HEWLETT PACKARD

MAINTENANCE MANUAL

30051A

UNIVERSAL INTERFACE (DIFFERENTIAL)

(FOR HP 3000 COMPUTER SYSTEMS)

Manual Part No. 30051-90001
Microfiche Part No. 30051-90005

Printed in U.S.A. 10/72
Update 5/76

Printed-Circuit Assembly:

30051-60001

Note

This manual should be retained with the documentation covering the subsystem in which this interface printed-circuit assembly is used.

HEWLETT-PACKARD
5303 STEVENS CREEK BLVD., SANTA CLARA, CALIFORNIA 95050

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the most recent date on which the technical material on any given page was altered. If a page is simply re-arranged due to a technical change on a previous page, it is not listed as a changed page. Within the manual, changes are marked with a vertical bar in the margin.

Pages	Effective Date
Title	May 1976
ii thru v	May 1976
vi thru ix	Oct 1972
Blank	Oct 1972
1-1 thru 1-2	Oct 1972
2-1 thru 2-7	Oct 1972
Blank	Oct 1972
3-1 thru 3-15	Oct 1972
Blank	Oct 1972
3-17	Oct 1972
Blank	Oct 1972
3-19 thru 3-21	Oct 1972
Blank	Oct 1972
3-23 thru 3-27	Oct 1972
Blank	Oct 1972
3-29 thru 3-34	Oct 1972
3-35 thru 3-36	Mar 1975
3-37 thru 3-41	Oct 1972
Blank	Oct 1972
3-43	Oct 1972
Blank	Oct 1972
3-45	Oct 1972
Blank	Oct 1972
4-1	Mar 1975
4-2 thru 4-3	Oct 1972
4-4	Mar 1975
4-5	Oct 1972
Blank	Oct 1972

PRINTING HISTORY

30051A

New editions incorporate all update material since the previous edition. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The date on the title page and back cover changes only when a new edition is published. If minor corrections and updates are incorporated, the manual is reprinted but neither the date on the title page and back cover nor the edition change.

First edition.....10/72
Second edition.....5/76

Note: This update is for the purpose of referencing the HP 3000 Series II documentation.

This manual contains maintenance information for the HP 30051A Universal Interface (Differential). The interface described in this manual is used as an integral part of the HP 3000 Computer System to provide I/O capability between the system and parallel data transfer devices requiring differential-logic signals.

The contents of this manual are organized in four sections as follows:

a. Section I contains general information relative to the interface physical features and specifications.

b. Section II contains operating parameters for the interface including a description of control word and status word formats.

c. Section III contains theory of operation for the interface.

d. Section IV contains servicing instructions with preventive and corrective maintenance information.

This manual should be retained and used with related documentation for the HP 3000 Series II Computer System or the pre-Series II HP 3000 Computer System. The related documentation for the HP 3000 Series II Computer System includes the following:

a. HP 3000 Series II Computer System Service Manual, part no. 30000-90018.

b. HP 3000 Series II Computer System Signal and Power Distribution Manual, part no. 30000-90021.

c. HP 3000 Series II Computer System Stand-Alone Universal Interface PCA, Card Reader/Punch PCA Diagnostic, part no. 30050-90012.

d. HP 3000 Computer Systems Support Log, part no. 03000-90117.

The related documentation for the pre-Series II HP 3000 Computer System includes the following:

a. HP 30001A CPU/IOP Maintenance Manual, part no. 30001-90003.

b. HP 30005A/30006A Memory Maintenance Manual, part no. 30005-90001.

c. HP 3000 Computer System Manuals of Diagnostics (MOD).

d. HP 3000 Computer System Detailed Diagrams Manual, part no. 03000-90023 (Diagram Set DD-501, part no. 30051-90004).

e. HP 3000 Computer System Illustrated Parts Breakdown (IPB) Manual, part no. 03000-90021.

f. HP 3000 Computer System PCA Cage Maintenance Manual, part no. 30002-90002.

g. HP 3000 Computer System Installation Manual, part no. 03000-90032.

h. System Configuration Package.

Note: Do not use the equipment described in this manual to interface serial data transfer or parallel data transfer devices requiring DTL/TTL compatible signals.

CONTENTS

Section			Page
I	GENERAL INFORMATION		
	1-1.	Introduction	1-1
	1-3.	General Description	1-1
	1-8.	Specifications	1-1
	1-10.	Identification	1-1
II	OPERATING PARAMETERS		
	2-1.	Introduction	2-1
	2-3.	General Information	2-1
	2-7.	System Parameters	2-1
	2-9.	CIO Instruction	2-1
	2-10.	RIO Instruction	2-1
	2-11.	SIN Instruction	2-2
	2-12.	SIO Instruction	2-2
	2-13.	SMSK Instruction	2-2
	2-14.	TIO Instruction	2-2
	2-15.	WIO Instruction	2-2
	2-16.	Direct I/O	2-2
	2-18.	Programmed I/O	2-2
	2-20.	Control Order	2-2
	2-21.	End Order	2-2
	2-22.	Interrupt Order	2-3
	2-23.	Jump Order	2-3
	2-24.	Read Order	2-3
	2-25.	Return Residue Order	2-3
	2-26.	Sense Order	2-3
	2-27.	Write Order	2-3
	2-28.	Control Word Format	2-3
	2-30.	Master Clear	2-3
	2-31.	Clear All Interrupts	2-3
	2-32.	Selective Interrupt Clear	2-3
	2-33.	Initiate Data Transfer	2-3
	2-34.	Device Control	2-3
	2-35.	Data Transfer Interrupt	2-4
	2-36.	Interrupt/Device Status	2-4
	2-37.	Byte Transfer	2-4
	2-38.	Enable Interrupt Request	2-4
	2-39.	Start Transfer Timer	2-4
	2-40.	Status Word Formats	2-4
	2-43.	Interface Status Byte	2-5
	2-45.	SIO OK	2-5
	2-46.	RIO, WIO OK	2-5
	2-47.	Interrupt Pending	2-5
	2-48.	Sequence Counter	2-5
	2-49.	Device Flag	2-5
	2-50.	Interrupt/Device Status	2-5
	2-51.	Permanent Zero	2-5
	2-52.	Interrupt Status Byte	2-6
	2-54.	Device Status Interrupts	2-6
	2-55.	Data Transfer Interrupt	2-7
	2-56.	Clear Interface Interrupt	2-7
	2-57.	I/O System Interrupt	2-7
	2-58.	Transfer Error Interrupt	2-7
	2-59.	Time Out Interrupt	2-7
	2-60.	Device Status Byte	2-7

CONTENTS (Continued)

Section			Page
III	THEORY OF OPERATION		
	3-1.	Introduction	3-1
	3-3.	System-Level Description	3-1
	3-11.	Functional-Level Description	3-1
	3-13.	Command Function	3-3
	3-15.	Direct Commands	3-3
	3-23.	Program Commands	3-7
	3-24.	Program Control Strobe Operation	3-7
	3-29.	Program Read Strobe Operation	3-8
	3-33.	Program Write Strobe Operation	3-10
	3-36.	Direct/Program Status Strobe Operations	3-10
	3-39.	"Not" Set Interrupt	3-13
	3-40.	Interrupt Processing	3-13
	3-45.	I/O Data Gates	3-14
	3-49.	Control Function	3-14
	3-51.	Device Control Logic	3-19
	3-53.	Sequence Counter Qualifier Logic	3-19
	3-60.	Sequence Counter	3-24
	3-62.	Device Command Logic	3-25
	3-63.	Device Flag Edge Detector	3-26
	3-65.	Device Control Word Logic	3-29
	3-67.	Device End Logic	3-30
	3-68.	Device End During a One-Word Data Transfer	3-30
	3-69.	Device End During a Two-Byte Data Transfer	3-30
	3-70.	Device Service Request Logic	3-32
	3-72.	Interrupt Request Logic	3-32
	3-74.	Data Transfer Timer	3-33
	3-76.	Device Interrupt Request Logic	3-34
	3-77.	I/O System and Interface Interrupt Request Logic	3-34
	3-78.	Clear Interrupt Decoder Logic	3-35
	3-80.	Data and Status Transfer Function	3-37
	3-82.	Status Logic and Control	3-37
	3-83.	Interface Status Byte	3-38
	3-91.	Device Status Byte	3-40
	3-92.	Interrupt Status Byte	3-40
	3-93.	Data In Logic	3-40
	3-94.	One-Word Data Transfer	3-40
	3-95.	Byte Transfer	3-40
	3-96.	Data Out Logic	3-40
	3-97.	Normal Write	3-41
	3-98.	Test Write	3-41
IV	MAINTENANCE		
	4-1.	Introduction	4-1
	4-3.	General Servicing Information	4-1
	4-5.	Safety Precautions	4-1
	4-9.	Wiring Information	4-1
	4-11.	Jumper Wire Configuration	4-1
	4-13.	Service Request Priority (W5)	4-1
	4-14.	Self-Test Power (W1)	4-1
	4-15.	Group Interrupt Mask (W4)	4-1
	4-16.	DRT Address Parity (W2)	4-1
	4-17.	Device Number (W6 through W13)	4-1
	4-18.	Required Servicing Equipment	4-1
	4-20.	Principal Servicing Points	4-3

CONTENTS (Continued)

Section		Page
4-23.	Preventive Maintenance	4-3
4-25.	Troubleshooting	4-4
4-28.	Corrective Maintenance	4-4

ILLUSTRATIONS

Figure	Title	Page
1-1.	HP 30051A Universal Interface (Differential)	1-1
2-1.	Control Word Format	2-4
2-2.	Status Word Format	2-6
2-3.	Interface/Interrupt Status Word Format	2-6
2-4.	Interface/Device Status Word Format	2-7
3-1.	Overall System Block Diagram	3-2
3-2.	Direct Command Functional Logic Diagram	3-4
3-3.	Program Control Strobe Command Functional Logic Diagram	3-7
3-4.	Program Strobe Timing	3-8
3-5.	Program Read Strobe Functional Logic Diagram	3-9
3-6.	Program Write Strobe Functional Logic Diagram	3-11
3-7.	Direct/Program Status Strobe Functional Logic Diagram	3-12
3-8.	Interrupt Processing Functional Logic Diagram	3-15
3-9.	Command Function I/O Data Gates Functional Logic Diagram	3-17
3-10.	Device Control Logic Block Diagram	3-19
3-11.	Response Mode and Pulse Mode Timing Diagrams	3-20
3-12.	Qualifier Q0 Simplified Diagram	3-20
3-13.	Control Function Sequence Counter Operational Flow Diagram	3-21
3-14.	Qualifier Q1 and Q3 Simplified Diagram	3-23
3-15.	Qualifier Q2 Simplified Diagram	3-23
3-16.	Sequence Counter Simplified Diagram	3-24
3-17.	Device Command Logic Block Diagram	3-25
3-18.	Device Flag Edge Detector	3-26
3-19.	Device Command Logic	3-27
3-20.	Device Control Word Logic	3-29
3-21.	Discrete Device Control Signals	3-30
3-22.	Device End Logic	3-31
3-23.	Device End Signal Timing Diagrams	3-32
3-24.	Device Service Request Logic	3-33
3-25.	Data Transfer Timer Interrupt Request Logic	3-34
3-26.	Device Interrupt Request Logic	3-35
3-27.	I/O System and Interface Interrupt Request Logic	3-36
3-28.	Clear Interrupt Decoder Logic	3-37
3-29.	Status Logic and Control	3-38
3-30.	Data In (from Device) Transfer Logic	3-43
3-31.	Data Out (to Device) Transfer Logic	3-45
4-1.	Interface PCA and Interconnection Cable Connector Pin Numbering Scheme	4-2
4-2.	Interface PCA Jumper Wire Locations	4-5

TABLES

Table	Title	Page
1-1.	Interface PCA Specifications	1-2
2-1.	Sample Program to Read Data	2-1
2-2.	Sample Program to Write Data	2-2
2-3.	Direct I/O Command Codes	2-3
2-4.	Interrupt Request Selective Clear	2-4
2-5.	Interrupt Service Routine	2-5
2-6.	Sequence Counter Status Bit Decoding	2-6
3-1.	Direct I/O Commands Decoding Table	3-4
3-2.	Command Function External Signal List	3-5
4-1.	Interface PCA to Interconnecting Cable Connector Pin Cross Reference and Function Index	4-3
4-2.	Connector J2 Jumper Wire Functions	4-4

SECTION I

GENERAL INFORMATION

1-1. INTRODUCTION.

1-2. This section describes the physical features and provides specifications and identification data for the HP 30051A Universal Interface (Differential) (figure 1-1). Related publications that may be required for operation of this interface are listed in the preface to this manual.

1-3. GENERAL DESCRIPTION.

1-4. The HP 30051A Universal Interface (Differential) provides interface capability between the HP 3000 Computer System and an external I/O device that uses parallel data transfer through differential logic circuits. Typical devices used with the HP 3000 Computer System include line printers, card punches, and paper tape readers and punches. The HP 30051A Universal Interface (Differential) consists of the following items:

a. Universal interface (differential) printed-circuit assembly, part no. 30051-60001.

b. Maintenance Manual, part no. 30051-90001.

1-5. The universal interface (differential) printed-circuit assembly (interface PCA) contains logic circuits for decoding and executing programmed instructions from the computer system. Executing programmed instructions includes sending 16-bit data and status words from the device to the computer system and 16-bit data and control words from the computer system to the device. When transferring data to the device, the interface PCA accepts up to 16 bits from the computer system and transfers these bits to the device in one word or in two bytes of up to 8 bits each. When accepting data from the device, the interface PCA receives either one word of up to 16 bits or two bytes of up to 8 bits each. These two bytes are then packed into one 16-bit word before transfer to the computer system. Software control of the interface PCA can be either direct (CPU-to-interface PCA) mode or SIO (IOP-to-multiplexer channel-to-interface PCA) mode. In both cases, the I/O operations performed between the device and interface PCA are identical.

1-6. Status information is transferred to the computer system in either of two 16-bit word formats. In both formats the high order eight bits (0 through 7) indicate status of the interface PCA logic circuits. The low order bits (8 through 15) are programmed through the control word format to indicate status of either the external device or the interrupt request circuit. Eight different actions of the device can be monitored through the device status byte. The interrupt request circuit contains eight different methods for the device and interface PCA to request an interrupt for service from the computer and each interrupt request provides a separate bit in the interrupt status byte.

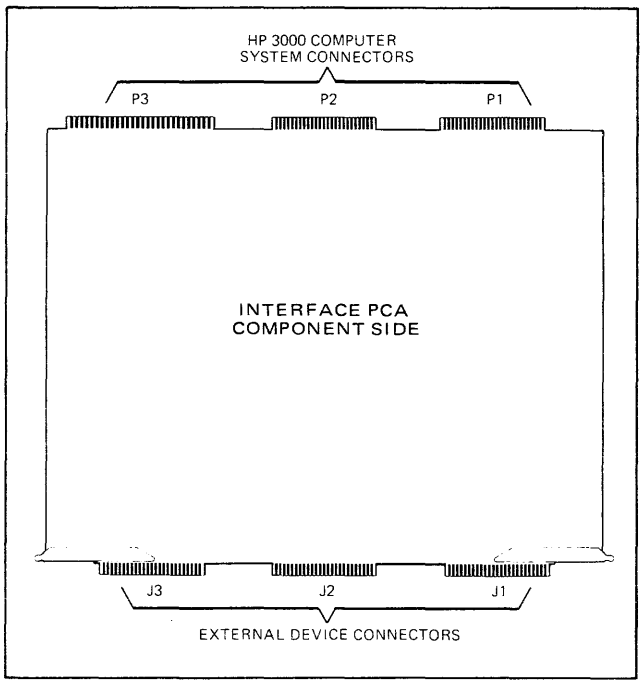


Figure 1-1. HP 30051A Universal Interface (Differential)

1-7. The 16-bit control word contains 5 bits that are transferrable directly to the device. The remaining bits are used on the interface PCA to control data and status transfers and interrupt request logic.

1-8. SPECIFICATIONS.

1-9. Specifications for the interface PCA are listed in table 1-1. Data transfer rates given in table 1-1 were measured between successive programmed strobes with the diagnostic hardware (self-test connector), part no. 30049-60001, installed and a clock rate of 200 nanoseconds. These rates are affected by interconnecting cable length and I/O circuit capacitance so that cables longer than 30 feet and/or increased capacitance will decrease the data transfer rate.

1-10. IDENTIFICATION.

1-11. Printed-circuit assemblies (PCA's) are identified by a part number etched on the PCA. Revisions to the PCA are

identified by a letter, a series code, and a division code (A-0000-00) marked beneath the part number on the PCA. The letter identifies the version of the etched trace pattern on the unloaded PCA. The four-digit series code pertains to the electrical characteristics and the positions of the components on the PCA. The two-digit division code identifies the division of Hewlett-Packard that manufactured the PCA.

Table 1-1. Interface PCA Specifications

CURRENT REQUIRED FROM COMPUTER POWER SUPPLIES:	
+5-volt supply:	3.5 amperes
METHOD OF DATA TRANSFER:	
To and from Device:	16-bit parallel, or two 8-bit bytes using byte transfer, through differential line drivers and receivers
To and from Computer:	16-bit parallel
Logic 1 level (high):	+2.5 Vdc minimum
Logic 0 level (low):	+0.5 Vdc maximum
DATA TRANSFER RATE: (Refer to paragraph 1-9.)	
Read Order:	
Packing Disabled	300k words per second
Packing Enabled	230k words per second
Write Order:	
Unpacking Disabled	210k words per second
Unpacking Enabled	180k words per second
DIMENSIONS:	
Depth:	11-1/2 in. (292.1 mm)
Width:	13-11/16 in. (352.425 mm)
Thickness (with Components):	5/8 in. (15.875 mm)
WEIGHT:	
Net:	1 lb, 4 oz (0.558 kg)

SECTION II OPERATING PARAMETERS

2-1. INTRODUCTION.

2-2. This section contains information regarding operation of the interface PCA in the HP 3000 Computer System. Descriptions are included for I/O instructions, direct I/O commands, and programmed I/O orders used by the interface PCA. The control word and status word formats are also described along with sample programming information. All information in this section is general in regard to programming data transfers with a particular device. To determine specific parameters for a particular device, refer to the applicable subsystem manual.

2-3. GENERAL INFORMATION.

2-4. The interface PCA permits bidirectional data transfer between an external device and the HP 3000 Computer System. This interface PCA operates under direct I/O commands from the CPU or under programmed I/O orders initiated through an SIO routine. In either case, the interface PCA performs the same functions and operation is identical to the external device.

2-5. The interface PCA has two sets of jumper wires that determine operating parameters. One set of jumper wires is located on the interface PCA and is used to select the device number, service request priority, group interrupt mask, and DRT address parity (see figure 4-2). The other set of jumper wires is located on the device cable connector which connects to connector J2 on the interface PCA. These jumpers select various operating features of the interface PCA (refer to table 4-2).

2-6. All required control information, data, and status is transferred between the HP 3000 Computer System and the interface PCA by way of the IOP and MUX buses. A block diagram showing the connections for these buses is provided in figure 3-1.

2-7. SYSTEM PARAMETERS.

2-8. Operation of the interface PCA is controlled by machine language I/O instructions. These instructions are supplied by a user program or the operating system. From either source, the response to any given instruction is the same. I/O instructions initiate either direct I/O operations (operations under direct control of the CPU) or programmed I/O operations (operations under control of the multiplexer channel). The following paragraphs describe the I/O instructions that affect the interface PCA and the resultant direct and/or programmed I/O operation.

2-9. CIO INSTRUCTION. The CIO instruction causes a 16-bit control word to be sent from the top-of-stack (TOS) register in the CPU/IOP module to the interface PCA. All 16 bits of the control word are sent with the CIO instruction, therefore, each command modifies all bits in the interface PCA control word buffers. The control word format is described in paragraph 2-28. If the CIO instruction is executed successfully, condition code CCE is set in the computer system.

2-10. RIO INSTRUCTION. The RIO instruction strobes a data word from the interface PCA data input buffers to the computer system memory (refer to table 2-1). The RIO instruction is preceded by a CIO instruction to transfer data

Table 2-1. Sample Program to Read Data

```
<<THIS ROUTINE READS TWO BYTES OF DATA FROM THE DEVICE>>
            TOS←DEVICENUMBER;
            TOS←%002004;                    <<ENABLE DATA TRANSFER>>
                                            <<ENABLE PACKING OPTION>>
            ASSEMBLE (CIO 1);
SELF:       IF <> THEN GO SELF;             <<TEST FOR CIO COMPLETED>>
TRYAGAIN:   ASSEMBLE (RIO Ø);
HANGUP:     IF < THEN GO HANGUP;            <<DEVICE DOES NOT RESPOND>>
            IF = THEN GO ONWARD;            <<RIO EXECUTED O.K., CONTINUE>>
            ASSEMBLE (DEL);                 <<DATA TRANSFER STILL IN PROGRESS>>
            GO TRY AGAIN;                   << DELETE STATUS WORD & TRY AGAIN>>
ONWARD:     DATA←TOS;                       << 2 BYTES IN PACKED FORMAT>>
            <<CONTINUE WITH MAIN PROGRAM>>
```

(control word bit 5). The interface PCA accepts a 16-bit word of data (or two 8-bit bytes if byte transfer is enabled) from the device and stores the data in the input buffer. If bit 1 of the status word is high, the RIO instruction reads the buffer contents into the computer system memory. When the RIO instruction is executed successfully, condition code CCE is set in the computer system.

2-11. SIN INSTRUCTION. The SIN instruction sets the I/O system interrupt (bit 13 of the interrupt status word) and causes an interrupt request to be initiated. If control word bit 14 is high and no other interrupts are active, the interface PCA will request an interrupt.

2-12. SIO INSTRUCTION. The SIO instruction initiates a microprogram routine that switches control of the interface PCA away from the CPU and allows the multiplexer channel to control I/O operations. This frees the CPU to do other word during routine I/O operations.

2-13. SMSK INSTRUCTION. The SMSK instruction sends a 16-bit mask word from the top-of-stack (TOS) register in the CPU/IOP module through the interface PCA data out lines. The mask word sets up all interface PCA's in the mask group to allow interrupts. Interface PCA's are assigned to mask groups by a jumper wire (see figure 4-2).

2-14. TIO INSTRUCTION. The TIO instruction causes the interface PCA to send a status word to the computer system. If the TIO instruction is internally microprogrammed as part of an RIO, WIO, or SIO instruction, the status word is sent to scratch pad 2 in the CPU. If the TIO instruction is part of the program code, the status word is sent to the TOS register in the CPU/IOP module. The status word format is described starting in paragraph 2-40.

2-15. WIO INSTRUCTION. The WIO instruction causes a 16-bit data word (or two 8-bit bytes if byte transfer is enabled) to be transferred from the computer system to the external device (refer to table 2-2). Bit 1 of the status word must be high before the WIO instruction is issued or the instruction is rejected. If the WIO instruction is executed successfully, condition code CCE is set in the computer system.

2-16. DIRECT I/O.

2-17. Direct I/O operations are under control of the CPU and result from decoding and executing machine language instructions. These instructions each cause the CPU to issue a 12-bit word over the IOP bus to all PCA's. This 12-bit word consists of a 3-bit direct I/O command (IOCMD), an 8-bit device number (DEVNO), and a service out (SO) bit. The IOCMD bits indicate which operation is required by a PCA (refer to table 2-3), the DEVNO bits indicate the PCA being addressed, and the SO bit indicates that a direct I/O command is on the IOP bus. The addressed PCA acknowledges receipt of the direct I/O command by returning a service in (SI) signal over the IOP bus.

2-18. PROGRAMMED I/O.

2-19. The I/O program is initiated as a result of a direct SIO instruction and once initiated is controlled by the multiplexer channel through the IOP and IOP bus. The multiplexer channel accepts the programmed I/O command orders one-at-a-time from the IOP, decodes the command orders and sends control strobe signals to the interface PCA. There are eight programmed I/O command orders that affect operation of the interface PCA. The command orders are CONTROL, END, INTERRUPT, JUMP, READ, RETURN RESIDUE, SENSE, and WRITE. How these command orders affect operation of the interface PCA is described in the following paragraphs.

2-20. CONTROL ORDER. This order causes a transfer of a 16-bit control word in the IOAW field from memory to the interface PCA. The control word format is identical to the CIO instruction control word format and is described in paragraph 2-28. To prevent loss of control word bits, all bits must be restored with each control word.

2-21. END ORDER. This order indicates the termination of an I/O program and causes the interface PCA to send a status word to the computer system memory location specified by the IOAW. The status word format is described in paragraph 2-40.

Table 2-2. Sample Program to Write Data

```
<<THIS ROUTINE WRITES TWO BYTES OF DATA TO THE DEVICE>>
            TOS←DEVICENUMBER;
            TOS←%000004;                    <<ENABLE UNPACKING OPTION>>
            ASSEMBLE (CIO 1);
SELF:       IF <> THEN GO SELF;             <<TEST FOR CIO COMPLETED>>
            TOS←DATA;                       << 2 BYTES IN PACKED FORMAT>>
            ASSEMBLE (WIO 1);
HANGUP:     IF <> THEN GO HANGUP;           <<TEST FOR WIO COMPLETED>>
            <<CONTINUED WITH MAIN PROGRAM>>
```

Table 2-3. Direct I/O Command Codes

COMMAND	$\overline{\text{IOCMD}}$			DESCRIPTION
	$\overline{00}$	$\overline{01}$	$\overline{02}$	
CIO	1	1	0	Control I/O. Transfers a 16-bit control word from the computer system to the interface PCA (see figure 2-1).
RESET INT	0	1	1	Reset Interrupt. Clears the interface PCA interrupt active condition but does not clear the conditional logic that specifies the cause of an interrupt request.
RIO	0	0	0	Read I/O. Transfers a 16-bit data word from the interface PCA to the computer system.
SIN	1	1	1	Set Interrupt. Sets the I/O system interrupt and causes an interrupt request to be initiated.
SIO	1	0	1	Start I/O. Initiates a microprogram routine that allows the multiplexer channel to control operation of the interface PCA.
SMSK	0	0	1	Set Mask. Transfers a mask word from top of stack (TOS) to all interface PCA's.
TIO	0	1	0	Test I/O. Transfers a 16-bit status word from the interface PCA to the computer system (see figures 2-2, 2-3, and 2-4).
WIO	1	0	0	Write I/O. Transfers a 16-bit data word from the computer system to the interface PCA.

2-22. INTERRUPT ORDER. This order sets the I/O system interrupt (bit 13 of the interrupt status word) and causes an interrupt request to be initiated. Then, if control word bit 14 is high and no other interrupts are active, the interface PCA will request an interrupt. There is no effect on the external device.

2-23. JUMP ORDER. This order causes the multiplexer channel to jump unconditionally to the address given in the IOAW for the next IOCW to be executed.

2-24. READ ORDER. This order causes the transfer of a block of data from the device to the computer system memory. The starting address in memory is specified by the IOAW. This transfer terminates normally if the device sends a Device End signal or the word count has gone to zero.

2-25. RETURN RESIDUE ORDER. This order has no effect on the interface PCA. The order returns the remainder of the word count to the IOAW. This information is used to determine the number of words transferred if the device terminates a READ or WRITE order before word count has gone to zero.

2-26. SENSE ORDER. This order causes a 16-bit status word to be transferred from the interface PCA to the computer by the IOAW. The status word format is described in paragraph 2-40.

2-27. WRITE ORDER. This order causes the transfer of a block of data from the computer system memory to the external device. The starting address of the block is specified by the IOAW. This transfer terminates normally if the device sends a Device End signal or the word count has gone to zero.

2-28. CONTROL WORD FORMAT.

2-29. The control word is a 16-bit word that is sent from the computer system to the interface PCA over the IOP bus data out lines. This control word is sent to the interface PCA after execution of a direct CIO instruction or I/O program CONTROL order. To ensure against loss of control bits, all of the control bits must be restored each time a control word is sent to the interface PCA. Figure 2-1 shows the control word format and the separate bits are explained in the following paragraphs.

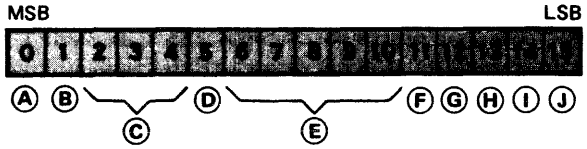
2-30. MASTER CLEAR. When high (logic 1), this bit initializes the interface PCA. Its action on the interface PCA is similar to pushing the I/O reset switch on the computer system console. Master clear should be used only when the condition of the interface PCA is unknown or when it is necessary to clear one of the interface status bits.

2-31. CLEAR ALL INTERRUPTS. This bit, when high, clears all processes on the interface PCA that could initiate an interrupt request. Master clear has the same effect on the interrupt processing logic. After this bit is issued, the interrupt pending bit (bit 2 of the status word) should be low indicating that all interrupt requests were cleared.

2-32. SELECTIVE INTERRUPT CLEAR. These three bits may be programmed to separately clear each of the eight conditions that can cause an interrupt request. Table 2-4 lists the bits configurations required to clear each interrupt request.

2-33. INITIATE DATA TRANSFER. This bit must be set high with a CIO instruction before every RIO instruction to signal the interface PCA to fetch another data word from the device. Set this bit high only when bits 0 and 1 of the status word are both high to indicate that an SIO is not in progress and data is not being transferred.

2-34. DEVICE CONTROL. These five bits are made available to the device through differential line drivers. On the interface PCA, these bits are stored in a buffer that is modified by every control word.



Ⓐ Master Clear. A logic 1 initializes interface PCA.

Ⓑ Clear All Interrupts. A logic 1 clears all processes on the interface PCA that could initiate an Interrupt Request signal.

Ⓒ Selective Interrupt Clear. These three bits are programmed in octal code to selectively clear each of the eight processes that can initiate an Interrupt Request signal (refer to table 2-4).

Ⓓ Initiate Data Transfer. This bit must be set to a logic 1 by a CIO instruction before every RIO instruction to signal the interface PCA to accept another data word from the device. Bits 0 and 1 of the status word must also be set to a logic 1.

Ⓔ Device Control. These five control bits are available to the external device through differential line drivers.

Ⓕ Data Transfer Interrupt. A logic 1 allows the interface PCA to issue an Interrupt Request signal at the completion of each data transfer sequence.

Ⓖ Interrupt/Device Status. This bit selects the lower eight bits (8 through 15) of the status word. A logic 1 selects device status, and a logic 0 selects interrupt status.

Ⓗ Enable Byte Transfer. A logic 1 enables byte transfer for packing during read and unpacking during write.

Ⓘ Enable Interrupt Request. A logic 1 enables the Interrupt Request signal.

Ⓙ Enable Transfer Timer. A logic 1 enables the transfer timer.

2186-4

Figure 2-1. Control Word Format

Table 2-4. Interrupt Request Selective Clear

CONTROL WORD BITS			INTERRUPT PROCESS CLEARED
2	3	4	
0	0	0	Inactive state, no action
0	0	1	Transfer Timer and Transfer Error, interrupt status bits 15 and 14
0	1	0	I/O System, interrupt status bit 13
0	1	1	Clear Interface, interrupt status bit 12
1	0	0	Data Transfer, interrupt status bit 11
1	0	1	Device Status bit 8, interrupt status bit 8
1	1	0	Device Status bit 9, interrupt status bit 9
1	1	1	Device Status bit 10, interrupt status bit 10

2-35. DATA TRANSFER INTERRUPT. This bit, when high, causes the interface PCA to issue an interrupt request at the completion of each data transfer sequence. Its purpose is to allow programming of data transfer instructions (WIO and RIO) to the device, continue with some other operation, and then return to transfer more data when signaled by the interrupt.

2-36. INTERRUPT/DEVICE STATUS. When this bit is low the interrupt status byte is enabled and when this bit is high the device status byte is enabled. Table 2-5 provides a typical interrupt service routine with changes in the stack shown for each step.

2-37. BYTE TRANSFER. When high, this bit allows the interface PCA to accept two data bytes of up to 8 bits each from the device and to pack them into a single word of up to 16 bits. This data word can then be transferred to the computer system memory. Also, the high byte transfer bit allows the interface PCA to accept a single word of up to 16 bits from the computer system memory and to transfer the word to the device in two bytes of up to 8 bits each.

2-38. ENABLE INTERRUPT REQUEST. This bit, when high, enables the interrupt request logic so that the interface PCA may request an interrupt.

2-39. START TRANSFER TIMER. This bit, when high, starts the data transfer timer. If the transfer timer is not cleared by a control word (see table 2-4) within 5 seconds, an interrupt request is initiated.

2-40. STATUS WORD FORMATS.

2-41. The status word is a 16-bit word that is fetched from the interface PCA after executing a TIO instruction or a SENSE order in an I/O program. When the TIO instruction is internally microprogrammed as part of an RIO, WIO, or SIO instruction, the status word is sent to scratch pad 2 of the CPU. If the TIO instruction is part of the program code, the status word is sent to the TOS register in the CPU/IOP module. The SENSE order is always the first part of an I/O program doubleword and results in the status word being stored in the second part of the same doubleword.

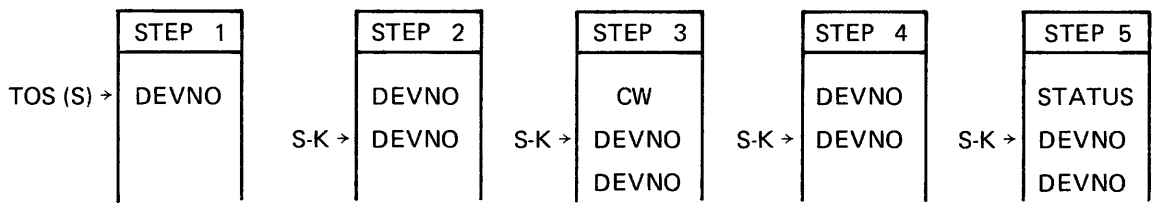
2-42. The 16-bit status word has two different formats with the upper eight bits (bits 0 through 7) being the same for either format. The lower eight bits (8 through 15) indicate either interrupt status or device status depending on control word bit 12. With control word bit 12 low, the status word contains the interrupt status byte and with control word bit 12 high, the status word contains the device status byte.

Table 2-5. Interrupt Service Routine

```
START:   *** Typical interrupt service routine to retrieve
         * a status word containing the device status
         * byte.

1        LOAD     DEVNO    <Push device number onto stack>
2        DUP               <Duplicate device number in next location>
3        LOAD     CW       <Push control word onto stack>
4        CIO      K (K=1)  < K = location of DEVNO relative to top of stack. In this case
                           DEVNO is one location below TOS (S-1) so K=1>
5        TIO      K (K=1)

         * Status word is now on the TOS. Remainder of
         * program can test status and respond
         * accordingly.
```



2-43. INTERFACE STATUS BYTE.

2-44. The interface status byte is contained in the upper eight bits (0 through 7) of every status word. These bits indicate the condition of several interface PCA functions (see figure 2-2). Each bit is explained in the following paragraphs.

2-45. SIO OK. When high, this status bit indicates that it is permissable to execute an SIO command. A low in this bit position indicates that an SIO routine is in progress or that the multiplexer channel cannot service the interface PCA.

2-46. RIO, WIO OK. This bit, when high, indicates that it is permissable to execute an RIO or WIO instruction to transfer data between the device and computer system. During data transfers this bit is held low to indicate that a data transfer is in progress.

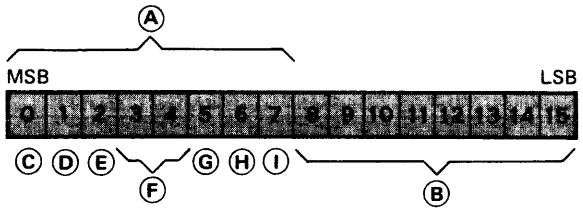
2-47. INTERRUPT PENDING. A high in this bit position indicates that at least one of the eight possible interface PCA functions is requesting an interrupt. When this bit is low it indicates that there are no active interrupt requests. Even when the interrupt pending bit is high, the interrupt request may be inhibited if control word bit 14 is low.

2-48. SEQUENCE COUNTER. These two bits indicate the condition of the interface PCA data transfer sequence counter. They are used primarily for troubleshooting the interface PCA during diagnostic routines. The two bits are decoded as described in table 2-6.

2-49. DEVICE FLAG. This bit is supplied by the external device in response to a request for operation. When high, it indicates that the I/O operation at the device is complete. Between I/O operations, this bit is normally low.

2-50. INTERRUPT/DEVICE STATUS. This bit is controlled by bit 12 of the control word and is used to indicate the content of the lower eight status bits (7 through 15). When this bit is high, the lower eight status bits are device status and when this bit is low, the lower eight status bits are interrupt status. Interrupt and device status bits are explained in the following paragraphs.

2-51. PERMANENT ZERO. Status bit 7 is permanently wired low.



Ⓐ Interface Status Byte. These upper eight bits always indicate the interface PCA status.

Ⓑ Interrupt/Device Status Byte. These lower eight bits indicate interrupt status when bit 6 is a logic 0 and indicate device status when bit 6 is a logic 1.

Ⓒ SIO OK. A logic 1 indicates that it is permissable to execute an SIO instruction.

Ⓓ RIO, WIO OK. A logic 1 indicates that it is permissable to execute an RIO or WIO instruction.

Ⓔ Interrupt Pending. A logic 1 indicates that at least one of the eight possible interface PCA operations is requesting an interrupt.

Ⓕ Sequence Counter. These two bits indicate the condition of the data transfer sequence counter.

Ⓖ Device Flag. A logic 1 indicates I/O operation has begun. Between I/O operations this bit is normally at logic 0.

Ⓗ Interrupt/Device Status. This bit indicates the contents of the lower eight status bits. A logic 1 indicates device status and a logic 0 indicates interrupt status.

Ⓘ Permanent Zero. Permanently wired to logic 0.

2186-5

Figure 2-2. Status Word Format

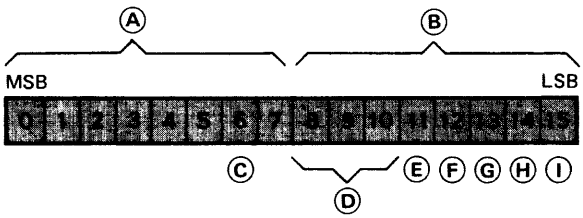
Table 2-6. Sequence Counter Status Bit Decoding

STATUS BIT		DESCRIPTION
3	4	
0	0	Sequence counter at rest.
1	0	Request for operation to device.
1	1	Begin operation from device.
0	1	Request for operation to begin transfer of second byte.

2-52. INTERRUPT STATUS BYTE.

2-53. The interrupt status byte is contained in bits 8 through 15 of the status word when bit 6 of the status word is low. Each bit of the interrupt status byte indicates a function on the interface PCA that can request an interrupt (see figure 2-3). When one or more of these bits are high, indicating an interrupt request, bit 2 of the status word is also high to indicate that an interrupt request is waiting for service. Also, each of the interrupt request conditions and its associated status bit can be cleared individually to ensure that no interrupt requests will be lost. Selective clearing of the interrupt requests is described in paragraph 2-32. The following paragraphs describe each of the interrupt status bits.

2-54. DEVICE STATUS INTERRUPTS. The device status interrupts are three independent bits in the interrupt status word. These bits (8, 9, and 10) are enabled by the same signals from the device that provide device status bits 8, 9, and 10. To activate the interrupt requests from the device status lines, the appropriate jumper wire must be



Ⓐ Interface Status Byte. These upper eight bits always indicate the interface PCA status.

Ⓑ Interrupt Status Byte. These lower eight bits indicate interrupt status when bit 6 is a logic 0.

Ⓒ Interrupt/Device Status. This bit indicates content of the lower eight status bits and must be a logic 0 for interrupt status.

Ⓓ Device Status Interrupts. Device status bits 8, 9, and 10 can be used to initiate interrupt requests. When used to initiate an interrupt request, device status bits 8, 9, and 10 also initiate interrupt status bits 8, 9, and 10 respectively.

Ⓔ Data Transfer Interrupt. A logic 1 indicates that control word bit 11 is a logic 1 and that an interrupt request will occur after each data transfer.

Ⓕ Clear Interface Interrupt. A logic 1 indicates that an I/O program was cancelled by the external device.

Ⓖ I/O System Interrupt. A logic 1 indicates that a direct SIN command or I/O program INTERRUPT code was executed.

Ⓗ Transfer Error Interrupt. A logic 1 indicates that an error has occurred in the transfer of data between the interface PCA and the computer system.

Ⓘ Transfer Timer Interrupt. A logic 1 indicates that the transfer timer was enabled by control word bit 15 and the time expired before the timer was cleared.

2186-6

Figure 2-3. Interface/Interrupt Status Word Format

installed in the mating connector to connector J2. The jumper wires designation are J2W4, J2W8, and J2W9 for interrupt status bits 8, 9, and 10 respectively (refer to table 4-2). When the appropriate jumper wire is installed, interrupt status bits 8 and 9 are enabled by the leading edge of device status signals 8 and 9, and interrupt status bit 10 is enabled by the trailing edge of device status bit 10.

2-55. DATA TRANSFER INTERRUPT. The data transfer interrupt is enabled by control word bit 11 and when enabled causes an interrupt request after each data transfer. Refer to paragraph 2-35.

2-56. CLEAR INTERFACE INTERRUPT. A clear interface interrupt is valid only during an I/O program when the multiplexer channel is controlling the interface PCA. The external device can completely abort the I/O program and enable this interrupt by sending a Clear Interface signal to the interface PCA.

2-57. I/O SYSTEM INTERRUPT. This bit, when high, indicates that a direct SIN instruction or an I/O program INTERRUPT order was executed and an interrupt requested.

2-58. TRANSFER ERROR INTERRUPT. This bit, when high, indicates that an error has occurred in the transfer of data between the interface PCA and the computer system. Once enabled, this bit must be cleared before another data transfer can occur. The conditions that cause this bit to go high are as follows:

a. Illegal address.

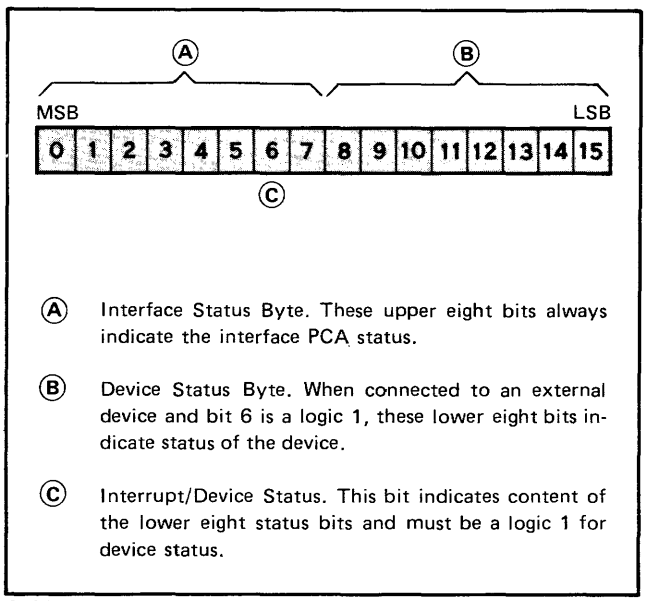
b. Memory parity error.

c. Parity error on data transfer to or from computer system memory.

2-59. TIME OUT INTERRUPT. This bit goes high if the transfer timer delay expires. The transfer timer is first started by control word bit 15 and then expires 5 seconds later if the timer is not cleared by a control word with bit 4 at a logic 1. If the timer is not cleared within the preset time limit, an interrupt is requested.

2-60. DEVICE STATUS BYTE.

2-61. The device status byte is contained in bits 8 through 15 of the status word when bit 6 of the status word is high. Each bit of the device status word indicates an input signal from the external device (see figure 2-4). Three of the bits (8, 9, and 10) are also connected to the interrupt status lines to enable the device to request an interrupt (refer to paragraph 2-54).



2186-7

Figure 2-4. Interface/Device Status Word Format

SECTION III THEORY OF OPERATION

3-1. INTRODUCTION.

3-2. This section contains system-level and functional-level descriptions of operation for the HP 30051A Universal Interface (Differential) PCA. The system-level description briefly describes interface PCA operations in relation to the HP 3000 Computer System. The functional-level description divides the interface PCA into functional circuit groups and provides a description of operation for each group.

3-3. SYSTEM-LEVEL DESCRIPTION.

3-4. The interface PCA is combined with any compatible device such as a line printer to form a subsystem of the HP 3000 Computer System (see figure 3-1). This subsystem can be any one of sixteen subsystems connected to each multiplexer channel in the overall system. Operation of the subsystem requires the transfer of commands, data, and status information between the interface card and the following system components:

a. Memory Module

b. Central Processor Unit (CPU)

c. Input/Output Processor (IOP)

d. Multiplexer Channel

3-5. The memory module contains the I/O drivers that are executed by the CPU, I/O programs that are transferred by the IOP to the multiplexer channel, and the device reference tables (DRT). I/O drivers contain direct instructions such as Read I/O (RIO), Write I/O (WIO), and Start I/O (SIO). As an I/O driver instruction is executed by the CPU, the CPU issues direct commands through the IOP and IOP bus directly to the addressed interface PCA. When the interface PCA accepts a direct command, it returns an acknowledge signal and performs the command. If the direct instruction was an SIO, the multiplexer channel assumes control of the interface PCA and the CPU is free to perform other functions.

3-6. An SIO instruction calls a routine that permits the IOP to transfer an I/O program, one instruction at a time, from the memory module to the multiplexer channel. The multiplexer channel then controls operation of the interface PCA. Instructions contained in the I/O program are similar to the instructions contained in the I/O driver and they perform many of the same functions such as Read I/O and Write I/O.

3-7. The multiplexer channel is a controller for up to 16 subsystems and contains a one-word storage location for each subsystem. Program instructions from the IOP are stored one at a time in the storage location for the addressed subsystem. Each program word is then decoded by the multiplexer channel and sent through the MUX CHAN bus to the appropriate interface PCA. When the interface PCA accepts the decoded program instruction, it returns an acknowledge signal to the IOP and performs the instruction. The interface PCA and multiplexer channel exchange signals that monitor progress in performing instructions.

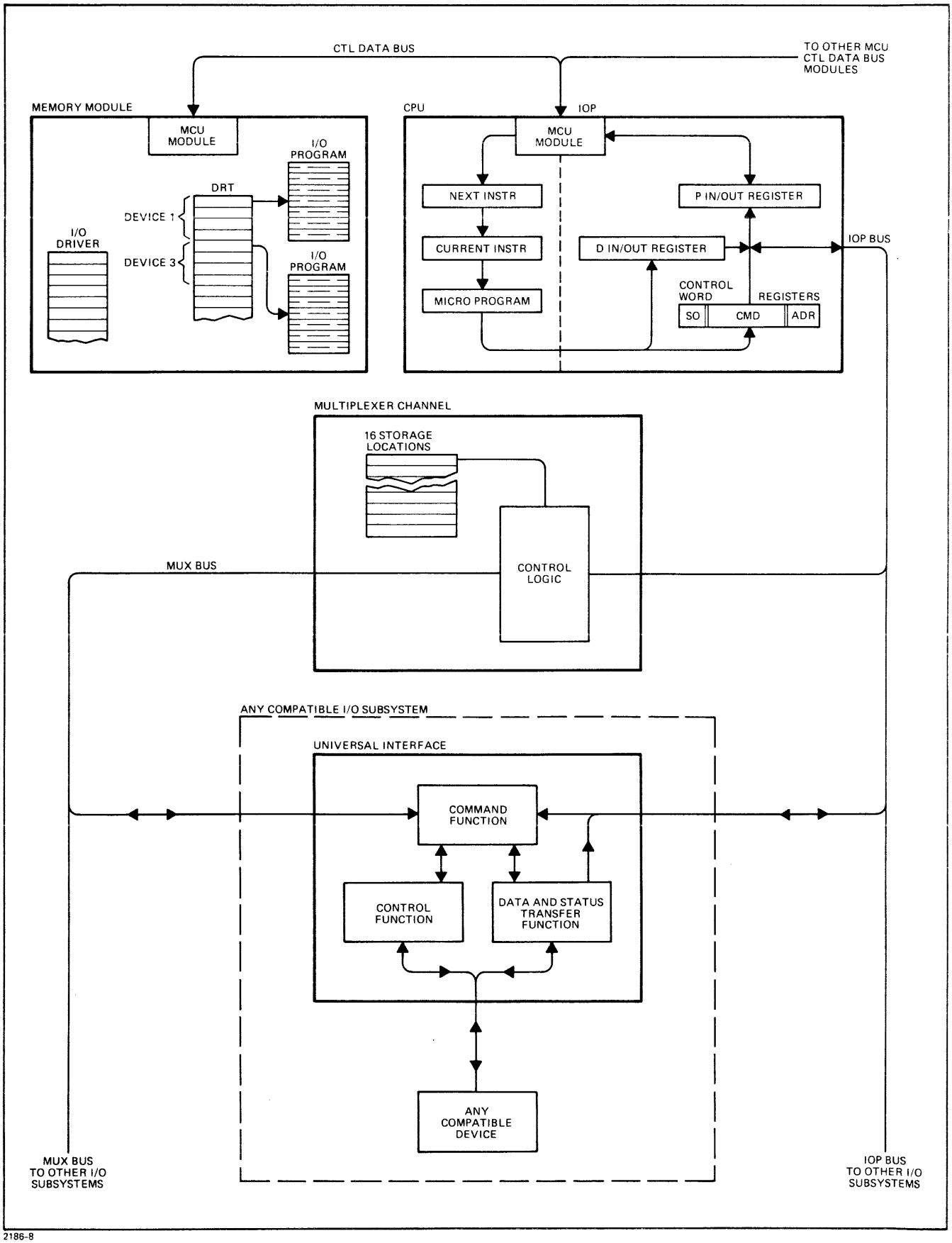
3-8. Data are transferred to and from the device through the interface PCA and IOP bus. This transfer occurs in response to either direct or program I/O instructions. The device number (address) and interrupt priority are determined by relocateable jumper wires on the interface PCA. An interrupt to transfer data to or from the device can be initiated by the operating program or by an interrupt request from the device and interface PCA. Interrupt requests are also initiated due to several different status conditions.

3-9. Status conditions for the interface PCA and device, including the conditions that cause an automatic request for interrupt, are transferred from the interface PCA through the IOP bus data lines. These status conditions are transferred as either of two 16-bit status words. One status word indicates the interface PCA status and the status of eight interrupt conditions. The second status word indicates the same interface PCA status as the first status word. However, the eight interrupt status bits are replaced by eight device status bits. Bit 12 of the control word determines which status word is transferred.

3-10. The device control word is a 16-bit word that is sent to the interface PCA through the IOP bus data lines. Five bits of the control word are available at output pins to control device operations. The remaining bits control functions on the interface PCA such as selecting the interrupt status or device status word. The interface card also has three discrete signals available for the device. These three discrete signals are Master Clear, Power Fail Warning, and Power On.

3-11. FUNCTIONAL-LEVEL DESCRIPTION.

3-12. The interface PCA performs three interrelated functions. These functions are (1) command, (2) control, and (3) data and status transfer (see figure 3-1). Each of these functions is divided in circuit groups and operation of the circuit groups is described in the following paragraphs. Simplified and detailed diagrams for the interface PCA are contained in the HP 3000 Computer System diagrams manuals. The simplified diagrams are set number SD-131 and the detailed diagrams are set number DD-501. Functional block diagrams, timing diagrams and partial schematics contained in this manual are referred to by figure number.



2186-8

Figure 3-1. Overall System Block Diagram

3-13. COMMAND FUNCTION.

3-14. The interface PCA command function is performed by the circuits that receive and process the direct commands from the CPU and the decoded program commands from the multiplexer channel. In addition, an interrupt processing circuit and the I/O data gates are contained with the command function circuits. Operation of these circuits is described in the following paragraphs.

3-15. DIRECT COMMANDS. Direct commands are executed by the CPU and sent through the IOP and IOP bus to the interface PCA. The interface PCA receives the direct commands as three bits of the I/O command (IOCMD) word. These three bits are decoded as shown in table 3-1. To enable the command decoder and initiate the command operation requires a series of interlocking operations. Since decoding of each direct command is similar, only the decoding of the SIO command is described in the following paragraphs.

3-16. When the CPU executes an SIO command, the resulting CPU microcode assembles a device address word, microcode test I/O, and service out. This combined 12-bit word, is sent through the S-bus to the IOP control word registers (see figure 3-1). The IOP, in turn, sends the 12-bit word through the IOP bus to the interface PCA command function circuits (see figure 3-2). These circuits receive the 12-bit word as $\overline{\text{DEVNO}}$, 8 bits; $\overline{\text{IOCMD}}$, 3 bits; and $\overline{\text{SO}}$, 1 bit. (Refer to table 3-2 for a list of external signal names and abbreviations.) The $\overline{\text{DEVNO}}$ bits are compared with the device number relocateable jumper wire configuration. Coincidence of the compared signals provides a True Compare signal that is combined with the SO signal to enable the command decoder.

3-17. The command decoder then decodes the $\overline{\text{IOCMD}}$ bits as a TIO which provides a $\overline{\text{D STATUS STB}}$ command to the status transfer function on the interface PCA. This command strobes a 16-bit status word onto the IOP bus. At the same time, the status transfer function provides an acknowledge signal ($\overline{\text{ACK 2}}$) that results in sending an $\overline{\text{SI}}$ signal to the IOP. The IOP then terminates the $\overline{\text{SO}}$ signal that was enabling the command decoder and transfers the 16-bit status word to scratch pad 2 of the CPU.

3-18. In the CPU, the microcode checks bit 0 of the status word to determine if the device and interface PCA are ready to transfer data. If the status indicates that data transfer can proceed, the microcode again assembles a $\overline{\text{DEVNO}}$, $\overline{\text{SO}}$, and $\overline{\text{IOCMD}}$ command word that is sent through the IOP and IOP bus to the interface PCA. The interface PCA command decoder is again enabled. However, the decoded command is now $\overline{\text{SIO}}$.

3-19. When a direct strobe command such as $\overline{\text{D WRITE}}$ STB is decoded, the interface PCA performs the commanded function and returns an acknowledge signal to the CPU. The acknowledge signal must be received before the CPU can proceed to the next command. However, decoding a $\overline{\text{SIO}}$ command starts a routine that allows the IOP to control the interface PCA operations and frees the CPU to perform other operations. Decoding the $\overline{\text{SIO}}$ command direct sets the $\overline{\text{SIO}}$ OK, Service Request, Enable Service Request, and Service Request Buffer flip-flops. Setting the $\overline{\text{SIO}}$ OK FF causes status bit 0 to go to logic 0. This indicates, during any subsequent status checks, that an SIO operation is in progress and prevents the CPU from sending another $\overline{\text{SIO}}$ command until the first operation is complete.

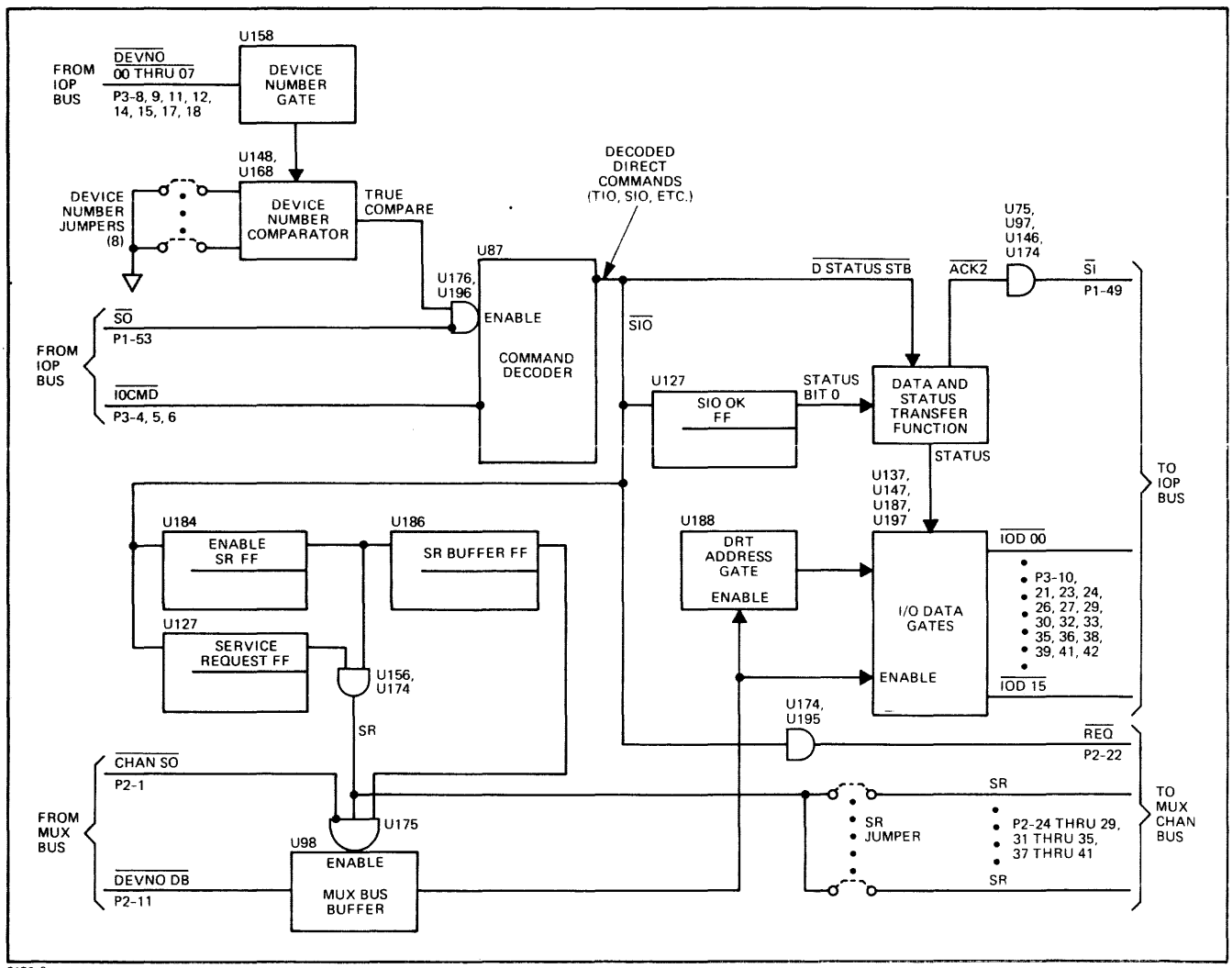
3-20. Setting the Service Request and Enable Service Request FF's causes an SR signal to be enabled. This SR signal can be jumper wired to any one of 16 output pins. The location of this jumper wire determines which device will be serviced by the multiplexer channel. Each interface PCA connected to the same multiplexer channel must have the jumper wire connected to a different output pin. The SR signal and a $\overline{\text{REQ}}$ signal are sent through to MUX bus to the multiplexer channel. At the multiplexer channel, the $\overline{\text{REQ}}$ signal causes an $\overline{\text{SI}}$ signal to be sent to the IOP. The IOP then terminates the $\overline{\text{SO}}$ signal to the interface PCA. Terminating the $\overline{\text{SO}}$ signal disables the interface PCA command decoder. Also, the $\overline{\text{REQ}}$ signal puts the multiplexer channel in the DRT fetch mode. The SR signal from the interface PCA causes the multiplexer channel to send a high Service Request signal to the IOP. This high Service Request signal starts the multiplexer channel DRT fetch operation and causes the IOP to issue a DATAPOLL IN signal on the IOP bus. The DATAPOLL IN signal propogates through each multiplexer channel until the highest priority multiplexer channel that is requesting service breaks the priority chain.

3-21. Once the multiplexer channel is being serviced by the IOP, the multiplexer channel overrides the SR signal from the interface PCA and at the same time sends a $\overline{\text{CHAN SO}}$ signal through the MUX bus to the interface PCA. This allows the set-side output from the Service Request Buffer FF, set when the $\overline{\text{SIO}}$ command was decoded, to enable eight of the MUX bus buffer gates on the interface PCA. Enabling these MUX bus buffer gates allows the $\overline{\text{DEVNO DB}}$ signal from the multiplexer channel to enable the DRT address and data gates so that the DRT address is available on the I/O data lines ($\overline{\text{IOD 00}}$ thru $\overline{\text{IOD 15}}$) to the IOP bus. The DRT address is developed by sending device number bits 00 through 07 ($\overline{\text{DEVNO 00}}$ thru $\overline{\text{DEVNO 07}}$) through the DRT address gates to I/O data lines $\overline{\text{IOD 06}}$ thru $\overline{\text{IOD 13}}$. This shift in bit positions multiplies the device number by four to provide the DRT address.

3-22. The multiplexer channel then sends an $\overline{\text{SI}}$ signal and a DRT fetch command to the IOP and the IOP accepts the DRT address from the IOP bus. The content of this address is a pointer to the first I/O program command. I/O program commands are then transferred one-at-a-time to the multiplexer channel where they are decoded and the decoded program commands are sent on to the interface PCA. Operation of the I/O program commands is described in the following paragraphs.

Table 3-1. Direct I/O Commands Decoding Table

IOCMD BITS			I/O DRIVER COMMAND	DECODED COMMAND	
00	01	02		NAME	ABBREVIATION
1	1	1	SIN	"Not" Set Interrupt	$\overline{\text{SET INT}}$
0	1	1	RESET INT	"Not" Reset Interrupt	$\overline{\text{RESET INT}}$
1	0	1	SIO	"Not" Start Input/Output	$\overline{\text{SIO}}$
0	0	1	SMSK	"Not" Set Mask	$\overline{\text{SMSK}}$
1	1	0	CIO	"Not" Direct Control Strobe	$\overline{\text{D CONT STB}}$
0	1	0	TIO	"Not" Direct Status Strobe	$\overline{\text{D STATUS STB}}$
1	0	0	WIO	"Not" Direct Write Strobe	$\overline{\text{D WRITE STB}}$
0	0	0	RIO	"Not" Direct Read Strobe	$\overline{\text{D READ STB}}$



2186-9

Figure 3-2. Direct Command Functional Logic Diagram

Table 3-2. Command Function External Signal List

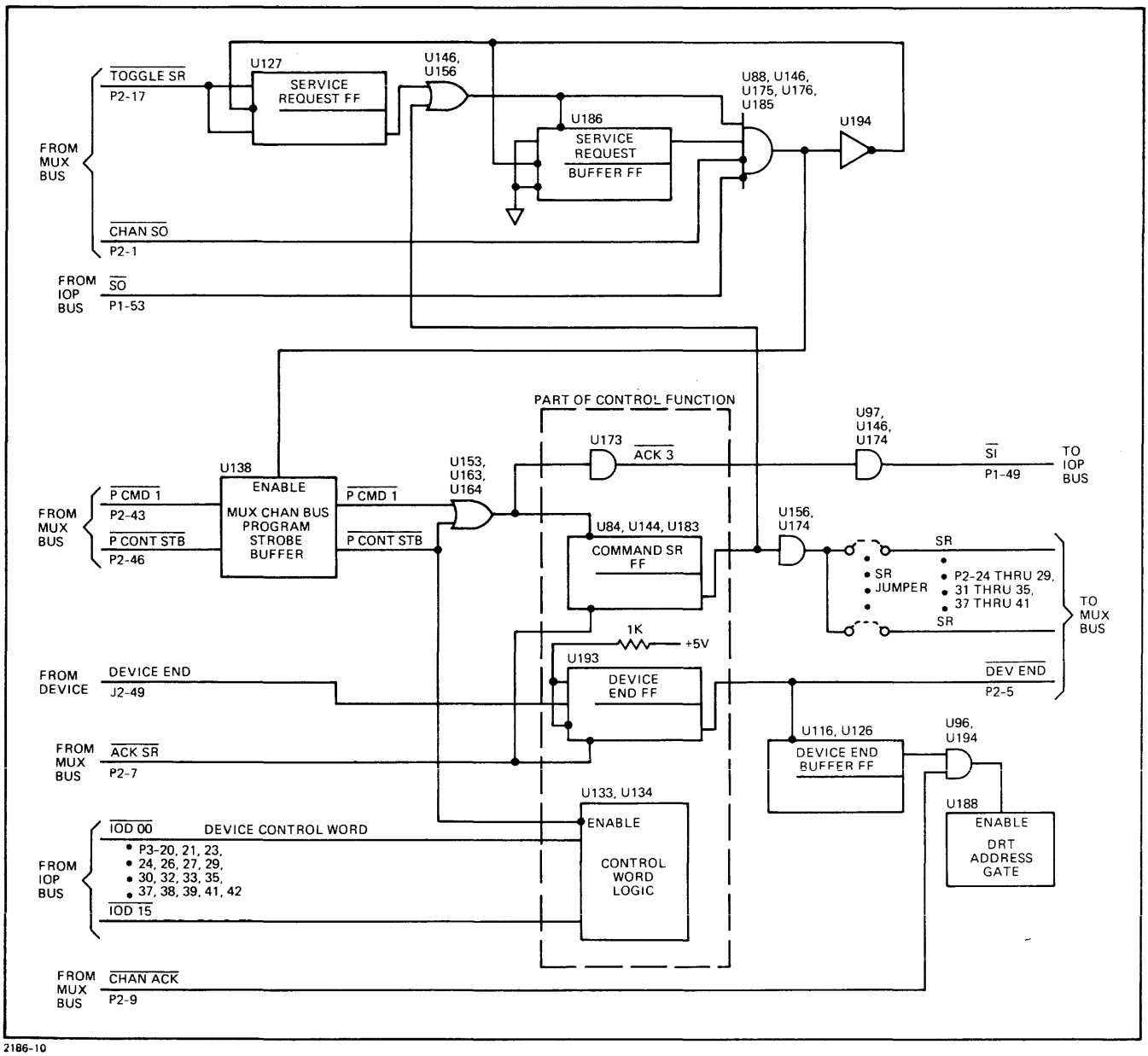
NO.	SIGNAL NAME	ABBREVIATION	BUS		CONNECTOR AND PIN NO.
			MUX	IOP	
1	"Not" Acknowledge Service Request	$\overline{\text{ACK SR}}$	X		P2-7
2	"Not" Channel Service Out	$\overline{\text{CHAN SO}}$	X		P2-1
3	"Not" Device Number 00 thru 07	$\overline{\text{DEVNO 00}}$ thru $\overline{\text{DEVNO 07}}$		X	P3-8,9,11,12, 14,15,17,18
4	"Not" Device End	$\overline{\text{DEV END}}$	X		P2-5
5	"Not" Device Number Data Base	$\overline{\text{DEVNO DB}}$	X		P2-11
6	"Not" End of Transfer	$\overline{\text{EOT}}$	X		P2-13
7	"Not" Input/Output Command Bits 00,01,02	$\overline{\text{IOCMD 00,01,02}}$		X	P3-4,5,6
8	"Not" Input/Output Data Bits 00 thru 15	$\overline{\text{IOD 00}}$ thru $\overline{\text{IOD 15}}$		X	P3-20,21,23, 24,26,27,29, 30,32,33,35, 36,38,39,41, 42
9	"Not" Input/Output Data Parity	$\overline{\text{IODPRTY}}$		X	P3-1
10	Input/Output Reset	IORESET		X	P1-11
11	"Not" Interrupt Acknowledge	$\overline{\text{INTACK}}$		X	P3-50
12	Interrupt Poll In	INTPOLL IN		X	P1-48
13	Interrupt Poll Out	INTPOLL OUT		X	P1-44
14	"Not" Interrupt Request	$\overline{\text{INTREQ}}$		X	P3-44
15	"Not" Jump Met	$\overline{\text{JMP MET}}$	X		P2-14
16	"Not" Module Control Unit Clocks	$\overline{\text{MCUCLKS}}$		X	P1-13
17	"Not" Channel Acknowledge	$\overline{\text{CHAN ACK}}$	X		P2-9
18	"Not" Program Command 1	$\overline{\text{P CMD 1}}$	X		P2-43
19	"Not" Program Control Strobe	$\overline{\text{P CONT STB}}$	X		P2-46
20	"Not" Program Read Strobe	$\overline{\text{P READ STB}}$	X		P2-50

Table 3-2. Command Function External Signal List (Continued)

NO.	SIGNAL NAME	ABBREVIATION	BUS		CONNECTOR AND PIN NO.
			MUX	IOP	
21	"Not" Program Status Strobe	$\overline{\text{P STATUS STB}}$	X		P2-45
22	"Not" Program Write Strobe	$\overline{\text{P WRITE STB}}$	X		P2-48
23	"Not" Read Next Word	$\overline{\text{RD NEXT WD}}$	X		P2-47
24	"Not" Request	$\overline{\text{REQ}}$	X		P2-22
25	"Not" Service In	$\overline{\text{SI}}$		X	P1-49
26	"Not" Start Input/Output Enable	$\overline{\text{SIO ENABLE}}$	X		P2-12
27	"Not" Service Out	$\overline{\text{SO}}$		X	P1-53
28	Service Request	SR	X		(1 of 16) P2-24 thru 29, 31 thru 35, 37 thru 41
29	"Not" Set Interrupt	$\overline{\text{SET INT}}$	X		P2-49
30	"Not" Set Jump	$\overline{\text{SET JMP}}$	X		P2-44
31	Toggle In Transfer	TOGGLE INXFER	X		P2-16
32	Toggle Out Transfer	TOGGLE OUTXFER	X		P2-18
33	Toggle Start Input/Output OK	TOGGLE SIO OK	X		P2-19
34	"Not" Toggle Service Request	$\overline{\text{TOGGLE SR}}$	X		P2-17
35	"Not" Transfer Error	$\overline{\text{XFER ERROR}}$	X		P2-21
36	RETURN	RET	X	X	P1-15, 16, 19, 20, 29, 30; P2-2, 4, 6, 8, 10, 15, 20, 23, 30, 36, 42; P3-3, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46, 49

3-23. PROGRAM COMMANDS. The program commands are contained in the first part of the I/O program doublewords. These commands are transferred through the IOP and the IOP bus to the multiplexer channel. The multiplexer channel decodes the program commands and sends decoded program strobe signals through the MUX bus to the interface PCA command function circuits. On the interface PCA, the program strobe signals cause the same read, write, status, and control operations as the equivalent direct strobe signals. However, program commands are only executed during an SIO routine. The program strobe signals sent to the interface PCA include P CONT STB, P READ STB, P WRITE STB, P STATUS STB, and SET INT (refer to table 3-2). Operations caused by these strobe signals are described in the following paragraphs.

3-24. Program Control Strobe Operation. The program control strobe operation takes a 16-bit device control word from the I/O program and loads the word into the interface PCA control word logic circuits (see figure 3-3). This operation begins after the Service Request and Service Request Buffer FF's were set by the SIO command and the interface PCA has sent the SR signal to the multiplexer channel. The multiplexer channel and IOP then sends CHAN SO and SO signals respectively to the interface PCA. These signals are combined to enable the MUX CHAN bus program strobe buffer (see figure 3-4).



2186-10

Figure 3-3. Program Control Strobe Command Functional Logic Diagram

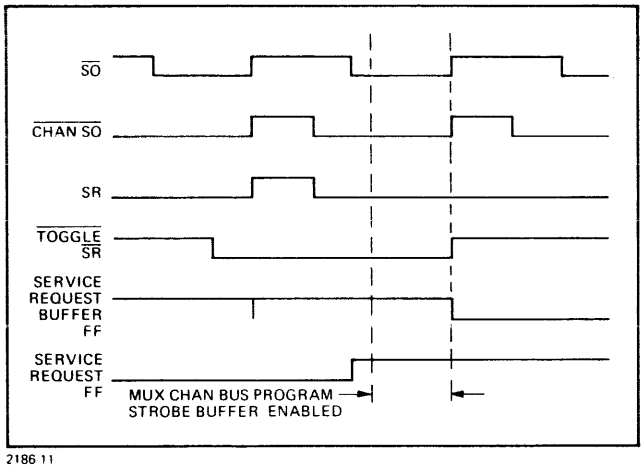


Figure 3-4. Program Strobe Timing

3-25. When the multiplexer channel subsequently decodes the first command (CIO) in the SIO routine, the first signal received from the multiplexer channel is TOGGLE SR. The TOGGLE SR signal causes the Service Request FF to clear and the Service Request Buffer FF is then cleared on the trailing edge of the CHAN SO or SO signal. The Service Request FF then remains clear until the next SIO initialization routine.

3-26. The MUX CHAN bus program strobe buffer remains enabled until the trailing edge of the CHAN SO or SO signal. While the buffer is enabled, the multiplexer channel decodes a P CMD 1 signal and sends this signal through the buffer on the interface PCA. This P CMD 1 signal is sent to the control function circuits where it provides an acknowledge signal (ACK 3) and direct sets the Command Service Request FF. The ACK 3 signal causes the interface PCA to send an SI signal to the IOP to acknowledge that the first part of the I/O program word was received and decoded. When the Command Service Request FF is set, an SR signal is sent to the multiplexer channel and the Service Request Buffer FF is direct set in preparation for the next program strobe signal.

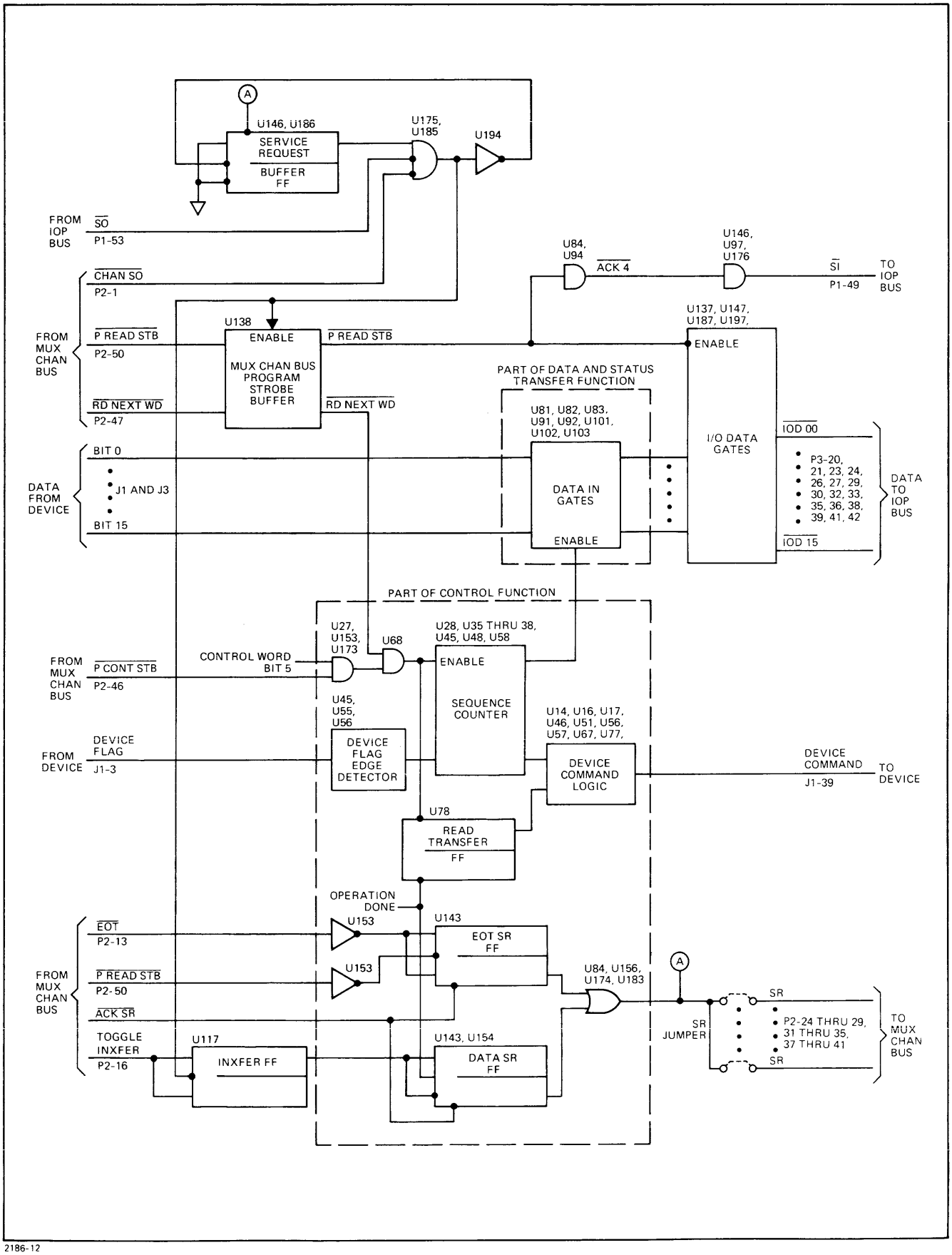
3-27. On receipt of the SR signal, the multiplexer channel initiates a fetch for the second part of the I/O program doubleword. Because the first part of the word was a CIO, the second part of the word is the 16-bit device control word. This device control word is sent to the IOP bus where it waits to be strobed onto the interface PCA. The multiplexer channel then sends an ACK SR signal to the interface PCA to direct clear the Command Service Request FF. Clearing the Command Service Request FF prepares the interface PCA for the next program command. The CHAN SO and SO signals again enable the MUX bus program strobe buffer and the multiplexer channel sends a P CONT STB signal through the buffer.

3-28. This P CONT STB signal is also sent to the control function circuits where it provides an ACK 3 signal, direct sets the Command Service Request FF, and enables the control word logic circuits. The ACK 3 signal results in sending an SI signal to the IOP and setting the Command Service Request FF sends an SR signal to the multiplexer channel. These signals indicate to the IOP and multiplexer channel that the P CONT STB signal was received. When the control word logic circuits are enabled, the 16-bit device control word is loaded from the IOP bus onto the interface PCA. The multiplexer channel then proceeds with a DRT fetch operation to fetch the first part of the next I/O program word. Use of the device control word is described with the control function.

3-29. Program Read Strobe Operation. The program read strobe operation strobes data from the device onto the IOP bus data lines. Data is received from the device in one 16-bit word or in two 8-bit bytes that are packed into one 16-bit word before transfer to the IOP bus. This operation begins when the multiplexer channel fetches and starts decoding a read command. The Service Request Buffer FF was set during the DRT fetch operation by the previous SR signal so that the SO and CHAN SO signals enable the MUX CHAN bus program strobe buffer (see figure 3-5). While the MUX CHAN bus program strobe buffer is enabled (see figure 3-4) the multiplexer channel sends a TOGGLE INXFER signal. The TOGGLE INXFER signal causes the INXFER FF to set and the flip-flop output is sent to the control function circuits to enable the Data Service Request FF. However, this flip-flop remains inactive until data is received from the device.

3-30. Data transfer from the device to the interface PCA is initiated if control word bit 5 is set during the previous control strobe operation or if a RD NEXT WD signal is decoded by the multiplexer channel and sent through the MUX CHAN bus program strobe buffer. In either case, the Read Transfer FF is direct set and the trailing edge of the same signal enables the sequence counter. The sequence counter steps the interface PCA through the data transfer operation that begins by sending a Device Command signal to the device. The device may have already received up to five bits of control information during the control strobe decoding operation. These control bits establish conditions for data transfer to or from the device, but the data transfer is initiated by the Device Command signal. Depending on the interface PCA connections, the Device Command signal can be sent until the device responds or for a preset pulse length.

3-31. When the device is ready to transfer data it responds to the Device Command signal by sending a Device Flag signal to the interface PCA. The sequence counter then enables the data input gates to receive data from the device. If two 8-bit bytes are being received, the sequence counter steps the circuits through a second Device Command and Device Flag operation so that the second byte is received. Finally, with the data loaded onto the interface PCA, the sequence counter returns to its starting point and an Operation Done signal is sent to the Read Transfer FF and Data Service Request FF. The Operation Done signal direct sets the Read Data Transfer FF to disable the device command logic until the sequence counter is again enabled. A



2186-12

Figure 3-5. Program Read Strobe Functional Logic Diagram

combination of the Operation Done signal and the output from the INXFER FF causes the Data Service Request FF to set.

3-32. Setting the Data Service Request FF sends an SR signal to direct set the Service Request Buffer FF and also through the MUX CHAN bus to the multiplexer channel. The SR signal indicates that data is ready to transfer from the interface PCA to the IOP bus so the multiplexer channel sends an ACK SR signal that direct clears the Data Service Request FF. Data transfer then continues as the multiplexer channel decodes a P READ STB signal and sends the signal through the MUX CHAN bus program strobe buffer. The P READ STB enables the I/O data gates to strobe the 16-bit data word onto the IOP bus data lines and at the same time provides an ACK 4 signal. This ACK 4 signal results in a SI signal being sent to the IOP to indicate that data is on the IOP bus and the interface PCA is ready for the next program strobe operation. After a block of data is transferred and an END order is programmed, the multiplexer channel sends a EOT signal to the interface PCA. This EOT signal is sent to the set-side input of the EOT SR FF and the flip-flop is then clocked by the last P READ STB signal. Setting the EOT SR FF also sends an SR signal to direct set the Data Service Request FF and through the MUX CHAN bus to multiplexer channel. This SR signal indicates that data transfer is complete and the multiplexer channel can fetch the next command.

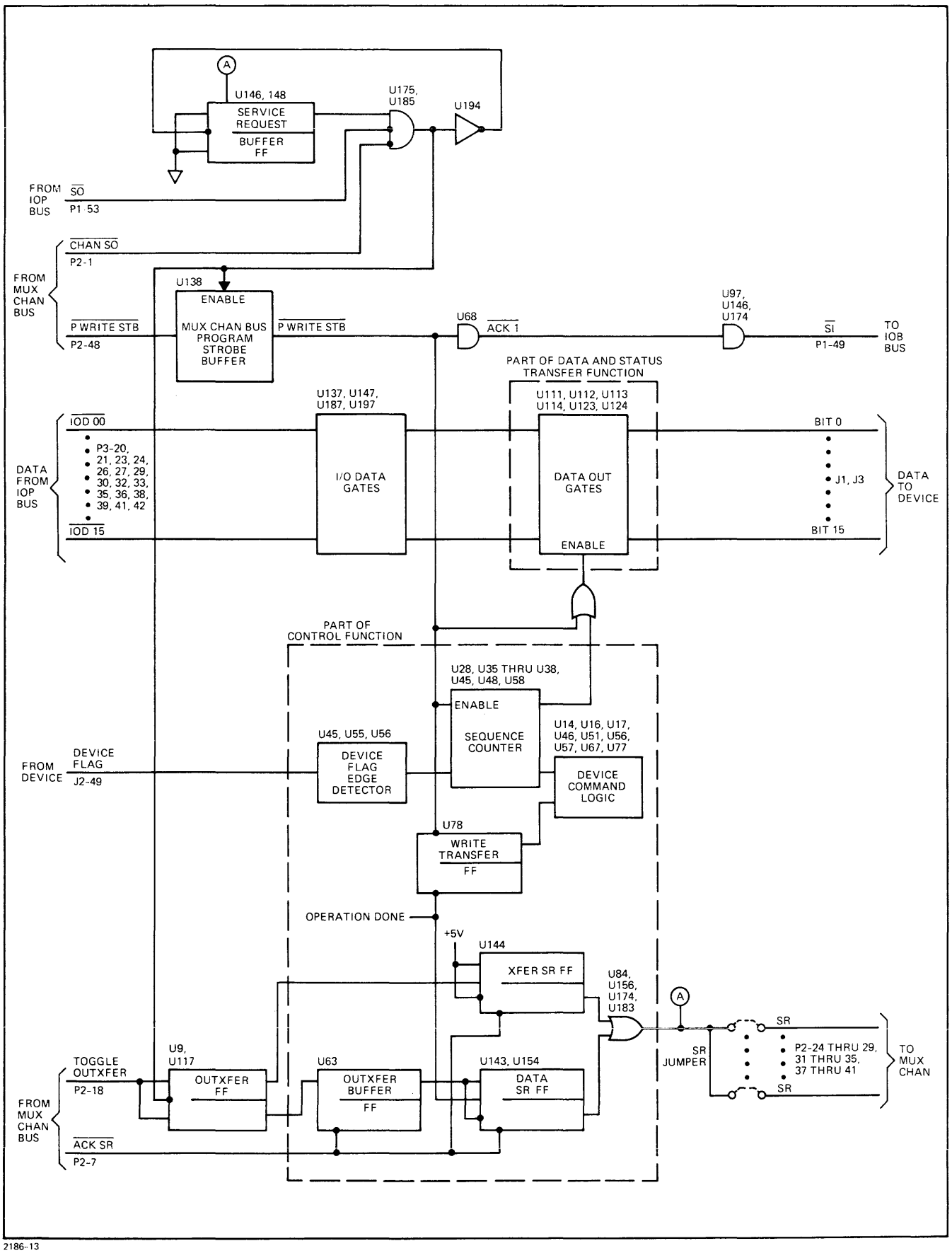
3-33. Program Write Strobe Operation. The program write strobe operation transfers data from the IOP bus, through the interface PCA data gates, and to the device. Data is sent to the device in one 16-bit word or in two 8-bit bytes if unpacking (byte transfer) is enabled. This operation begins when the multiplexer channel fetches and starts decoding a write command. The Service Request Buffer FF was set during the DRT fetch operation by the previous SR signal so that the SO and CHAN SO signals enable the MUX CHAN bus program strobe buffer (see figure 3-6). While the MUX CHAN bus program strobe buffer is enabled (see figure 3-4) the multiplexer channel sends a TOGGLE OUTXFER signal that sets the OUTXFER FF. Output from the OUTXFER FF is sent to the control function circuits to enable the OUTXFER Buffer FF and the XFER SR FF. The OUTXFER Buffer FF keeps a signal on the input to the Data Service Request FF in case the OUTXFER FF is cleared before data transfer is complete. Setting the XFER SR FF sends an SR signal to direct set the Service Request Buffer FF. The SR signal is also sent to the multiplexer channel to indicate that the interface PCA is ready to receive data from the IOP bus.

3-34. The I/O data gates that receive the 16-bit data word from the IOP bus are permanently enabled so that the data is sent through the gates to the data output gates in the data and status transfer function. This data is strobed onto the interface PCA output lines whenever a P WRITE STB is decoded by the multiplexer channel and sent through the MUX CHAN bus program strobe buffer. The same P WRITE STB signal provides an ACK 1 signal, enables the sequence counter, and direct sets the Write Transfer FF. An SI signal results from the ACK 1 signal and is sent to the IOP to indicate that data is being taken from the IOP bus. Enabling the sequence counter steps the interface PCA through a data transfer operation that sends data to the device. This operation begins by sending a Device Command signal to the device. The device may have already received up to five bits of control information during the control strobe decoding operation. These control bits establish conditions for data transfer to or from the device, but the data transfer is initiated by the Device Command signal. Depending on the interface PCA connections, the Device Command signal can be sent until the device responds or for a preset pulse length.

3-35. When the device is ready to receive data it responds to the Device Command by sending a Device Flag signal to the interface PCA. Data is already on the interface PCA output lines and can be accepted by the device as one 16-bit word or can be unpacked under control of the sequence counter and made available to the device as two 8-bit bytes. In eigher case the sequence counter returns to its starting point and an Operation Done signal is sent to the Write Transfer FF and the Data Service Request FF. The Write Transfer FF is direct set to disable the device command logic until the sequence counter is again enabled. A combination of the Operation Done signal and the output from the OUTXFER Buffer FF causes the Data Service Request FF to set. Setting the Data Service Request FF sends an SR signal to direct set the Service Request Buffer FF and also go through the MUX CHAN bus to the multiplexer channel. At the multiplexer channel the SR signal indicates the interface PCA is ready for the next program strobe operation.

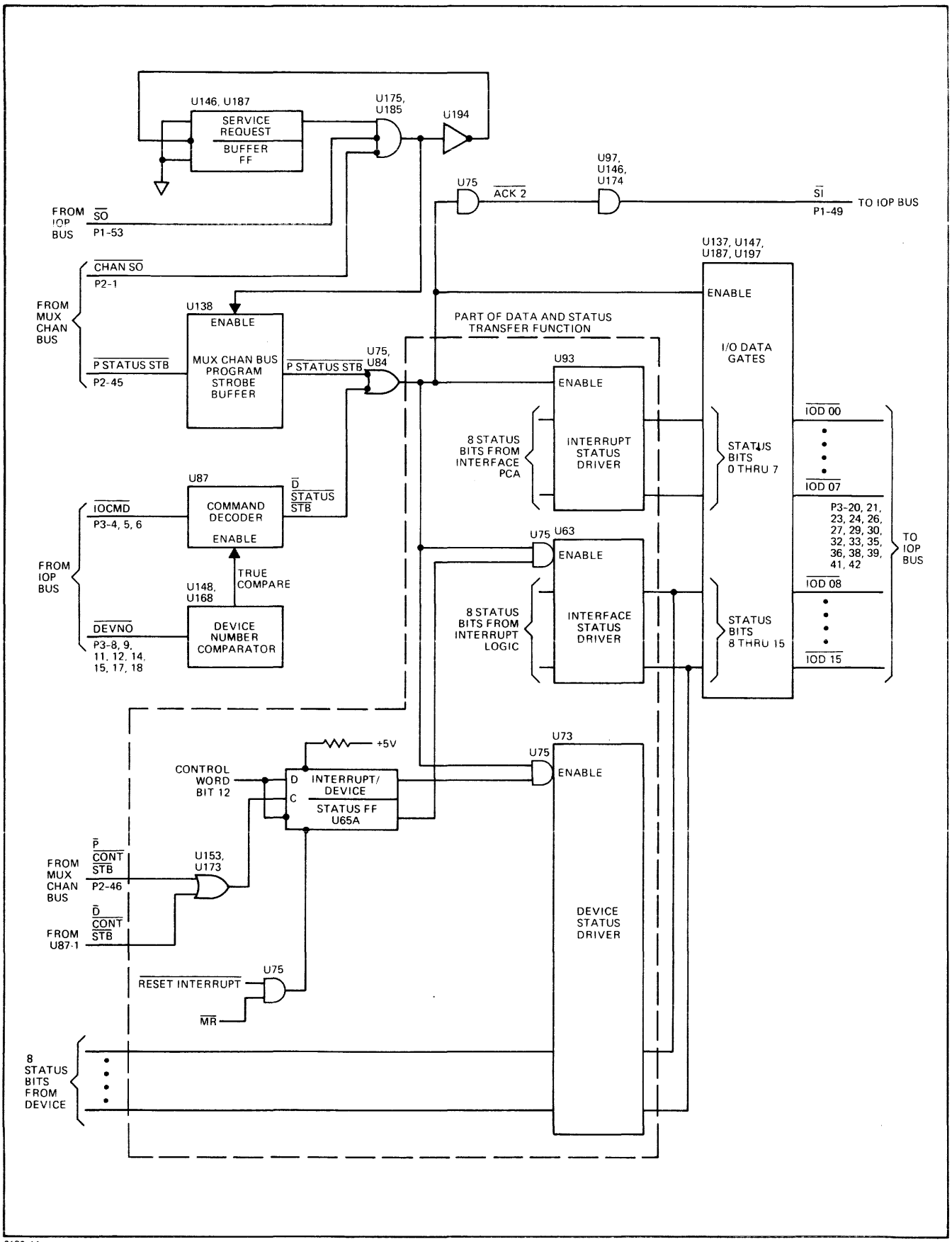
3-36. Direct/Program Status Strobe Operations. The direct/program status strobe operation transfers a 16-bit status word from the interface PCA to the IOP bus data lines. This status word is divided into two 8-bit bytes with bits 0 through 7 always indicating the interface PCA status such as sequence counter count and condition of the Device Flag signal. Bits 8 through 15 indicate either interrupt request status or device status depending on the condition of the Interrupt/Device Status FF (see figure 3-7). With the flip-flop cleared, the interrupt status register is enabled by either a direct status strobe or a program status strobe. These interrupt status bits indicate eight possible conditions that can request an interrupt. To read device status the flip-flop is set by control word bit 12 during the previous control word strobe operation. The eight device bits are routed directly from the external device to the interface PCA device status driver.

3-37. The status strobe operation begins when the interface PCA receives either a direct TIO command that the command decoder converts to a D STATUS STB signal or, during I/O program execution, the



2186-13

Figure 3-6. Program Write Strobe Functional Logic Diagram



2186-14

Figure 3-7. Direct/Program Status Strobe Functional Logic Diagram

multiplexer channel sends a P STATUS STB signal. Figure 3-4 shows the signals that enable the MUX CHAN bus program strobe buffer. On the interface PCA, the D STATUS STB and P STATUS STB signals provide the same operations. The active signal enables either the interrupt or device status register, enables the interface PCA status register and I/O data gates, and provides an ACK 2 signal. This ACK 2 signal results in an SI signal that indicates to the IOP that the 16-bit status word is available on the IOP bus.

3-38. From the IOP bus data lines, the status word is sent to one of three possible storage locations. The storage location is determined by the instruction used to fetch the status word from the interface PCA. A direct TIO instruction pushes the status word to the top of the stack. A microcode TIO instruction comes from assembling direct instructions such as WIO and RIO and result in sending the status word to scratch pad 2 in the CPU. I/O program orders, such as SENSE and END, are always the first part of an I/O program doubleword and store the status word in the second part of the same doubleword.

3-39. "Not" Set Interrupt. There are two "not" Set Interrupt signals on the interface PCA. One of the "not" Set Interrupt signals is provided when a direct SIN instruction causes the CPU to issue a direct I/O command to the interface PCA. This direct SIN command is decoded by the interface PCA command decoder as a "not" Set Interrupt signal (see figure 3-8). The other "not" Set Interrupt signal is provided during programmed I/O when the multiplexer channel receives and begins decoding an END order or INTERRUPT order. Decoding an END order results first in sending a TOGGLE SIO OK signal through the MUX CHAN bus to the interface PCA. This signal clears the SIO OK FF and sets status bit 0 to logic 1. On subsequent status checks, a logic 1 at status bit 0 indicates that the interface PCA is not performing an SIO routine. Bit 4 of the END order is then checked. If bit 4 is a logic 1 or when the multiplexer channel is decoding INTERRUPT order, it sends a SET INT signal to the interface PCA. On the interface PCA, when the MUX CHAN bus program strobe buffer is enabled, the SET INT signal provides the other "not" Set Interrupt signal. Whether the "not" Set Interrupt signal results from a direct instruction or programmed order, the resultant signal is sent to the control function interrupt logic where it sets the Set Interrupt FF. The Set Interrupt FF provides an output to bit 13 of the interrupt status word and also sends an Interrupt Request signal to the interrupt processing circuit of the command function. When the interrupt is subsequently processed, status bit 13 indicates the cause for the interrupt request. The interrupt processing circuit is described starting in the following paragraph.

3-40. INTERRUPT PROCESSING. The interrupt processing circuits of the command function accept Interrupt Request signals from the control function interrupt logic circuits. When conditions are met for an interrupt, the interrupt processing circuits store the Interrupt Request signal and enable the interface PCA to request service from the CPU.

3-41. The interrupt processing circuit contains three flip-flops and associated logic circuits. These flip-flops are called Interrupt Mask, Interrupt Latch, and Interrupt Active (see figure 3-8). Setting the Interrupt Mask FF qualifies one condition for setting the Interrupt Latch FF. If the Interrupt Mask FF is cleared, interrupts cannot occur. The Interrupt Mask FF input is connected to a jumper wire at the interface PCA side of the data out (to device) gates. These gates are always enabled, but do provide isolation between the interface PCA and IOP bus data lines. Connections are provided to connect the jumper wire from the Interrupt Mask FF input to any of the 16 I/O data lines, to +5 Vdc, or to a signal return. A +5 Vdc connection or a connection to any I/O data line that is high allows the Interrupt Mask FF to set on the positive-going edge of a clock input signal. The flip-flop is also set by an MR signal that occurs following an I/O Reset signal from the IOP or following a direct or program control strobe with control word bit 0 at logic 1. Clearing the Interrupt Mask FF requires an input connection to signal return or to any I/O data line that is low when the clock input occurs. Connecting the Interrupt Mask FF to an I/O data line allows the I/O driver set mask (SMSK) instruction to control the state of the flip-flop. When the SMSK instruction is executed, the 16-bit word on top of the stack is output over the IOP bus to the I/O data out gates. The resulting SET MASK command is decoded and combined with the SO signal to clock the Interrupt Mask FF. This allows the Interrupt Mask FF to be set and/or cleared by any selected bit of the 16-bit word on the IOP bus.

3-42. The set-side output from the Interrupt Mask FF is combined with the Interrupt Request signal and the resulting signal is applied to the input of the Interrupt Latch FF. Output from the Interrupt Latch FF follows the input as long as the clock input remains high. When the clock input goes low, the flip-flop is latched in the state indicated by the last input signal level. A high input signal, when the clock input goes low, latches the flip-flop in the set state. The clock input to the Interrupt Latch FF is provided by the INTPOLL IN signal from the IOP bus. As long as the flip-flop input remains low, the INTPOLL IN signal propogates through one gate on the interface PCA and is returned to the IOP bus as the INTPOLL OUT signal. If the Interrupt Latch FF is latched in the set state, the clear-side output inhibits the INTPOLL OUT signal. This prevents an interrupt from any lower priority device until the Interrupt Latch FF is cleared.

3-43. Output from the Interrupt Mask FF and the Interrupt Request signal are also combined and inverted and then sent to the IOP as the INTREQ signal. The INTREQ signal can be sent by any interface PCA and the IOP responds by sending the INTPOLL IN signal to

all interface PCA's. On the interface PCA that is requesting an interrupt, the INTPOLL IN signal latches the Interrupt Latch FF and, after a short delay, enables the device number gate. The device number gate output is sent through IOP bus lines DEVNO 00 thru DEVNO 07 to the IOP. The delayed INTPOLL IN signal is also inverted to direct set the Interrupt Active FF. Setting the Interrupt Active FF sends an INT ACK signal to the IOP to indicate that the device requesting an interrupt has its address available on the IOP bus lines.

3-44. The IOP drops the INTPOLL IN signal upon receipt of the INT ACK signal and, the clear-side output from the Interrupt Active FF is sent to the control function to inhibit the Interrupt Request signal. The Interrupt Latch FF is then unlatched; however, the Interrupt Active FF remains set until the interrupt request is serviced and the IOP sends an IORESET signal to the interface PCA.

3-45. I/O DATA GATES. The I/O data gate circuits of the command logic provide input and output isolation for the data, status, and control signals routed between the interface PCA and IOP. In addition, these circuits contain the device number gates, device number comparator, DRT address gate, device number relocateable jumpers, and parity jumper (see figure 3-9). The data-in gates receive data or status bits from the interface PCA data and status transfer function and, when the gates are enabled, route the data or status bits to IOP bus data lines IOD 00 thru IOD 15. These data-in gates are enabled to route the data or status bits by a D READ STB, D STATUS STB, P READ STB, or D STATUS STB signal. The DRT address is also routed through the data-in gates to the IOP bus and the gates are then enabled by a DEVNO DB or DEV END signal. Operation of the DRT address gate is described in the following paragraphs.

3-46. The data-out gates receive data and control word bits from IOP bus data lines IOD 00 thru IOD 15 and, since the gates are always enabled, route the data and control word bits directly to the data and status transfer or control function. During execution of a direct SMSK instruction the data-out gates also receive a 16-bit word from the top of the stack. This word is used to control the Interrupt Mask FF in the interrupt processing circuits. When this feature is being used the interrupt mask jumper wire is connected to one of the input bit signal lines.

3-47. The device number gates and comparator, DRT address gate, and device number jumpers all work together to provide addresses to the IOP. First, the device number is determined by the connection of the eight relocateable device number jumper wires. This jumper wire configuration provides an eight bit word to the device number gates and comparator and the DRT address gate. When data or status is sent from the interface PCA to the IOP bus, an interrupt must be requested to enable the device number gates. Enabling the device number gates sends the eight bit device number through IOP bus device number lines DEVNO 00 thru DEVNO 07 to the IOP. Output from the same gate that enables the device number gate is normally low when an interrupt is not being requested. This low output enables the device number comparator. If a direct command (IOCMD) is sent to the interface PCA, it is accompanied by a device number (DEVNO) and an SO signal. The DEVNO is sent through IOP bus lines DEVNO 00 thru DEVNO 07 and through the comparator gates to the comparator circuits. This DEVNO is compared to the device number jumper wire configuration and, if all eight bits match, the comparator provides a True Compare output signal. The True Compare signal is combined with the SO signal to enable the command decoder and allow direct commands to control the interface PCA.

3-48. The device reference table (DRT) address gate is enabled while the multiplexer channel does a DRT fetch operation. This occurs when an SIO instruction is decoded and the multiplexer channel sends DEVNO DB signal to the interface PCA. The multiplexer channel also does a DRT fetch operation after receiving a DEV END signal from the interface PCA. A DEV END signal terminates the I/O program command that is in operation and the multiplexer channel is advanced to fetch the next I/O program instruction. The multiplexer channel then sends a CHAN ACK signal to the interface PCA. This CHAN ACK signal is combined with the DEV END signal to enable the DRT address gate. When the DRT address gate is enabled, the device number jumper wire configuration is reflected through the gate to the data-in lines. Bits 00 through 07 of the device number are connected to bits 06 through 13 of the data-in lines. This connection provides a shift that multiplies the device number by four to obtain the DRT address. The same signal that enables the DRT address gate also enables the data-in gates so the DRT address is available to the IOP on IOP bus data lines IOD 06 thru IOD 13.

3-49. CONTROL FUNCTION.

3-50. The interface PCA control function contains circuits that perform the control operations for the interface PCA and external device and provide requests for service and interrupts to the HP 3000 Computer System. These circuits consist of the device control logic, device end logic, device service request logic, and interrupt request logic. Operation of these circuits is described in the following paragraphs. Simplified and detailed diagrams for the interface PCA are contained in the HP 3000 Computer System diagrams manuals. The simplified diagrams are set number SD-131 and the detailed diagrams are set number DD-501. Functional block diagrams, timing diagrams, and partial schematics contained in this manual are referenced by figure number.

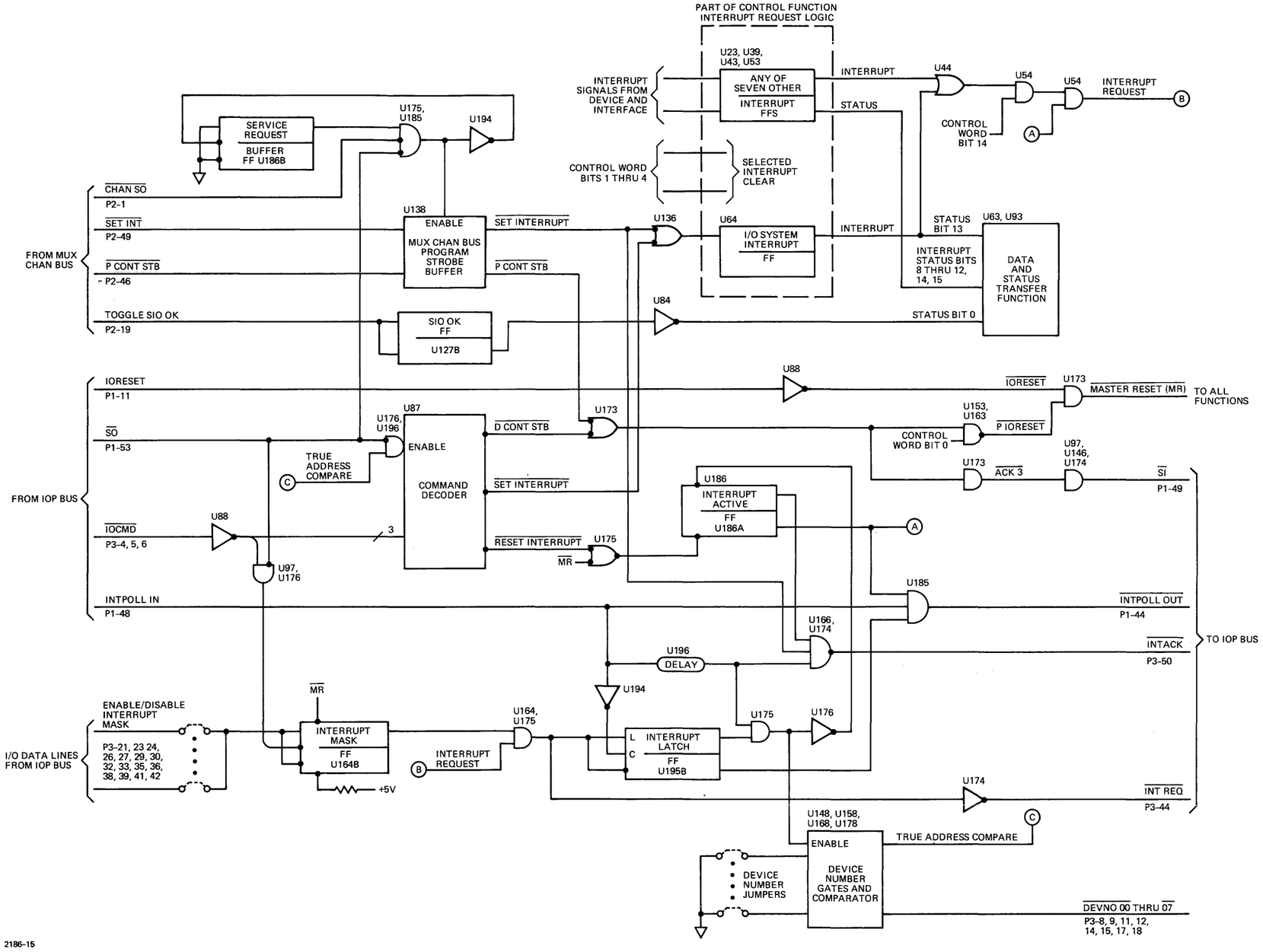


Figure 3-8. Interrupt Processing Functional Logic Diagram

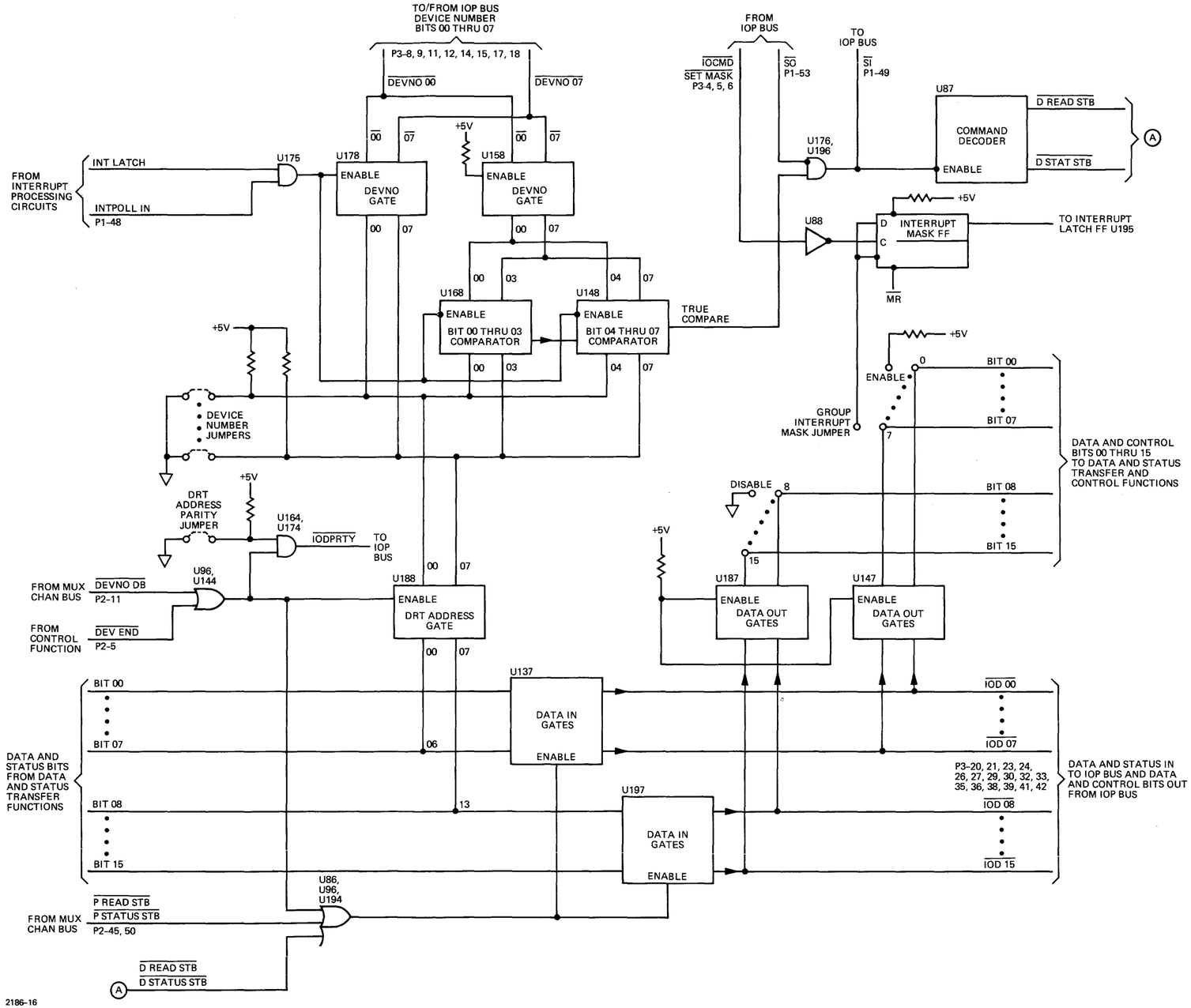


Figure 3-9. Command Function I/O Data Gates Functional Logic Diagram

3-51. DEVICE CONTROL LOGIC. The device control logic consists of the data transfer sequence counter, device command logic, device flag edge detector, and the device control word logic (see figure 3-10). These circuits transfer commands for data transfer from the HP 3000 Computer System to an external device and can also transfer up to five discrete control bits to an external device. The transfer of data to or from a device is controlled by two signals. These two signals are the Device Command signal that is sent to the device to request an I/O operation and the Device Flag siganl that is received from the device to acknowledge the start and/or finish of an I/O operation. Two modes of operation are possible with these two signals. The two modes are called response mode and pulse mode. In the response mode the Device Flag signal responds during the Device Command duty cycle and in the pulse mode the Device Flag response is independent of the Device Command duty cycle (see figure 3-11). These modes are selectable by jumper wires in the mating connector cover of connector J2.

3-52. The data transfer sequence counter (counter) receives data transfer strobes from the command function that initiate the start of data transfer. The counter then advances through four counts depending on the necessary qualifiers being present before each advancement. Output from the counter initiates the Device Command at the proper counts and also provides enabling signals to the data gates in the data and status transfer function. Qualifiers necessary to advance the counter are provided by the device flag edge detector circuit. The counter advances on either the leading or trailing edge of the Device Flag signal, depending on the jumper wire configuration in the mating connector cover of connector J2. Figure 3-13 shows the sequence of operations for the counter and qualifiers necessary to advance the count and transfer data.

3-53. Sequence Counter Qualifier Logic. The sequence counter qualifier logic provides conditions which must be satisfied before the sequence counter can advance to the next count. This qualifier logic contains the following three flip-flops with their associated gating circuits: Qualifier Q0, Read Transfer, and Write Transfer. Setting the Qualifier Q0 FF starts the sequence counter. Since the flip-flop set side input is connected to a positive voltage, the flip-flop sets whenever a positive-going signal is detected at the clock input (see figure 3-12). The clock input goes positive, setting the Qualifier Q0 FF, at the trailing edge of any of the following signals:

a. $\overline{\text{D WRITE STB}}$.

b. $\overline{\text{P WRITE STB}}$.

c. $\overline{\text{RD NEXT WD}}$.

d. $\overline{\text{P CONT STB}}$ (with control word bit 5).

e. $\overline{\text{D CONT STB}}$ (with control word bit 5).

3-54. The Qualifier Q0 FF set-side output enables the sequence counter to advance from count 0 to count 1. The Count 0 signal then goes low and direct clears the Qualifier Q0 FF. This flip-flop can also be cleared by the "not" Master Reset ($\overline{\text{MR}}$) signal that results from an HP 3000 Computer System reset.

3-55. The device sends a Device Flag signal to indicate the beginning of an I/O operation and terminates the signal to indicate that the I/O operation is complete (see figure 3-11). Leading and trailing edges of this Device Flag signal are used to enable Qualifiers Q1, Q2, and Q3 which, in-turn, advance the sequence counter through counts 2, 3, and return to 0 respectively. Qualifier Q1 indicates the beginning of an I/O operation and occurs on the leading edge of the Device Flag signal when jumper wire J2W2 is removed from the mating connector of J2, or on the trailing edge of the Device Flag signal when J2W2 is installed (see figure 3-14).

3-56. Qualifier Q2 indicates the end of an I/O operation and occurs on the leading edge of the Device Flag signal when jumper wire J2W6 is installed on the mating connector of J2, or on the trailing edge of the Device Flag signal when J2W6 is removed (see figure 3-15). In some applications it is desirable to have Qualifiers Q1 and Q2 occur on the same edge of the Device Flag signal. When only the leading edge is used, jumper wire J2W6 is installed and J2W2 is removed. For using the trailing edge only,

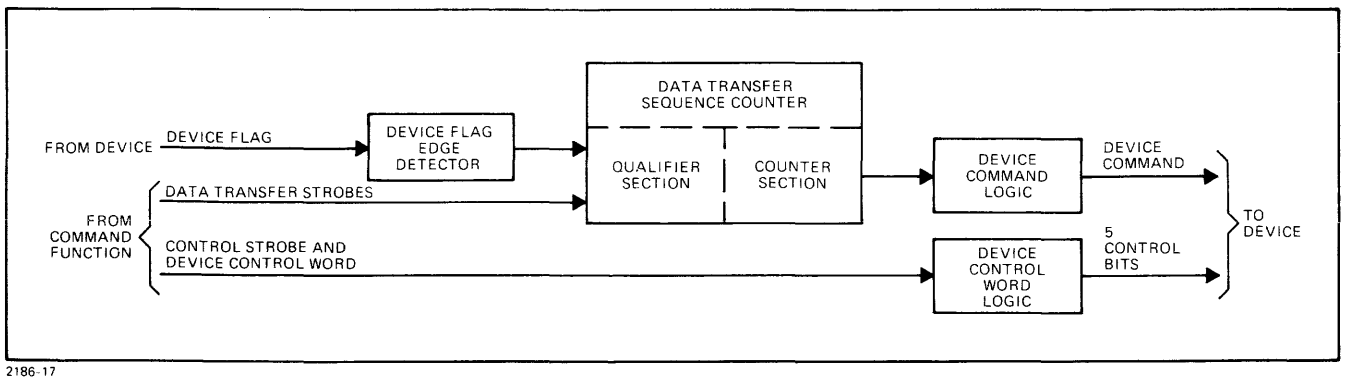


Figure 3-10. Device Control Logic Block Diagram

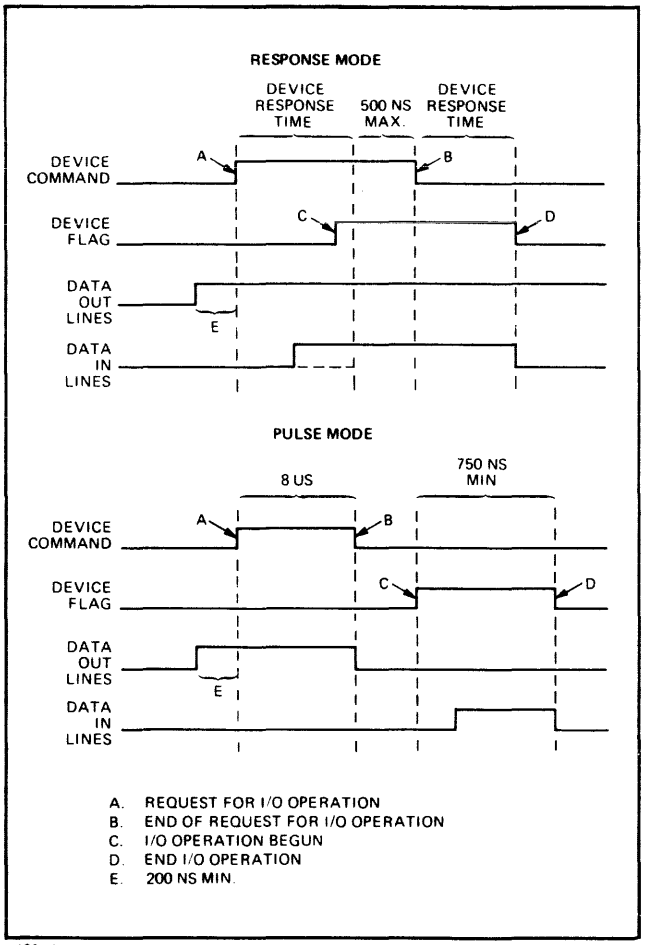


Figure 3-11. Response Mode and Pulse Mode Timing Diagrams

jumper wire J2W2 is installed and J2W6 is removed. In either case, the output of gate U56A is always high; therefore, Qualifier Q2 is always high. Qualifier Q2 will also go high if the device sends a Device End signal. This allows the sequence counter to continue counting through to count 0 in the event of a device end.

3-57. Qualifier Q3 is used to advance the sequence counter during a byte transfer. Byte transfer is enabled by control word bit 13 when data is being transferred to or from the device in two eight bit words. To transfer each byte (eight bit word) the Device Command signal is sent to the device and the device returns a Device Flag signal. The second Device Flag signal indicates the beginning of the operation to transfer the second byte and, depending on whether jumper wire J2W2 is removed or installed in the mating connector of J2, Qualifier Q3 occurs on leading or trailing edge of the signal (see figure 3-14). When byte transfer is not enabled, the low Byte Transfer signal keeps Qualifier Q3 always high. Qualifier Q3 will also go high if the device sends a Device End signal. This allows the sequence counter to continue counting through to count 0 in the event of a device end.

3-58. When Qualifier Q3 enables the sequence counter to return to count 0, the Count 0 signal is combined with the Qualifier Q2 signal to provide an Operation Done signal to the interface PCA logic (see figure 3-15). The Qualifier Q2 signal is always high when gate U56A is enabled or remains high until Device Flag 2 FF is cleared. During a 16-bit data transfer to or from a device, Device Flag 2 FF is cleared by the Count 1 signal from the sequence counter. This allows the flip-flop to be set by the Device Flag signal and remain set until the start of the next counting sequence. During a byte transfer when the Device Flag

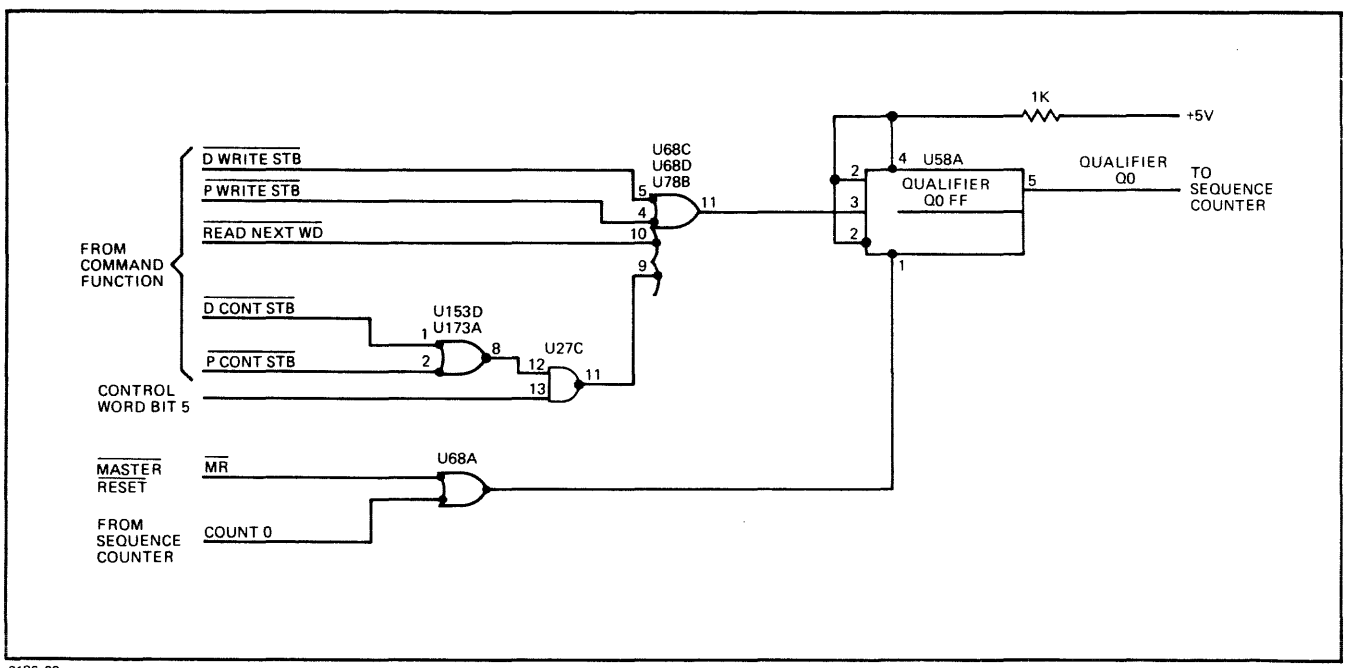
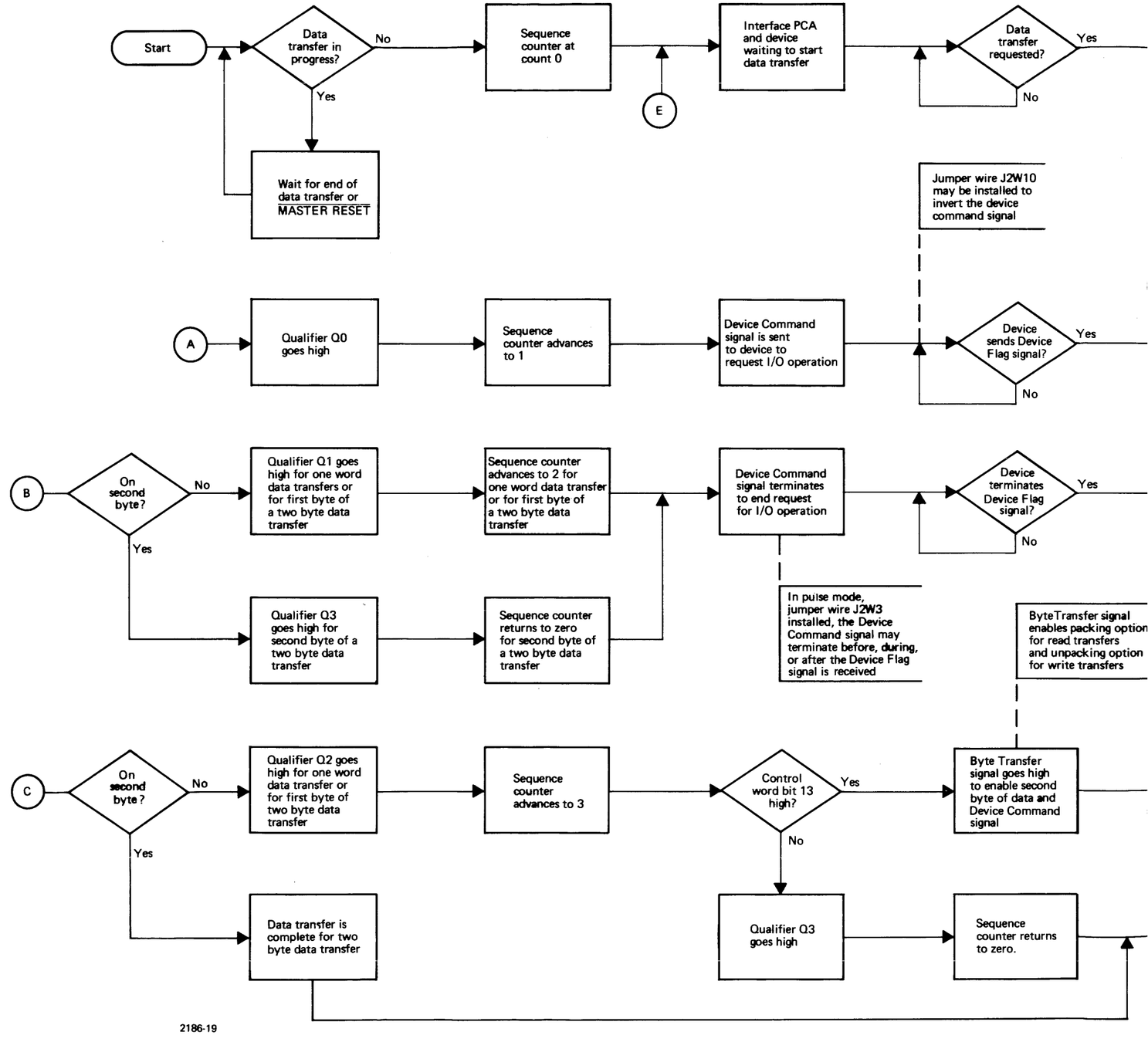


Figure 3-12. Qualifier Q0 Simplified Diagram


Start
Data transfer in progress?
No
Yes
Wait for end of data transfer or MASTER RESET
Sequence counter at count 0
E
Interface PCA and device waiting to start data transfer
Data transfer requested?
Yes
No
Jumper wire J2W10 may be installed to invert the device command signal
A
Qualifier Q0 goes high
Sequence counter advances to 1
Device Command signal is sent to device to request I/O operation
Device sends Device Flag signal?
Yes
No
B
On second byte?
No
Yes
Qualifier Q1 goes high for one word data transfers or for first byte of a two byte data transfer
Sequence counter advances to 2 for one word data transfer or for first byte of a two byte data transfer
Device Command signal terminates to end request for I/O operation
Device terminates Device Flag signal?
Yes
No
In pulse mode, jumper wire J2W3 installed, the Device Command signal may terminate before, during, or after the Device Flag signal is received
Qualifier Q3 goes high for second byte of a two byte data transfer
Sequence counter returns to zero for second byte of a two byte data transfer
ByteTransfer signal enables packing option for read transfers and unpacking option for write transfers
C
On second byte?
No
Yes
Qualifier Q2 goes high for one word data transfer or for first byte of two byte data transfer
Sequence counter advances to 3
Control word bit 13 high?
Yes
No
Byte Transfer signal goes high to enable second byte of data and Device Command signal
Data transfer is complete for two byte data transfer
Qualifier Q3 goes high
Sequence counter returns to zero.

2186-19

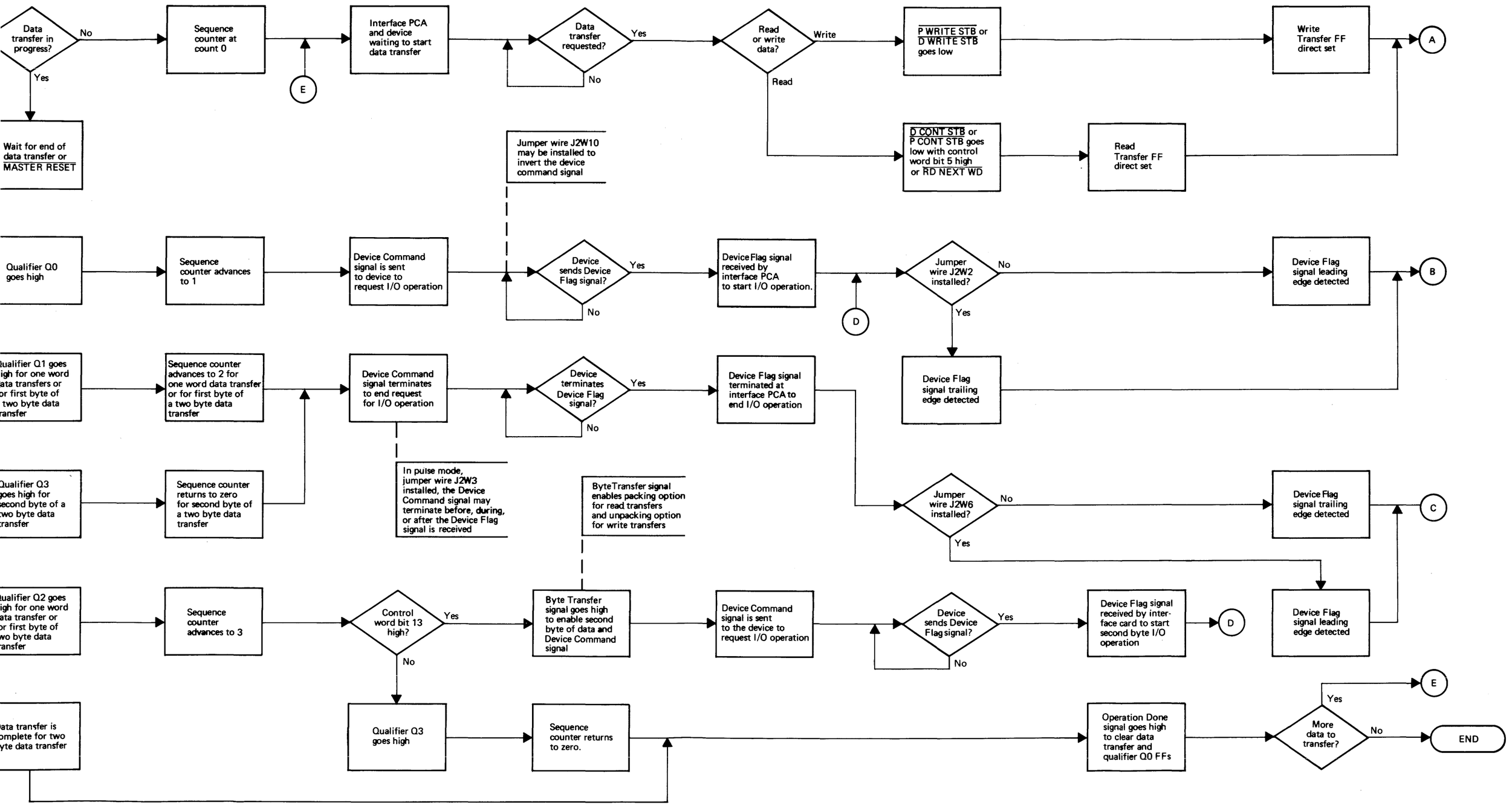
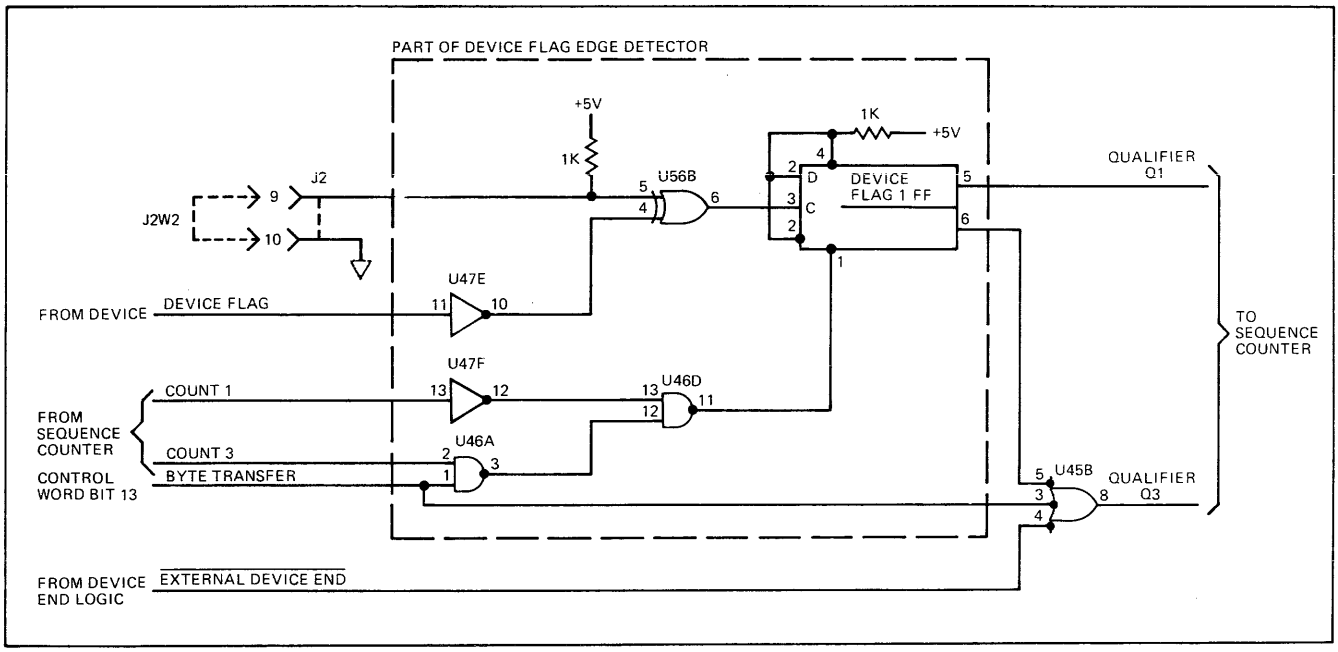
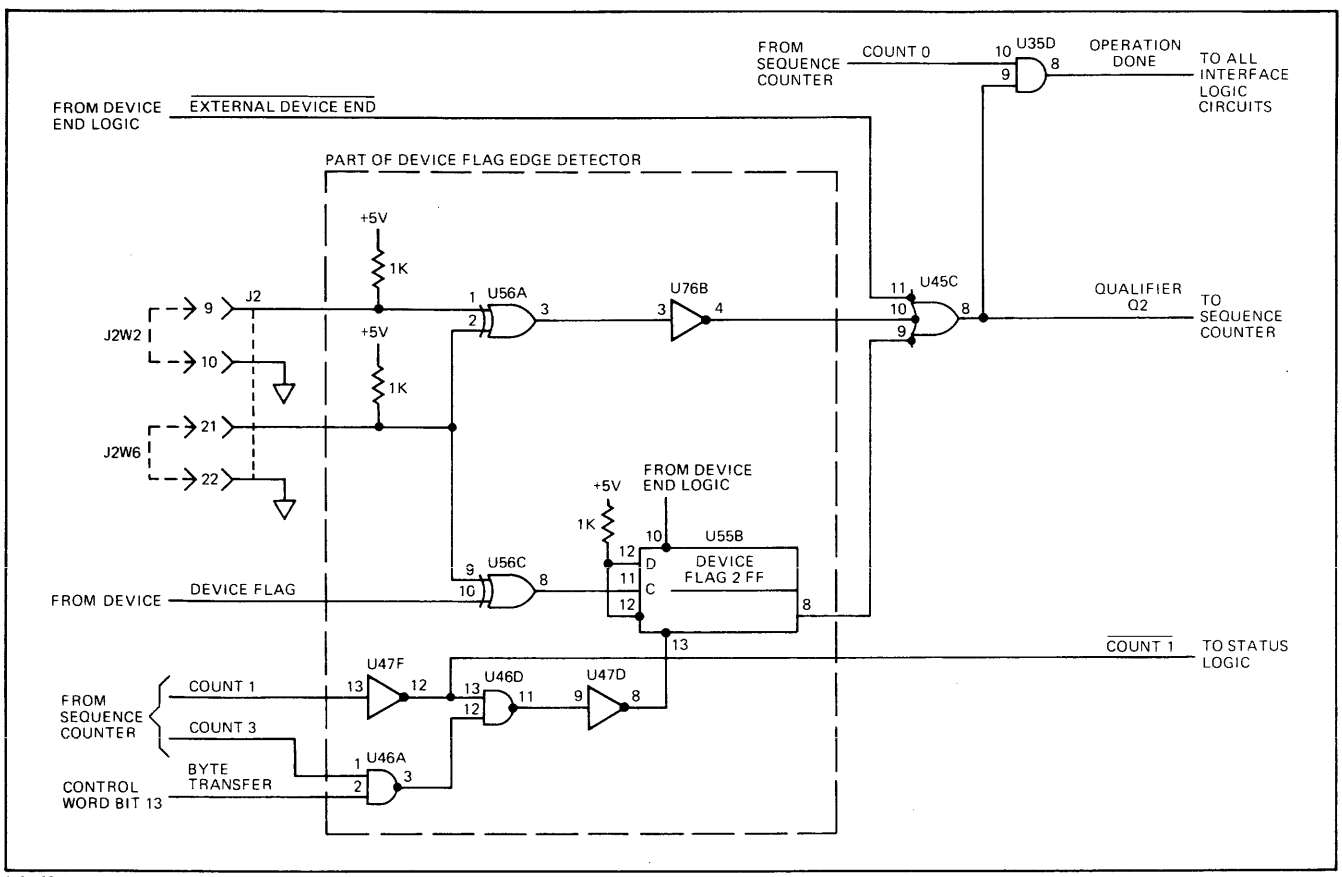


Figure 3-13. Control Function Sequence Counter Operational Flow Diagram



2186-21

Figure 3-14. Qualifier Q1 and Q3 Simplified Diagram



2186-22

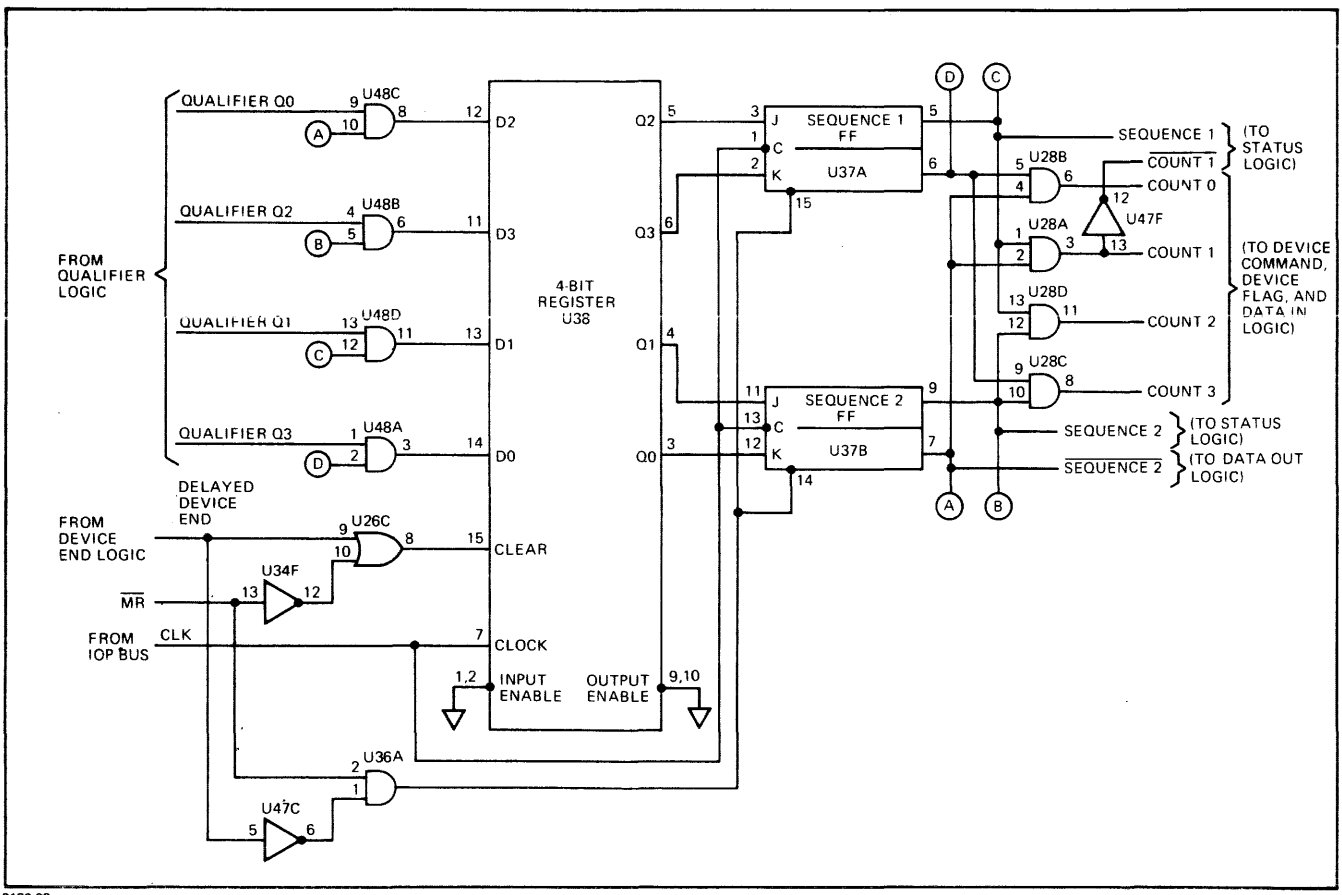
Figure 3-15. Qualifier Q2 Simplified Diagram

signal is sent twice (once for each eight bit transfer), Device Flag 2 FF is cleared by the Count 1 signal, set by the first Device Flag signal, cleared by the Count 3 signal combined with the Byte Transfer signal, and set again by the second Device Flag signal. If the Device Flag 2 FF is being set by the trailing edge of the Device Flag signal, the sequence counter can return to zero before Qualifier Q2 goes high. Qualifier Q2 then prevents the Operation Done signal from going high until the second byte of data is transferred.

3-59. The signals listed in paragraph 3-53, "a" and "b" are also used to direct set the Write Transfer FF while the signals in "c", "d", and "e" are also used to direct set the Read Transfer FF. Output from these flip-flops is combined with the output from the sequence counter to enable the Device Command logic. Only one of the flip-flops is set, depending on the read or write function, but either flip-flop is cleared by the Operation Done signal or by a "not" Master Reset ($\overline{MR}$) signal. The appropriate flip-flop is set for each I/O transfer operation and must be set again for each repeat operation.

3-60. Sequence Counter. The sequence counter steps the interface PCA and device through a data transfer operation. This counter contains a four bit register and two flip-flops with their associated gating circuits (see figure 3-16). Qualifier signals Q0 through Q3 control the sequence counter and the counter is stepped by the CLK signal received from the IOP bus $\overline{MCU\ CLKS}$ line. When the proper qualifier signal is high, the counter advances on the trailing edge of the next CLK signal. The counter has four output signals called Count 0 through Count 3. Count 0 is high and the other three output signals are low when the counter is inactive between data transfers. The counter advances making Count 1 high and the other three output signals low when Qualifier Q0 signal goes high followed by a CLK signal. Advancing continues as Qualifier signals Q1 through Q3 go high followed by a CLK signal. Qualifier Q3 returns the counter to its inactive state with the Count 0 signal high. The counter is also set to Count 0 when both flip-flops are direct cleared by either a "not" Master Reset ($\overline{MR}$) signal that results from a HP 3000 Computer clear or a Device End signal from the device end logic. These same signals also clear the four bit register.

3-61. Output signals from the sequence counter are used to control the Device Command logic and to enable the data transfer logic. To begin a data transfer operation the Count 1 signal is combined with the output from the Read Transfer FF or Write Transfer FF and the combined output enables the Device Command signal that is sent to the external device. During a byte transfer, the same transfer



2186-23

Figure 3-16. Sequence Counter Simplified Diagram

flip-flop outputs are combined with the sequence counter Count 3 signal to enable the Device Command signal and initiate transfer of the second eight bit byte during the same transfer operation. When a byte transfer is not enabled, the sequence counter steps through Count 3 and returns to Count 0.

3-62. Device Command Logic. The device command logic provides a Device Command signal that requests the device to start an I/O data transfer. This logic is controlled by the sequence counter and also receives the Read Transfer, Write Transfer, and Byte Transfer signals. Read or Write Transfer signals are provided by the qualifier logic, and the Byte Transfer signal is provided by control word bit 13 (see figure 3-17). The device command logic contains the read and write data control logic, Write Delay OS (one-shot multivibrator), Command Pulse OS, and associated logic circuits. Signals received by the read and write data control logic circuits determine whether data is to be transferred from the device or to the device. These signals also determine whether data is to be transferred in one word of up to 16 bits or in two bytes of up to 8 bits each. When two bytes are transferred in one data transfer operation, the Device Command signal is enabled twice (once for each eight bit byte). During a write transfer the Device Command signal is delayed for a minimum of 200 nanoseconds to allow data to settle on the data out lines (see figure 3-11). This delay is controlled by the Write Delay OS. When the Device Command is being used in the pulse mode, the Command Pulse OS controls the pulse width. The Device Command signal is sent to the device through a gating circuit that allows selection of a positive-going or negative-going pulse. The following paragraphs provide details of operation of the device command logic.

a. The read and write data control logic circuits receive the output signal from either the Read Transfer FF or Write Transfer FF. These flip-flops are contained in the qualifier logic circuit. One of these flip-flops is set by the I/O direct command or other signals that initiate a data transfer. The resulting Read or Write Transfer signal is combined with the Count 1 signal from the sequence counter to initiate the Device Command signal (see figure 3-19). For 16-bit data transfers, the Device Command signal is initiated once for each data transfer operation. To transfer two bytes of up to eight bits each, the Device Command signal is initiated twice for each data transfer operation. The second Device Command signal occurs if the Byte Transfer signal is enabled by control word bit 13. This Byte Transfer signal is first combined with the Read or Write Transfer signal and then with the Count 3 signal from the sequence counter. This initiates the second Device Command signal. For read transfers, Device Command signals are sent directly to the response or pulse mode selection logic circuit. For write data transfers, Device Command signals are sent to the Write Delay OS.

b. The Write Delay OS provides an adjustable time delay from 200 nanoseconds. This time delay allows data to settle on the data lines before the Device Command signal is sent to the device. The circuit contains the Write Delay OS which has external time delay components and also contains inverter U27A with associated components and gate U46B (see figure 3-19). The one-shot time delay can be adjusted by the addition of a capacitor between pins 5 (+) and 6 (-) in the mating connector to J1. Inverter U27A with its associated components prevents the output of gate U46B from going low before the Write Delay OS is set. The output from gate U46B is sent to the response or pulse mode logic circuit.

c. The response or pulse mode logic provides the selection of either response mode or pulse mode operation and also provides a pulse for pulse mode operation (see figure 3-11). This logic circuit contains a gating circuit with jumper wire J2W3 and the Command Pulse OS

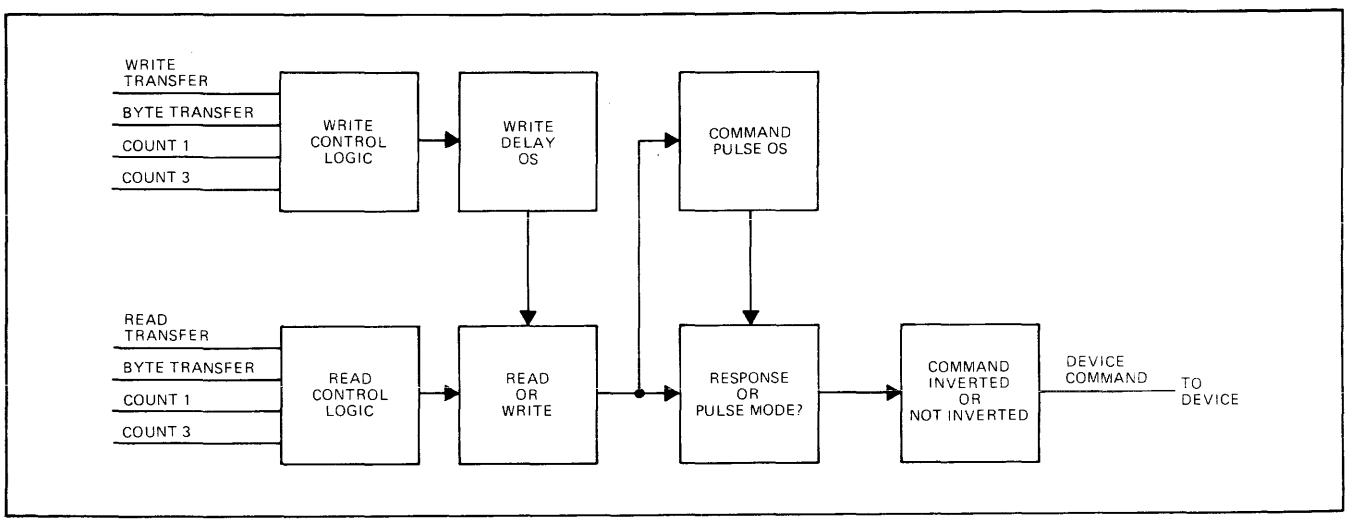


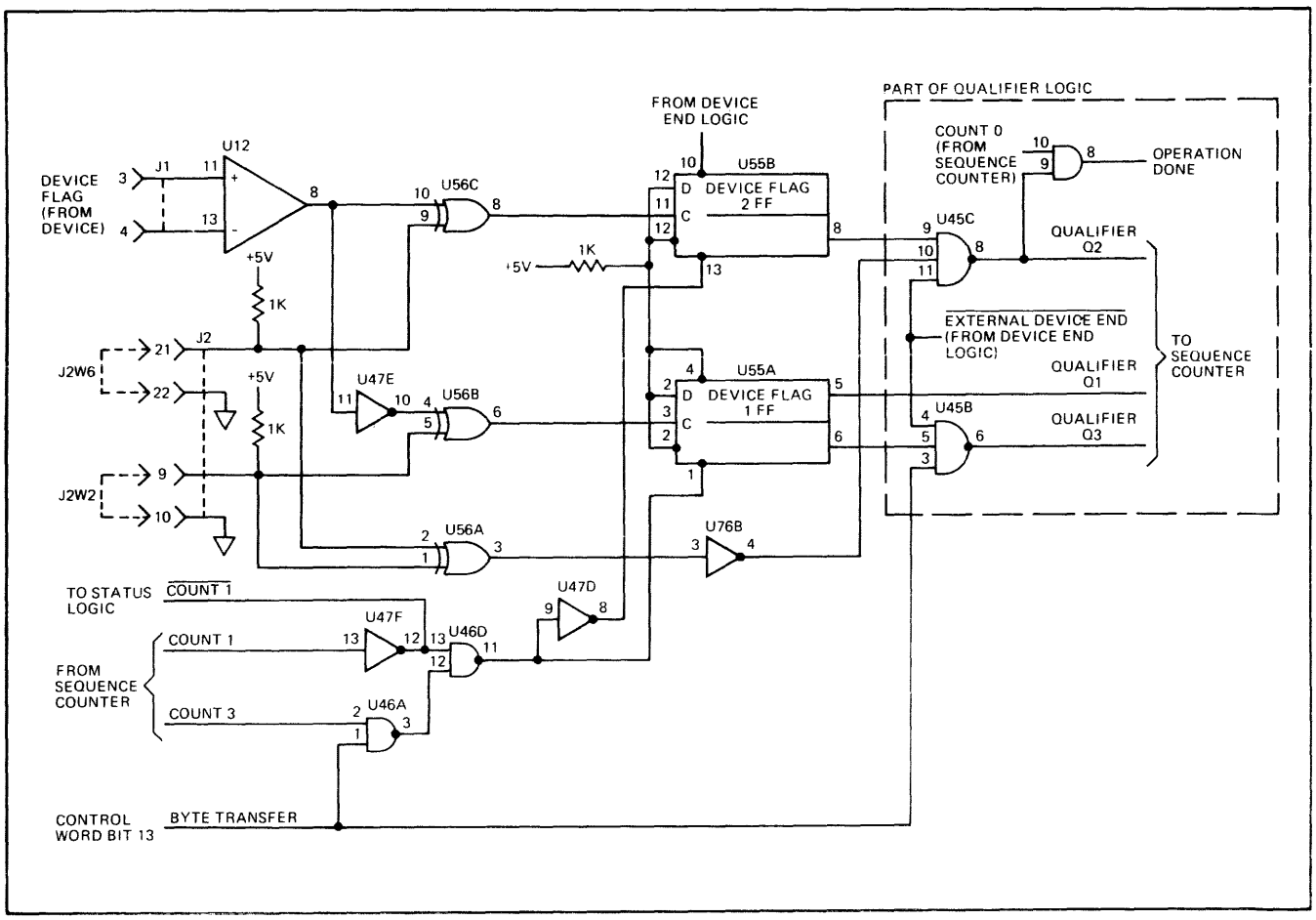
Figure 3-17. Device Command Logic Block Diagram

(see figure 3-19). Connection pins for jumper wire J2W3 are contained in the mating connector for J2. With the jumper removed, the Device Command signal is sent directly through gate U67F to the output inverter logic. If jumper wire J2W3 is installed, gate U67F is disabled and gate U67A is enabled by the output pulse from the Command Pulse OS. The Command Pulse OS has external time delay components. These components are selected to provide an 8 microsecond Device Command pulse. This pulse is then sent to the output inverter circuit.

d. The Device Command signal inverter logic provides an inverted or not inverted signal to the output circuit. This inverter logic circuit contains an "exclusive or" gate and jumper wire J2W10 (see figure 3-19). By definition, the "exclusive or" gate output is high only when one input is high and the output is low when both inputs are high or low at the same time. With jumper wire J2W10 removed, the gate output remains low until a Device Command signal is received. This is the not inverted signal. If jumper wire J2W10 is installed, the output signal is inverted. The output is sent through a differential line driver to the device.

3-63. Device Flag Edge Detector. The device flag edge detector circuit receives the Device Flag signal from the external device and, after detecting the preselected edges, provides qualifying conditions for the qualifier logic circuit. This device flag edge detector contains three "exclusive or" gates, Device Flag 1 FF, Device Flag 2 FF, and associated logic circuits (see figure 3-18). The three "exclusive or" gates are controlled by the Device Flag signal from the device and by jumper wires J2W2 and J2W6. Connection pins for these jumpers are located in the mating connector for J2. By definition, the output from an "exclusive or" gate is high when either input is high and the output is low when both inputs are high or low. This arrangement of jumper wires and gates allows the jumper wire configuration to determine the conditions that set the Device Flag 1 and 2 FF's. For example, when jumper wires J2W2 and J2W6 are both removed, Device Flag 1 FF sets on the leading edge of the Device Flag signal and Device Flag 2 FF sets on the trailing edge of the Device Flag signal.

3-64. Both device flag flip-flops are direct cleared by the Count 1 signal from the sequence counter. Device Flag 2 FF is cleared when the counter advances from Count 0 to Count 1 and Device Flag 1 FF is cleared when the counter



2186-26

Figure 3-18. Device Flag Edge Detector

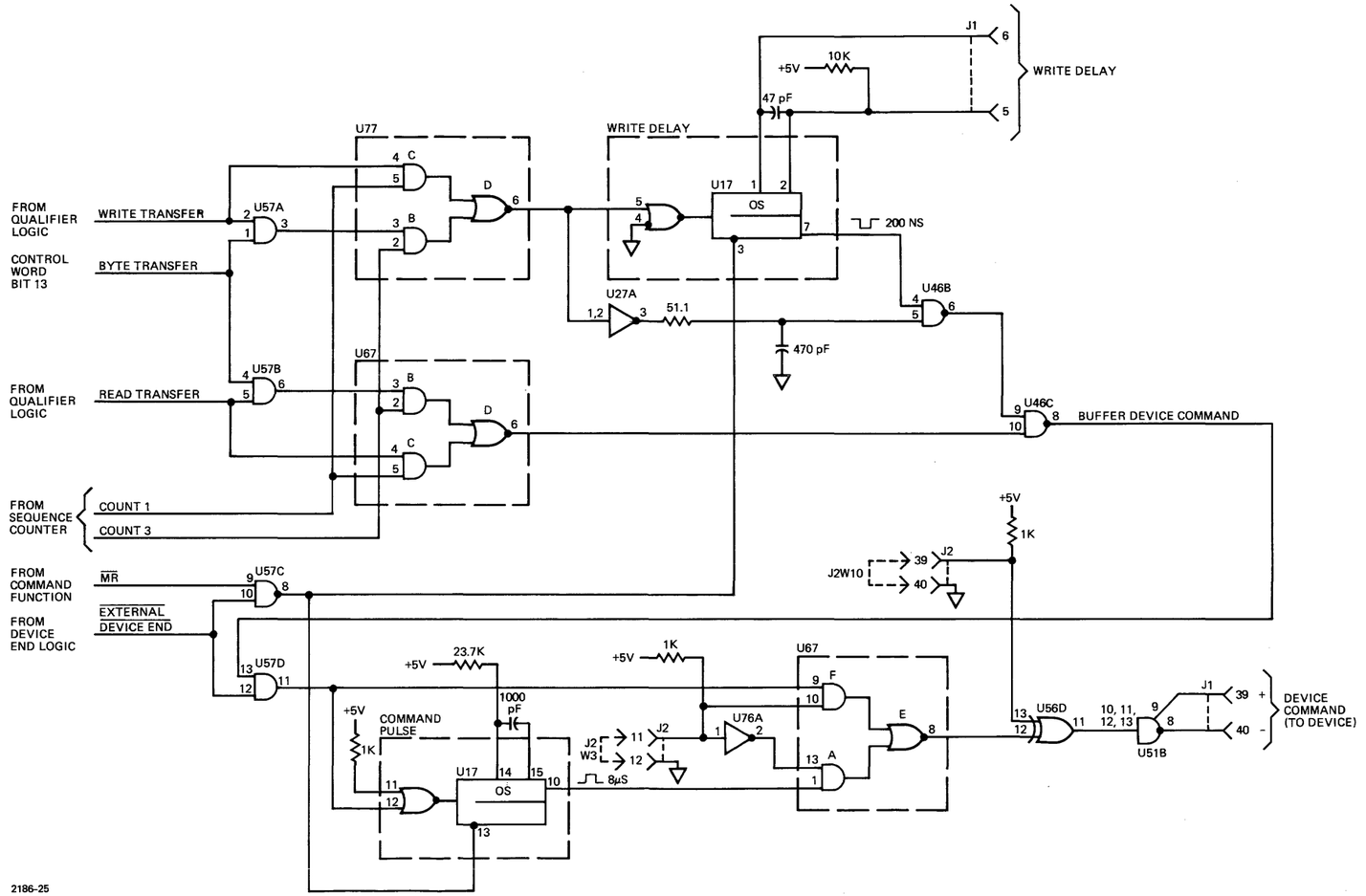


Figure 3-19. Device Command Logic

advances from Count 1 to Count 2. When the Byte Transfer signal is enabled by control word bit 13, a second Device Flag signal is received that sets both flip-flops a second time during the same data transfer operation. To direct clear the flip-flops a second time, the Count 3 signal from the sequence counter is combined with the Byte Transfer signal. The Device Flag 2 FF is then cleared when the counter advances from Count 2 to Count 3 and the Device Flag 1 FF is cleared when the counter returns from Count 3 to Count 0.

3-65. Device Control Word Logic. The device control word logic contains the logic circuits that receive the 16-bit control word from the IOP bus. Five of the control bits are made available to the external device and the remaining bits control operation of the interface PCA (see figure 3-20).

3-66. In addition to the control word bits, the interface PCA provides three discrete signals to the device and power for the self-test hood (see figure 3-21). The power for the self-test hood is sent through jumper wire W1. This prevents power from being available at the output pin unless W1 is installed. Power is also available at connector pin J1-35. This power is used by the self-test hood to determine the interface PCA logic configuration.

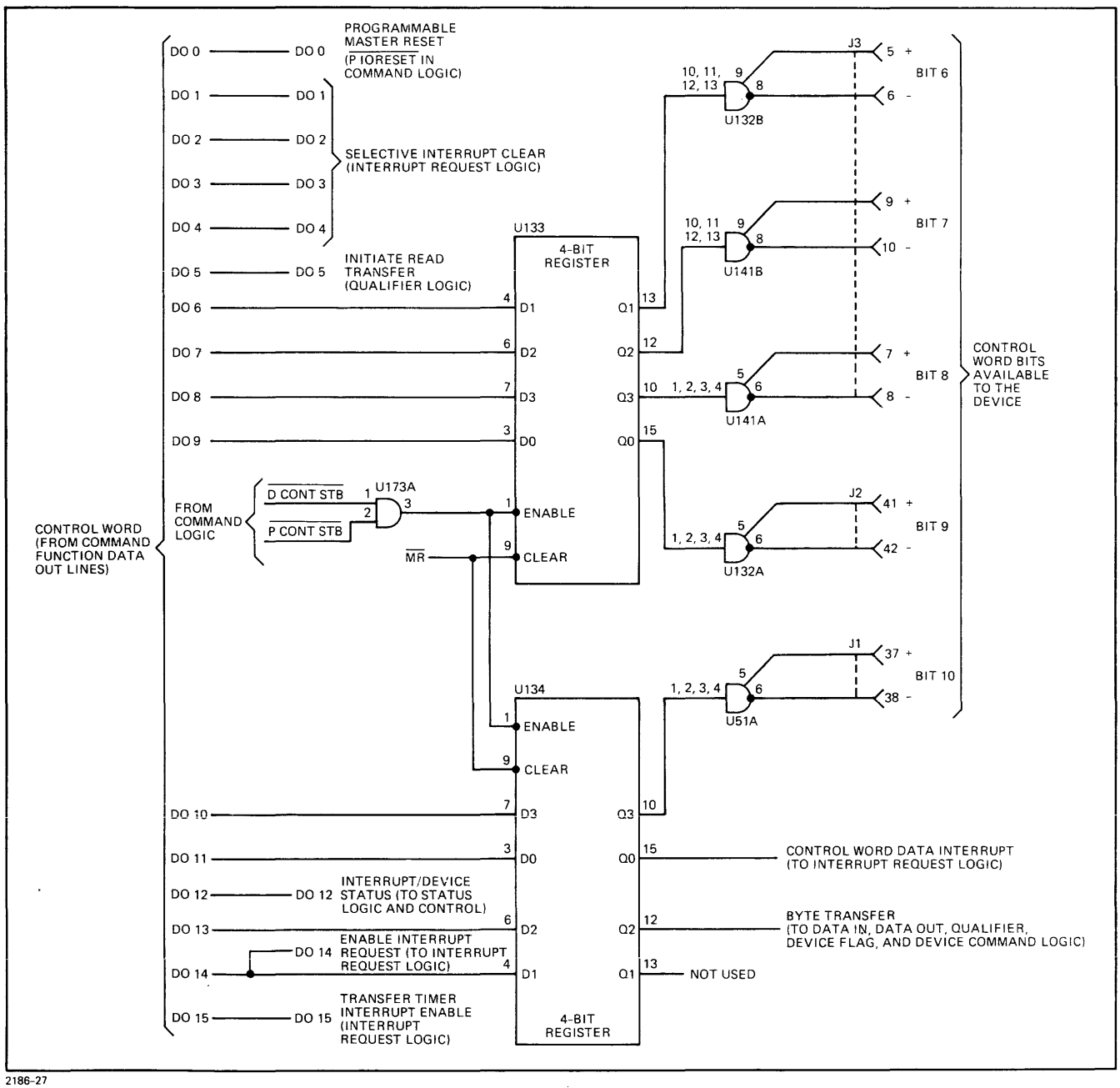
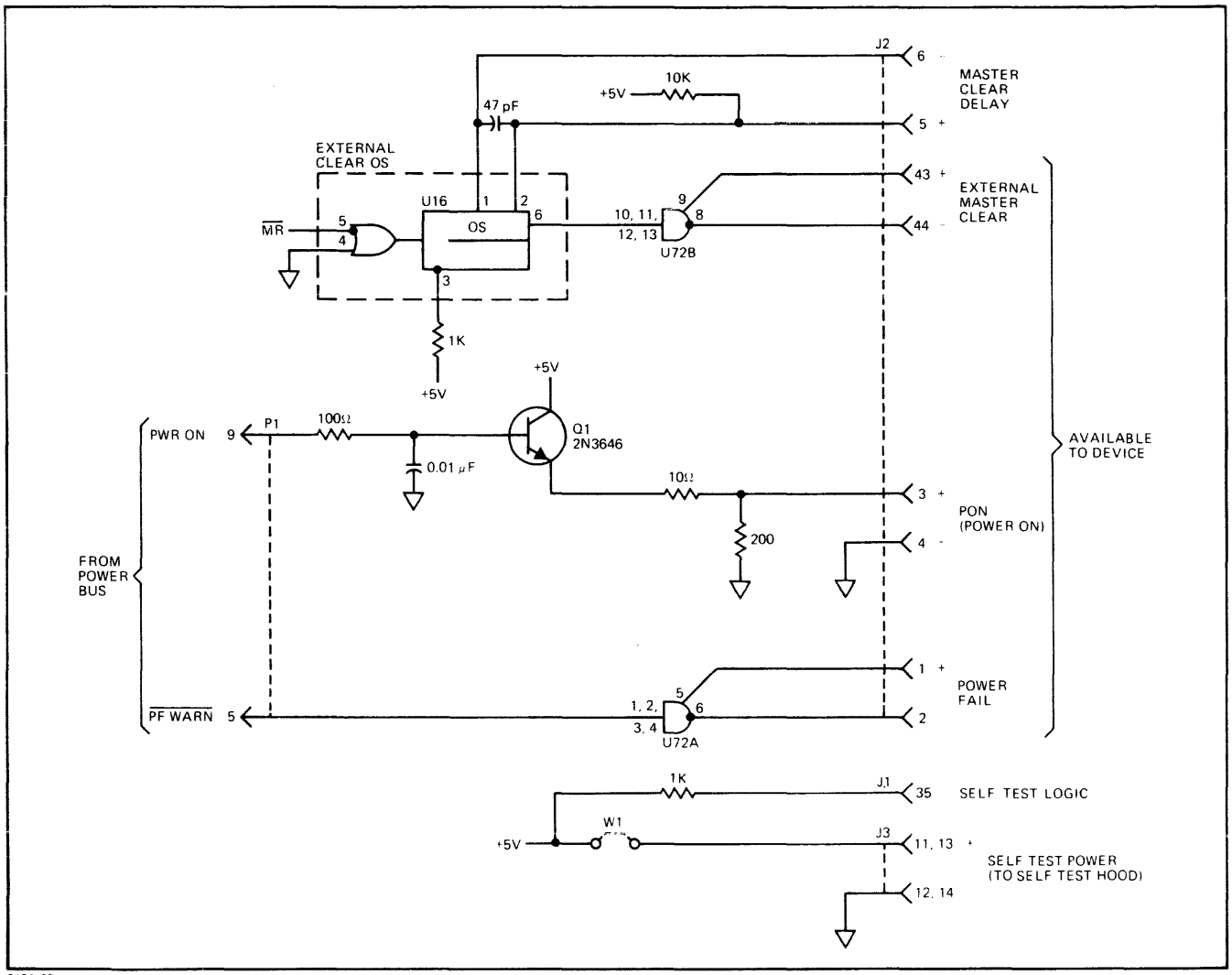


Figure 3-20. Device Control Word Logic



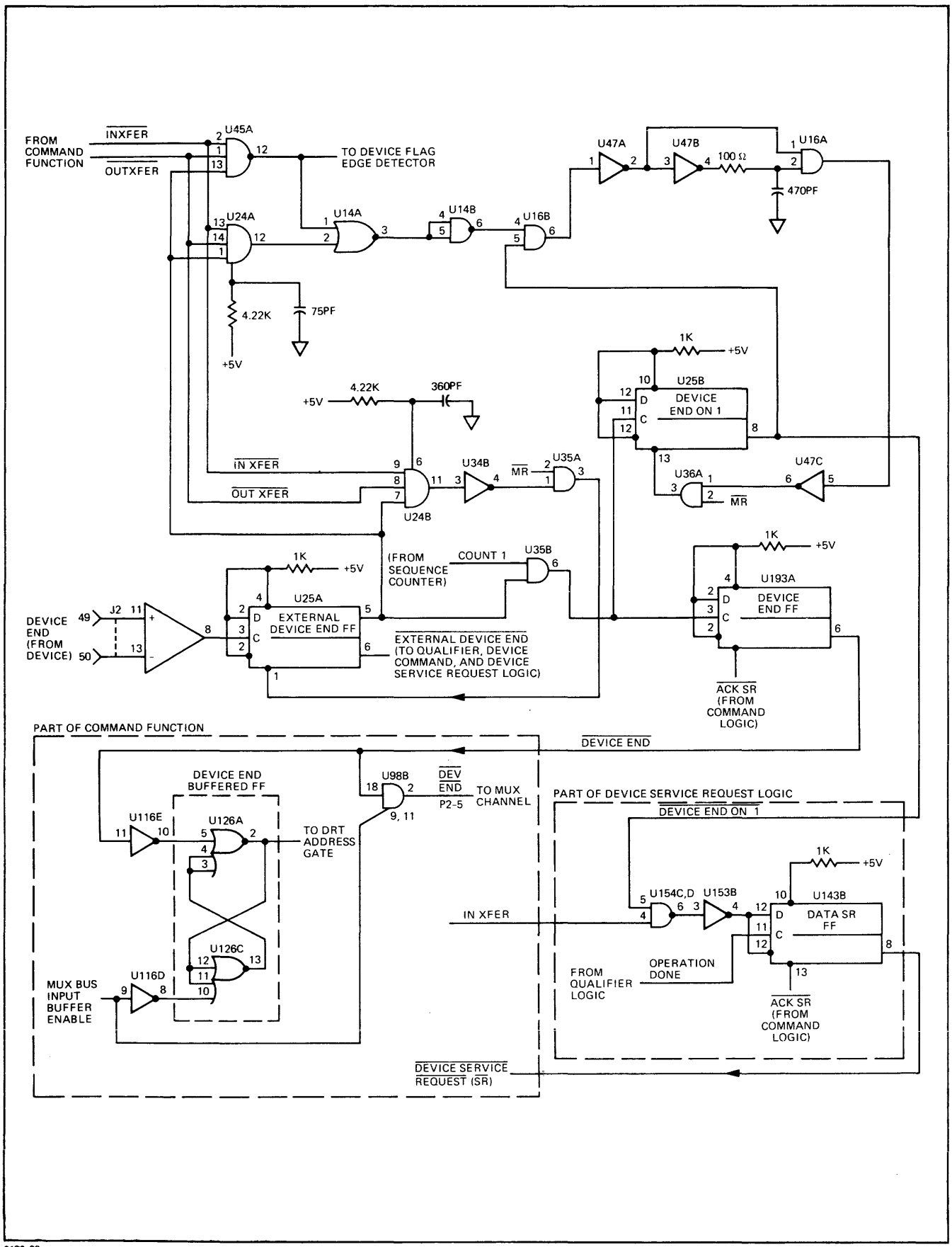
2186-28

Figure 3-21. Discrete Device Control Signals

3-67. DEVICE END LOGIC. The device end logic receives a Device End signal from the external device and, during an I/O program data transfer, causes the instruction being performed to be terminated. The system then proceeds to perform the next command in the I/O program. Direct data transfer commands (RIO and WIO) are not affected by the Device End signal. The Device End signal is sent to the interface PCA only during an I/O data transfer and in response to a Device Command signal from the interface PCA. When this occurs following the first byte of a two-byte data transfer, the second byte is transferred to memory in bits 0 through 7 of the data word and the data transfer is then terminated. If the Device End signal is received during data-chaining of an I/O program data transfer, all data-chained requests are terminated until the next I/O program word. The device end logic contains the External Device End, Device End On 1, and Device End FF's with their associated logic circuits (see figure 3-22).

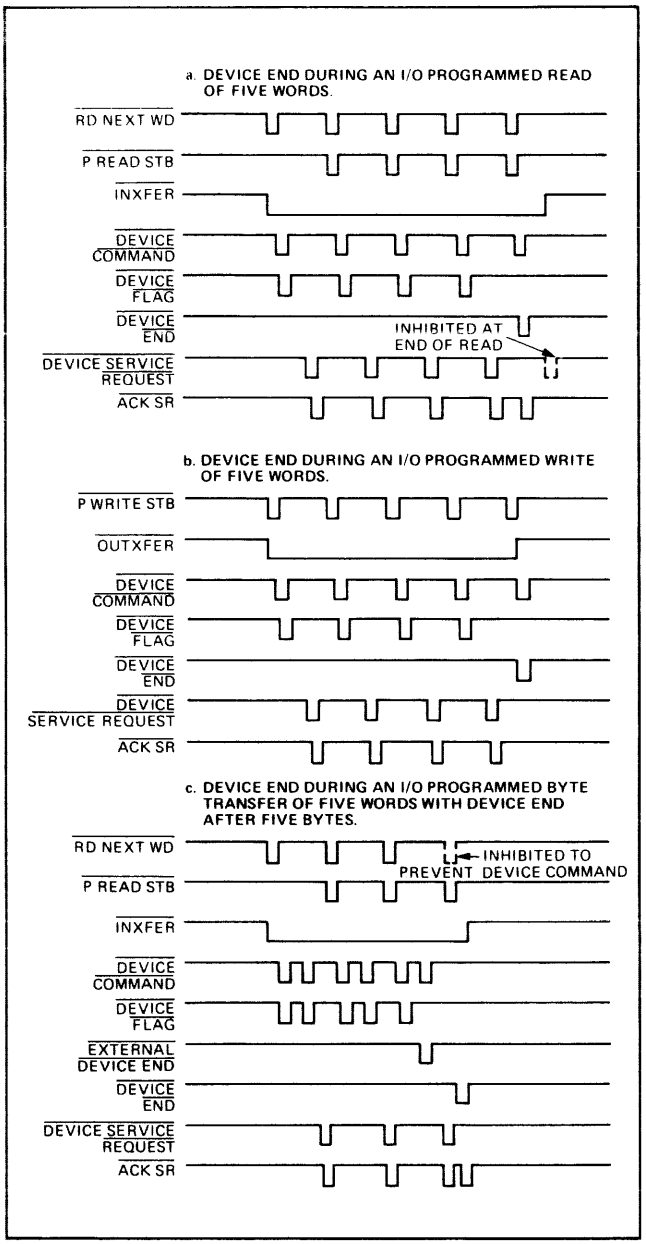
3-68. Device End During a One-Word Data Transfer. During a one-word (up to 16 bits) data transfer, the Device End signal is received from the device during sequence counter Count 1 and in response to a Device Command signal being sent to the device. The Device End signal inhibits the Device Command line and sends a $\overline{\text{DEV END}}$ signal through the MUX CHAN bus to the multiplexer channel. An $\overline{\text{ACK SR}}$ signal is returned from the multiplexer channel to acknowledge the $\overline{\text{DEV END}}$ signal (see figure 3-23).

3-69. Device End During a Two-Byte Data Transfer. During a two-byte (up to eight bits per byte) data transfers the first byte transfers in the same manner as a one-word transfer (refer to paragraph 3-68). However, the first Device Command signal causes a valid data transfer that loads the first byte into both the upper and lower byte positions (bits 0 through 7 and 8 through 15 respectively) of the data-in



2186-29

Figure 3-22. Device End Logic



2186-30

Figure 3-23. Device End Signal Timing Diagrams

registers. When the sequence counter reaches count 3, the Device Command signal again goes high and then the device returns a Device End signal that causes the lower half of the data-in register to be loaded with zeros. The Device Command signal is then inhibited by the Device End signal and a "not" Device Service Request signal is sent to the multiplexer channel to allow the last byte of data to be sent to memory. If this was the last byte of data to be transferred, the interface PCA sends a "not" Device Service Request signal through the MUX CHAN bus to the multiplexer channel and the "not" Device End signal is inhibited. If there is more data to be transferred the interface PCA sends the "not" Device End signal to the multiplexer channel.

3-70. DEVICE SERVICE REQUEST LOGIC. The device service request logic is used only when data transfers are controlled by the multiplexer channel (I/O program). This logic provides a "not" Device Service Request ($\overline{\text{DEV SR}}$) signal to the multiplexer channel for each of the following conditions (see figure 3-24):

a. The last programmed command sent by either a $\overline{\text{P CMD 1}}$ or $\overline{\text{P CONT STB}}$ has been executed.

b. The interface PCA has data ready to transfer to memory.

c. The interface PCA is ready to receive data from memory.

d. When the in transfer or out transfer operation is finished and the device is ready for the next command.

3-71. The $\overline{\text{DEV SR}}$ signal is sent through the command logic circuits to the multiplexer channel when any one of the following four flip-flops is set: Command SR, Transfer SR, Data SR, and EOT SR. These flip-flops are direct cleared when the $\overline{\text{DEV SR}}$ signal is acknowledged by the multiplexer channel and an $\overline{\text{ACK SR}}$ signal is returned. Jumper wire J2W1 can be installed to allow the Command SR FF to be set by the Device Status Bit 11 signal. This feature is used when an external device requires a long response time. Otherwise, jumper wire J2W1 is not connected.

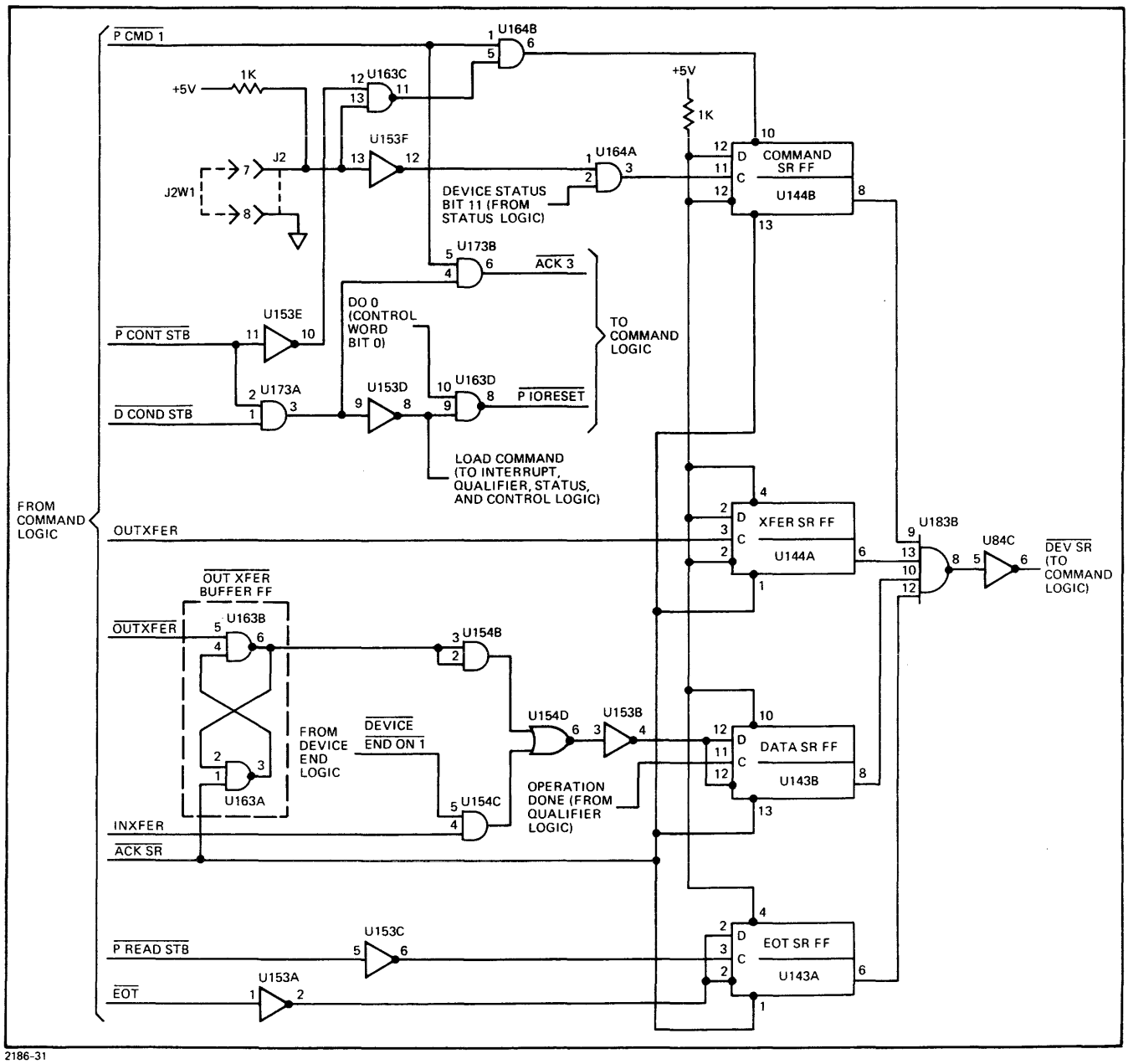
3-72. INTERRUPT REQUEST LOGIC. The interrupt request logic provides the Interrupt Request signal to the interrupt processing circuits of the command function. This Interrupt Request signal is stored by the interrupt processing circuits and, when conditions are met for an interrupt to occur, the Interrupt Request signal enables the interface PCA to request service from the CPU. An Interrupt Request signal can be initiated by eight different conditions on the interface PCA and the same conditions also provide status information for the interrupt status word. The eight conditions that initiate the Interrupt Request signal and provide status information are as follows:

a. Time Out. The data transfer timer times out before data transfer is complete.

b. Transfer Error. The multiplexer channel sends an $\overline{\text{XFER ERROR}}$ signal because of an error in data transfer between the interface PCA and IOP.

c. I/O System Interrupt. A $\overline{\text{SET INT}}$ signal is initiated by the command function because of a programmed interrupt request.

d. Clear Interface. The device sends a Clear Interface signal to the interface PCA to indicate an I/O program was cancelled by the device.



2186-31

Figure 3-24. Device Service Request Logic

e. Data Transfer. Data transfer interrupts are enabled by setting control word bit 11 to logic 1. An interrupt request is then initiated after each data transfer when the sequence counter goes from count 3 to count 0.

f. Device Interrupts. Three interrupt circuits are available to the device. These circuits can be connected through the device status lines to any signal from the device. The signals would then set device status bits 8, 9, and/or 10, and would also provide Interrupt Request signals and interrupt status bits.

3-73. Each of the eight conditions that initiate an Interrupt Request signal is processed by a separate circuit that contains at least one flip-flop. After initiating an interrupt, it is usually desirable to clear only the circuit that enabled the last interrupt request. The circuits are selectively cleared through the clear interrupt decoder circuit and control word bits DO1 through DO4. The circuits that enable the interrupt requests and the clear interrupt decoder are described in the following paragraphs.

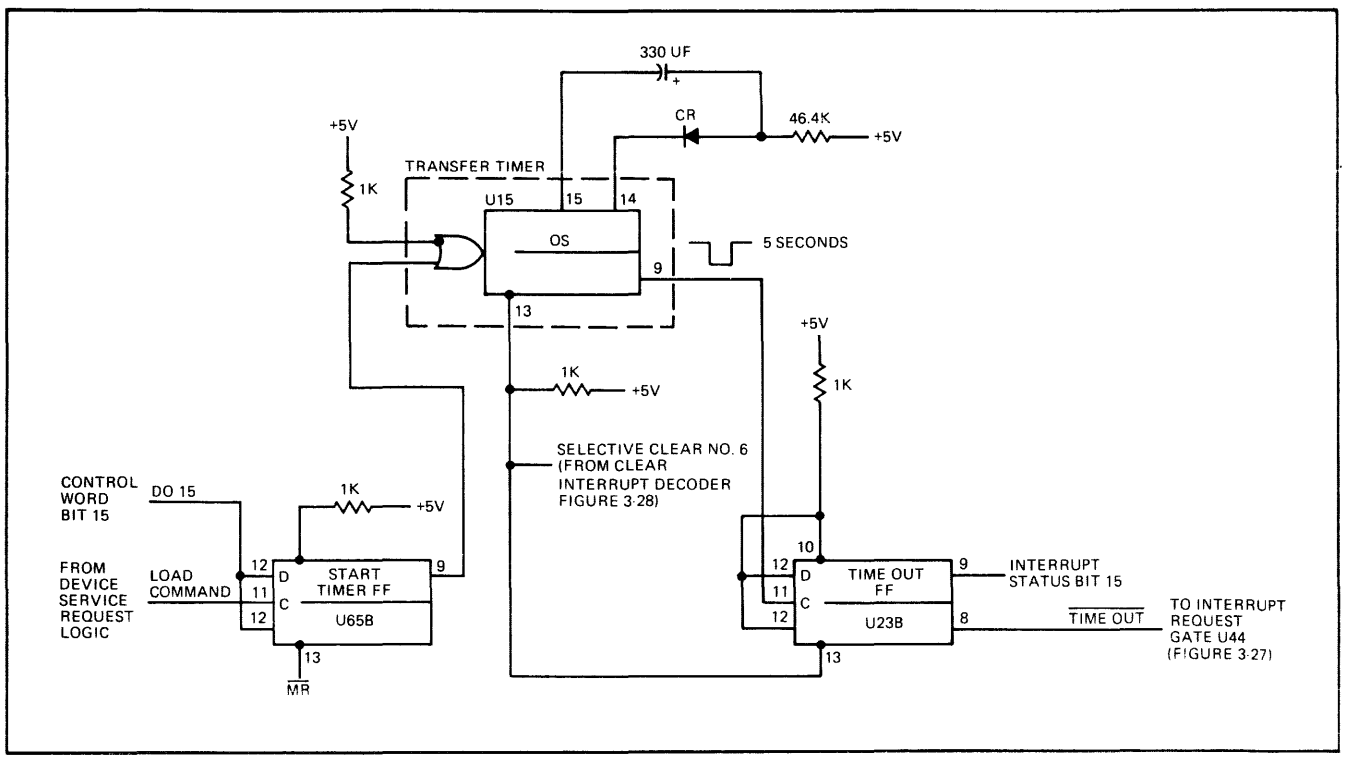
3-74. Data Transfer Timer. The data transfer timer enables an interrupt request to be initiated if the external

device fails to respond within a predetermined time. To start the timer, control word bit DO15 is sent as a logic 1. Once started, the timer will time out and enable an interrupt request unless the timer is cleared within five seconds. The timer is cleared by sending a second control word with bit DO15 at a logic 0 and selective clear number 6 enabled (control word bits DO2, DO3, and DO4 in bit pattern 001). Clearing the timer can also be accomplished by sending control word bit DO1 as a logic 1 (clear all interrupts) or control word bit DO0 as a logic 1 (master reset).

3-75. The data transfer timer interrupt logic contains a Start Timer FF, Transfer Timer OS (one-shot multivibrator), and Time Out Interrupt FF with associated logic circuits (see figure 3-25). To initiate the timer circuit, control word bit DO15 is sent as a logic 1. This causes the Start Timer FF to set and its set-side output is then used to start the Transfer Timer OS. The Start Timer FF remains set until control word bit DO15 is sent as a logic 0 or until a master clear (control word bit DO0 at logic 1) is sent to direct clear the FF. To restart the Transfer Timer OS, the Start Timer FF must be cleared and then set again by control word bit DO15 at a logic 1. If the Transfer Timer OS times out (after five seconds), its output sets the Time Out FF. Setting the Time Out FF sets bit 15 of the interrupt status word to a logic 1 and enables the interrupt pending bit (bit 2) of the interface status word.

3-76. Device Interrupt Request Logic. The device interrupt request logic provides three ways for the external device to request an interrupt. Signals from the device suitable for initiating an interrupt, such as a signal indicating data is available or a request for service, are connected to the device status input lines for Device Status Bits 8, 9, and 10. These lines are also connected to the device interrupt request logic. The device interrupt request logic contains three flip-flops and connections for jumper wires J2W4, J2W8, and J2W9 (see figure 3-26). When the jumper wires are not connected, the flip-flops remain in the set state and interrupts are not requested. If a jumper wire is installed, its respective flip-flop is cleared when the device status signal occurs at the clock input. Device Status signals for bits 8 and 9 cause their flip-flops to clear on the signals leading edge. The signal for bit 10 causes its flip-flop to clear on the signals trailing edge. Clearing any of the three flip-flops causes the set-side output to go low and initiate an interrupt request while the clear-side output goes high to provide an interrupt status bit. To clear the interrupt request, the cleared flip-flop is direct set by a Selective Clear signal.

3-77. I/O System and Interface Interrupt Request Logic. The I/O system and interface interrupt request logic provides four conditions for initiating the Interrupt Request signal. Each condition is provided through a flip-flop (see figure 3-27). When a flip-flop is set, its clear-side output provides a signal to initiate the Interrupt Request signal and



2186-32

Figure 3-25. Data Transfer Timer Interrupt Request Logic

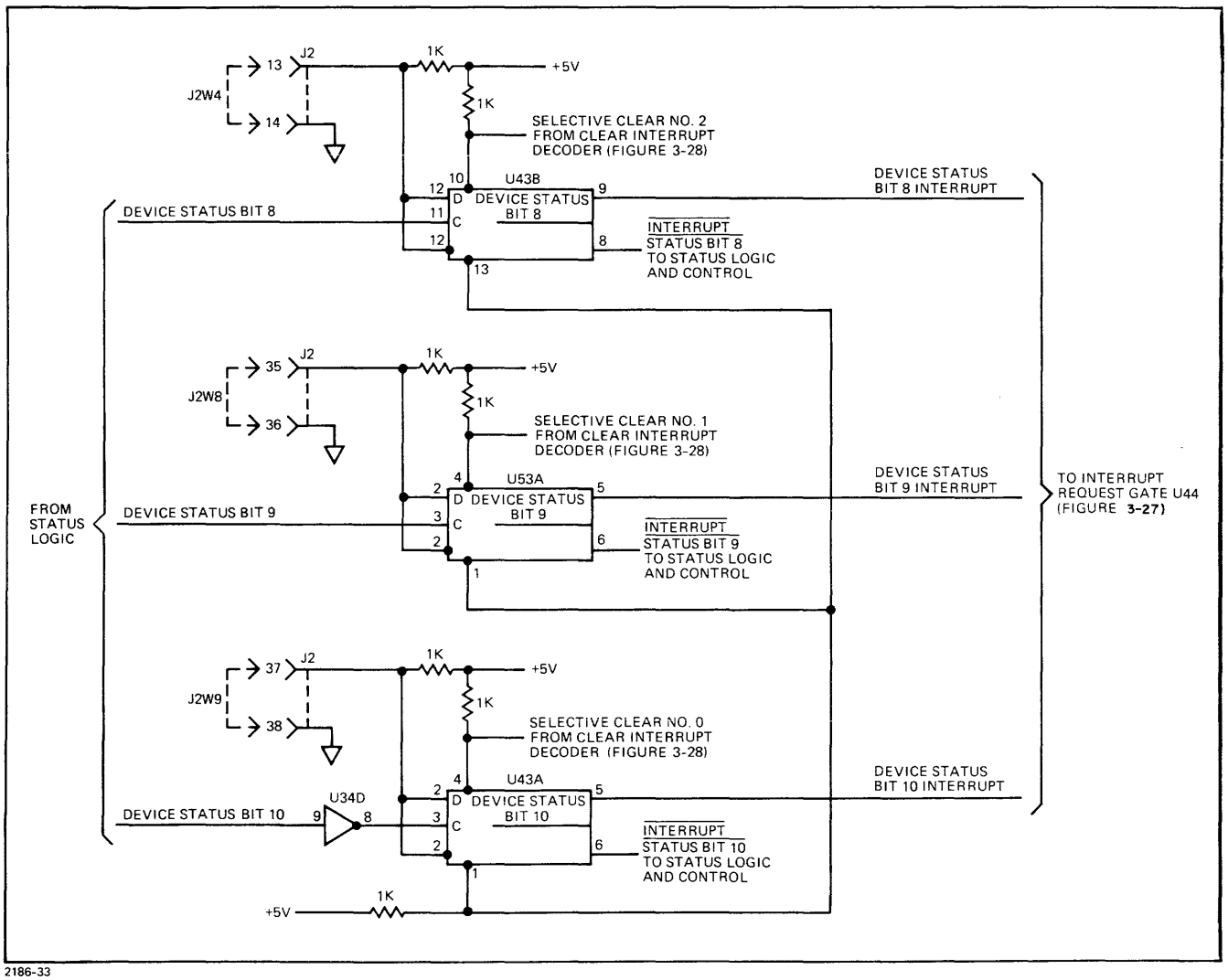


Figure 3-26. Device Interrupt Request Logic

its set-side output provides a separate bit in the interrupt status word. After being set, these flip-flops are cleared by separate Selective Clear signals. Setting any of the flip-flops enables gate U44 and provides the Interrupt Pending signal that sets interface status bit 2. The HP 3000 Computer can detect if an interrupt request is pending by scanning this interface status bit. With the Interrupt Pending signal high, an interrupt request can be enabled by control word bit DO14. Sending control word bit DO14 as a logic 1 sets the Interrupt Request FF and the flip-flop set-side output enables gate U54B. The Interrupt Request FF remains set until control word bit DO14 is sent as a logic 0 or until it is direct cleared by a "not" Master Reset ($\overline{MR}$) signal or a "not" Set Interrupt Active ($\overline{SET\ INT\ ACT}$) signal. A $\overline{MR}$ signal results from sending control word bit DO0 as a logic 1. A $\overline{SET\ INT\ ACT}$ signal occurs upon receipt of an interrupt poll in response to an interrupt request by this PCA. When gate U54B is enabled by the Interrupt Request FF set-side output and the Interrupt Pending signal, an Interrupt Request signal is enabled. This Interrupt Request signal is sent through gate U54C where the signal can be inhibited by a "not" Interrupt Active signal if a previous interrupt request is still being processed. From gate U54C, the Interrupt Request signal is sent to the interrupt processing circuits of the command function. Two flip-flop outputs, "not" Transfer Error Interrupt and "not" Clear Interface Interrupt, will also initiate a $\overline{CLR\ IL}$ signal that is used by the command function to clear the interface logic and inhibit service requests.

3-78. Clear Interrupt Decoder Logic. The clear interrupt decoder logic provides selective clearing of the eight circuits that can initiate an Interrupt Request

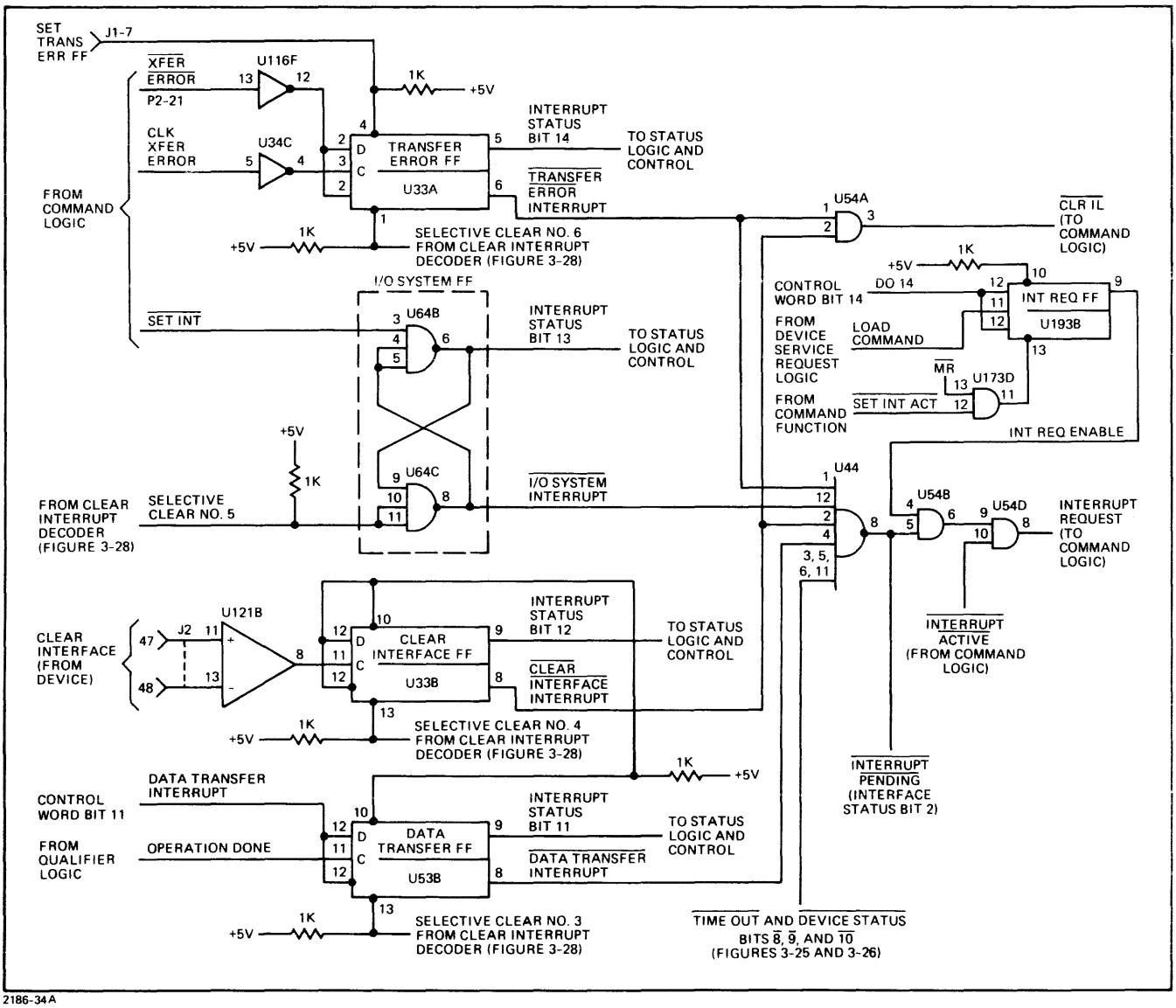


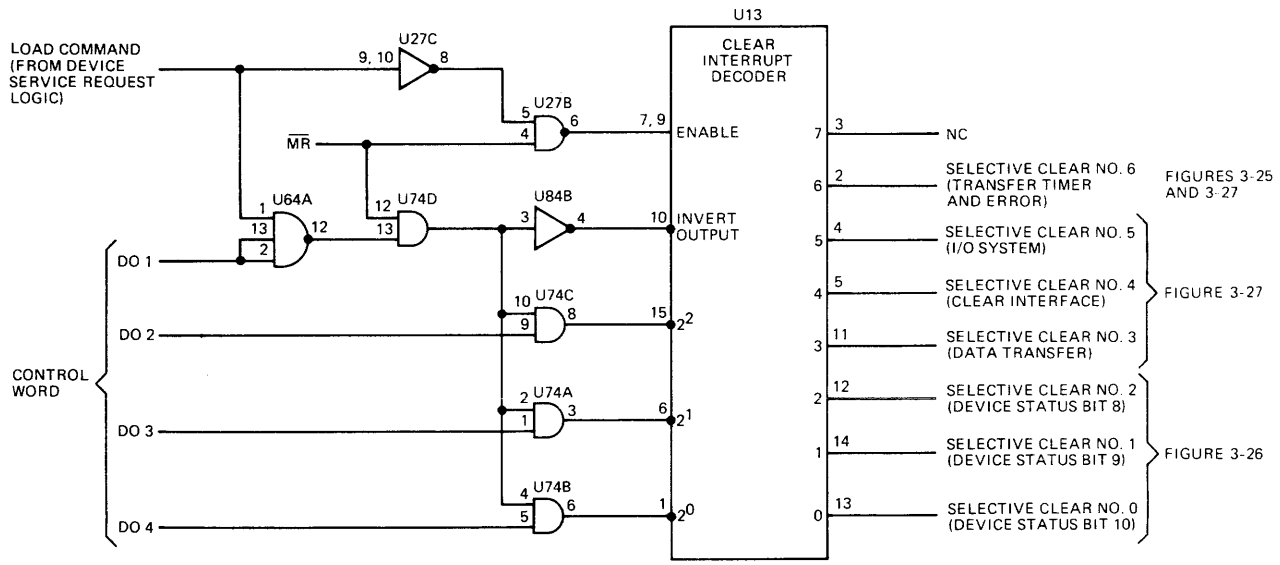
Figure 3-27. I/O System and Interface Interrupt Request Logic

signal. This logic contains a binary-to-octal decoder circuit and associated logic gates (see figure 3-28). The decoder circuit is enabled by a Load Command signal that results from a $\overline{\text{D CONT STB}}$ or $\overline{\text{P CONT STB}}$ signal from the command function. While the decoder is enabled, control word bits DO1 through DO4 are sent as inputs to the decoder. If control word bit DO1 is high, it is combined with the Load Command signal to inhibit the other decoder inputs and then to invert the decoder output. This sends decoder outputs 0 through 6 all low, which clears all of the interrupt request circuits and sets bits 8 through 15 of the interrupt status word to logic 0. The Interrupt Pending signal is also low which sets status word bit 2 to logic 0 to indicate all interrupt requests are cleared. These same operations occur following a system master reset ($\overline{\text{MR}}$).

3-79. Control word bits DO2 through DO4 provide the binary inputs to the decoder when control word bit DO1 is low (refer to the table in figure 3-28). These bits are used to selectively clear the interrupt request circuits. This allows each circuit to be cleared individually so that an interrupt will not be lost when multiple interrupts are being requested. For example, to clear the transfer timer and transfer error interrupt request, the control word is sent with bits DO1 through DO3 low and bit DO4 high. Decoder output number 6 would then be pulled low while all of the other outputs remained high. Other interrupt request circuits are cleared by changing the binary input to the decoder. To determine if an interrupt request may require clearing, status bit 2 (interrupt pending) is checked, and to determine which interrupt request circuit to clear, the interrupt status word (bits 8 through 15) is checked.

INPUTS				OUTPUT							
2^0	2^1	2^2	INVERT OUTPUT	0	1	2	3	4	5	6	7
0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1	1	1	0	1
0	1	0	0	1	1	1	1	1	0	1	1
1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	0	1	1	1	0	1	1	1	1
1	0	1	0	1	1	0	1	1	1	1	1
0	1	1	0	1	0	1	1	1	1	1	1
1	1	1	0	0	1	1	1	1	1	1	1
0	0	0	1	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
0	1	0	1	0	0	0	0	0	1	0	0
1	1	0	1	0	0	0	0	1	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0
1	0	1	1	0	0	1	0	0	0	0	0
0	1	1	1	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

NOTES:
1 = HIGH
0 = LOW



2186-35

Figure 3-28. Clear Interrupt Decoder Logic

3-80. DATA AND STATUS TRANSFER FUNCTION.

3-81. The data and status transfer function contains circuits that transfer data and status through the interface PCA to and from the external device and HP 3000 Computer System. These circuits consist of the status logic and control, data in logic, and data out logic. Operation of these circuits is described in following paragraphs. Simplified and detailed diagrams for the interface PCA are contained in the HP 3000 Computer System diagrams manuals. The simplified diagrams are set number SD-131 and the detailed diagrams are set number DD-501. Functional block diagrams, timing diagrams and partial schematics contained in this manual are referenced by figure number.

3-82. STATUS LOGIC AND CONTROL. The status logic and control consists of three eight bit drivers and a status control circuit containing the Interrupt/Device Status FF (see figure 3-29). These circuits provide a choice of two 16-bit status words that may be fetched from the interface PCA by either direct or program commands. Both 16-bit status words contain the same data in bits 0 through 7 and these bits are called the interface status byte. Status bits 8 through 15 can be either the interrupt status byte or device status byte depending on control word bit DO12. When control word bit DO12 is high (logic 1) the status word contains the interface status byte and the device status byte. With control word bit DO12 low (logic 0) the status word contains the interface status byte and interrupt status byte.

3-83. Interface Status Byte. The interface status byte is made up from signals developed on the interface PCA that indicate the operating condition of the PCA. These signals are inverted by the interface status driver so that when the driver is enabled, a high input signal provides a logic 0 output status bit. The status bit lines are connected to the data-in lines in the command function which are, in turn, connected to the IOP bus. Brief descriptions of the interface status bits are provided in the following paragraphs.

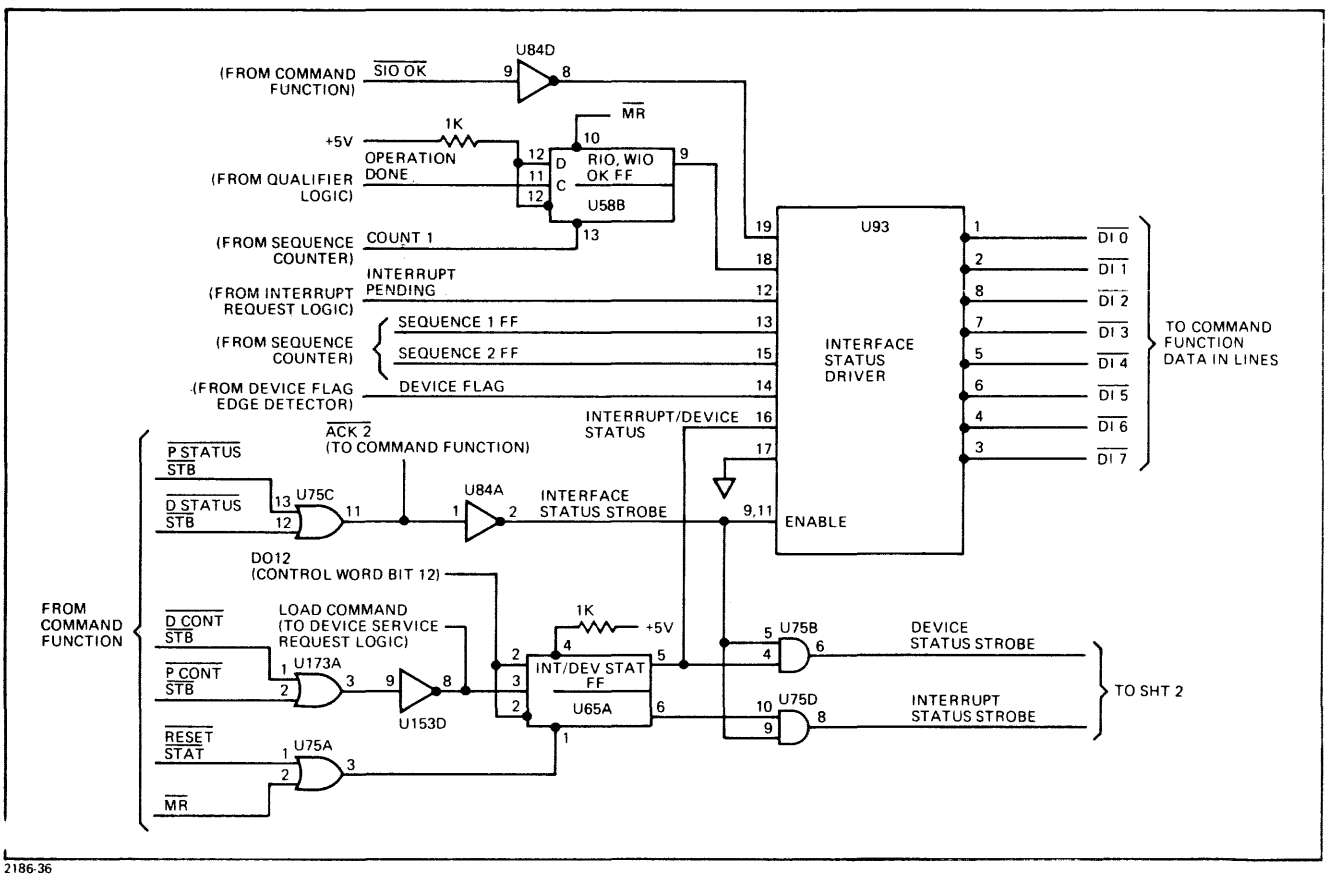
3-84. Status bit 0 is provided by the $\overline{\text{SIO OK}}$ signal from the command function. This signal is normally high and therefore bit 0 is normally a logic 1 when it is all right to proceed with a start I/O (SIO) operation. If bit 0 is a logic 0, an SIO operation is in progress or the multiplexer channel is not connected to the interface PCA.

3-85. Status bit 1 is provided by the RIO, WIO OK FF. This flip-flop is set by the Operation Done signal when the previous data transfer is complete. As long as the flip-flop remains set, bit 1 is a logic 1 indicating that it is all right to proceed with a read or write command. The flip-flop is cleared by the Count 1 signal from the sequence counter. When the flip-flop is cleared, bit 1 goes to logic 0 indicating that a read or write command is being executed.

3-86. Status bit 2 is provided by the Interrupt Pending signal from interrupt request logic. This signal goes low causing the bit 2 output to be a logic 1 when any interrupt request is waiting for service. The signal will remain low until all interrupt requests are cleared.

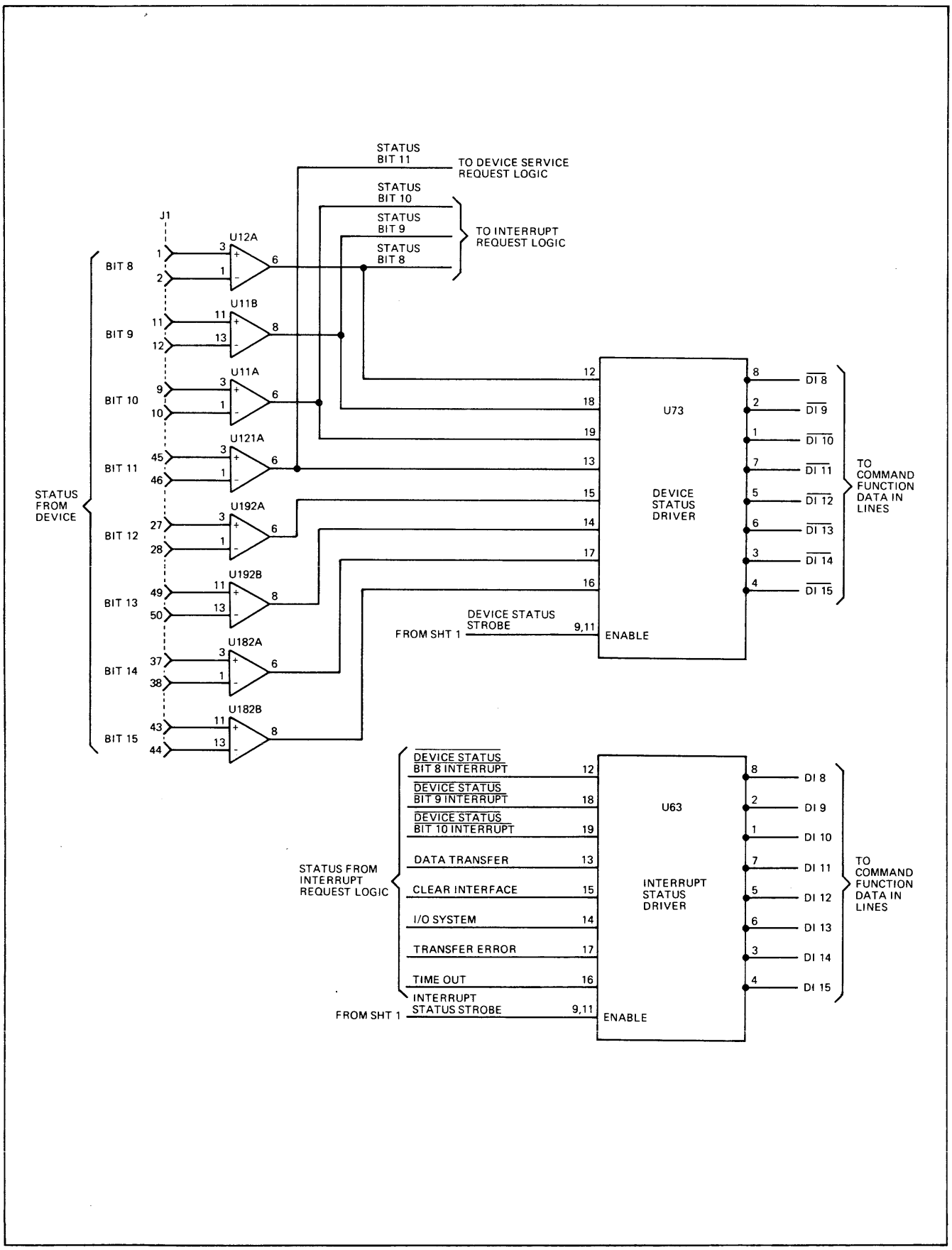
3-87. Status bits 3 and 4 are provided by the set-side output from the two sequence counter flip-flops. These bits are used mainly for troubleshooting when the interface PCA fails to complete a data transfer. The two bits can be decoded as follows:

BIT 3	BIT 4	DESCRIPTION
0	0	Sequence counter inactive, no data transfer in progress.
1	0	Request for operation sent to device.
1	1	Device responds to begin I/O operation.
0	1	Request for operation to begin byte transfer of second byte.



2186-36

Figure 3-29. Status Logic and Control (Sheet 1 of 2)



2186-37

Figure 3-29. Status Logic and Control (Sheet 2 of 2)

3-88. Status bit 5 is provided by the Device Flag signal from the device flag edge detector circuit. This bit is also used for troubleshooting to determine if the external device is responding to a request for I/O operation. During the I/O operation portion of the data transfer this bit is normally a logic 1.

3-89. Status bit 6 is provided by the Interrupt/Device status FF and indicates whether the lower half of the status word (bits 8 through 15) contains interrupt status or device status. If bit 6 is a logic 1 the status word contains the device status byte and if bit 6 is a logic 0 the status word contains the interrupt status byte.

3-90. Status bit 7 is connected to signal return and is permanently a logic 0.

3-91. Device Status Byte. The device status byte contains status signals developed by the external device to indicate the operating condition of the device. Up to eight different signals can be connected to the interface PCA through the differential line receivers. These signals are inverted by the device status driver when the driver is enabled by a Load Command signal with control word bit 12 (DO12) at logic 1. Four of the signals from the device may also be connected to other interface PCA circuits. The differential receiver output lines for bits 8, 9, 10 are connected to flip-flops in the interrupt request logic so that these signals can be used to develop Interrupt Request signals. Output from the differential receiver for bit 11 is connected to a flip-flop in the device service request logic so that this signal can be used to develop a Service Request signal.

3-92. Interrupt Status Byte. The interrupt status byte contains the signals from eight different flip-flops in the interrupt request logic. These signals indicate which circuits are requesting an interrupt. The signals are inverted by the interrupt status driver when the driver is enabled by a Load Command signal with control word bit 12 (DO12) at logic 0. The interrupt status driver is also enabled by a "not" Reset Interrupt or "not" Master Reset signal from the command function. For a description of the interrupt status bits, refer to paragraph 3-72.

3-93. DATA-IN LOGIC. The data-in logic consists of the circuits used to transfer data from the external device through the interface PCA to the command function data-in lines. These circuits consist of input signal differential receivers, multiplexers, registers, drivers, and control logic (see figure 3-30). This logic allows the interface PCA to accept data from the device in the form of one word of up to 16 bits or in the form of two bytes of up to eight bits each. When the byte transfer is used, the two 8-bit bytes are packed into one 16-bit word before being transferred to the command function data-in lines. The data-in transfer, whether one word or two bytes, occurs as a sequence of operations. First, data is transferred to the four-bit registers and then the data is strobed onto the data-in lines by a $\overline{\text{D READ STB}}$ or $\overline{\text{P READ STB}}$ signal. During byte transfers, an operation is added to do a double transfer of data into the four-bit registers. The following paragraphs describe one word and byte transfer operations.

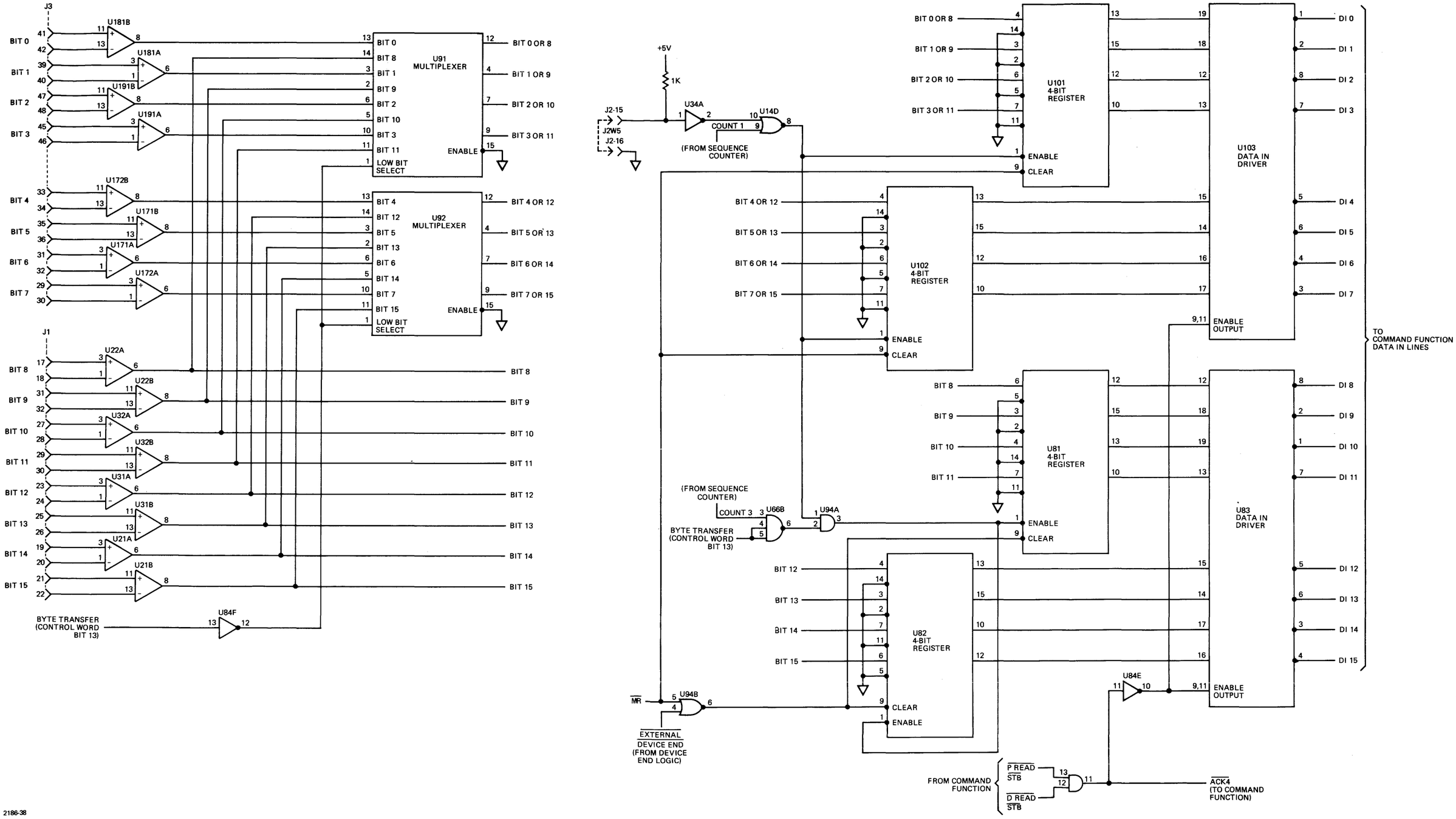
3-94. One-Word Data Transfer. For one-word data transfer, the Byte Transfer signal (control word bit 13) is low. This enables the multiplexers to pass the upper data byte (bits 0 through 7) from the differential receivers to the four-bit registers. The four-bit registers can be permanently enabled to accept data by installing jumper wire J2W5 or they can be enabled by the Count 1 signal from the sequence counter. When the Count 1 signal is used (J2W4 is not installed), the sequence counter controls the data transfer sequence and each output word is locked into the buffer when the sequence counter advances from count 1 to count 2.

3-95. Byte Transfer. For byte transfer where two eight-bit bytes are packed for transfer to the data-in lines, the Byte Transfer signal (control word bit 13) must be high. This enables the multiplexers to pass the first data byte (bits 8 through 15) from the differential receivers to the four-bit registers. The four-bit registers are then enabled by the Count 1 signal. Jumper wire J2W5 cannot be used during byte transfers. At this time, four-bit registers U101 and U102 contain the same eight data bits as four-bit registers U81 and U82. The second eight-bit byte is then received over the same input lines (bits 8 through 15). However, only four-bit registers U81 and U82 are enabled by the Count 3 signal from the sequence counter and the Byte Transfer signal. After both pairs of registers are loaded, a $\overline{\text{P READ STB}}$ or $\overline{\text{D READ STB}}$ signal transfers the full 16 bits to the data-in lines.

3-96. DATA OUT LOGIC. The data out logic consists of circuits used to transfer data from the command function data-out lines through the interface PCA and to the external device. These circuits consist of registers, multiplexers, differential line drivers, and control logic (see figure 3-31). This logic allows to interface PCA to transfer data to the external device in the form of one word of up to 16 bits or in the form of two bytes of up to 8 bits each. When byte transfer is used, the interface PCA unpacks one word of up to 16 bits and transfers the word to the device in two bytes of up to 8 bits each. The data-out transfer, whether one word or two bytes, occurs as a sequence of operations. First, the registers are enabled by a $\overline{\text{D WRITE STB}}$ or $\overline{\text{P WRITE STB}}$ signal from the command function and any data on the 16 data-out lines is stored in the registers. Outputs from the registers are connected to the multiplexer circuits and outputs from registers U114 and U124 are also connected directly to differential line drivers. The multiplexer circuit select input is normally low so that when byte transfer is not used (control word bit 13 is logic 0), data bits 8 through 15 are transferred through the multiplexer circuit to the differential line drivers. Since data bits 0 through 7 are connected directly to differential line drivers, the full 16-bit word is transferred. When byte transfer is enabled (control word bit 13 is a logic 1), the interface PCA data-out logic can operate in two different modes; normal write and test write. The following paragraphs describe operations in the two write modes.

3-97. Normal Write. For normal write operation, jumper wire J2W7 is not installed in the mating connector of J2. This allows the Write Transfer FF output and Byte Transfer signal to partially enable the Byte Write Select FF. The Write Transfer FF is direct set by either the $\overline{\text{D WRITE STB}}$ or $\overline{\text{P WRITE STB}}$ signal from the command function. This same signal initializes the qualifier logic so that the sequence counter advances from count 0 to count 1 (refer to figure 3-13). When the sequence counter returns to count 0, the Write Transfer FF is cleared by the Operation Done signal. The Byte Transfer signal goes high when control word bit 13 is a logic 1 and remains high until the control word is modified by a new control word. With the Byte Transfer signal high, the input to Byte Write Select FF, U66C-10, is high during sequence counter counts 1, 2, and 3. The other signal required to enable the Byte Write Select FF is provided by counts 0 and 1 of the sequence counter. This allows the Byte Write Select FF to provide, through inverter U116C, a high select signal to the multiplexer circuits during sequence counter counts 0 and 1 and a low select signal during counts 2 and 3. The high select signal causes bits 0 through 7 to be transferred to the device and the low select signal causes bits 8 through 15 to be sent to the device.

3-98. Test Write. Test write is enabled when jumper wire J2W7 is installed in the mating connector of J2. This allows the Byte Transfer signal and sequence counter counts 2 and 3 to enable the Byte Write Select FF without having the Write Transfer FF enabled. Therefore, the Byte Write Select FF is enabled during counts 2 and 3 of every byte transfer whether the byte transfer is for a read or write operation. Whatever data is on the multiplexer input lines is transferred to the device as if a normal write operation was performed. This feature is used during diagnostic testing to loop the output data back to the input and thereby check the packing and unpacking operations.



2186-38

Figure 3-30. Data In (from Device) Transfer Logic

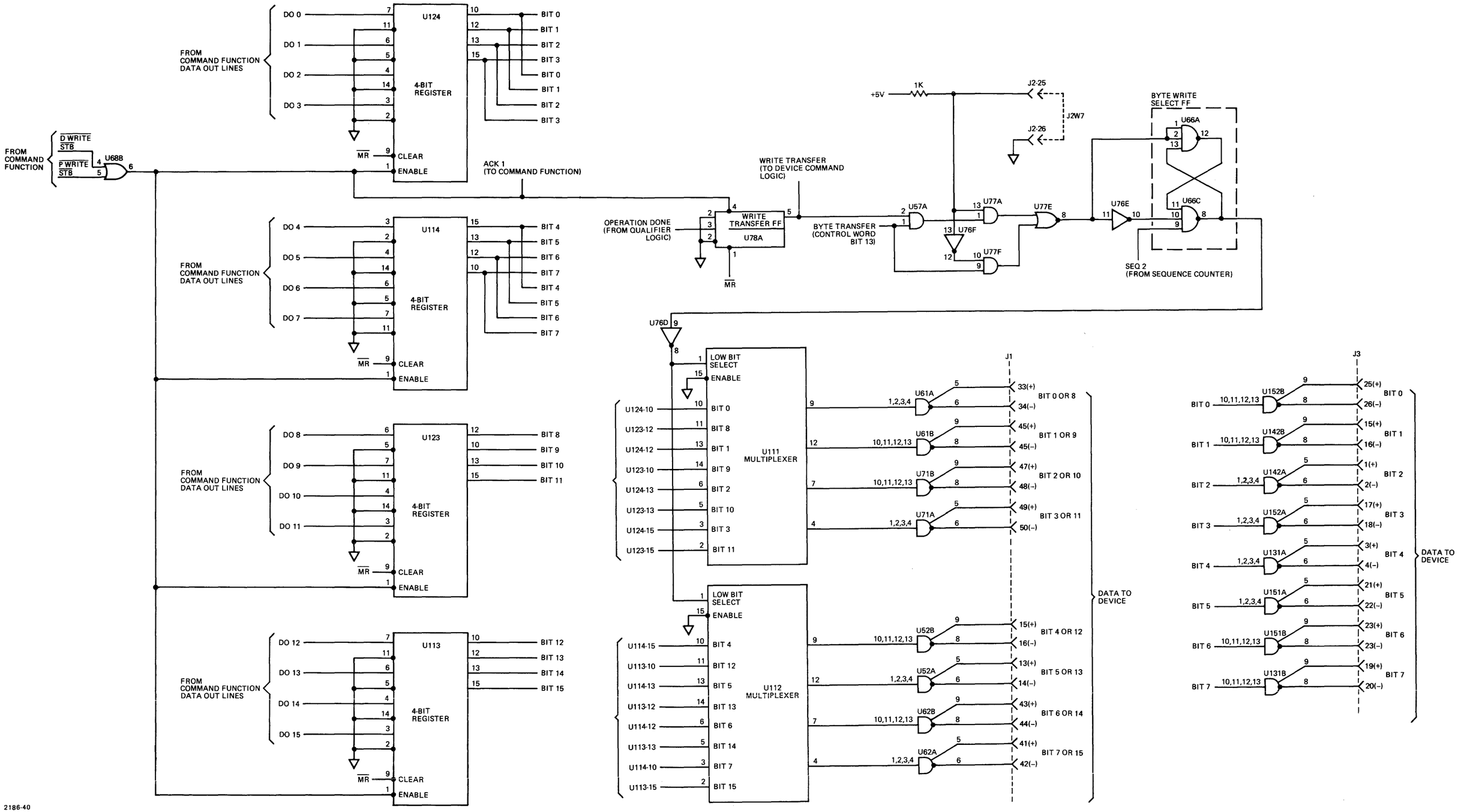


Figure 3-31. Data Out (to Device) Transfer Logic

4-1. INTRODUCTION.

4-2. This section contains general servicing information, preventive maintenance instructions, troubleshooting information, and corrective maintenance procedures applicable to the interface PCA. A logic diagram for the PCA is provided in the *System Diagrams Set.* Parts information for servicing and replacement is contained in the *HP 3000 Computer System Illustrated Parts Breakdown Manual.*

4-3. GENERAL SERVICING INFORMATION.

4-4. Paragraph 4-5 through 4-22 contain servicing information that may be commonly referred to from other areas of this manual.

4-5. SAFETY PRECAUTIONS.

4-6. When the interface PCA is being installed, removed, or placed on an extender PCA to perform maintenance and troubleshooting procedures, the computer system DC POWER switch must be set to STANDBY to remove power from the PCA connectors.

4-7. When the interface PCA is being connected to an external device, make sure there is no power at the interface PCA connectors and that the device power is off.

4-8. Failure to observe these precautions may result in damage to the interface PCA or the computer system connectors.

4-9. WIRING INFORMATION.

4-10. The interface PCA wiring information consists of the data necessary to connect the PCA in the HP 3000 computer system and to an interconnecting cable assembly. Figure 4-1 shows the numbering scheme for the interface PCA and interconnecting cable connector and table 4-1 provides a cross reference list of mating connector and interface PCA pin numbers. Table 4-2 lists functions for the jumper wires contained in the mating connector to connector J2. For details of interconnecting cable fabrication, refer to the *HP 3000 Computer System Installation Manual.*

4-11. JUMPER WIRE CONFIGURATION.

4-12. The interface PCA contains locations for installation of 12 jumper wires. These jumper wires are used to select various operating conditions and are located on the component side of the interface PCA as shown in figure 4-2. The following paragraphs describe the use of each jumper wire that can be installed on the interface PCA.

4-13. SERVICE REQUEST PRIORITY (W5). Jumper wire W5 selects one of 16 service request (SR) lines to the HP 30035A Multiplexer Channel. Each interface PCA serviced by the same multiplexer channel must be connected through a different service request line.

4-14. SELF-TEST POWER (W1). Jumper wire W1 is installed to provide power to the Diagnostic Hardware, part number 30049-60001. The diagnostic hardware is used for testing the interface PCA. When testing is complete, jumper wire W1 is removed.

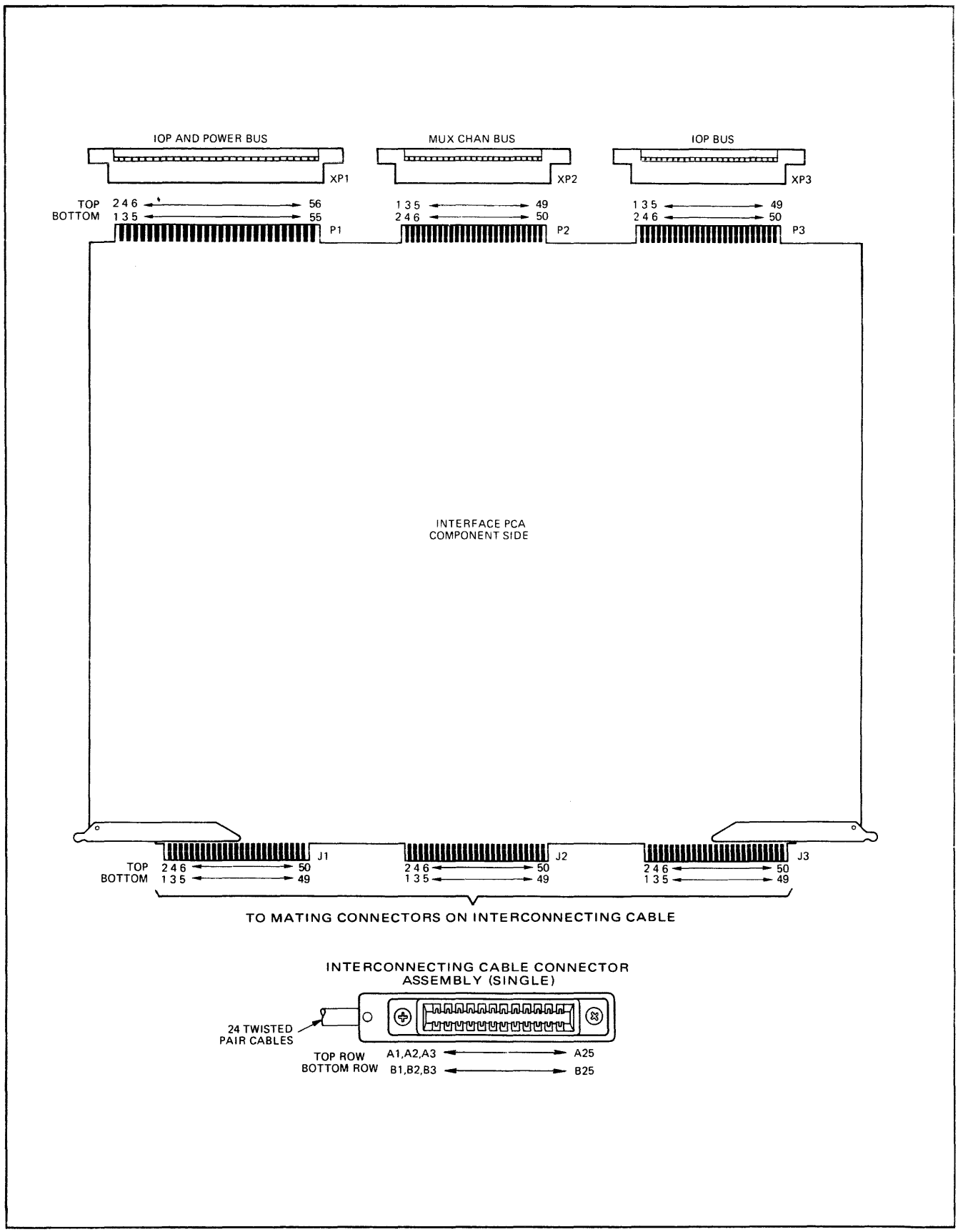
4-15. GROUP INTERRUPT MASK (W4). Jumper wire W4 selects the group interrupt mask. It can be connected to any of 16 data-out lines, to a permanent enable connection, or a permanent disable connection.

4-16. DRT ADDRESS PARITY (W2). Jumper wire W2 selects the DRT address parity depending on the configuration of jumper wires W6 through W13. For odd parity, this jumper wire is removed or installed, as required, so that the total number of positions where jumper wires (W2 and W6 through W13) are not installed is odd.

4-17. DEVICE NUMBER (W6 THROUGH W13). The configuration of these eight jumper wires determines the device number (address) and DRT address.

4-18. REQUIRED SERVICING EQUIPMENT.

4-19. A special test hood is required for diagnostic testing of the universal interface PCA when the PCA is not connected to an external I/O device. The special test hood is the HP 30049B Diagnostic Hardware for the Universal Interface. Descriptive information regarding the diagnostic hardware (test hood) is contained in a separate manual, part number 30049-90001. Diagnostic procedures using the test hood are contained in the *HP 3000 Computer System Manual of Diagnostics.* When diagnostic testing is performed with the interface card connected to a device, the test hood is not required and a different diagnostic test procedure is required to test the card and device together. The correct procedure is referenced from the applicable subsystem manual. The only other servicing equipment required is supplied with the HP 3000 Computer System.



2186-42

Figure 4-1. Interface PCA and Interconnecting Cable Connector Pin Numbering Scheme

Table 4-1. Interface PCA to Interconnecting Cable Connector Pin Cross Reference and Function Index

MATING CONNECTOR PIN NO.	INTERFACE CARD CONNECTORS			
	PIN NO.	J1 FUNCTION	J2 FUNCTION	J3 FUNCTION
A1,B1	1,2	Device Status Bit 8	Power Fail	Data Out Bit 2
A2,B2	3,4	Device Flag	Power On (PON), Ret	Data Out Bit 4
A3,B3	5,6	Write Delay	Master Clear Delay, Ret	Control Word Bit 6
A4,B4	7,8	Set Trans Err FF, Ret	Jumper Wire J2W1, Ret	Control Word Bit 8
A5,B5	9,10	Device Status Bit 10	Jumper Wire J2W2, Ret	Control Word Bit 7
A6,B6	11,12	Device Status Bit 9	Jumper Wire J2W3, Ret	Self-Test +5V, Ret
A7,B7	13,14	Data Out Bit 13	Jumper Wire J2W4, Ret	Self-Test +5V, Ret
A8,B8	15,16	Data Out Bit 12	Jumper Wire J2W5, Ret	Data Out Bit 1
A9,B9	17,18	Data In Bit 8	NC	Data Out Bit 3
A10,B10	19,20	Data In Bit 14	NC	Data Out Bit 7
A11,B11	21,22	Data In Bit 15	Jumper Wire J2W6, Ret	Data Out Bit 5
A12,B12	23,24	Data In Bit 12	NC	Data Out Bit 6
A13,B13	25,26	Data In Bit 13	Jumper Wire J2W7, Ret	Data Out Bit 0
A14,B14	27,28	Data In Bit 10	NC	Device Status Bit 12
A15,B15	29,30	Data In Bit 11	NC	Data In Bit 7
A16,B16	31,32	Data In Bit 9	NC	Data In Bit 6
A17,B17	33,34	Data Out Bit 8	NC	Data In Bit 4
A18,B18	35,36	Self-Test Logic, NC	Jumper Wire J2W8, Ret	Data In Bit 5
A19,B19	37,38	Control Word Bit 10	Jumper Wire J2W9, Ret	Device Status Bit 14
A20,B20	39,40	Device Command	Jumper Wire J2W10, Ret	Data In Bit 1
A21,B21	41,42	Data Out Bit 15	Control Word Bit 9	Data In Bit 0
A22,B22	43,44	Data Out Bit 14	Master Clear	Device Status Bit 15
A23,B23	45,46	Data Out Bit 9	Device Status Bit 11	Data In Bit 3
A24,B24	47,48	Data Out Bit 10	Clear Interface	Data In Bit 2
A25,B25	49,50	Data Out Bit 11	Device End	Device Status Bit 13

NOTES:

1. All connections to and from a device are made through differential line drivers and receivers. In the pin no. columns, positive (+) connections are listed first with the negative (-) connection for the same signal listed following the comma.

2. When used, jumper wires are connected in the mating connector hood between the two pins listed on the same line.

3. Abbreviation NC indicates no connection.

4-20. PRINCIPAL SERVICING POINTS.

4-21. Principal servicing points for the interface PCA are the interconnecting cable connectors J1, J2, and J3. These connectors carry all of the signals transferred between the interface PCA and external device. A list of signal names and connector pin numbers is provided in table 4-1.

4-22. The interface PCA does not contain fuses or adjustable controls.

4-23. PREVENTIVE MAINTENANCE.

4-24. Preventive maintenance for the interface PCA should be performed when the preventive maintenance routines for the HP 3000 Computer System are performed. Preventive maintenance consists of inspecting the interface PCA and the interconnecting cable assembly for burned or broken components, loose connections, and deteriorated insulating materials.

Table 4-2. Connector J2 Jumper Wire Functions

DESIGNATION	PIN NO.		WITH JUMPER WIRE REMOVED (NORMAL)	WITH JUMPER WIRE INSTALLED
	FROM	TO		
J2W1	7	8	I/O program continues normally after program control strobe.	I/O program halts after a program control stobe and waits for Device Status Bit 11 to continue operation.
J2W2	9	10	Leading edge of Device Flag signal advances sequence counter from Count 1 to Count 2.	Trailing edge of Device Flag signal advances sequence counter from Count 1 to Count 2.
J2W3	11	12	Device Command signal operates in response mode.	Device Command signal operates in pulse mode.
J2W4	13	14	Inhibits Device Status Bit 8 from causing an interrupt request.	Permits interrupt request at trailing edge of Device Status Bit 8.
J2W5	15	16	Data input registers enabled by sequence counter counts 1 and 3.	Data input registers permanently enabled.
J2W6	21	22	Trailing edge of Device Flag signal advances sequence counter from Count 2 to Count 3.	Leading edge of Device Flag signal advances sequence counter from Count 2 to Count 3.
J2W7	25	26	Normal write transfer operation.	Test write transfer operation.
J2W8	35	36	Inhibits Device Status Bit 9 from causing an interrupt request.	Permits interrupt request at trailing edge of Device Status Bit 9.
J2W9	37	38	Inhibits Device Status Bit 10 from causing an interrupt request.	Permits interrupt request at leading edge of Device Status Bit 10.
J2W10	39	40	Device Command sginal not inverted.	Device Command signal logic inverted with respect to Data Out lines.

4-25. TROUBLESHOOTING.

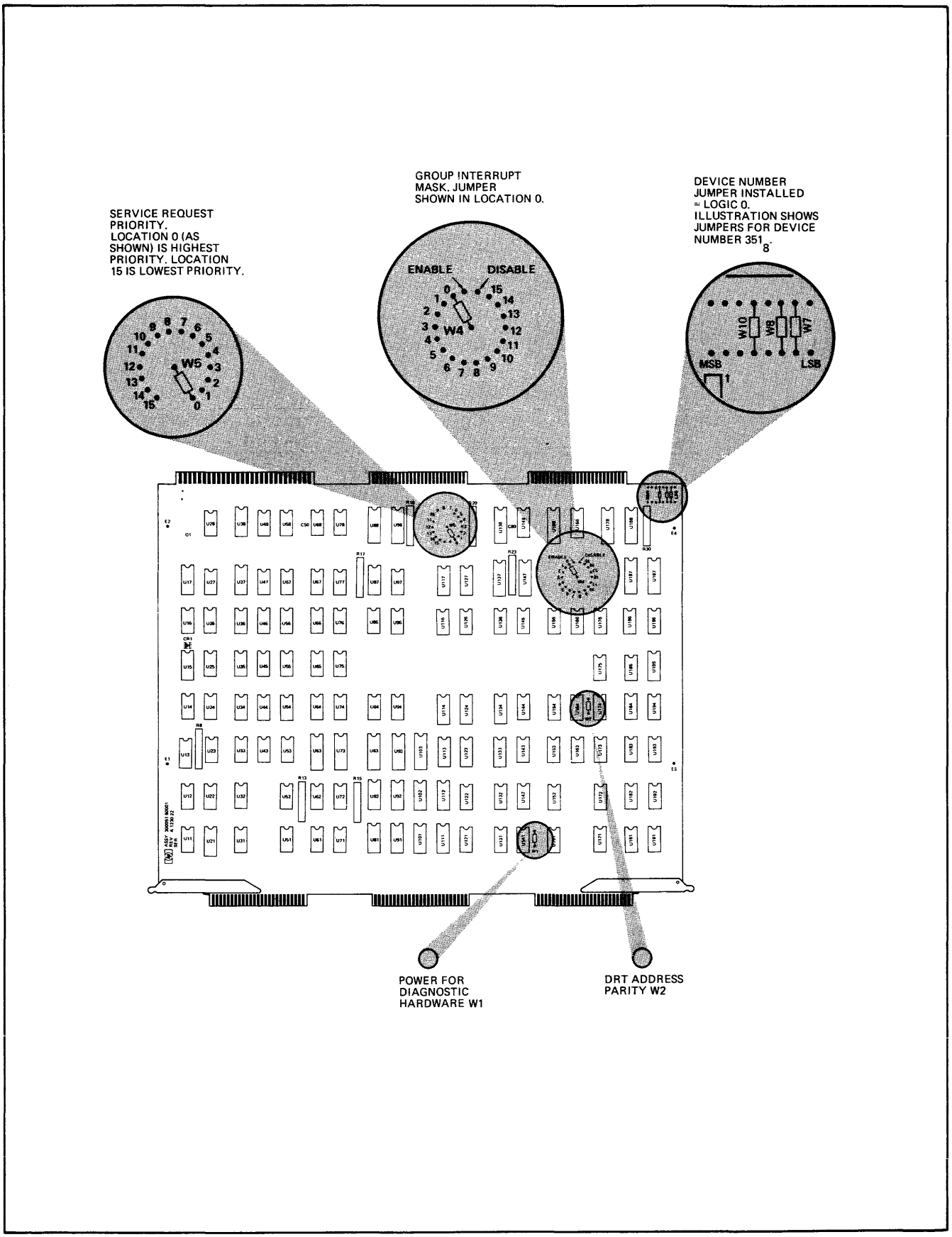
4-26. The interface PCA may be checked using the Diagnostic Program Procedures contained in the *Manual of Diagnostics.* This diagnostic checks the interface PCA functions through the HP 30049B Diagnostic Hardware for the Universal Interface PCA (self-test hood). If the interface PCA is connected in a subsystem, the applicable subsystem diagnostic program procedures should be performed.

4-27. Troubleshooting for the interface PCA is accomplished by performing the tests in the diagnostic program and analyzing any error halts that occur as the test is being run. Continuity checks of the interconnecting cable are performed using table 4-1. To further isolate a trouble, refer to the logic diagram for the PCA.

For parts location and replacement information refer to the *HP 3000 Computer System Illustrated Parts Breakdown Manual,* part number 03000-90021.

4-28. CORRECTIVE MAINTENANCE.

4-29. Corrective maintenance, as applicable to the interface PCA, is performed in the same manner as corrective maintenance for other printed-circuit assemblies in the HP 3000 computer system. No special procedures are required for the interface PCA.



2186-3

Figure 4-2. Interface PCA Jumper Wire Locations