sequence (indicating Control Unit Busy), Select Out must be dropped and must not be raised until Address Out has been dropped.

2. Hold Out. Hold Out is a line from the DX10 to the control unit and is used in conjunction with Select Out to provide synchronization of control unit selection.

   Hold Out is also used to minimize the propagation of the fall of Select Out by purging the Select Out signal from the Select Out signal path.

   Therefore, once Hold Out drops, it must not rise for at least 4 μs. The minimum downtime of this signal may be optionally adjusted at installation time to a minimum of 2 μs to handle high-speed channel configurations. In all cases, the DX10 must be capable of providing the 4-μs timing for general system configurations. To allow time for the falling of the interface signals of a sequence, Hold Out must not rise sooner than 1.5 μs after the fall of Operational In.

3. Select In. Select In is a line that extends the select signal from the jumper in the terminator block to the DX10. It provides a return path to the DX10 for the Select Out signal. It is blocked by the control unit from rising when the control unit decodes and recognizes its address (0_{16}). If Select In rises at the DX10 during a selection sequence, it indicates to the DX10 that the control unit failed to select.

Operational In is a line from the control unit to the DX10 and is used to signal that a tape unit has been selected (except during a Control Unit Busy sequence). It must stay up for the duration of the selection. The selected unit is identified by the address byte transmitted over Bus In.

The rise of Operational In indicates that the control unit and a tape unit are both selected and communicating with the DX10. Operational In can rise only when the incoming Select Out to the control unit is up and the outgoing Select In is down, i.e., the control unit may raise Operational In, except for the Control Unit Busy sequence, only in response to the rise of Select Out, and must block Select In from emanating to the DX10. Operational In can drop only after Select Out drops.

When Operational In is raised for a particular signal sequence, it must stay up until all required information is transmitted between the DX10 and the control unit. If Select Out is down, Operational In must drop after the rise of the outbound tag associated with the transfer of the last byte of information. With the exception of Request In, all inbound signals from the DX10 must be down within 1.5 μs of the fall of Operational In at the control unit.

Address In is a tag line from the control unit to the DX10 which tells the DX10 that the address of the currently selected tape unit has been placed on Bus In. The DX10 responds to Address In with Command Out.

The rise of Address In indicates that the address of the currently selected tape unit is available on the Bus In lines. Address In must stay up until the rise of Command Out. Address In must then fall so that Command Out may fall. Address In cannot be up at the same time that any other inbound tag line is active.

Command Out is a tag line that is used by the DX10 to start the selected tape unit following Address In, Status In, or Service In. Sending Command Out in response to Address In during the initial selection sequence indicates to the control unit that the DX10 has placed a command byte on the Bus Out lines. (The command byte has a fixed format.) The rise of Command Out indicates to the control unit that information on Bus In is no longer required to be valid. Command Out must stay up until the fall of the associated Address In, Status In, or Service In. It cannot be up concurrently with any other outbound tag line, except possibly during an interface disconnect sequence, in which case Address Out may be up.

Sending Command Out in response to Address In during a control unit-initiated sequence means to proceed. In the case of a DX10 initial selection sequence, Command Out indicates that Bus Out defines the operational command to be performed. Command Out in response to Service In always means stop. Command Out in response to Status In means stack.

When Command Out is raised to indicate proceed, stack, or stop, Bus Out must have a byte of all 0s, but need not have correct parity. Bus Out is not checked for parity or decoded by the control unit under these circumstances.

Status In is a tag line from the control unit to the DX10 notifying it of when the control unit has placed status information on Bus In. The status byte has a fixed format and contains bits describing the current status at the control unit. The DX10 responds with either Service Out or Command Out, depending on whether or not it accepted the status byte.

The rise of Status In indicates that a byte of status information is available on Bus In. Status In cannot be up concurrently with any other inbound tag line. Status In must stay up until the rise of an out-tag, or (in the Control Unit Busy sequence) until Select Out falls. It must then fall so that the responding out-tag may fall. During the Control Unit Busy sequence, status information on Bus In must be valid until Select Out (Hold Out) falls.

Service Out is a tag line from the DX10 to the control unit in response to Service In or Status In. Service Out indicates to the control unit that the DX10 has accepted the status information on the Bus In lines or has provided the data requested by Service In on the Bus Out lines.

When Service Out is sent during a Read or Sense operation in response to Service In or to Status In, the Service Out signal must rise after the DX10 accepts the information on Bus In. In these cases, the rise of Service Out indicates that the information is no longer required to be valid (on Bus In) or is not associated with any information on Bus Out.

When Service Out is sent in response to Service In during a Write or Control operation, it indicates that the DX10 has provided the requested information on Bus Out. In this case, the signal must rise after the information is placed on the bus. Service Out must stay up until the fall of the associated Service In or Status In. Service Out cannot be up concurrently with any other out-tag except during an interface-disconnect sequence (when Address Out may be up).

A Service Out response to Status In while Suppress Out is up indicates to the control unit that the operation is being chained and that this status is accepted by the DX10.

Service In is a tag line from the control unit to the DX10 and is used to inform the DX10 when the selected tape drive wants to transmit or receive a byte of information. The nature of the information associated with Service In depends on the operation. The DX10 must respond to Service In with Service Out, Command Out, or, during an interface disconnect, with Address Out.

During Read, Read-Backward, and Sense operations, Service In rises when information is available on the Bus In lines. During Write and Control operations, Service In rises when additional information is required on the Bus Out lines. Service In cannot be up concurrently with any other inbound tag line. Service In must stay up until the rise of either Service Out, Command Out, or Address Out. If the DX10 does not respond in time to Service In, an overrun condition occurs. In such a case, Service In'must not drop if an out-tag has not risen, and it must not rise if Service Out has not dropped.

An overrun condition causes both the Unit-Check status indicator and the Overrun Unit sense indicator to be set. Data transfer stops after an overrun condition.

Suppress Out is a line from the DX10 to the control unit used both alone and in conjunction with the out-tag lines to provide the following special functions: suppress data, suppress status, command chaining, and selective reset.

The following summarizes the selection controls and tag lines.

1. Except for Address Out, no more than one out-tag may be up at any given time during the interface-disconnect sequence.

2. No more than one in-tag may be up at any given time.

3. An in-tag will rise only when all out-tags are down, except for the Control Unit Busy sequence.

4. An in-tag will fall only after the rise of a responding out-tag, except for Status In in the Control Unit Busy sequence.

5. Service Out and Command Out may rise only in response to the up level of an in-tag.

6. Address Out for a channel-initiated selection sequence may rise only when Select In and Select Out are down at the channel.

7. Once Address Out and Select Out have risen for a channel-initiated selection sequence, Address Out must stay up until after the rise of Select In or Operational In or the fall of Status In.

8. Once Address Out has risen for the interface-disconnect control sequence, it must not drop until Operational In drops.

9. None of the out lines, except Suppress Out, have meaning when Operational Out is down.

10. Select Out can rise only if Operational In and Select In are down.

11. Operational In cannot fall until either:

    a. Select Out falls and an out-tag response is sent for the last in-tag of any given signal sequence.

    b. Operational Out falls.

c. An interface-disconnect sequence is given.

12. Operational In cannot rise unless Operational Out is up and must drop if Operational Out'drops.

5.7.2 Channel Bus Operation

Operation of the Channel Bus is under PDP-8/A Microprocessor control. For any Data Transfer operation, three distinct signal sequences are necessary to initiate, utilize, and terminate the Channel Bus. These sequences include initial (control and tape drive) selection, data transfer, and ending (channel termination) procedures. In addition to DX10 Data Channel-initiated sequences, two sequences – Control Unit Busy and Control Unit-Initiated Service Request – are used to activate the Channel Bus. These sequences are shown in-Figure 5-28 and are executed by the DX10 (under control of the microprocessor) as follows.

5.7.2.1 Initial-Selection Sequence – To initiate an operation, the microprocessor places the address byte on the Bus Out lines and raises Address Out. The control unit decodes the address on the bus. To be acceptable, the address must have correct parity.

The microprocessor then issues Select Out; the control unit blocks its propagation and raises the Operational In line. When Operational In rises, the microprocessor responds by dropping Address Out. After Address Out falls and the tape unit address is on Bus In, Address In may rise. For a byte multiplex operation, Hold Out with Select Out may drop any time after this point. After the microprocessor checks the address, it responds by placing the device command on Bus Out and raising the Command Out line. The control unit processes the command and drops Address In, which allows Command Out to fall. After Command Out drops, the control unit places status information on Bus In and raises Status In. (Note that if selection had been for a Start I/O instruction, sufficient information is available at this point to complete instruction execution.)

If the microprocessor accepts this status condition, it responds with Service Out. Service Out allows Status In to fall, completing the initial selection sequence. A Command Out response from the microprocessor also allows Status In to fall.

NOTE

A response of Command Out to Status In cannot prevent the execution of an immediate command.

An immediate-type command or command-immediate is a command whose execution meets the following conditions:

1. Execution requires no more information than that in the command byte; that is, no data or information bytes are transferred.

2. Channel-end time coincides with initial status time, and, on a normal operation, at least channel end instead of zero status will be in the initial status byte.

   If the device is operating during a channel-initiated selection sequence, the control unit presents Busy Status. When the control unit has, for the addressed device, status information outstanding from a previous operation or an externally initiated status condition, it presents the Busy Status (except to the all-zero command), along with the other status conditions in the status byte.

If the command is rejected by the control unit, for example, as a result of the detection of an invalid command, the control unit presents the Unit-Check status condition. No operation is initiated in the control unit and no ending status is generated.

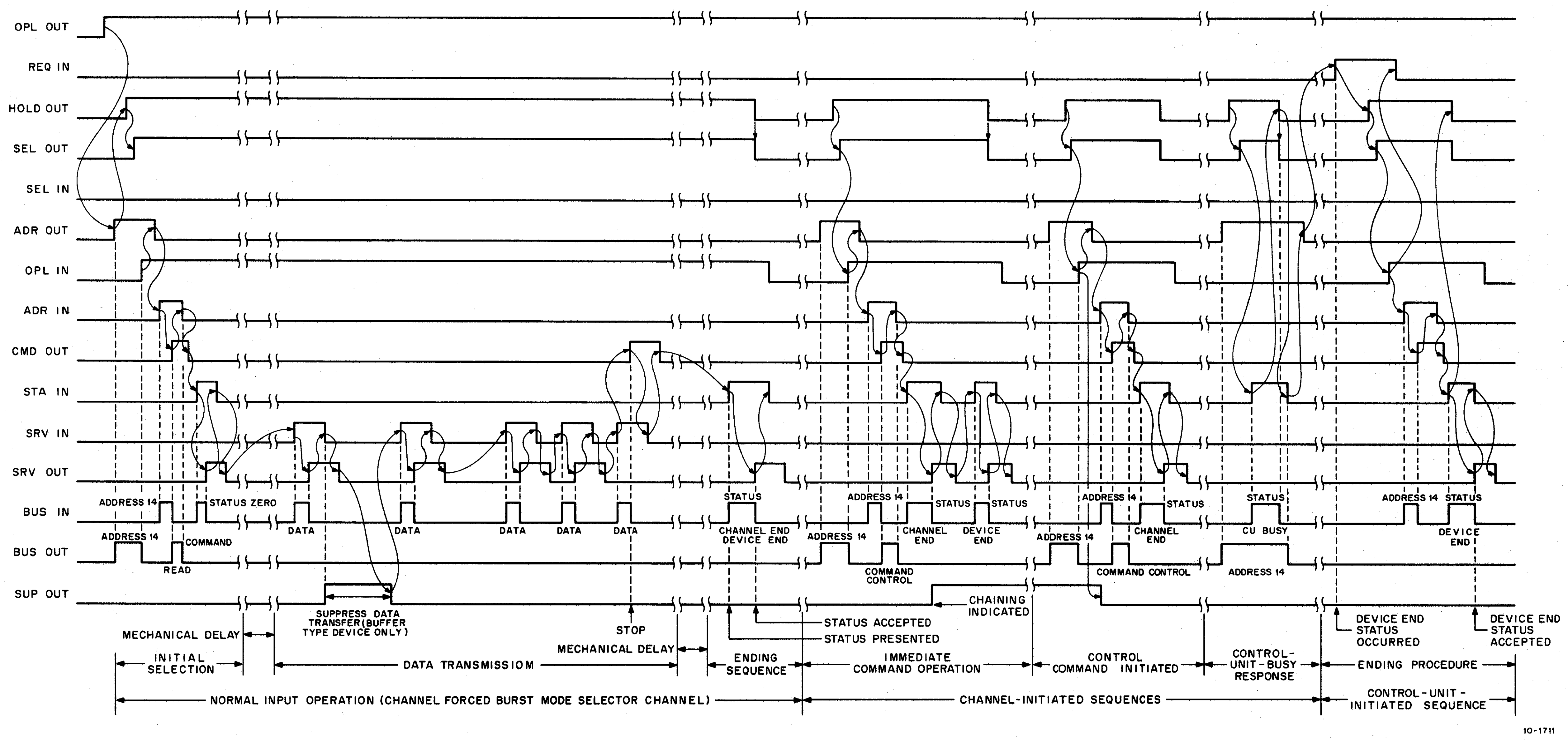


Figure 5-28 Selector Channel Timing

5.7.2.2 Control Unit Busy Sequence – If a device is addressed and the control unit is busy or has status pending for a device other than the one addressed, the control unit responds with a status byte indicating the busy condition. The control unit can present this status byte in either of two ways, depending on the control unit configuration; it can present status information (as in the initial selection sequence) or it can respond with the (shorter) Control Unit Busy sequence.

NOTE

The Control Unit Busy sequence is not used in response to an initial-selection sequence addressed to a device for which chaining has just been indicated.

The Control Unit Busy sequence begins when the microprocessor places the device address on the Bus Out lines and raises Address Out. Select Out is then raised. The control unit decodes the address on the Bus Out lines. When Select Out rises, the control unit blocks the propagation of Select In, places the Busy Status byte on Bus In, and raises Status In. Operational In is not raised.

After accepting the status byte, the microprocessor drops Select Out. The control unit responds by dropping Status In and disconnects from the interface. The microprocessor must keep Address Out up until Status In drops, thus completing the Control Unit Busy sequence.

5.7.2.3 Control Unit-Initiated Sequence (Service Request) – When the control unit requires service, it raises Request In to the DX10. The next time Select Out rises at the control unit, and no selection is being attempted by the DX10 (Address Out down), the control unit places the address of the device on Bus In, and signals both Address In and Operational In. When the microprocessor recognizes the address, Command Out is sent to the control unit, indicating proceed. After Address In drops, the microprocessor responds by dropping Command Out.

If the service request is for data, the sequence proceeds as described in Paragraph 5.7.2.4. The control unit-initiated selection for data transfer occurs in Byte Multiplex mode.

If the service request is for status information, the sequence proceeds as defined for the status cycle in the Paragraph 5.7.2.5.

5.7.2.4 Data Transfer Sequence – Data transfer may be requested by the control unit after a selection sequence. To transmit data to the DX10, the control unit places a data byte on Bus In and raises Service In; the tag and the validity of Bus In must be maintained until an outbound tag is raised in response.

To request data from the DX10, Service In is raised; the microprocessor places the data on Bus Out and signals with Service Out. The DX10 maintains the validity of Bus Out until Service In falls. After Service In falls, the microprocessor responds by dropping Service Out.

After selection, the control unit remains connected to the DX10 for the duration of the transfer of information. The information can be a single byte of data, a status report, an initiation of a new command, a string of data bytes, or a complete operation from initiation to reception of the final status report.

The duration of the connection is under control of both the DX10 and the control unit. To enable the DX10 to control the duration of the connection, the control unit cannot disconnect from the interface before Select Out (Hold Out) falls. However, the control unit may preserve the logical connection after the DX10 permits the control unit to disconnect (Select Out hold out down) by holding up Operational In. In this manner, the control unit can force Burst mode.

Depending on the duration of the connection, one of two modes of operation is established: Byte Multiplex or Burst. (These modes are established so that the program can schedule concurrent execution of multiple I/O operations.)

If Operational In remains up for longer than the byte multiplexing time-out limit, selection is in Burst mode. If the selection time is less than this time-out limit, the selection is in Byte Multiplex mode.

The Byte Multiplex mode is the normal mode for low-speed I/O devices; however, all I/O devices are designed to operate in Burst mode when required by the channel. Channels that are not capable of operation in Byte Multiplex mode force Burst mode by holding up Select Out (Hold Out) until presentation of channel-end status conditions. The transfer of one or more data bytes during a single interface sequence, where the time contributed by the control unit is less than 32 μs, is considered Byte Multiplex mode.

Burst mode is the normal mode of operation for high-speed (TU70-type) I/O devices. These devices force Burst mode (by holding up Operational In) when attached to a channel capable of byte multiplex operation. Medium-speed or buffered I/O devices, which may normally operate in either mode as determined by channel data rate capabilities, are equipped with a manual or programmable switch to select the mode of operation. The switch setting is overridden when Burst mode is forced by the channel. An interface disconnect executed by the channel overrides the Force Burst Mode condition of a control unit.

Some channels can tolerate an absence of data transfer during a Burst Mode operation, such as when reading a long gap on tape, for not more than approximately one-half minute. Equipment malfunction may be indicated when an absence of data transfer exceeds this time.

5.7.2.5 Ending Procedure and Asynchronous Status – The ending procedure may be initiated by either the control unit or the DX10. If the procedure is initiated by the control unit, the end of operation is completed in one signal sequence, assuming that both channel-end and device-end status conditions occur together. If the procedure is initiated by the DX10, the control unit may still require time to reach the point where the proper status information is available, in which case a second signal sequence is necessary to complete the ending procedure.

One of three situations may exist at the initiation of the ending procedure (assume selection is already obtained):

1. The DX10 recognizes the end of an operation before the tape unit reaches its ending point. In this situation, whenever the control unit requires service again, the control unit raises the Service In line. The microprocessor responds with Command Out, indicating stop. The control unit drops Service In and proceeds to its normal ending point without requesting further service. When the tape unit reaches the point at which it would normally send Channel End, the control unit places the ending status on Bus In and raises Status In. The microprocessor responds with Service Out, unless it is necessary to stack the status condition, in which case the microprocessor responds with Command Out.

2. The DX10 and the tape unit recognize the end of an operation simultaneously.

3. The tape unit recognizes the end of an operation before the DX10 reaches the end of an operation.

For 2 and 3, status information is available at the control unit. The control unit places the ending status on Bus In and raises Status In.

If Device End does not occur with Channel End, Device End is presented when it is available, and an additional status sequence is required.

Some status conditions are unrelated to any previous program-initiated command. One of these conditions is Device End, which is generated when the corresponding device goes from the Not Ready to the Ready state. These status conditions are handled in the same way as any other status information, and are subject to the same rules as far as presentation to the DX10 and stacking are concerned.

5.7.3 Channel Addressing

Addressing of a specific tape drive for a Data Transfer operation is accomplished by sending an 8-bit drive address byte over the Bus Out lines. At the time of installation, each tape drive is assigned a 4-bit (hexadecimal) device address, ranging from 0_{16} to 7_{16} for a 1 by 8 system configuration, or 0_{16} to F_{16} for a 2 by 16 configuration. The drive address is the low-order hexadecimal character and the control unit (which is always assigned address 0_{16} is the high-order character. Thus tape drive addresses in a 1 by 8 system configuration will range from 00_{16} to 07_{16}, and in a 2 by 16 configuration will be addressed from 00_{16} to $1F_{16}$.

NOTE

The latter configuration is available as an option on request and is not presently provided for in the DX10 microcode or in the Channel Program software.

5.7.3.1 Addressing Decoding – The TX01 Control Unit recognizes a device address that meets the following conditions:

1. The address has correct parity.
2. The address is assigned to the control unit.

The control unit does not respond to any address outside its assigned set. For example, the control unit is designed to control devices that have only bits 0000–0111 in the low-order position of the device address. It does not recognize addresses that have 1000–1111 in these bit positions. If the control unit does not respond to an address (Select Out is propagated through the control unit, and back to the channel as Select In), the device appears to be nonoperational. Nonoperational may also include (in addition to addresses outside an assigned set):

1. A device address not installed.
2. A device address partitioned out of the system by the program, operator, or service engineer (offline, disabled, etc.).

The control unit does respond to those addresses in the set for which the corresponding devices are either:

1. Ready
2. Not ready, but that can be made ready by means of an ordinary manual intervention. (A not-ready device is indicated by the Unit-Check status and Intervention-Required sense indicators.)

If the control unit receives an address for which no device is installed, the Unit-Check status indicator will be turned on [as well as the appropriate sense indicator(s)].

CHAPTER 6
MAINTENANCE

6.1 INTRODUCTION

The DX10 Data Channel has been designed as a high-reliability interface between a DECsystem-10 Computer System and the TU70 Magnetic Tape Subsystem. Accordingly, all operating, functional, and interface circuits in the DX10 are modular (board-mounted) to permit rapid replacement in the event of malfunction. With the exception of the cabinet-mounted maintenance panel, the three operating power supplies, and the cabinet cooling blowers, all modules can be removed and replaced without the aid of any special tools.

6.2 FUNCTIONAL TESTING

DX10/TU70 Subsystem fault detection is performed automatically using diagnostic programs. These programs have been designed to test three main operational areas as follows:

1. DX10 Programmable Data Channel. This diagnostic consists of two parts and is used to completely test the following functional areas of the DX10:

   I/O Bus interface
   IBUS interface
   All IBUS registers, counters, etc.
   PDP-8/A interface
   PDP-8/A CPU and Control Storage Memory
   PDP-8/A IOT instructions
   DECsystem-10 Memory interface
   DX10 data transfer logic and SILO (data channel)

   This diagnostic program runs independently of the TU70 Subsystem.

2. DX10/TU70 Subsystem. This diagnostic program tests Channel Bus connections, the TX01/DX10 interface, and the microprocessor firmware (microcode, IOT decoders, etc.). All microcode and Channel Program instructions are tested for proper operation during the running of this diagnostic.

3. DECsystem-10/TU70 Subsystem. This diagnostic program exercises the TU70 Subsystem under maximum dynamic loading conditions to ensure reliable system operation with the DECsystem-10 on-line.

Use of each of these diagnostic programs is described, together with the associated setup and module replacement procedures, in the individual writeups associated with each diagnostic program. The identifying nomenclature for each of these three programs is as follows:

1. PDP-10 DX10 Programmable Data Channel Diagnostic

   Part 1: MAINDEC-10-DDDXA
   Part 2: MAINDEC-10-DDDXB

2. PDP-10 TU70 Magnetic Tape Subsystem Diagnostic (MAINDEC-10-DDTUA)

3. PDP-10 TU70 Magnetic Tape Subsystem Reliability Diagnostic (MAINDEC-10-DDTUB)

When diagnostic testing of the DX10 Data Channel is necessary, mount the DX10 Diagnostic Distribution tape on a tape drive and press the READ IN switch on the DECsystem-10 operator's console. To run the diagnostic test, follow the instructions printed out on the console teletypewriter.

6.3 INDICATORS AND SWITCHES

The DX10 maintenance panel is used to select various registers, counters, etc. for transfer onto the IBUS and then to the panel indicators for verification purposes when the DX10 is not operating (the microprocessor is not running and the DECsystem-10 CPU is not attempting to read or write into any of the DX10 registers, counters, microprocessor control storage memory, etc.). First, select the register (counter, etc.) to be observed on the ADDRESS switches. (Read the chart affixed to the maintenance panel for addresses of various registers and counters.) When the proper address has been selected, press the EXAMINE pushbutton on the maintenance panel; the contents of the selected register, etc. will be displayed on the DATA (18:35) LED indicators. Selecting register address 17 will display all 0s, unless the IBUS is being activated because of a ground, a bus driver malfunction, etc. Selecting an address between 40 and 77 tests all of the LED drivers by displaying all 8s automatically, without having to press the EXAMINE pushbutton.

6.4 ON-LINE TESTS

The only on-line tests prescribed for isolating DX10 malfunctions are the DXA and DXB and TUA and TUB Diagnostic Programs. These are to be run in a prescribed sequence as indicated by the instructions contained in the related documents.

6.5 OFF-LINE TESTS

No off-line testing has been prescribed for the DX10 Data Channel.

6.6 BACKPLANE SIGNAL TEST POINTS

Table 6-1 lists the DX10 backplane signal test points.

Table 6-1
DX10 Backplane Signal Test Points

Signal	Module M85XX	Backplane Test Point
IB 18 L	95	10 AE2
IB 19		AD2
IB 20		AL2
IB 21		AK2
IB 22		AS2
IB 23		AR2
IB 24		AV2
IB 25		BF1
IB 26		BK2
IB 27		MB2
IB 28		AH2
IB 29		AF2
IB 30		AP1
IB 31		AM2
IB 32		AV2
IB 33		AT2
IB 34		BE2
IB 35 L	95	10 BD2
IOB DATAO H	98	12 EV2
IOB DATAI H		EV1
IOB CONO CLR H		ET2
IOB CONI H		FU2
IOB SELECT		ES2
RSEL 0H		ED2
RSEL 0L		AD1
RSEL 1H		ED1
RSEL 1L		AC1
RSEL 2H		EB1
RSEL 2L		AB2
RSEL 3H		EB2
RSEL 3L	98	12 AA1
CROBAR H*	97	8 FD2
IOC GATE FROM BUS L	98	12 AM1
IOC GATE FROM BUS H	98	12 AN1
IOC GATE TO BUS L	96	13 EN2
CBO OUTPUT READY (SILO) H	99	9 DA1
CBC BLOCK DONE H	99	9 EN2
SRV IN H	96	13 CK2
SRV OUT H	96	13 CF2
PULSE LA H	97	8 DR2
MEM START L	97	8 AJ2
STOP L	97	8 DS2
RUN L	97	8 BV2
CBC DATA REQ (1) H	98	12 EM1
CT 8A MEM REQ (1) H	98	12 FD2
MC MEM CYCLE H	98	12 DS2
ADR ACK L*	95	10 ED1
ACKN L*		10,11 DV1
RD RQ L*		11 FM1
WR RQ L*		11 FK1
WR RS L*		11 FJ1
RD RS L*		10 ED1
REQN L*	95	10 FL1
RQ CYC FAST L		10 FM1

*Negative Logic Signal (0 V to −3 V)

Reader's Comments

DX10 DATA CHANNEL
MAINTENANCE MANUAL
EK-DX10-MM-001

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

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

What features are most useful? ______________________________

What faults do you find with the manual? ______________________________

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

Does it satisfy *your* needs? ______________ Why? ______________________________

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

Please describe your position. ______________________________

Name ______________________ Organization ______________________

Street ______________________ Department ______________________

City ______________ State ______________ Zip or Country ______________

CUT OUT ON DOTTED LINE

— — — — — — — — — — — 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

digital