INPUT/OUTPUT PROCESSOR (IOP)
Model 8000, 8001

IPU CONSOLE IOP

I/O EXPANSION CHASSIS
Model 8910

Technical Manual

August 1984

Publication Order Number: 303-000170-100

GOULD
Electronics

This manual is supplied without representation or warranty of any kind. Gould Inc., S.E.L. Computer Systems Division therefore assumes no responsibility and shall have no liability of any kind arising from the supply or use of this publication or any material contained herein.

HISTORY

The Input/Output Processor (IOP) Model 8000 Technical Manual, Publication Order Number **303-000170-000**, was printed August, 1980.

Publication Order Number **303-000170-100** (Revision 1) was printed March, 1981.

Publication Order Number **303-000170-101** (Change 1 to Revision 1) was printed May, 1982.

Publication Order Number **303-000170-102** (Change 2 to Revision 1) was printed December, 1982.

Publication Order Number **303-000170-103** (Change 3 to Revision 1) was printed April, 1983.

Publication Order Number **303-000170-104** (Change 4 to Revision 1) was printed August, 1984. The updated manual contains the following pages:

	* Change Number
Title page	4
Copyright page	0
History page Change 4	4
iii/iv Change 3	3
iii/iv Change 2	2
iii Change 1	1
iv	1
v	3
vi through x	1
xi/xii	3
xiii/xiv	0
1-1 through 1-7/1-8	3
2-1 through 2-4	0
2-5	1
2-6	3
2-7	3
2-8 through 2-17/2-18	0
3-1 through 3-4	0
3-5	3
3-6	0
3-7 through 3-9	3
3-10 through 3-31	0
3-32	3
3-33 through 3-48	0
4-1	3
4-2 through 4-13	0
4-14	1
4-15	4
4-16 through 4-16A/4-16B	3
4-17 through 4-29/4-30	0
5-1 through 5-7/5-8	0
A-1 through A-2	0
A-3	3
A-4 through A-7/A-8	0
B-1 through B-4	0
IN-1 through IN-4	0

* Zero in this column indicates an original page.

On a change page, the portion of the page affected by the latest change is indicated by a vertical bar in the outer margin of the page. However, a completely changed page will not have a full length bar, but will have the change notation by the page number.

HISTORY

The Input/Output Processor (IOP) Model 8000 Technical Manual, Publication Order Number **303-000170-000**, was printed August, 1980.

Publication Order Number **303-000170-100** (Revision 1) was printed March, 1981.

Publication Order Number **303-000170-101** (Change 1 to Revision 1) was printed May, 1982.

Publication Order Number **303-000170-102** (Change 2 to Revision 1) was printed December, 1982.

Publication Order Number **303-000170-103** (Change 3 to Revision 1) was printed April, 1983. The updated manual contains the following pages:

	* Change Number
Title page	3
Copyright page	0
iii/iv Change 3	3
iii/iv Change 2	2
iii Change 1	1
iv	1
v	3
vi through x	1
xi/xii	3
xiii/xiv	0
1-1 through 1-7/1-8	3
2-1 through 2-4	0
2-5	1
2-6	3
2-7	3
2-8 through 2-17/2-18	0
3-1 through 3-4	0

	* Change Number
3-5	3
3-6	0
3-7 through 3-9	3
3-10 through 3-31	0
3-32	3
3-33 through 3-48	0
4-1	3
4-2 through 4-13	0
4-14 through 4-15	1
4-16 through 4-16A/4-16B	3
4-17 through 4-29/4-30	0
5-1 through 5-7/5-8	0
A-1 through A-2	0
A-3	3
A-4 through A-7/A-8	0
B-1 through B-4	0
IN-1 through IN-4	0

* Zero in this column indicates an original page.

On a change page, the portion of the page affected by the latest change is indicated by a vertical bar in the outer margin of the page. However, a completely changed page will not have a full length bar, but will have the change notation by the page number.

HISTORY

The Input/Output Processor (IOP) Model 8000-A and I/O Expansion Chassis Model 8910 Technical Manual, Publication Order Number **303-000170-000**, was printed August, 1980.

Publication Order Number **303-000170-100** (Revision 1) was printed March, 1981.

Publication Order Number **303-000170-101** (Change 1 to Revision 1) was printed April, 1982.

Publication Order Number **303-000170-102** (Change 2 to Revision 1) was printed December, 1982. The updated manual contains the following pages:

	* Change Number
Title page	2
Copyright page	0
iii/iv Change 2	2
iii Change 1	1
iii	0
v through x	1
xi/xii	2
1-1	2
1-2 through 1-6	0
2-1 through 2-4	0
2-5	1
2-6 through 2-16	0
3-1 through 3-6	0
3-7	2
3-8	0
3-9	2
3-10	1
3-11 through 3-31	0
3-32	2
3-33 through 3-46	0
4-1 through 4-13	0
4-14 through 4-15	1
4-16 through 4-16A/4-16B	2
4-17 through 4-29	0
5-1 through 5-8	0
A-1 through A-8	0
B-1 through B-4	0
IN-1 through IN-4	0

* Zero in this column indicates an original page.

On a change page, the portion of the page affected by the latest change is indicated by a vertical bar in the outer margin of the page. However, a completely changed page will not have a full length bar, but will have the change notation by the page number.

MANUAL HISTORY AND REVISION INSTRUCTIONS

PUBLICATION NO. 303-000170-101

EQUIPMENT: IOP & IO Expansion Chassis

PURPOSE: To revise and update manual to incorporate changes for Model 8000-A.

MANUAL HISTORY

REVISION TYPE*	REVISION NO.	DATE ISSUED
Original	000	8/80
C	1	10/80
R	100	3/81
C	1	4/82

REVISION TYPE*	REVISION NO.	DATE ISSUED

*C = Change
R = Revision

REVISION INSTRUCTIONS: Delete and add pages as shown on the following table.

DELETE	ADD
Title Page	Title Page, Change 1
iii through xi/xii	iii through xi/xii, Change 1
2-5 and 2-6	2-5, Change 1 and 2-6
3-7 through 3-10	3-7, Change 1 through 3-10, Change 1
3-32	3-32, Change 1
4-14 through 4-16	4-14 through 4-16, Change 1
	4-16A/4-16B, Change 1

CONTENTS

Chapter Page

1 GENERAL DESCRIPTION

2 MICROPROGRAM

3 THEORY OF OPERATION

4 OPERATING AND PROGRAMMING

Chapter Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

GENERAL DESCRIPTION

1.1 Introduction

This technical manual contains or references information concerning the Model 8000 and 8001 Input/Output Processor (IOP), the Model 8910 I/O Expansion Chassis and the IPU Console IOP. They are designed and manufactured by Gould Inc., S.E.L. Computer Systems Division, Fort Lauderdale, Florida.

The information contained in this manual is presented in the following order:

Chapter 1 - General Description
Chapter 2 - Microprogramming
Chapter 3 - Theory of Operation
Chapter 4 - Operation and Programming
Chapter 5 - I/O Expansion Chassis (Optional)

In this manual, all references to IOP are applicable to IOP Models 8000, 8001, and IPU Console IOP, except that clock override and IPL panel functions are not supported by the IPU Console IOP.

An input/output processor (IOP) is an I/O multiplexing channel consisting of the following interrelated elements: a SelBUS interface, a multipupose bus (MPB) interface, the IOP proper, and system control panel (SCP) device dependent logic. The SelBUS interface provides the communications path between the IOP and the CPU, or the IOP and memory. The IOP proper has a control memory that contains the microprogram (firmware) for controlling the SelBUS and MPB interfaces. The IOP circuits consist of control logic for operating the MPB, the SCP interface, and the receiver/drivers necessary to communicate with the MPB controllers.

The IPU Console IOP is part of the IPU Console option (Model 3615). The IPU Console option consists of an alphanumeric CRT, the IPU Console IOP board, and cabling. The primary function of the IPU Console option is to provide an IPU with panel functions. It provides all the panel functions except initial program load (IPL) and clock override. It should be noted that the IPU console is dedicated for operation with an IPU and cannot be accessed by the CPU. This option is not available for all systems with an IPU.

Note

> The IPU Console IOP board is not interchangeable with the IOP Models 8000 and 8001.

The I/O Expansion Chassis provides power and the additional slots required to expand the MPB of a computer system to accept additional I/O controller boards.

1.2 SelBUS Description

The SelBUS is a high-performance, synchronous-time-division multiplexed bus that functions at a continuous rate of 26.67 million bytes per second. The completely passive SelBUS is physically a part of the CPU chassis backplane.

1.2.1 SelBUS Lines

The bidirectional SelBUS consists of 148 parallel lines: 32 data, 24 address, 4 memory bus controller inhibit, 2 identification (ID) tag (ECHO), 23 SelBUS priority, and 39 control and tag lines. Other lines on the SelBUS include power, ground, and spares.

The 32 data lines permit the simultaneous transfer of four eight-bit bytes or one full 32-bit word on the SelBUS. The 24 address lines address the selected IOP or memory.

The 23 SelBUS priority lines are unitarily coded and are used to establish access priority to the SelBUS, with a priority polling operation during each 150 nanosecond SelBUS cycle.

1.2.2 SelBUS Protocol

The SelBUS can handle several types of transfers between the central processing unit (CPU) memory and input/output processor (IOP). These transfers are listed below:

Write data or order transfer (WDOT)
Read data transfer (RDT)
Interrupt control transfer (ICT)
Read status transfer (RSTX)
Advance read status transfer (ARSTX)
Advance interrupt control transfer (AICT)
Data return transfer (DRT)
Error transfer (ET)
Memory write transfer (MWT)
Memory read transfer (MRT)
Memory read and lock transfer (MRLT)
Memory instruction read transfer (MIRT)

The protocol for handling these various types of transfers is essentially the same; basically, four steps are required:

1. Testing of the inhibit lines.
2. Priority polling on the SelBUS.
3. The transfer itself.
4. A transfer acknowledge (TA).

For a SelBUS device to communicate with memory during a read or write operation, the device must test the status of the inhibit line associated with the accessed integrated memory module (IMM). However, to initiate a series of memory transfers, a SelBUS device bypasses the testing operation for the first transfer and polls for SelBUS priority. If the device has the highest SelBUS priority, it executes its transfer during the next SelBUS cycle. If the IMM is not busy, the transfer is accepted, and a TA indication with a coded ID tag is returned to the device. If the IMM is busy, the transfer is rejected; however, the coded ID tag is still returned to the device. The ID tags tell the SelBUS device which IMM inhibit lines to test before attempting its next transfer.

The SelBUS priority polling operation does not use centralized priority polling logic. Rather, each SelBUS device is equipped with its own priority logic. During the SelBUS priority polling operation, each device wanting access to the SelBUS compares its priority with the priorities of the other devices. Access to the SelBUS, for the next 150 nanosecond cycle, is then granted to the highest active device on the SelBUS.

The actual SelBUS transfer is accomplished in one 150 nanosecond cycle. The destination address is placed on the 24 address lines, and data are placed on the 32 data lines. For a memory read operation, the destination address is used for the address of the SelBUS device requesting the data. After the data are fetched from memory, the device address that was buffered in the IMM is used to route the data back to the requesting device.

Therefore, a complete memory read operation requires two transfers: one to transfer the address of the memory location and another to return the data to the requesting device. Memory write operations require only one transfer on the SelBUS. This single transfer simultaneously transfers data and the memory address from the requesting device to the IMM during one 150-nanosecond cycle.

1.3 Physical Description

The input/output processor (IOP) is an, I/O multiplexing channel. The IOP combines the functions of a real-time option module (RTOM), system control panel (SCP), and system operator console device. These functions all share the same SelBUS interface and microprocessor. The controllers of the IOP are implemented on 12.5-by-15-inch boards. These boards plug into the multipurpose bus (MPB) slots of a split backplane or an I/O expansion chassis. Using this type of board connection, high packing density can be achieved.

The RTOM and SCP functions are implemented using the same SelBUS protocol that is currently being used. The MPB uses a subset of Extended I/O protocol. The IOP is able to operate as the SCP, RTOM, and multiplexing channel for this type of system. A jumper is provided on the IOP to allow selection of the SCP functions. If the RTOM function is required, the IOP physical address must be 72 (IPU) or 7E (CPU). This address enables the MPB to respond to address 72 or 7E and the RTOM to address 73 (IPU) or 7F (CPU). If the RTOM function is not desired, the physical address must be set to any address lower than 78. However, the IOP still uses two physical addresses (even and odd pair) on the SelBUS. This addressing method allows maximum system flexibility without additional hardware on the IOP board.

The IOP is implemented using four 2901 bit slice processors. The 2901 provides the required speed, power, and parallel processing capabilities. This processor and its three state bus allow shared processing for all of the functions and power to control the four interfaces resident on the IOP. Firmware instructions are provided to load a CRAM daughter board from the master or control console slave initial program load (IPL) ports. These CRAM commands are for future expansion and are not presently supported. The IOP and controller costs are reduced by using a single large scale integration (LSI) interface chip and/or proven bus protocol to perform the MPB functions.

The SelBUS interface is a dual bidirectional interface used for burst and nonburst mode operation. It utilizes two physical addresses on the SelBUS and allows for simultaneous data and command transfers from memory and the central processing unit (CPU). It can also be used for two simultaneous data transfers from memory. This operation increases system throughput by allowing the IOP to prefetch the next input/output command doubleword (IOCD) during a data-chained operation.

The master SCP/operator console device interface is a serial RS232C or current loop two-wire interface. This interface, with the addition of the current loop interface, is identical to the control console slave port. Data is formatted and output to the cathode ray tube (CRT) terminal in a serial manner. Switching between SCP and operator console modes is achieved using specific commands. These commands are entered into the operator console via the CRT. Anytime the system halts, the CRT automatically switches to the SCP mode, and the data on the screen is updated. Outputting data in the serial format allows the serial ports to be used for CRT or hard copy devices.

The control console slave SCP/Operator Console interface supports a full duplex RS232C modem interface. This port may be used locally or through a modem to control, monitor, or IPL the system. This action allows maximum flexibility for the master or control console slave user to exercise the system. Since the IOP may also be the system control device, macrodiagnostics or software programs may also be loaded from devices on the system and executed using this port. The control console slave port is activated by the master port so that it can operate in the SCP mode. A command is provided to return control to only the master port. This action provides system protection from the control console slave user.

The IOP MPB interface provides a type of FIFO buffering of data, allowing the IOP microengine time to perform RTOM and SCP functions during a high-speed (1.5M bytes/sec) burst mode transfer. The MPB is a medium-speed asynchronous bus. The MPB protocol allows easy implementation and low-cost bus interfaces on each of the 16 possible controllers. Each of the controllers may operate up to 16 separate devices; however, a total of 124 devices is allowed on any one MPB. Device addresses FC through FF are used by the system control panel ports and may not be used on the MPB. Multiple IOPs may be used to expand the number of controlled devices. IOP-to-IOP communication is not supported on the MPB.

1.4 Functional Description

The input/output processor (IOP) implements both the system control panel (SCP) and operator console functions in an interleaved manner. These two functions are available to I/O devices at each of the two ports on the IOP.

One port, the master, is designed for the support of a master device, such as a cathode ray tube (CRT) terminal, teletypewriter (TTY), or full duplex dedicated line modem.

The other port, the control console slave, also supports a complete set of control lines to interface with an asynchronous modem. Using this modem line interface, it is possible to perform SCP functions from a remote I/O device.

The displays on each of the CRTs (master and control console slave) are under software control in the operator console mode. In the SCP mode, after a reset, both master and control console slave displays have SCP information unless the control console slave CRT is deactivated from the master CRT. Activation of the control console slave CRT causes both the master and control console slave CRTs to display identical SCP information.

However, only one mode or the other can be activated at any one time. Automatic switching from the SCP mode to the operator console mode occurs when the RUN command is entered into the SCP or whenever the CPU is placed into the run mode. CPU transitions from RUN to HALT cause the CRT to be switched to the SCP mode and current data to be output to the display.

When both terminals are active in the SCP mode they operate in the echo mode. That is, when a character is input on a CRT it is stored by the firmware and then sent to each of the ports. As each CRT receives the echoed character, it is displayed on the screen. Since the character is echoed to both ports, characters input on one terminal appear on both terminals.

If either operator begins to input a message and the other operator inputs a character, the two characters appear sequentially on the screen. Thus, the command may be rejected as being incorrect. This action serves as an indication that the master operator wishes to take control. In the operator console mode, each port is independent and under software control.

The master and control console slave ports use a universal asynchronous receiver/transmitter (UART) integrated circuit to handle the serial formatting.

The master and control console slave ports have a choice of RS232C or current loop interface. Both ports support full duplex modem controls.

Each port has a set of jumpers to determine operating characteristics. These jumpers are read during the power-up system reset routine and used to initialize the UART. The master and control console slave ports have four jumpers dedicated to the selection of baud rates. Note that when the system is operating in the SCP mode, the same information is successively written to each port and output at the slower port rate. The ports operate independently in the operator console mode and will output data at their own rate; therefore, it is not necessary to set both ports to the same baud rate.

An eight-bit American Standard Code for Information Interchange (ASCII) code, with the most significant bit equal to zero, is used in conjunction with the UART. This particular bit arrangement looks like an ordinary seven-bit ASCII code; however, if a seven-bit ASCII code were used, the UART would treat it as incomplete data.

The SCP, implemented on the IOP, supports the initial program load (IPL) function. The two commands relating to the IPL function are as follows:

IPLCR	IPL device is at default address.
IPL=XXXXCR	IPL device is at address XXXX.

The IPL device may be connected locally, on the multipurpose bus (MPB), to the SelBUS, the master terminal port of the IOP, or the control console slave port. At a distance, it may be connected over a modem line to one of the two ports.

To use one of the IOP I/O ports as the IPL device, either the other port must be connected to a terminal or a device performing the IPL, or it must be able to act as a terminal so that the necessary commands can be input for the IPL operation.

The input of one of the IPLCR or IPL=XXXXCR commands causes the IOP firmware to notify the CPU to initiate its IPL routine. This routine sets up an input/output command doubleword (IOCD) at memory address 0000 with the command chain and suppress incorrect length (SIL) flags set. The IOCD calls for a binary read of 120 bytes, beginning at memory address 0000. The central processing unit (CPU) performs an initial load random access memory (RAM), with a dummy interrupt level, to communicate with the SCP. A command is then sent to the addressed device to start the I/O process. With the initial records read from the IPL device, a predetermined program status doubleword (PSD) is written into memory (beginning at memory address 0000) followed by the necessary number of chained IOCDs to enable loading of the entire IPL block. The PSD indicates where the software control should go at the end of the IPL sequence.

The IPL function is treated as a read operation by the IOP, except for notification of the CPU to initiate the IPL firmware sequence.

The task of initial configuration load is no longer a part of the IPL procedure but, instead, is left to the discretion of the software programmer. The IPU Console IOP does not support IPL.

1.5 IPU Console IOP Differences

The differences between the IPU Console IOP and other IOP's are listed below and then mentioned wherever applicable throughout this manual. These differences are as follows:

1. The IPL panel function is not available on the IPU Console IOP.
2. The clock override function is not supported by the IPU Console IOP.
3. The I/O classes associated with an IPU Console IOP have different designations than the standard I/O classes although they are identical in function. The I/O classes for the IPU Console IOP and the corresponding standard I/O class are listed as follows:

IPU Console IOP Class	Corresponding Standard I/O Class
7	F
B	3
6	E

4. The physical address settings for the panel and RTOM functions are different for an IPU Console IOP. They are defined in Table 1-1.

Table 1-1. Physical Address Settings

	Physical Address	
	Standard IOP	IPU Console IOP
RTOM	7F	73
SCP	7E	72

1.6 Specifications and Leading Particulars

Specifications and leading particulars are listed in Table 1-2.

Table 1-2
Specifications and Leading Particulars

Characteristics	Specifications
Physical dimension	15 inches wide by 18 inches deep
Burst transfer rate	1.5 megabyte/sec (maximum)
Word size	
Data bus	16 bits
Microinstruction	48 bits
Directly addressable memory	16 megabytes (maximum)
Controller burst time	537 microseconds
Interrupts	17 (16 associated with the real-time option module (RTOM) and 1 associated with the input/output processor (IOP))
Protocol	
Input/Output processor (IOP)	Extended I/O
Standard MPB controllers	8 line async communications controller, Line Printer/Floppy Disc controller, etc.
Orders	Pulse and level
MPB bus length	20 feet.

CHAPTER 2

MICROPROGRAM

2.1 Firmware Control Character Sequences

The three separate significant input/output processor (IOP) port command sequences are listed below:

1. @@P - Switches the ports to the system control panel (SCP) mode.
2. @@C - Switches the ports to the operator console mode.
3. @@A - Operator console attention.

2.2 Input/Output Processor (IOP) Master and Control Console Slave Port Operation

On power-up, the ports are initialized in the system control panel (SCP) mode, with the secondary panel enabled. In order to disable the control console slave port as an additional SCP it is necessary to input 'PRIP (CR)' on either SCP port.

The command sequence @@P is not recognized on the control console slave port when the secondary panel is not enabled.

After the information is passed to the CPU, the ports are switched to the operator console mode during IPL. To switch back to the SCP mode it is necessary to subsequently issue the @@P command.

When connected to a modem and the communication line drops during an IPL sequence (modem carrier on drops), the operator console maintains its data terminal ready (DTR) line and automatically reconnects the communications line. If the communication line drops in the SCP mode, the SCP function of the control console slave port is automatically disabled.

The read with command sequence recognition command is used by the software for a read operation from the operator console devices.

In this mode, all data, including a command sequence, are passed to memory before a mode switch or attention command occurs. If no input/output command doubleword (IOCD) is pending for the operator console device, the command sequence is recognized even though no data is passed to memory.

Loading through these ports requires that the data stream be scanned and protected from unwanted control sequences.

Upon power-up, the ring detection circuitry is disabled and will continue to be disabled (even in the operator console mode) until a disconnect command is issued to the read or write subchannel of the port. The ring indication is only reported on the read subchannels as attention status and cannot be reported a second time until a disconnect command to that port has been issued.

Ring detection and the subsequent reporting of attention status to the software is only enabled when the IOP is in the operator console mode.

2.3 System Control Panel (SCP) Firmware

The SCP firmware items are listed below:

Master reset routine
Data interrupt routine
Idle routine
SCP command decode routine
Console receive routine
Operator error routine
Nonpresent memory error routine
IOP console routine
SelBUS error routine
ASCII-to-binary conversion routine
Binary-to-ASCII conversion routine
Address stop clear register routine
Stop routines
SCP operator console mode selection routine
Clear memory routine
Write to CRAM routine (future expansion)
Address CRAM at a specific address routine
Read control switches routine
Write to control switches routine
Read effective address routine
Execute CRAM routine (future expansion)
Read general purpose registers routine
Halt routine
Initial program load routine
Memory address read routine
Memory address read virtual routine
Memory data write routine
Message routine
Clock override routine
Primary/secondary panel routine
Read program status doubleword routine
Write program status doubleword routine
Read program status word routine
Write to general purpose register A (GPR007) routine
Write program status word routine
Reset command routine
Run routine
Instruction step routine
Advance read status transfer/read status transfer routine
Data return transfer routine
Write data or order transfer routine
Display update routine
Convert number field to binary routine

2.3.1 Master Reset Routine

The master reset routine resets the universal asynchronous receiver/transmitters (UARTs). The UARTs are initialized by loading their internal registers with the appropriate parameters needed for proper operation. Baud rate jumpers, on the device interface (DI) board, are then read and loaded into the UARTs. For proper terminal operation, the jumpered baud rates must match the terminal baud rates. The master reset routine also resets the address stop compare registers and enables the run/halt transition interrupts.

2.3.2 Data Interrupt Routine

With the SCP in the panel mode, the data interrupt routine reads American Standard Code for Information Interchange (ASCII) UART characters. As the ASCII characters are input in the form of a command, they are stored in the command buffer. An ASCII character counter and a register, in scratch RAM, keep track of the next available location in the command buffer. These registers are incremented after each character is read.

The data interrupt routine then compares each incoming character with an ASCII carriage return (CR) indication. If the CR indication is at the end of a command, the command is decoded and the routine continues processing the incoming data. If the CR indication is the first ASCII character, the data interrupt routine checks for a previously entered command in the command buffer (at this time, the command buffer has not been written over). If any of the following commands are detected in the command buffer, they are processed: control RAM address (CRMA), memory address (MA=), MA virtual (MAV=), memory data (MD=), or STEP. The memory address buffer is incremented, where applicable, during the processing of these commands. If these commands are not in the command buffer, the data interrupt routine branches to the operator error routine.

If an equals (=) character is detected as the first character inputted, the firmware checks for a CRMD or MD command in the buffer. If either of these commands are in the buffer the ASCII count is adjusted to place the new numerical value in the correct location of the buffer. If either command is not detected, the routine branches to the operator error routine.

The data interrupt routine then checks the first character for a line feed (LF) indication. If the LF command is detected, the command, which is stored in the command buffer, is repeated.

2.3.3 Idle Routine

The idle routine has two parts: idleloop for the SCP idle mode and idle for the operator console idle mode. The SCP idle mode checks the scratch RAM flag register to see if the SCP needs to output nonpresent memory or operator error messages, slashes, or a full display of data. The SCP idle mode also checks to see if a program status doubleword (PSD) operation is in progress.

The operator console idle mode reads and initiates the operator console device command stored in memory (input/output command double word, etc.), transmits data, and checks for a ring in indication.

2.3.4 SCP Command Decode Routine

After a CR indication is detected, the SCP command decode routine decodes the command in the command buffer and vectors the program to the individual command routine.

The SCP command decode routine also checks for the central processing unit (CPU) halt command. The halt command is required before certain SCP commands can be processed.

If a valid command is not decoded or a halt indication not obtained, the SCP command decode routine branches to the operator error routine.

2.3.5 Console Receive Routine

The console receive routine reads the operator console mode characters input to the UARTs and looks for the @@P character sequence (to switch to the SCP mode) or the @@A character sequence (for an operator console attention indication). It then processes these operator console device input characters.

2.3.6 Operator Error Routine

The operator error routine outputs OPERATOR ERROR (CR) // to the SCP CRTs when an invalid command is input to the SCP.

2.3.7 Nonpresent Memory Routine

The nonpresent memory error routine outputs NON PRES MEM (CR) // to the SCP CRTs when an invalid memory address is input to the SCP.

2.3.8 SelBUS Error Routine

The SelBUS error routine outputs SELBUS ERR (CR) // to the SCP CRTs whenever SCP SelBUS protocol is not followed (i.e., a WDOT is received in response to a DRT when a WDOT is not expected or an incorrect RSTX is received by the SCP firmware).

2.3.9 ASCII-To-Binary Conversion Routine

The ASCII-to-binary conversion routine converts the ASCII hexadecimal characters in the most significant byte of register one into a four-bit binary number. This binary number is then stored in nibble one of register one.

2.3.10 Binary-to-ASCII Conversion Routine

The binary-to-ASCII conversion routine converts binary coded hexadecimal, in nibble one of register one, to ASCII code. This ASCII-coded number is then stored in the most significant byte of register one.

2.3.11 Address Stop Clear Register (AS, IS, RS, and WS) Routine

The address stop clear register routine clears the compare register and disables the stop command for the address compare, instruction fetch, read, and write operand instructions.

2.3.12 Stop (AS, IS, RS, and WS) Routines

The stop routines set up the compare address for the address stop, instruction fetch stop, and read and write operand stop instructions.

2.3.11 Address Stop Clear Register (AS, IS, RS, and WS) Routine

The address stop clear register routine clears the compare register and disables the stop command for the address compare, instruction fetch, read, and write operand instructions.

2.3.12 Stop (AS, IS, RS, and WS) Routines

The stop routines set up the compare address for the address stop, instruction fetch stop, and read and write operand stop instructions.

2.3.13 SCP/Operator Console Mode Selection Routine

The @@P command sets the SCP mode flag. The @@C command resets the SCP mode flag. The resetting of the SCP mode flag causes the firmware to switch to the idle routine so that the operator console mode can be entered.

2.3.14 Clear Memory (CLE) Routine

The clear memory routine clears the first contiguous block of main memory by writing zeros into main memory until a nonpresent memory error condition is detected. A nonpresent memory error indication is not displayed, and the clear memory (CLE) command is terminated.

2.3.15 DELETED

2.3.16 DELETED

2.3.17 Read Control Switches (CS) Routine

The read control switches (CS) command reads and displays the dedicated control switches memory address that is sent to the SCP during the execution of a reset command.

2.3.18 Write To Control Switches (CS) Routine

The write to control switches (CS=XXXX(CR) command writes converted data (XXXX) into the dedicated control switch address. The data are then displayed on the SCP cathode ray tube (CRT) terminal.

2.3.19 Read Effective Address (EA) Routine

The read effective address routine requests the effective address of the instruction currently in the program status word (PSW). The effective address is fetched from the CPU and displayed on the SCP CRT.

2.3.20 DELETED

2.3.21 Read General Purpose Registers (GPR) Routine

The read general purpose registers routine requests that the data stored in each of the general purpose registers (GPRs) be displayed on the SCP CRT. The data is first converted into ASCII code before they are displayed on the SCP CRT.

2.3.22 Halt (HALT) Routine

If the CPU is in the run mode, the halt routine requests that the CPU halt and wait for a new program status word (PSW). Note that the PSW is actually word one of the program status doubleword (PSD). If the CPU is already in the halt mode, the halt routine requests and displays the current PSW.

2.3.23 Initial Program Load (IPL) Routine

The IPL routine sends an IPL pulse, with device address 0000, to the CPU. This action causes the CPU to use a default IPL address. Note that the IPL pulse causes the CPU to send an ARSTX/RSTX pair requesting a send IPL data command.

For the IPL=XXXX command, XXXX is converted from ASCII code to binary and stored in the send data register so that it can be sent to the CPU as the IPL device address. The IPU Console IOP does not provide this function.

2.3.24 Memory Address Read (MA) Routine

The memory address read (MA) routine converts the input memory address from ASCII code to binary, stores the converted memory address in the binary address buffer (for future use), and requests memory data. The memory address and the data at that address are then displayed on the SCP CRT.

2.3.25 Memory Address Read Virtual (MAV) Routine

The memory address read virtual (MAV) routine requests the CPU to convert the input memory address (virtual) to a real address. The real address is then used to read the data contained at that memory address. The real address and the data at this address are then displayed on the SCP CRT.

2.3.26 Memory Data Write (MD) Routine

The memory data write routine converts input memory data from ASCII code to binary. It then writes the data into memory at the current location indicated by the binary address buffer. (Note that the memory location stored in the binary address buffer comes from the MA instruction.)

If any memory reference commands have been executed by the SCP between the MA and memory data (MD) instructions, the data may have been written into the wrong memory location. Therefore, care must be exercised by the SCP operator.

2.3.27 Message (MSGE) Routine

The MSGE routine allows interoperator messages to be displayed on the SCP CRTs. Carriage returns may be inserted by the operators to input new lines. This action will not execute a mode change.

To exit the MSGE mode and return to the SCP mode, two successive carriage returns must be entered into the SCP. Note that the message characters are not listed as SCP command characters.

2.3.28 Clock Override (OVR) Routine

The clock override routine pulses the clock override line to the CPU, causing the clock override flip-flop to toggle. This action causes 'OVR' to be displayed on the SCP CRT if the clock override condition has been set, or removed from the display if the condition has been reset. The IPU Console IOP does not support OVR.

2.3.29 Primary/Secondary Panel Routine

When only the master port is to be placed into the SCP mode, the operator enters 'PRIP(CR)' into the operator console. The PRIP command initiates the primary panel routine and disables the control console slave port. When both the master and control console slave ports are to be placed into the SCP mode, the operator enters 'SECP' into the operator console, thus initiating the secondary panel routine.

2.3.30 Read Program Status Doubleword (PSD) Routine

The read PSD routine reads the second word of the CPU PSD. The PSD flag is set so that the first word of the PSD will also be read and the resulting doubleword displayed on the SCP CRT.

2.3.31 Write PSD Routine

The write PSD routine converts the input data from ASCII code to binary. It also writes the new second word of the PSD to the CPU, using the SCP bus protocol (ARSTX/RSTX send function, data return transfer (DRT), ARSTX/RSTX send data, and DRT sequence).

2.3.32 Read Program Status Word (PSW) Routine

The read PSW routine initiates a request to the CPU for a PSW (first word of a PSD). It then checks to see if a PSD read operation is in progress. If a PSD read operation is not in progress, the next instruction, as indicated in the PSW, is read and the PSW and the instruction are displayed on the SCP CRT.

2.3.33 Write To GPR Register A (GPR0-7) Routine

The write to GPR register A routine converts the input ASCII-coded data to binary. The data are then written into GPR 'A', using the CPU SCP bus protocol (ARSTX/RSTX send function, DRT, ARSTX/RSTX send data, and DRT sequence).

2.3.34 Write PSW (PSW) Routine

The write PSW routine converts input data from ASCII code to binary and writes the new PSW information (PSD word one) to the CPU, using the SCP bus protocol.

2.3.35 Reset Command (RST) Routine

The reset command routine sends a reset pulse to the CPU.

2.3.36 Run (RUN) Routine

The run routine causes the SCP to switch to the operator console mode. If the CPU is in the halt mode, the CPU run/halt line is pulsed. If the CPU is in the run mode, the CPU run/halt line is not pulsed and zeros are then output in the number fields of the SCP CRT display.

2.3.37 Instruction Step (STEP) Routine

Using the CPU SCP bus protocol, the instruction step routine requests that the CPU execute the next instruction.

2.3.38 Advance Read Status Transfer/Read Status Transfer (ARSTX/RSTX) Routine

The ARSTX/RSTX routine causes the ready line to be pulled in response to an ARSTX instruction. This instruction indicates to the CPU that the SCP is ready for an RSTX.

The flag register is checked to see if an ARSTX is expected (i.e., requested ARSTX send function in response to the firmware pulling the CPU SCP attention line).

If the ARSTX is unexpected (the CPU halts and has PSW information to send), the ARSTX/RSTX routine prepares the DRT to return for the PSW.

The type of ARSTX/RSTX pair is then identified (send function, send data, send IPL data, send function for control switches, or stop acknowledged).

For a send function ARSTX/RSTX pair, a branch to the DRT routine occurs to send a prepared response to the ARSTX/RSTX pair. This prepared response occurs during the execution of various SCP commands.

For a send data ARSTX/RSTX pair, the send data register data (previously loaded by the SCP command sequence) are sent as the DRT.

The send data IPL ARSTX/RSTX pair is treated in the same manner as the send data ARSTX/RSTX pair.

The send function for control switches ARSTX/RSTX pair occurs after the reset command is executed. The ARSTX/RSTX routine prepares the appropriate DRT response and branches to the DRT routine.

The acknowledge address stop ARSTX/RSTX pair responds by requesting PSW information through a DRT.

2.3.40 Write Data or Order Transfer (WDOT) Routine

The WDOT routine receives returned information (PSWs, PSDs, GPR contents, etc.) from the CPU. Binary data are converted to ASCII code and placed into the ASCII data buffer. Once the data are assembled, the WDOT routine branches to the appropriate SCP command routine to process the data.

2.3.41 Display Update Routine

The display update routine checks the various lines of the CPU and sets up the appropriate responses into the display register for the machine state indicators. The contents of the display register are output to the UARTs.

2.3.42 Convert Number Field To Binary Routine

The convert number field to binary routine converts the entire number field into binary data using the ASCII-to-Binary Conversion Routine discussed earlier. In the binary conversion, procedure binary buffers 0, 1, and 2 are utilized. Data are then stored in the binary buffer of scratch RAM.

The address of the ASCII data buffer, in scratch RAM, is loaded into binary register three. For a more convenient display layout, the binary number is stored, starting with binary buffer 0, with the MSB first.

2.4 Input/Output Processor (IOP) Microword Formats

The IOP uses a 48-bit-wide microword that allows multiple functions to be performed in a single instruction. For example, only one instruction is required to load a literal into the arithmetic logic unit (ALU), execute a data word test to set up a branch, and issue more orders. The 48-bit-wide microword has seven different formats (0 through 6). Each format allows certain functions to be performed. A complete test structure appears in formats 0 and 4. Formats 1, 2, and 3 contain a minitest field (three enables and a single bit indicating 0 or 1). In essence, the single bit actually indicates test 0 or test 16. The general format for the 48-bit-wide microword is shown in Figure 2-1.

Bits 0 through 2, with bit 0 the most significant bit (MSB), contain the number of the format (0 through 6).

Bits 3 through 7, with bit 3 the MSB, contain the source address external to the ALU. Bits 4 through 7 can also contain the internal register A address, as shown in Table 2-1.

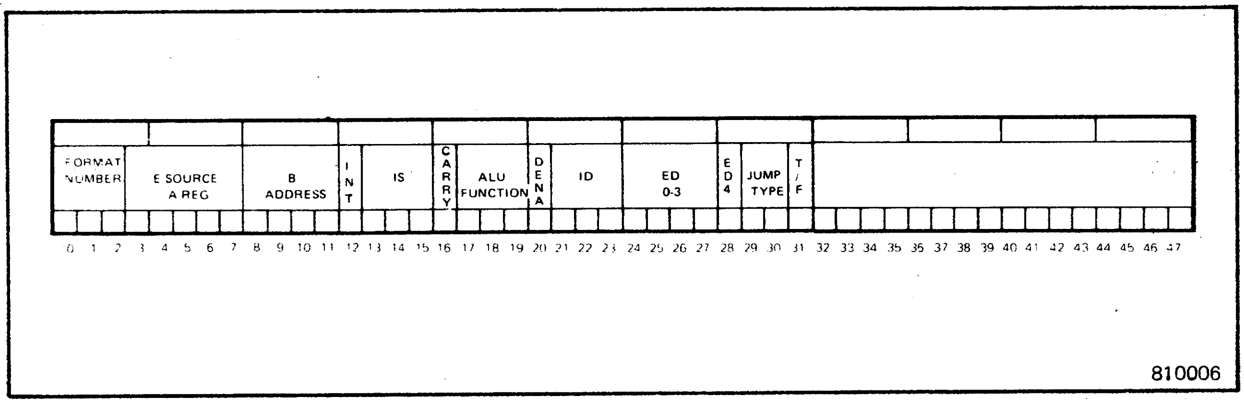


Figure 2-1. IOP General Format for the 48-Bit-Wide Microword

Note that both the source address and the internal register A address can be used in one instruction if bits 4 through 7 match.

Bits 8 through 11, with bit 08 the MSB, contain the internal 2901 register B address, as shown in Table 2-2.

Table 2-1
Internal 2901 Register A Address

Value	DBUS Source/2901 Register A Address
0	NOS/R(0)
1	R(1)
2	CPU data register upper/R(2)
3	CPU data register lower/R(3)
4	Burst data read upper/R(4)
5	Burst data read lower/R(5)
6	Nonburst data read upper/R(6)
7	Nonburst data read lower/R(7)
8	FIFO data/R(8)
9	FIFO swap/R(9)
A	FIFO counter/R(A)
B	Read data/R(B)
C	Read UART switches/R(C)
D	Read rate switches/R(D)
E	Parallel poll register/R(E)
F	R(F)
10	CPU destination register
11	CPU destination register
12	RAM data
13	RAM direct data
14	RAM swap data
15	RAM swap direct data
16	RAM counter

Table 2-2
Internal 2901 Register B Address

Value	2901 Register B Address
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	A
B	B
C	C
D	D
E	E
F	F

Bit 12, when equal to one, enables the interrupt disable circuitry.

Bits 13 through 15 contain the ALU internal source (2901 decode) information, as shown in Table 2-3.

Bit 16, when equal to one, forces a carry bit onto the ALU function information.

Bits 17 through 19 contain the ALU (2901) function information, as shown in Table 2-4. Note that bit 16 is also shown as either 1, 0, or X. Bit 17 is the MSB.

Bit 20, when equal to one, forces the ALU output onto the data bus (DBUS).

Bits 21 through 23, with bit 23 the MSB, specify the internal ALU destination for the 2901, as shown in Table 2-5.

Table 2-3
ALU Internal Source Information

Value	ALU Internal Source Information (R,S)
0	R(A), Q
1	R(A), R(B)
2	0, Q
3	0, R(B)
4	0, R(A)
5	LIT or EXT SRC, R(A)
6	LIT or EXT SRC, Q
7	LIT or EXT SRC, 0

Table 2-4
ALU Function Information

Bit 16	Value of Bits 17-19	ALU Function
0	0	R+S
1	0	R+S+1
0	1	S-R-1
1	1	S-R
0	2	R-S-1
1	2	R-2
X	3	R or S
X	4	R and S
X	5	R and S
X	6	R EX-OR S
X	7	R EX-NOR S

Table 2-5
Internal ALU Destination Information

Value	ALU Destination
0	Q
1	NOD
2	R(B)
3	R(B)
4	Right shift R(B) and Q individually
5	Right shift R (B)
6	Left shift R(B) and Q individually
7	Left shift R(B)

Bits 24 through 27 contain the four least significant bits (LSBs) of the external DBUS destination, as shown in Table 2-6.

Bit 28 contains the MSB of the external DBUS destination.

Bits 29 through 30 identify the type of jump used, as defined below:

0 - Normal branch
1 - Computed branch
2 - Push program counter (PC) and branch
3 - Pop PC

Table 2-6
External DBUS Destination

Value	DBUS Destination
0	DBUS
1	Not used
2	Nonburst data write upper
3	Nonburst data write lower
4	Data write upper byte
5	Data write lower byte
6	Not used
7	Not used
8	MAR upper
9	MAR uower
A	MAC upper
B	MAC lower
C	RAM data
D	RAM (12)
E	RAM counter
F	Not used
10	Upper address halt
11	Lower address halt
12	FIFO data
13	FIFO swap
14	FIFO counter
15	Load data
16	Load control

Bit 31 is the true/false (T/F) bit. When this bit is zero, a branch condition true exists.

Bits 32 through 47 specify the remaining functions associated with the seven formats.

2.4.1 Format 0

Format 0 contains information for the eight-bit test, four-bit (mini) pulse order, and four-bit hop address, as shown in Figure 2-2. Format 0 does not contain the individual enable signals for the pulse order group; therefore, when this format is decoded, only pulse order group A (bits 40 through 43) is produced.

Bits 00 through 31 contain the same information that was previously discussed.

Bits 32 through 34 enable the test group.

Bits 35 through 39 specify the test number.

Bits 40 through 43 contain the four-bit pulse order group A information.

Bits 44 through 47 contain the four-bit hop address. Note that all branches are absolute.

2.4.2 Format 1

Format 1 contains information for the four-bit test, eight-bit level order, and four-bit hop address, as shown in Figure 2-3. Format 1 is the only microword format that is associated with level orders.

Bits 00 through 31 contain the same information that was previously discussed.

Bits 32 through 34 enable the test group.

Bit 35 specifies the test type, as shown below:

0 for test 0
1 for test 16

Bit 36 is the set/reset (S/R) bit associated with the level order field.

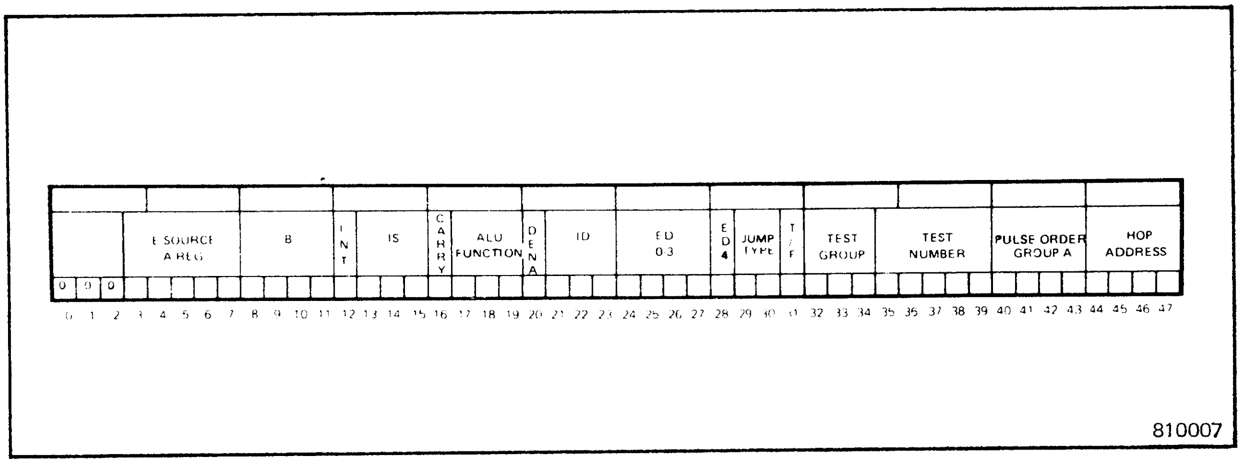


Figure 2-2. IOP Microword Format 0

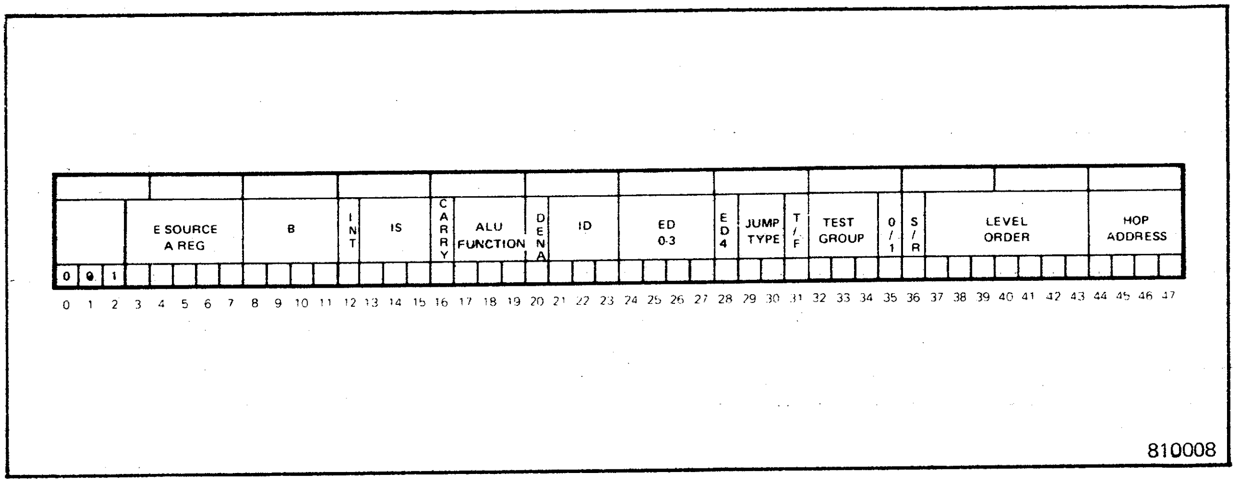


Figure 2-3. IOP Microword Format 1

Bits 37 through 43 (MSB=bit 37) specify the level order used. Level orders can be set high or low, using bit 36 (the S/R bit).

Bits 37 through 43 are used as a one-out-of-128 decode, thus allowing a total of 128 level orders.

Bits 44 through 47 contain the four-bit hop address. Note that all branches are absolute.

2.4.3 Format 2

Format 2 contains information for the four-bit test, eight-bit pulse order (large pulse order field), and four-bit hop address, as shown in Figure 2-4. Microword format 2 can be split into two different four-bit fields; bits 40 through 43 are actually a decode of 1-of-16. Bits 36 through 39 determine which of the four pulse orders can be produced simultaneously, using a single bit enable field. Note that if all four pulse orders (pulse order groups A, B, C, and D) are simultaneously required, they all need the same decode address of the LSB.

Bits 00 through 31 contain the same information that was previously discussed.

Bits 32 through 34 specify the test group of the four-bit test.

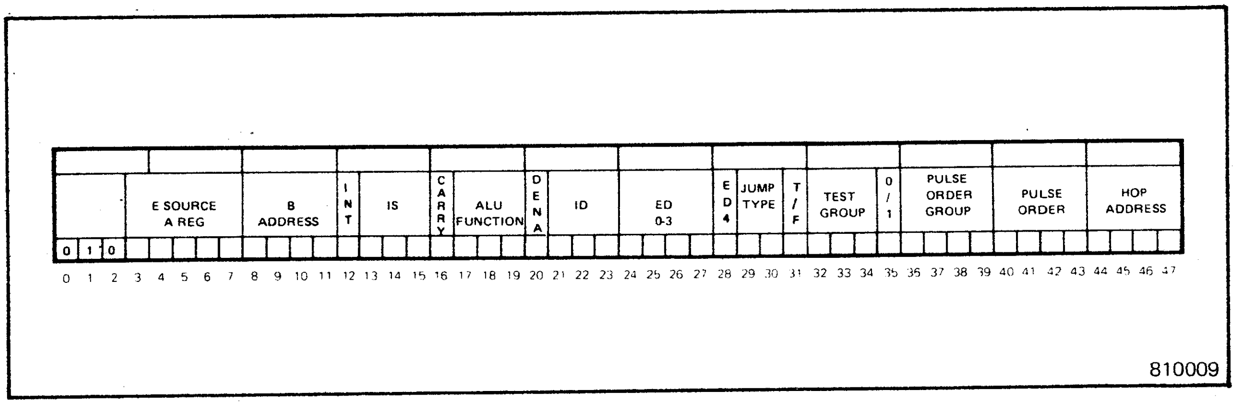


Figure 2-4. IOP Microword Format 2

Bit 35 specifies the test type, as shown below:

0 for test 0
1 for test 16

Bits 36 through 39 specify the pulse order group field (A, B, C, or D) that will be used to produce a pulse order.

Bits 40 through 43 specify the pulse order used.

Bits 44 through 47 contain the four-bit hop address. Note that all branches are absolute.

2.4.4 Format 3

Format 3 contains information for the four-bit test, eight-bit literal, and four-bit hop address, as shown in Figure 2-5.

Bits 00 through 31 contain the same information that was previously discussed.

Bits 32 through 34 specify the test group of the four-bit test.

Bit 35 specifies the test type, as shown below:

0 for test 0
1 for test 16

Bits 36 through 43 contain the eight-bit literal, which will be right-justified and zero-filled when sent to the ALU 16-bit data path. This literal cannot be used to address random access memory (RAM).

Bits 44 through 47 contain the four-bit hop address. Note that all branches are absolute.

2.4.5 Format 4

Format 4 contains information for the eight-bit branch, as shown in Figure 2-6.

Bits 00 through 31 contain the same information that was previously discussed.

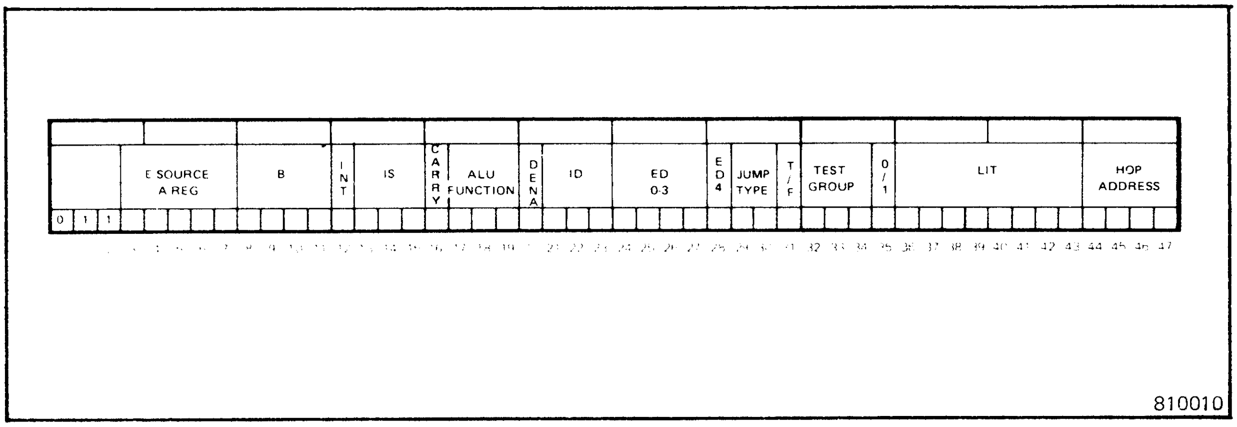


Figure 2-5. IOP Microword Format 3

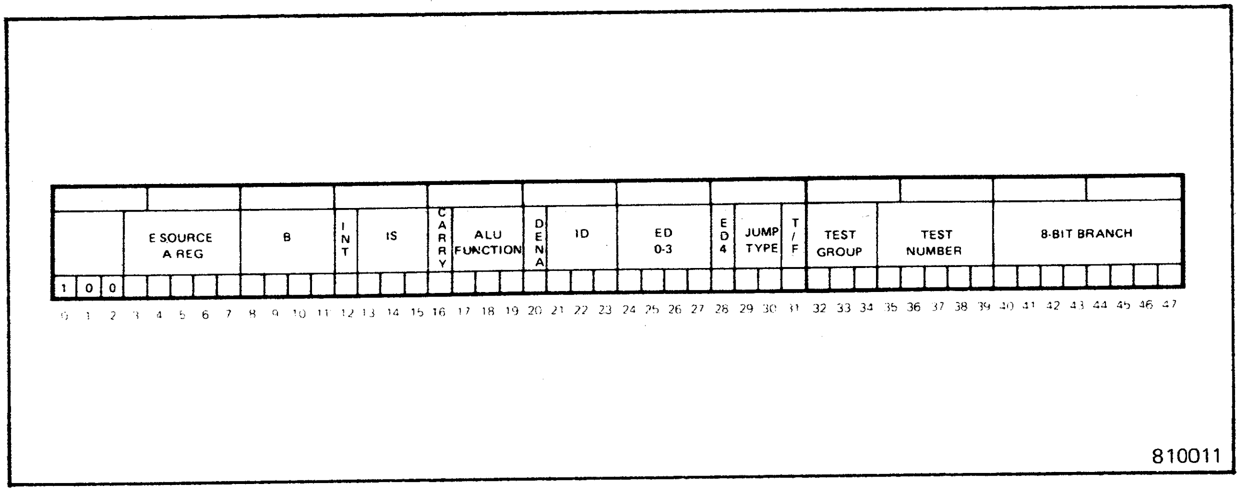


Figure 2-6. IOP Microword Format 4

Bits 32 through 34 specify the test group of the eight-bit test.

Bits 35 through 39 contain the test number.

Bits 40 through 47 contain the eight-bit branch address. Note that all branches are absolute.

2.4.6 Format 5

Format 5 contains information for the 16-bit literal, as shown in Figure 2-7.

Bits 00 through 31 contain the same information that was previously discussed.

Bits 32 through 47 contain the 16-bit literal. This field may also be used to address the scratch RAM.

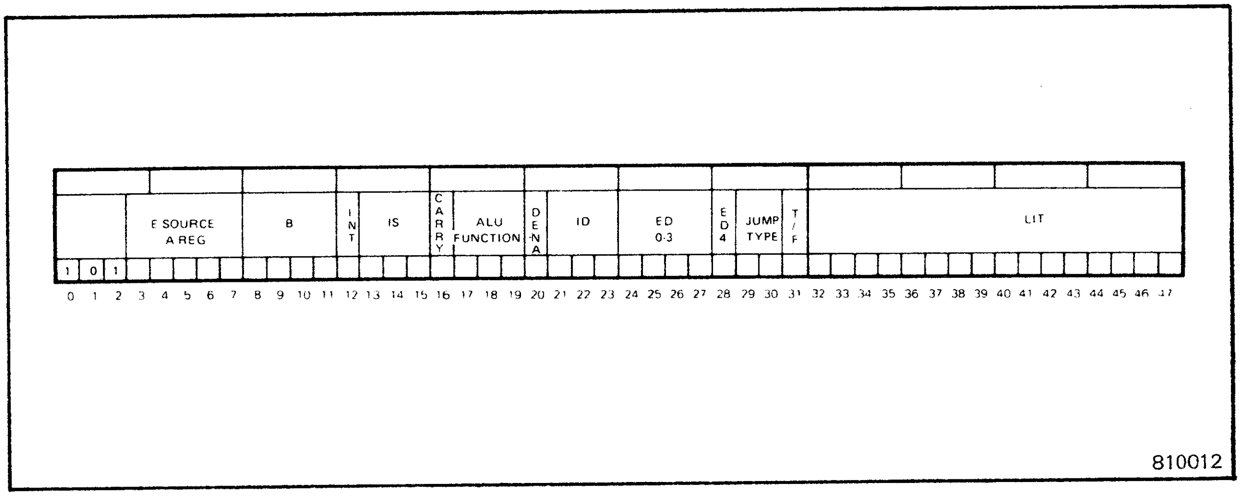


Figure 2-7. IOP Microword Format 5

2.4.7 Format 6

Format 6 is a 16-bit branch (always branch), which does not require a test. Format 6 is shown in Figure 2-8.

Bits 00 through 31 contain the same information that was previously discussed.

Bits 32 through 47 contain the 16-bit branch address. Note that all branches are absolute.

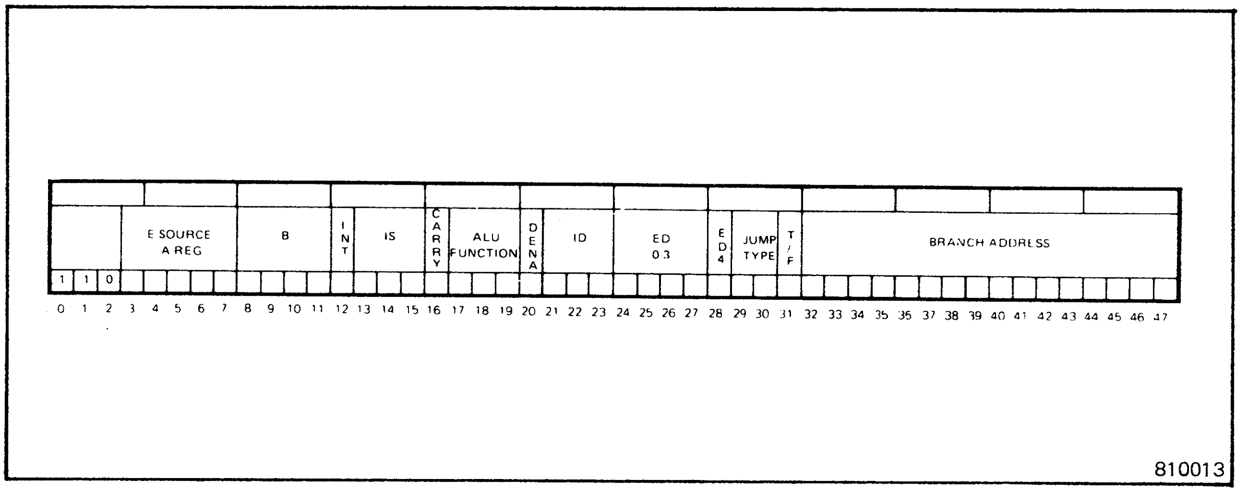


Figure 2-8. IOP Microword Format 6

CHAPTER 3

THEORY OF OPERATION

3.1 Introduction

This chapter discusses, in general and in detail, the major functional sections of and the operations performed by the input/output processor (IOP).

3.2 Functional Theory of Operation

This portion of the manual provides a description of the major functional sections of the input/output processor (IOP). The block diagram of the IOP, shown in Figure 3-1 (sheets 1-3), is provided as an aid in understanding the general description of the IOP. The IOP Drawings Manual should also be used to understand the functionality of the IOP.

The IOP proper is contained on a full-size board that interfaces with the SelBUS. This board is accompanied by an interface (IOP interface) board on the pin side of the SelBUS. The IOP board set combines real-time option module (RTOM), system control panel (SCP) functions, an operator console, with a 1.5M byte multipurpose bus (MPB) on a device interface (DI) board. The IOP interface board also contains two asynchronous interfaces and an interval timer. Interface to all multipurpose bus controllers is provided, using a 50-conductor flat cable.

The IOP hardware items are listed below:

1. A SelBUS interface
2. An IOP microengine
3. A multipurpose bus interface
4. An SCP interface
5. RTOM/interval timer

The IOP uses a 48-bit-wide microword. This size microword allows multiple functions to be performed in a single instruction. For example, only one instruction is required to load a literal into the arithmetic logic unit (ALU) and execute a data word test to set up a branch.

3.2.1 SelBUS Interface

The circuitry for the SelBUS interface circuit, shown in Figure 3-1, sheet 1, is listed below:

1. SelBUS receiver/data input file
2. SelBUS driver/data output file
3. SelBUS drivers, memory address register (MAR), and memory address counter (MAC)
4. SelBUS receiver and destination input register
5. Address compare/address stop register
6. SelBUS transfer control (including address recognition)
7. Interrupt poll and control functions

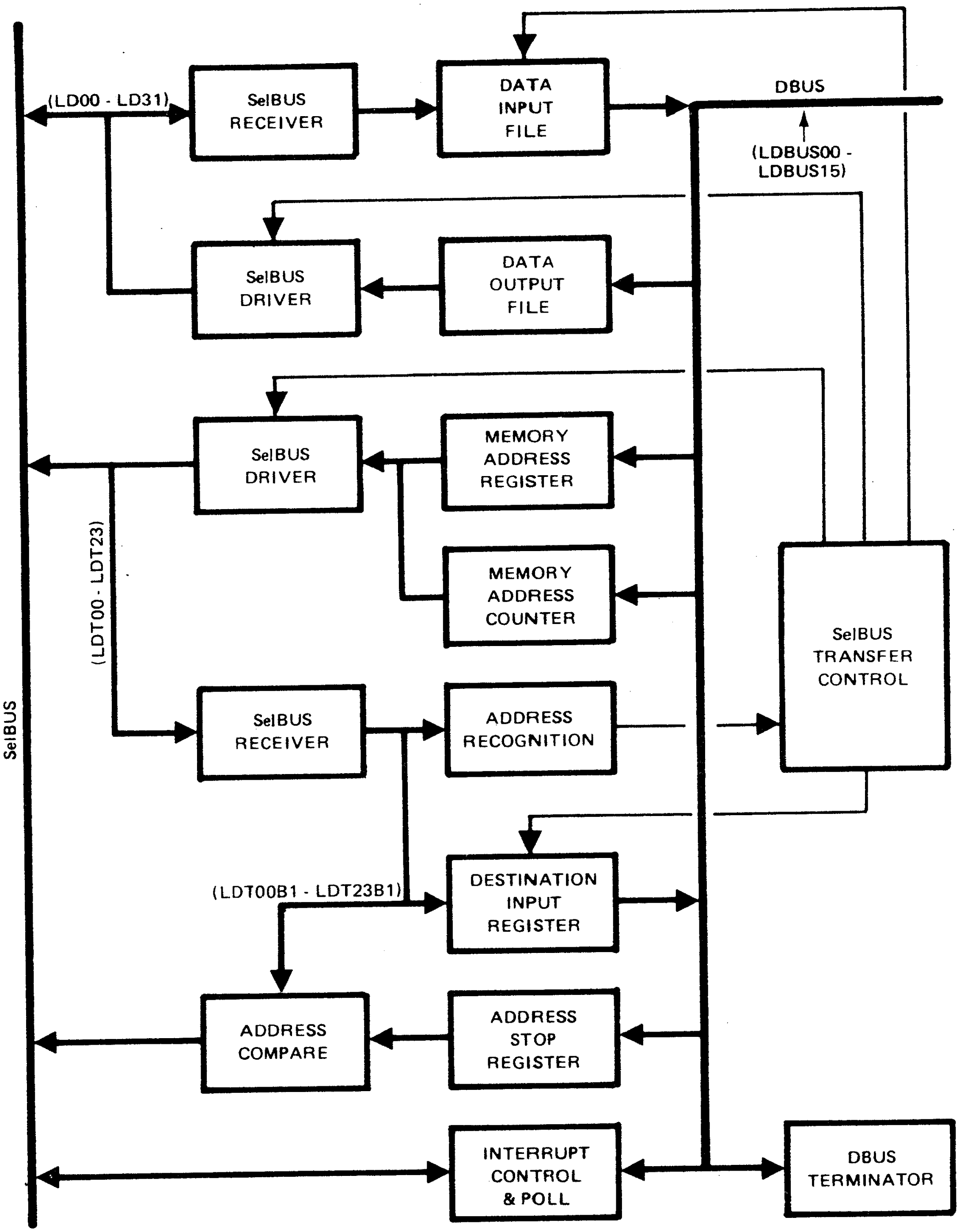


Figure 3-1. IOP Block Diagram (Sheet 1 of 3)

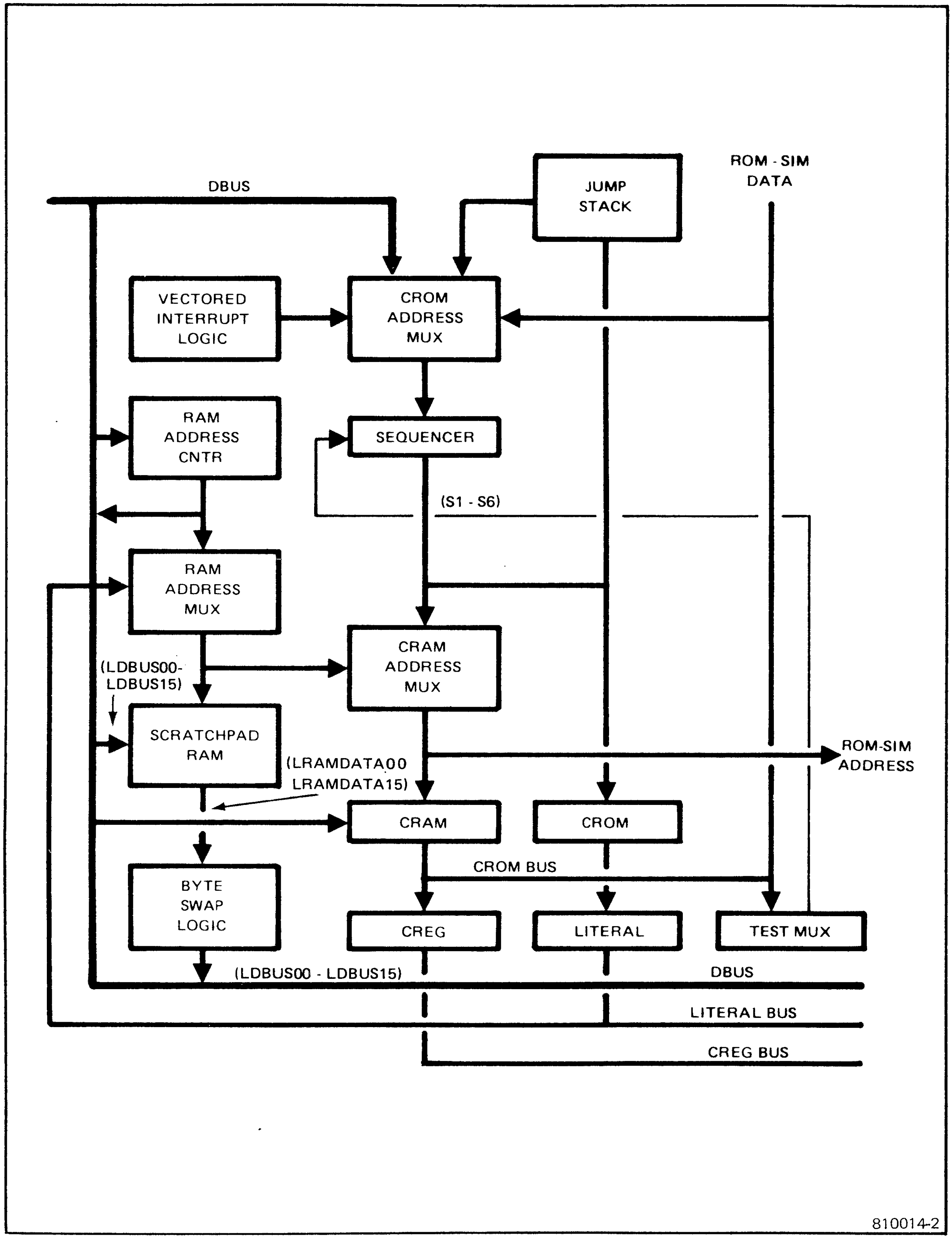


Figure 3-1. IOP Block Diagram (Sheet 2 of 3)

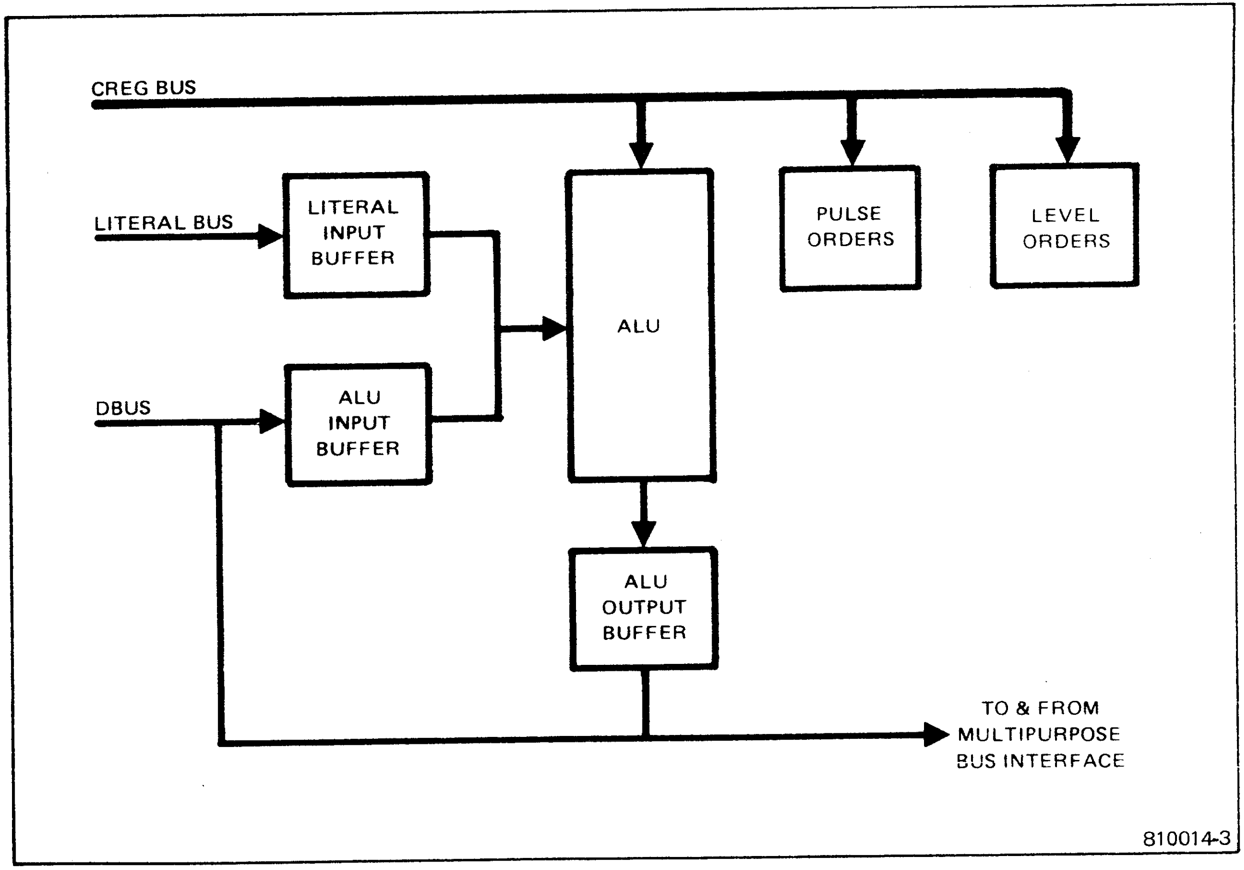


Figure 3-1. IOP Block Diagram (Sheet 3 of 3)

3.2.1.1 SelBUS Receiver/Data Input File

The input data receivers obtain data from the SelBUS. The data are then placed into the DBUS via the data input file register.

Data bits 00 through 31 (LD00 through LD31) are applied to octal latch circuits in the SelBUS receiver circuit. With the last 35 nsec of the LLATCH clock high, the data at the input of the octal latches are propagated through the data input file and ignored.

With the LLATCH clock low, the data on the data bus (DBUS) are latched for 140 nsec. After 140 nsec, the circuits are unlatched to allow new data to be sent to them. The latched data are also sent to 4-bit-wide by 4-words-deep registers.

3.2.1.2 SelBUS Driver/Data Output File

The SelBUS driver/data output file is similar to the SelBUS receiver/data input file except that the data is taken from the DBUS and placed onto the SelBUS via the data output file. Data bits LDBUS00 through LDBUS15 are applied to the registers and are either latched or ignored.

3.2.1.3 SelBUS Drivers, Memory Address Register, and Memory Address Counter

There are two sources of memory addressing on the IOP board: a register and a counter. The memory address counter (MAC) is preset and incremented by words to perform

3.2.1.3 SelBUS Drivers, Memory Address Register, and Memory Address Counter

There are two sources of memory addressing on the IOP board: a register and a counter. The memory address counter (MAC) is preset and incremented by words to perform memory fetches and memory address storage. The memory address register (MAR) and MAC are both 24 bits wide. With 24 destination bits on the SelBUS backplane, up to 16M bytes of memory can be addressed. The MAR and MAC require two full clock cycles for loading or unloading the 24 bits from/to the IOP data bus (DBUS). The MAC is used exclusively for burst mode memory transfers, and the MAR is used for nonburst mode and CPU transfers.

The MAR is utilized in single transfers. Single transfers occur when queuing the CPU for status transfers or single byte (word) transfers to memory when the IOP is in the nonburst data mode. The MAR also drives the destination bus.

After an address is loaded into the MAC it can be constantly incremented or decremented (by words), as required, to change the address during high-speed burst mode memory transfers.

The information contained in the MAR and MAC is placed onto the SelBUS via the SelBUS driver circuits.

3.2.1.4 SelBUS Receiver and Destination Input Register

Since only a single load is allowed on the SelBUS, the destination input data (LDT00 through LDT23) are applied to a noninverting buffer register to obtain the proper fan-out. Since this action is always necessary, the buffer register is constantly enabled.

A latch circuit saves the destination information during CPU transfers (commands). If the destination information is to be tested by the firmware, it must be latched in the destination input register.

The buffered (same polarity) destination information (LDT00B1 to LDT23B1) is sent to the address compare register and address recognition logic (part of the SelBUS control logic).

3.2.1.5 Address Compare/Address Stop Register

The address compare logic is used by the system control panel (SCP) functions to halt the CPU when a read address compare, write address compare, or instruction fetch address compare instruction is executed.

The destination bus (LDT00B1-LDT23B1) is constantly being monitored and compared with data stored in the address compare register. The address compare register is loaded from the internal DBUS via the address stop register. The comparator, used in the address compare halt logic circuit, has only one output (A=B).

To halt the CPU during a read or write of a certain address, that specific address and certain qualifiers (to indicate an operand read stop, operand write stop, instruction stop, or any combinaton of the three) are loaded into the address compare register. The three outputs are then ORed together since they are mutually exclusive. A line is then driven back to the CPU telling it to halt. This function is part of the system control panel interface (SCPI) board. A separate line is driven by the IPU Console IOP to halt the IPU.

The address comparator logic of the SelBUS interface is used during input transfers to the interface. The address comparator determines when a transfer, on the SelBUS, is destined for the IOP. The determination is based on a comparison of destination bits 9 through 14 and the physical address of the IOP. If the two addresses compare, and the memory tag signal and CPU address bit are false, the SelBUS transfer is considered for this particular IOP and the SelBUS interface stores the contents of the transfer at the end of the transfer cycle.

The IOP uses the middle 8 bits (8 through 15) on the 24 bit destination address bus as the physical (board) hexadecimal encoded address. Note that bit eight of the destination address bus is only used by the CPU. If bit eight is equal to one, the IOP is to communicate with the CPU. If bit 8 is equal to 0, the next 7 bits (9 through 15) of the destination address bus are used as the physical address of the IOP. Also note that since the IOP always responds to an even/odd address pair (i.e. 70/71) there is no jumper for destination address bus bit 15 - it is said to be implied by the nature of the IOP addressing scheme.

On the IOP board there is an eight position jumper block associated with destination address bits 8 through 15. Jumpers 1 through 6 (destination address bits 9 through 14) are used to enable or not enable the six destination address bits shown in Table 3-1. Jumper seven is used to enable or disable the system panel functions of the IOP. Jumper eight is not used.

The six address jumpers allow the input to the address comparitor to "look" for a high true signal. From the destination address bus the signal passes through a set of inverters so that a high true signal can be applied to the A input of the comparator. Pull-up resistors are used to place a low true (false) signal at the B input of the comparator. This arrangement causes the comparator to "look" for a false signal on the destination address bus.

When a jumper is placed in the jumper block it grounds the pull-up resistor. This action pulls the associated inverter input down and causes the output of the inverter to go high. Now the address comparator is "looking" for a high bit value on the destination bus.

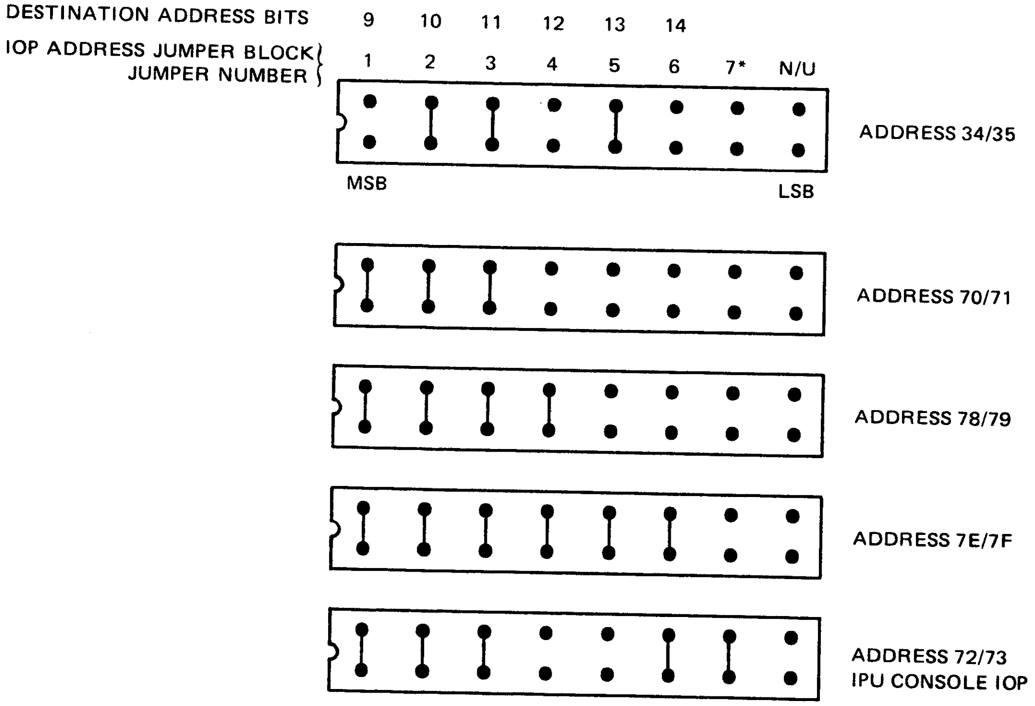
The address comparator logic is composed of one comparator integrated circuit (IC), seven jumpers, and several gates. The seven jumpers must be set up at the time of installation to reflect the IOP's physical address. This address is compared, by the comparator circuit, with destination bus bits 9 through 14 (LDT09 through LDT14) from the SelBUS. If a comparison is obtained, the comparator IC generates an A = B signal.

The gates test for the following conditions: a logic low level on the transfer (LTX) tag signal line; logic high levels on the memory (LMEM) tag signal line and CPU address bit (LDT08); and the address bits all equal to zero. A comparison for a logic high-condition is performed by comparing the signal to TV, the comparison for a logic low condition is performed by comparing the signal to ground. The true (low) condition of the transfer tag signal line indicates that a valid transfer is on the SelBUS and the false (high) condition of the memory tag signal and CPU address bit (destination bus bit 8) indicates that the transfer is not intended for CPU or memory.

If all the comparisons are true, the address comparator assumes that the SelBUS transfer is for this particular IOP and it generates the transfer input signal in the true or high logic condition. The transfer input signal enables the SelBUS interface control logic. The address comparison occurs during the middle of a SelBUS transfer cycle and the comparison must be true by the end of the transfer cycle.

The least significant bit of the address, destination bus bit 15 (LDT15), selects between one of two data registers (burst mode or non-burst mode) and the LCPU bit selects the CPU (command) register.

Table 3-1
Typical IOP Address Jumper block - Jumper Configuration



IOP TYPE	LOCATION	
MODEL 8000	K15	X8
MODEL 8001	K16	X8
IPU CONSOLE IOP	K16	X8

830587

* Note insert jumper 7 to enable control panel functions and remove jumper 7 for 32/7X systems.

3.2.1.6 SelBUS Transfer Control

The SelBUS transfer control logic is a high speed hardware sequencer that controls all transfers through the SelBUS interface and maintains the proper tag bits. The address recognition portion of the SelBUS transfer control logic informs the IOP that the CPU or memory is passing data or commands to it.

The SelBUS interface bus transfer priority polling logic determines and obtains transfer priority for output transfers to the SelBUS. For discussion purposes the priority polling logic can be divided into two functional logic sections as shown below:

1. The priority determination logic that determines the interfaces relative priority. This logic consists of a group of jumpers.
2. The priority polling control logic that executes a polling sequence to obtain bus transfer priority at the command of the microprocessor.

The priority determination logic receives or generates 23 priority signals (LPR00 and HPR01 through HPR22) from/to the SelBUS. The receive logic is composed of three groups of inverters with seven inverters in each group. Each individual inverter receives a priority signal from the SelBUS. The outputs of the inverters are sent to three groups of jumpers with seven jumpers in each group. These three groups of jumpers assign the priority of all higher priority devices that interface to the SelBUS. High priorities are assigned by simply installing their respective jumpers. If, for example, a particular IOP is assigned priority 03 (the HPR03 signal line), then jumpers for priority lines 01 and 02 must be installed to give them priority over line 03 as shown in Table 3-2. The HPR03 jumper must be left out along with the jumpers for all lower priority devices. In this example, if priority 10 is to have priority over line 03, then priority 10 line jumper must be installed.

The outputs of each of the groups of priority input jumpers are wired-ORed together to generate the lost poll bus 1 and 2 (LLOSTPOLLBUS1 and LLOSTPOLLBUS2) signals. These two signals, along with the LPR00 signal, are ORed to generate the HLOSTPOLLBUS signal to the priority polling control logic. The lost poll bus signals indicate that a higher priority transfer is demanding the SelBUS and this condition inhibits an attempt to obtain the SelBUS by lower priority devices until all higher priority transfers are complete.

The priority generation logic consists of two output transmitters and three groups of jumpers. One of the two output transmitters is used to directly generate the priority 00 (LPR00) signal for the SelBUS during the transfer cycle immediately preceding the cycle in which the output transfer is executed, if the output transfer is a data return transfer (DRT) addressed to the CPU. In this particular case where the IOP is generating a DRT to the CPU, normal bus transfer priorities are by-passed and the transfer is made at priority 00 (highest priority level). The second of the two output transmitters is only used during output transfers to the memory. This output generates the poll memory (HPOLLMEM) signal. The HPOLLMEM signal is sent to the three groups of jumpers to generate one of 22 priority signals. These signals are on the SelBUS during the bus transfer cycle immediately preceding the cycle in which the memory transfer is executed, if no higher priority transfer is in progress.

The IPU Console IOP does not generate LPR00; however, it does poll the SelBUS for DRTs to the IPU using the priority assigned to the IPU Console IOP.

The actual priority signal generated depends upon the jumper configuration in the three groups of jumpers. The jumpers must be configured such that the jumper for the priority level to which the IOP is assigned, is installed and all the other jumpers are left out. For example, if the IOP is assigned to priority 03, the jumper for the HPR03 output signal must be installed and the remaining jumpers left out as shown in Table 3-2. These sets of conditions cause the HPOLLMEM signal to generate the HPR03 signal on the SelBUS.

Table 3-2
SelBUS Transfer Polling Logic - Jumper Assignment

Priority Detection	Model 8000	Model 8001& IPU Console IOP	Priority Generation	Model 8000	Model 8001& IPU Console IOP
HPR01 o———o HPR02 o———o HPR03 o———o HPR04 o———o HPR05 o———o HPR06 o———o HPR07 o———o	E-21 X4	E-23 X4	HPR01 o o HPR02 o o HPR03 o o HPR04 o o HPR05 o o HPR06 o o HPR07 o o HPR08 o o	A-25 X1	A-23 X1
HPR08 o———o HPR09 o o HPR10 o o HPR11 o o HPR12 o o HPR13 o o HPR14 o o	E-22 X5	E-24 X5	HPR09 o———o HPR10 o o HPR11 o o HPR12 o o HPR13 o o HPR14 o o HPR15 o o	A-26 X2	A-24 X2
HPR15 o o HPR16 o o HPR17 o o HPR18 o o HPR19 o o HPR20 o o HPR21 o o	E-23 X6	E-25 X6	HPR16 o o HPR17 o o HPR18 o o HPR19 o o HPR20 o o HPR21 o o HPR22 o o	A-27 X3	A-25 X3
NOTE: IOP SelBUS Priority nine(9) Shown.					

The HPR03 signal indicates that the IOP is going to transfer data to the SelBUS during the next transfer cycle. The HPR03 signal then inhibits any lower priority devices from trying to transfer data to the SelBUS during the next transfer cycle. If a higher priority transfer was scheduled for the next transfer cycle, the HPOLLMEM signal is inhibited and the IOP would not be able to transfer data until the higher priority transfer was completed.

3.2.1.7 Interrupt Poll and Control Functions

The interrupt poll and control logic presents the IOP priority interrupt to the SelBUS when requested to do so by the IOP microengine.

3.2.2 IOP Microengine

The IOP microengine, shown in Figure 3-1, sheets 2 and 3, consists of the control random access memory (CRAM), control read only memory (CROM), sequencer, arithmetic logic unit (ALU), random access memory (RAM), literal buffer, and associated control logic.

The microengine hardware circuitry is listed below:

1. CROM/CRAM address multiplexer (mux)
2. Sequencer
3. Vectored interrupt logic
4. CRAM
5. Control PROMs
6. Control register (CREG)
7. Literal buffer
8. RAM address counter
9. Scratch pad RAM
10. RAM data byte swapping logic
11. ALU
12. Jump control stack (J-Stack)
13. DBUS terminator

3.2.2.1 CROM/CRAM Address Mux

There are five different types of sequencer operations: four branch and one increment, with the increment operation the most common.

The four sources of branch addresses that can be input to the CROM/CRAM address mux are:

1. CROM bits
2. DBUS data
3. External stack
4. Vectored interrupt

The CROM bits contain the branch address stored in the microword. When a branch operation is performed, the branch address is normally obtained from the CROM word.

The DBUS source of branch addresses is used for a computed branch. In a computed branch operation, the program branches to some location based on the result of some calculation from the ALU.

After the branch operation is completed, the external stack allows a program to return to the address it left. The external stack can store up to 16 return addresses in a last-in-first-out (LIFO) arrangement.

The vectored interrupt branch address results when a vectored interrupt is executed.

In addition to branching, the sequencer can perform a straight increment operation.

The CRAM address mux consists of two eight-to-four multiplexers (quad two-to-one) providing two sources of addresses to the CRAM. One source, the CROM address, is used during an execution cycle. When executing from the CRAM, the address is sourced from the CROM address output.

The other source, the RAM address counter (RAC), is used when the CRAM is to be loaded. Using these two sources, it is possible to simultaneously load the CRAM while executing a command from CROM. To execute a command from CRAM, the addressing scheme is switched so that the data are now obtained from the address established by the CROM address mux.

3.2.2.2 Sequencer

Five of the output bits of the sequencer are combined to provide a PROM bank select signal. There are two banks of 2K-by-48-bit PROMS. The CRAM is at the high end of the addressing capability.

The CRAM uses some locations of the CROM and is loaded from the serial ports through main CROM. A 16-bit branch then allows the IOP to go to that area of memory and execute the command at that address.

The CROM address stack portion of the sequencer is a 16-deep-by-4-bit-wide memory element with an operating speed of 20 nsec. Since only one group of CROM address stacks is being used, the chip select (CS) input is always enabled (tied to ground).

The enable stack write (LENSTKWRT) signal is used when a push stack operation is performed. The push stack operation causes the input of the stack to be written into the stack at particular locations. The stack address counter addresses those particular locations. The stack address counter is an up-down counter, which is initially set to zero at power-up. Each time data are pushed onto the stack, the stack address counter is incremented. This action changes the address of the stack. When an address is brought out of the stack (popped), the stack address counter is decremented. The output of the CROM address stack is permanently enabled, since it is connected to the CROM address multiplexer. With this type of arrangement, a 20 nsec response time (for 16 bits) is maintained.

IOP sequencer control logic is used to control the branch or increment functions of the sequencer. During an increment operation, the sequencer takes its output value and, during the next clock cycle, adds one to it. This incremented value is then output and used for the next fetch cycle.

During branch operation, the sequencer replaces the least significant four or eight bits, or all 16 bits, with a branch address. The branch address might come from the microword, DBUS, vectored interrupt logic, or jump control stack (J-stack).

When a four-bit branch operation is performed, the most significant 12 bits do not change. The output of the test logic determines if a branch or increment operation is to be performed. If a true output results, a branch operation is performed. A false output results in an increment operation.

To optimize speed (shortest path out of the test logic), only one level of time delay occurs between the test output and the sequencer control lines.

There are six control lines (S1 through S6) to the sequencer. Since the sequencer is separated into four 4-bit fields, the S5 and S6 control lines are split into four parts: S5A, B, C, and D, and S6A, B, C, and D. S5A contains the most significant four bits and S5D the least significant four bits. S1 and S2 are used to distinguish between DBUS branches and all other branches. S3 and S4 are hardwired to ground. With this arrangement, only a minimum of the sequencer circuit is utilized. This scheme is also used because the sequencer only contains a four-bit-deep internal stack. S5C and S6C enable the data for an eight-bit branch. S5D and S6D (with S5D always tied to ground) are used with any branch.

3.2.2.3 Vectored Interrupt Logic

The vectored interrupt logic allows up to 15 different levels of interrupts to be sent to twin priority encoders. If vectored interrupt level three is set, the output of that chip is a binary encoded three on the A0, A1, and A2 lines. While vectored interrupt level three is activated, no other lower priority level can be encoded. For example, if vectored interrupt level seven polls for encoding, it is ignored. However, if a vectored interrupt level of zero, one, or two polls for encoding, it overrides vector interrupt level three. A 16-bit priority encoding scheme is obtained by gating the output of the twin eight-bit priority encoders.

The last input to the twin priority encoders is tied to +V to differentiate between zero interrupt level and zero output (no interrupt level).

The gated output of the priority encoders is sent to a less-than, equal-to, or greater-than magnitude comparator and a hexadecimal latch. The magnitude comparator compares the output of the priority encoder to the output of the hexadecimal latch. If the outputs are the same, an A=B output is obtained. This output is a "do nothing" state, which the magnitude comparator is in most of the time. When a higher level interrupt is applied to the A inputs, as opposed to the B inputs of the comparator, an A is greater than B output signal results.

The output signal from the comparator enables a gate that generates a vectored interrupt signal and allows the hexadecimal latch to be loaded with the higher priority interrupt level. This action, in turn, steers the adress inputs to the CROM address mux and allows entry into a certain section of the CROM (vectored entry into CROM).

Once the particular vectored interrupt subroutine is completed, the clear interrupt signal (LSTKINTCLR) must be enabled to clear the hexadecimal latch. The source of this particular interrupt is assumed to have disappeared by this time, due to some firmware interaction. When the program is ready to branch back to its main subroutine, the LSTKINTCLR signal is enabled to reset the hexadecimal latch to zero. This action places zeros into the B input of the comparator. If any value besides zero is at the A input of the comparator, another vectored interrupt subroutine is immediately entered. If a zero is applied to the A input, another vectored interrupt is not generated and the program continues to be executed as it was before the first interrupt occurred.

The disable interrupt signal (LDISINT) is a dedicated bit (bit 12) of the microword, which can be set or reset on an instruction-by-instruction basis. The LDISINT signal is used when the operation being performed is so important that an interrupt cannot be allowed.

The vectored interrupt logic provides a method of quickly going from an idleloop, or lesser part of a particular subroutine, into a more critical subroutine. For example, instead of being in an idleloop performing a sequence of tests (Are CPU data here?, Is interrupt here?, Is real-time clock here?, etc.), the interrupt level for that particular test is enabled.

There are two special interrupt levels associated with the priority encoders: CPU data here (LCPUDATAHERE) and program interrupt (LPROGINT). The LCPUDATAHERE interrupt level is processed immedately, except when one of three higher interrupt priority levels are received. One of these higher interrupt priority levels is the LPROGINT level. The LPROGINT interrupt is a level order that can be set to lock out any lower level priority interrupt. The two priority levels above the LPROGINT level are zero (IOP time-out) and one (FIFO service request). Both of these interrupt levels are concerned with the multipurpose bus (MPB).

The normal way of getting out of a particular interrupt level encompasses a twofold operation: first, the interrupt stimulus must be turned off (removed), and second, a clear stack interrupt pulse order must be issued. The clear stack interrupt pulse order clears (forces to zero) the stack register. With the stimulus removed and the stack register cleared, there is a zero out of the stack register and a zero applied to the magnitude comparator; therefore, there is no comparison (A=B) and no interrupt generated.

3.2.2.4 Control RAM

The CRAM is a future option and is not provided for in this design. The basic CRAM chip is a 4-bit-wide-by-256-word-deep RAM element with separate input and output ports. The separate ports are necessary since the CRAM is being loaded from the DBUS and output onto the CROM bus.

Manipulating the read/write and address lines allows the CRAM to be written into, using the RAC as a source of address, and then switched to the CROM address mux to execute the addressed command.

3.2.2.5 Control PROM

The 2K-by-4-bit control PROM requires 11 bits of addressing (to obtain the 2K capacity) and a bank select input. The control PROM is a standard PROM with a response time of 60 nanoseconds.

3.2.2.6 Control Register

Each time that the sequencer is incremented or addressed, 25 nsec are required for its output to stabilize. The output of the sequencer is then sent to the PROMs. The PROMs take 60 nsec to provide an output. When an output is obtained from the PROMs, 85 nsec has elapsed. On the next H3 clock edge (the 150 nsec point), the output of the CRAM is loaded into the control register (CREG). This action causes a change from the CROM cycle bit to a CREG cycle bit. Most of the command execution, except test and branch operations, is performed during the CREG cycle.

The test and branch operations are performed during the CROM cycle. Changing from a CROM cycle bit to a CREG cycle bit creates a pipeline machine where operations are performed in stages, thus shortening the actual execution time. This speed-up of execution time is accomplished by branching on the CROM cycle rather than waiting one more clock period and branching on the CREG cycle.

3.2.2.7 Literal Buffer

The literal buffer circuit removes bits from the microword and forces them on the ALU input bus to form a constant that is easily programmed into the machine. CROM bits 32 to 47 are programmed by the microprogrammer. These bits may be combined with data in the ALUs for a masking operation, or loaded into RAM, or set up as bits for a bus transfer or other miscellaneous operations. Data from CROM bits 32 to 47 are latched into flip-flops every clock cycle. When these data are to be used as a literal, the enable literal (LENLITERAL16) line is pulled. The LENLITERAL16 line is used to produce a six-bit literal from the literal buffer.

There is also a format that produces an eight-bit literal. This eight-bit literal takes half of its bits from CROM bits 36 to 39 and the other half from CROM bits 40 to 43.

The literal may also be used as a direct address for the scratch pad RAM.

3.2.2.8 RAM Address Counter

Depending on which pulse order is used - either increment RAM address counter (LINCRAC) or decrement RAM address counter (LDECRAC) - the RAM address counter (RAC) either counts up or down. The RAC can be preloaded with any number using the load RAM address counter (LLDRAC) signal. The RAC is also clocked every cycle by the H3 clock H3CLKXX); however, since the enable P (ENP) line of the RAC is controlled by a gate, there is only one true up or down count. This up or down count only occurs when either the increment or decrement pulse order is received.

The enable T (ENT) line comes from the lower order stages of the RAC, the carry-out of the lowest stage is connected to the ENT input of the next stage, and the carry-out of this stage is connected to the ENT input of the highest stage.

The RAC contains 12 bits, the lower eight bits used for the CRAM. The scratch pad RAM, which is a separate block of logic, is a full 4K deep and, therefore, requires the full 12 address bits.

3.2.2.9 Scratch Pad RAM

The scratch pad RAM consists of 16 4K-deep-by-one-bit-wide RAM chips requiring a full 12 address bits. The scratch pad RAM is controlled (turned on/off) by the source and destination decode logic. The input to the scratch pad RAM is labeled LDBUS00 through LDBUS15. The output of the scratch pad RAM is labeled LRAMDATA00 through LRAMDATA15. The scratch pad RAM is addressed through the RAM address mux by the RAM address counter and CREG bits 36 through 47.

3.2.2.10 RAM Data Byte Swapping Logic

The RAM data byte swapping logic allows for two options: bringing data through all the way to the DBUS in order (LDBUS00 through LDBUS15) or taking the most significant byte and shifting it to the least significant position while simultaneously shifting the least significant byte to the most significant position. This action allows a swap of bytes, which is desirable when talking to byte-oriented devices connected to the MPB. Two bytes of information can be stored in RAM and one byte loaded; then, a swap of bytes can occur and the other byte of data can be loaded. This action allows twice as much data to be stored in a given location without masking by the ALU.

The byte swapping logic is set up on a halfword basis. This type of set-up allows the recouping of information from byte-oriented devices. The output of the RAM can also be left on when using byte swapping logic, thus providing a speed-up of the RAM.

3.2.2.11 ALU

The ALU consists of four IM2901A four-bit microprocessor chips. The IM2901A, shown in Figure 3-2, consists of a 16-word-by-4-bit two-port RAM, an internal high-speed ALU, and the required shifting, decoding, and multiplexing circuits.

Data enters the IM2901A via the ALU input bus at the direct data inputs, D0 through D3. CREG bits 13 through 15 are applied to the ALU source operand decode circuit at I2 through I0; GREG bits 17 through 19 are sent to the ALU function decode circuit at I5 through I3. CREG bits 21 though 23 are applied to the ALU destination decode circuit at I8 through I6.

The first four address bits (CREG4 through CREG7) are applied to the 16-bit-by-2-port RAM via the A word address lines (A0 through A3). The second four address bits (CREG8 through CREG11) are applied to the 16-bit by 2-port RAM via the B word address lines (B0 through B3).

The modified data leave the IM2901A at Y3 through Y0 and are then placed onto the DBUS. The IM2901A is clocked via the CP line by the H3 clock.

The ALU look-ahead carry logic portion of the ALU provides a means of fast carry by allowing four ALU chips to be tied together. Normally, the carry output of one ALU is connected to the carry input of the next stage, etc.. This type of connecting scheme causes the ripple time (the time to get from the data input of the first stage to the carry output of the last stage) to be approximately 20 nsec per stage. This time is entirely too long for this particular application; therefore, the normal ripple carry logic is replaced with the ALU look-ahead carry logic.

The fast carry chip cuts the ripple time from 60 to 10 nsec. The 10 nsec ripple time factor now allows an ALU plus carry operation and a test operation to be performed during the same clock period, instead of waiting another clock period to latch the results.

3.2.2.12 Jump Control Stack

Since the jump control stack (J-stack) consists of four 16-deep by 4-bit-wide register stacks, a register addressing scheme is required. This register addressing scheme is accomplished using a counter. The control circuitry for the counter comes from a decoder and a pulse order to reset the stack. During the power-up sequence, zeros are loaded into the decoder, resetting the counter.

3.2.2.13 DBUS Terminator

A 330/220 ohm pull up/down resistor is provided to terminate each line of the 16-bit DBUS.

3.2.3 Pulse and Level Orders, Test Field, and Source and Destination Decode

The bus source decode and bus destination decode circuits are used in conjunction with the data bus (DBUS). There is only one DBUS for the entire IOP. Each function

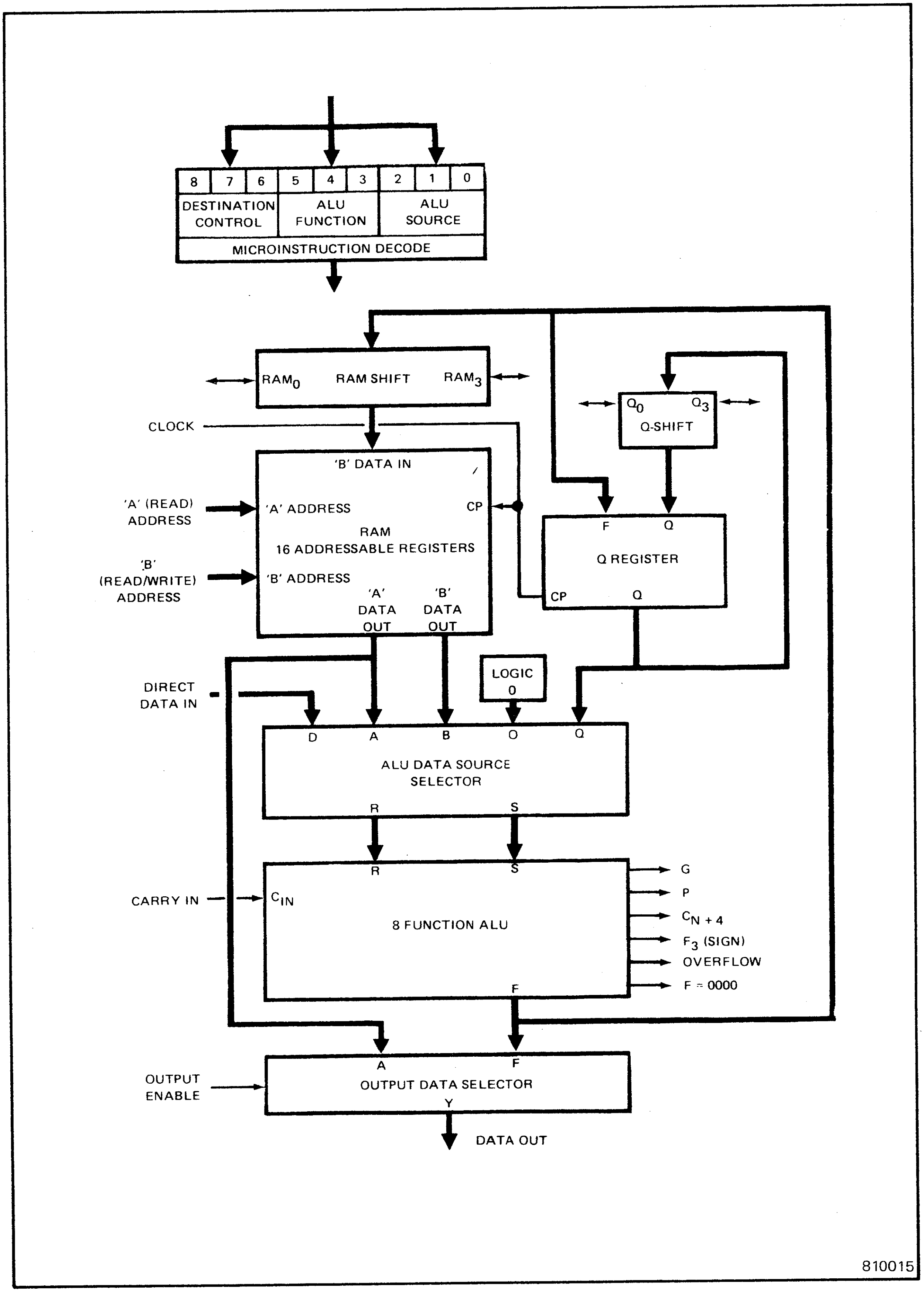


Figure 3-2. The 2901 Microprocessor

connected to the DBUS either sends or receives data. The IOP microengine determines which block is to transmit data and which block is to receive the data via the source and destination decode logic. This decision is based upon tests performed and/or microinterrupts received. Data can be sent to, or received from, the DBUS without going through the ALU.

The pulse and level orders, test field, and source and destination decode circuitry are listed below:

1. IOP test structure
2. IOP mini-mux test structure
3. Test control logic
4. Test summation circuitry
5. Source decode circuitry
6. Destination decode circuitry
7. Pulse order decoders
8. Level order select logic

3.2.3.1 IOP Test Structure

The IOP test structure is enabled during the CROM cycle, which is synchronized to the clock-3 edge of the H3 clock. There are three separate 32-bit-wide banks of tests. These banks are individually enabled so that the result of test one, two, and/or three can be ORed together. After the tests are performed, a branch or increment command is initiated.

3.2.3.2 IOP Mini-Mux Test Structure

A complete test structure appears in microword format 0 and format 4. Formats 1, 2, and 3 contain a mini-test field (three enables and a single bit indicating 0 or 1). In essence, the single bit actually indicates test 0 or test 16. Because of timing constraints imposed on the IOP, the mini-mux test structure was developed. The mini-mux test structure takes the six low true inputs from the mini-test field (A0 and A16, B0 and B16, C0 and C16) and applies them to the eight-to-four multiplexer. Note that only six of the eight inputs to the multiplexer 1A, 1B, 2A, 2B, 3A, and 3B are being used.

To obtain a true/complement output from the multiplexer, the 1Y, 2Y and 3Y outputs are applied to three exclusive OR circuits. CROM bit 31 tests for a true (high) or false (low) condition; therefore, it is considered a true/false bit in the microword.

3.2.3.3 Test Control Logic

The test control logic is used to select the correct test mux for the particular test. The CROM01, CROM02, and format 5, 6, and 7 signals are used to turn the corresponding fields on when required and off when not required.

Format 6 is a 16-bit branch (always branch) that does not require a test; therefore, the test fields must once again be turned off. At this time, there is no format 7. If, for some reason, a format 7 is programmed, the sequencer just increments and continues as if format 7 did not exist.

3.2.3.4 Test Summation Circuitry

The test summary circuit is used to tie together all of the high and low test lines that are applied to the sequencer control circuit.

3.2.3.5 Source Decode Circuitry

The source decode circuit allows certain groups of logic to place (source) data on the internal DBUS.

3.2.3.6 Destination Decode Circuitry

The destination decode circuit determines the destination of data on the DBUS.

3.2.3.7 Pulse Order Decoders

The pulse order decode circuit utilizes two types of fields: the F0 format (minipulse order field) and the F2 format (large pulse order field).

The F2 format can be split into two different four-bit fields; bits 40 through 43 are actually a decode of 1-of-16. Bits 36 through 39 determine which of the four fields will produce a pulse order. Up to four pulse orders can be produced simultaneously, using a single bit enable as was used in the test field. Note that if all four pulse orders (groups A, B, C, and D) are simultaneously required, they all need the same decode address out of the least significant bits.

A pulse order is a low true signal, 75 nsec long, starting halfway through the cycle and going through to the end of the cycle.

The F0 format does not contain the individual enable signals; therefore, as soon as F0 is decoded, only pulse order group A is produced.

3.2.3.8 Level Order Select Logic

There is only one microword field (format 1) associated with level orders. This field contains eight bits (CREG36 through CREG43). CREG36 is a set/reset bit. Level orders can either be set high or low. Bits 37 through 43 are used as a 1-of-128 decode. This arrangement allows a total of 128 level orders.

The level order select logic circuit uses eight 8-bit addressable latches. All level orders are reset low during system reset and are set/reset halfway through the cycle.

3.2.4 IOP Interface

Since the IOP combines the functions of an RTOM, SCP, and system operator console, the IOP interface circuitry must be able to properly interconnect several devices to the IOP. A simplified block diagram of the IOP interface circuit is shown in Figure 3-3. The IOP interface circuit contains control panel logic, interval time logic, system control panel control logic, MPB logic (with associated circuitry), and a clock distribution circuit.

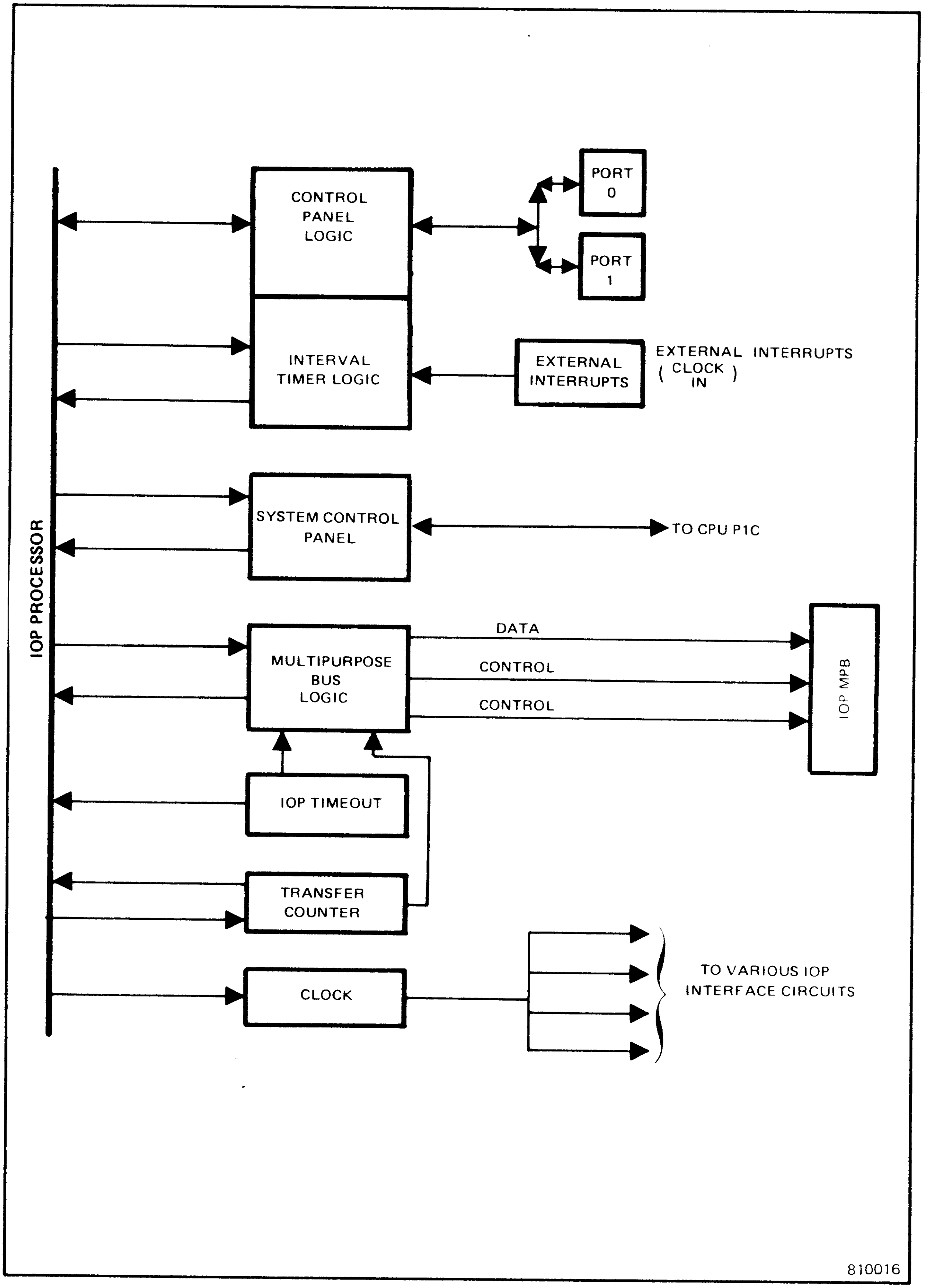


Figure 3-3. Simplified Block Diagram of IOP Interface Circuits

The circuitry for the IOP interface is listed below:

1. Multipurpose Bus Logic
2. IOP Time-Out circuitry
3. IOP MPB
4. Clock
5. Transfer Counter
6. Control Panel/Interval Timer Logic
7. Ports 0 and 1
8. External Interrupts
9. System Control Panel Interface Circuit

3.2.4.1 Multipurpose Bus Logic

The circuitry for the multipurpose bus logic is listed below:

1. Input buffers
2. Input/output ready circuit
3. Input first-in-first-out (FIFO) clock
4. Data swapping circuitry for the input FIFO
5. Output drivers
6. Output FIFO
7. Data swapping circuitry for the output FIFO
8. Parity generator and checker circuit
9. Input/output transfer difference counter
10. Transceiver transmit/receive mode controlling circuit
11. Output FIFO data/command interfacing circuit
12. Handshake circuitry
13. Stop communications signal generator

3.2.4.1.1 Input Buffers

The input buffers place low true data from the IOP DBUS (LDBOO through LDB15) onto the buffered IOP DBUS (LBDB00B through LDB15B). Note that the "B" suffix indicates the DBUS has been buffered. This buffered data is then distributed on the IOP DI board. Since pins 1 and 19 of the input buffers are hardwired to +V and ground respectively, the input buffers are permanently enabled.

3.2.4.1.2 Input/Output Ready Circuit

The four output FIFO input ready signals are ANDed together, producing the output FIFO write enable (HOFIFOWTENA) signal. This signal is used to send a write command to the FIFO (the FIFO uses two load clock inputs - one must be held high, while the other strobes in the data).

The four output FIFO output ready signals are NANDed together, producing the low true output FIFO enable (LOFENA) signal. This signal is used to control the output enable circuit of the FIFOs. LOFENA is inverted, producing the output bus data available (HOBDATAAVAIL) signal used in the three-wire handshake control circuitry.

The output ready circuit monitors the status of the first word location in the FIFO. The first word location is that area from which data are sent to the output of the input/output ready circuit. The unload clock circuit must be high to enable the output ready circuit; therefore, during a read operation (from the IOP), the input FIFO data available LIFDATAAVAIL) line is held false (high).

3.2.4.1.3 Input FIFO Clock

The multipurpose bus (MPB) data lines (LDI00 through LDI15) are the input signal lines for the input first-in-first-out buffer (FIFO). The output signals are sent to latches on the data swapping circuitry for the input FIFO. These latches rearrange the output of the input FIFO according to certain software commands. The signals resulting from the rearrangement are LIF00B through LIF15B.

When the input FIFO clear line - write to input FIFO order (HWTTOIBORD) - is low, it prevents a write operation into the FIFO. The HWTTOIBORD signal clears the FIFO. A high on this line also enables the write to input buffer (HWTTOIB) signal so that data can be clocked into the input FIFO.

A timing signal applied to the FIFOs clocks the data from the input FIFO into the input of the byte swapping registers. This signal originates from the IOP as either read direct (LRDDIR) or read delay (LRDDELAY).

Writing into the input FIFO is controlled by signals applied to the CKA and CKB inputs. The CKA signal (HWTINFIFO) clocks the data (LDI00 through LDI15) from the MPB into the input FIFO. The HWTINFIFO signal is controlled by the three-wire handshake circuitry. Note that the HIFIFOWTENA signal must be held high at the same time the data is being clocked into the input FIFO.

The CKB signal (HIFIFOWTENA) allows a write operation to occur, since it must be held high while the HWTINFIFO signal clocks the data. The HIFIFOWTENA signal is derived from the input ready signals (from each of the four input FIFOs).

3.2.4.1.4 Data Swapping Circuitry for the Input FIFO

The data swapping circuitry for the input FIFO is connected between the input FIFO and the IOP DBUS drivers. The IOP firmware is capable of writing information into either even (bytes 0 or 2) or odd (bytes 1 or 3) byte boundaries.

If the input DBUS writes into the odd byte boundary and the data swapping circuit is not used, the IOP firmware must write the first byte into an odd byte in memory. It then must constantly rearrange the bytes until the burst is complete. This necessary action allows sequential data to be placed into memory. Therefore, by using the data swapping circuit, the necessity is eliminated for IOP firmware to constantly rearrange the bytes input from the IOP interface.

The IOP source decode read direct (LRDDIR) signal is used when the information from the IOP DI is sent to memory on even byte boundaries.

The LRDDIR signal allows the information input by the IOP interface (bytes 0 and 1) to be sent through the alignment flip-flops and retain its integrity as bytes 0 and 1. These bytes are then sent to memory in the same byte order they had in the IOP interface.

If the information is to be sent to memory on odd byte boundaries, the IOP source decode read delay (LRDDELAY) signal chooses an alternate path through the alignment flip-flops. The LRDDELAY signal causes byte 1 to be latched. Byte 0 is sent through one of the alignment flip-flops and appears as byte 1 on the DBUS.

The IOP firmware is designed to recognize the DBUS information in byte 0 as invalid. Therefore, it is not stored in memory. This action occurs as a result of the LRDDELAY signal.

The data swapping circuit automatically realigns the data input on even boundaries by placing the first byte into an odd byte boundary. Byte 1 of the IOP DI information is then latched. On the next input FIFO read command, bytes 2 and 3 are output from the input FIFO. The information sent to the IOP DBUS is byte 1 (stored in the latch) and byte 2 (presently coming from the input FIFO). Byte 3 (also presently coming from the input FIFO) is then stored in the latch.

3.2.4.1.5 Output Drivers

The noninverting output drivers are between the input FIFO registers and the IOP DBUS. These drivers are controlled by the read input buffer (LRDIB and HRDIB) signals from the data swapping circuit for the input FIFO.

3.2.4.1.6 Output FIFO

The output bus data available (HOBDATAAVAIL) signal is the summation of information on the output FIFO ready lines (OFOR1 through OFOR4) from the four output FIFOs. This signal is applied to the three-wire handshake control logic.

The output FIFO write enable (HOFIFOWTENA) signal is the summation of information on the four input ready lines associated with the output FIFO. This signal must be active (high) to allow a write to output FIFO command to occur. The HOFIFOWTENA signal is applied to the CLKB input of the FIFOs.

The HOFIFOWTENA signal readys the output FIFO so that the HWTTOOB signal can clock the data into it. The HOFIFOWTENA signal is a summation of the output FIFO's input-ready lines.

The output FIFO drivers are permanently enabled.

The write to output buffer (HWTTOOB) signal allows input data to be clocked into the output FIFO. HWTT00B results from either the LWTDIR or LWTDELAY source decode signals being conditioned by the no clock output buffer (LNOCLOCKOB) signal. Note that the LNOCLOCKOB signal controls the byte swapping circuit.

During IOP power-up, the level orders are initially low signals; therefore, both the input and output FIFOs are cleared. Firmware design allows only one of the FIFOs to be active at a time.

While writing to the output FIFO, the input FIFO's clear line is low, keeping it cleared. Conversely, while writing to the input FIFO, the output FIFO's clear line is low, keeping it cleared.

3.2.4.1.7 Data Swapping Circuitry for the Output FIFO

The data swapping circuitry for the output FIFO is located in front of the stage where the data are clocked into the output FIFO. The LNOCLKOB signal prevents data from being written into the output FIFO during a swapping sequence. The write delay (LWTDELAY) signal (source decode from IOP) clocks data into a set of octal latches so that they can be held until the LNOCLKOB signal allows the data to be written into the output FIFO. The write directly (LWTDIR) signal allows the data to be written directly into the output FIFO.

The swapping of data before the output FIFO consists of switching the eight most significant bits (MSBs) with the eight least significant bits (LSBs).

3.2.4.1.8 Parity Generator and Checker Circuit

The parity generate/check circuit is a single-parity generator connected to the MPB. This circuit constantly generates parity bits to check on the flow of data (in both directions) on the MPB. For output data, the parity bits are sent out with the data. For input data, the circuit blocks the outgoing parity bits and checks incoming data for a parity error.

Since the IOP and IOP DI boards are required to handle single or double byte transfers, there are two parity generator and checking circuits. The two circuits are required when transmitting double byte information. During a single byte transmission, the second byte coming from the IOP is unusable, since it is not in any known state.

The second circuit is not necessary when receiving single byte information, because the data source from the IOP MPB is not driven and is pulled high by the terminating resistors.

When the IOP is in the transmit mode and sending a single byte over the IOP MPB, the LHALFWORD level order signal is received from the IOP. This action causes multiplexer A inputs from the first parity generator/checker circuit to be selected.

When the IOP is in the transmit mode, the send (LSEND) signal from the gate is low, placing the second parity generator/checker circuit in the odd parity mode. If the data sent to the second parity generator/checker circuit is odd, the odd summation output is activated, sending an odd parity indication to the other end of the IOP MPB.

The parity odd (LPARITYODD) input signal from the IOP MPB is generated by the parity generator and checking circuitry on the controller. The condition of the LPARITYODD line is determined by the parity of the bytes transmitted by the controller.

When the IOP is in the receive mode (LSEND is high), the LPARITYODD signal is sent to the gate. This action places the second parity generator/checker circuit in either the even or odd parity check mode, depending upon the state of the LPARITYODD line.

The LHALFWORD line is low only when single byte data is transmitted. In this case, the input data is eight bits wide. The first parity generator/checker circuit receives the eight bits applied to it and generates a summation odd or even signal, depending upon the parity of the data it is receiving.

The summation odd line of the second parity generator/checker circuit is directly connected to the 'I' input of the first parity generator/checker circuit. These integrated circuits receive the IOP data bits on the LDI00 through LDI15 lines and generate a parity signal, depending upon the parity of the data applied to the parity generator/checker circuits. These signals are then applied to the multiplexer. The multiplexer must first be placed in the double byte mode by a high on the LHALFWORD line. The B inputs to the multiplexer are then selected and sent to the IOP.

The LPARODD signal from the multiplexer is used when the IOP is in the receive mode. It is sent to the input FIFO. This parity signal is clocked into the input FIFO at the same time that the byte or bytes coming from the IOP are clocked into the input FIFO.

3.2.4.1.9 Input/Output Transfer Difference Counter

The input/output transfer difference counter circuit keeps an accurate account of the number of bytes that are either loaded into or read out of the FIFOs.

A counter for each FIFO (input and output) keeps track of the number of bytes being transferred into or out of each respective FIFO.

The IOP receives two interrupt levels: high and low. The high interrupt level informs the firmware to service the FIFO transfer sequence. This action is taken because either the output FIFO contains less than four bytes or the input FIFO contains more than 12 bytes of data. A high-level interrupt is issued to request service, since either the output FIFO contains less than twelve bytes or the input FIFO contains more than four bytes of data. These interrupts prevent the overloading of the FIFOs, which, in turn, prevents any transferred data from being lost.

3.2.4.1.10 Transceiver Transmit/Receive Mode Controlling Circuit

The parallel poll (HPARPOLLORD) signal (a level order from the IOP) issues the IOP output attention (LCOATN) signal and blocks the MPB data drivers (HTXCHBUS) from transmitting any data.

When the IOP places the DI board in the transmit mode (data are transferred from the DI board to the MPB), the IOP executes the write from output buffer level order (HWTFOBLORD). This signal, along with a NOT parallel poll order, and a NOT inhibit line, activates the DI board MPB drivers.

While sending data from the output FIFO to the MPB, the IOP output data available (LCODAV) line is being sourced and looking for the no data accepted (LNDAC) and the not ready for data (LNRFD) signals from the receiving controller. When the IOP becomes aware that the LNDAC line from the controller is going high (at some time after the LNRFD line goes low, depending upon the controller), the DI issues the data accepted (HDAC) signal to clock the transfer counter and I/O transfer difference counter down by one count. The HDAV signal is a synchronous 150 nsec pulse that clears the LCODAV line 75 nsec after the HDAC signal is issued. This action readies the DI board to transfer the next byte of data. This sequence of events continues as long as data are available and the receiving controller continues to sequence through the three-wire handshake signals.

The data continues to be available and is only stopped when the IOP terminates by emptying the output FIFO and causing an end or identify (EOI) to occur. The write from output buffer level order (HWTFOBLORD) is then reset.

The LSENDEOI signal is generated when the transceivers are in the parallel poll mode. The LSENDEOI signal is also generated when the IOP is in the transmit mode and a LENDMSG signal from the transfer counter indicates that the count has reached maximum and there are no more data to be transferred.

On IOP power-up, the interface clear order (LIFCORD) signal is generated, resetting the IOP output data available (CODAV) circuitry and the initial NDAC flip-flop.

When the IOP wants to receive data, the write to input buffer level order (HWTTOIBORD) is set and the DI board waits for the data available (LDAV) signal. The IOP DI shapes the DAV signal into a 150-nanosecond pulse and issues a write to the input FIFO (HWTTOIB) signal to clock the data into the input FIFO. At the same time, the DI signals that it has accepted the data and is ready to accept more data during the next clock period.

When the NRFD line goes low, the transmitting controller recognizes this condition and forces the data available (DAV) line high at some later time.

3.2.4.1.11 Output FIFO Data/Command Interfacing Circuit

The IOP interface attention (LCIATN) line is generated, using the pull attention (LPULLATN) signal from the output FIFO circuit and the HPARPOLLORD signal from the IOP. The LPULLATN signal comes from the output of the output FIFO and is generated at the input to the output FIFO by the LPULLATN order. The LPULLATN signal is active true (low) when the information sent to the output FIFO is a command rather than data.

3.2.4.1.12 Handshake Circuitry

The prerequisites for starting a three-wire handshake sequence are low NDAC and NRFD lines.

When the IOP DI board circuitry is in the receive mode, it will be sending a high IOP output not ready for data (LCONRFD) signal to the transmitting controller. The transmitting controller becomes aware that the DI board circuitry is ready to receive data; therefore, it issues a low data available (LDAV) signal to the IOP. The IOP DI board circuitry recognizes the DAV signal, latches it, shapes it into a 150-nsec-long pulse, and approximately 450 nsec later, issues a high IOP output no data accepted (CONDAC) signal on the LCONDAC line. At the same time, a low LCONRFD signal is issued. When the LNRFD line goes low, the transmitting controller recognizes the condition and forces the LDAV line low at some later time.

The HIFIFOWTENA signal is high, since a write operation is being enabled into the input FIFO. Since the input FIFO is cleared by a low from the HWTTOIBORD level order, it will be ready to accept data.

The HWTTOIBORD signal is now high, since the IOP has issued this order to begin a receive sequence. The LDAV line is also low, since it is a prerequisite for beginning a three-wire handshake sequence.

These sets of conditions produce a 150-ns-shaped pulse (HWTTOIB) that clocks the data presently on the IOP into the input FIFO. These sets of conditions are also the source of the LCONDAC signal. The LCONDAC signal is set its high state approximately 150 ns after the data are clocked onto the IOP.

When the IOP is unable to receive data, it causes the LNRFD and LNDAC lines to go low.

The LPAROLLORD signal is an inversion of the HPARPOLLORD signal from the P1C plug on the IOP board.

3.2.4.1.13 Stop Communications Signal Generator

The stop communications (LSTOPCOMM) signal is generated using the IOP output end or identify (LCOEOI) or LEOI signals. The LCOEOI signal is sourced by the IOP. The LEOI signal is received from the MPB DI bus.

If a LCOEOI signal is not being sent (LCOEOI line high) and a low is received on the LEOI line while a transfer is in progress, the stop command (HSTOPCOMM) will be issued. The LEOI signal results from several other conditions: a receiving data device malfunction; a turned-off receiving device not receiving any more data; or full controllers buffers. Regardless of the condition, the low output indicates that the controller cannot receive any more data. Therefore, the HSTOPCOMM line indicates that no more data should be transferred. This procedure is followed rather than waiting for an IOP time-out to occur.

3.2.4.2 IOP Time-Out Circuitry

The IOP time-out timer is preloaded with a count of 512 when the IOP controller issues a time output start (LTIMOUTSTART) signal. The time-out counters begin counting at the H3 clock rate when the LTIMOUTSTART line goes high. A carry is generated whenever the count reaches 4095. The 4095 count minus the preload count of 512 (3583) multiplied by 150 nsec equals 537.45 microseconds, or the IOP time-out rate. The IOP time-out (HCLTIMEOT) signal is then sent to the IOP interrupt circuitry.

3.2.4.3 Input/Output Processor Multipurpose Bus

The circuitry for the IOP MPB is listed below:

1. Transceiver circuitry for MPB control signals
2. Data signal transceivers

3.2.4.3.1 Transceiver Circuitry for MPB Control Signals

The inhibit (LINHIBIT) line from the MPB prevents all bus drivers from transmitting data from the DI board to the MPB, except the free running clock (L5.0688MHZ) signal. The remainder of the MPB signals, except the 150 nsec clock (L150NSCLK), attention (LATN), and interface clear (LIFC) signals, are further conditioned by the parallel poll order (LPARPOLLORD) and HWTFOBLORD (level order from the IOP) signals. When the IOP is in the parallel poll mode, the transmit multipurpose bus (HTXCHBUS) signal is low. Note that both the LEOI and LATN lines are also low at this time. The remaining IOP lines are deactivated from the MP.

3.2.4.3.2 Data Signal Transceivers

The output FIFO data lines (LCO00 through LCO15) are enabled by the HTXCHBUS signal so that they can transmit. These lines then become the MPB data lines (LDI00 through LDI15) when the IOP is in the transmit mode. When the IOP is in the receive mode, LDI00 through LDI15 are directly attached to the input FIFO, and the HTXCHBUS line is low. These sets of conditions control the drivers on the IOP DI board.

3.2.4.4 Clock

The IOP DI board receives a 150-ns lock signal from the IOP via P1C-02. The input clock freerunning (LICLKFREERUN) signal is not conditioned by the stop system clock (LSTSC) signal. However, the LSTSC signal has some delays that differentiate this clock from the SelBUS backplane clock.

A set of gates is used to condition the clock with LSTSC to produce two clock (L2CLK and L2CLOCK) signals. These two signals become the source for all the three-clock (H3CLK) signals. The H3CLK signals distribute the system clock (150-ns pulse) throughout the IOP DI board circuitry. In addition, there is an H2CLK signal produced with gate conditioning by the LSTSC signal.

The L150NSCLK signal sent to the MPB backplane is the L2CLK signal, driven by a driver and conditioned by the inhibit (HINHIBIT) signal. Note that this clock signal is a function of several gate delays and does not match any of the SelBUS backplane clocks.

3.2.4.5 Transfer Counter

The transfer counter (a four-bit synchronous up/down binary counter) is used as a down counter with a preset function. The transfer counter is clocked by the H3 system clock (H3CLK) signal.

Since the IOP DBUS uses low true data, the transfer count sent to the transfer counter is in complement form. The end-of-message (EOM) signal must be generated while the last byte of data is being clocked into the FIFO; therefore, the transfer count (in complement form) must be one greater than the true transfer count.

The destination decode load transfer count (LLDTFCNT) signal originates from the IOP destination decode logic, is selected by the HCREG25, 26 and 27 signals, and is enabled by the HCREG24 and 25 signals. The LLDTFCNT signal causes the counter to be loaded with the count that is presently on the buffered IOP bus (LDB00B through LDB15B).

The enable transfer count (LENATFCNT) signal, when true (low), allows the transfer counter to initiate a down count (the transfer count in complement form). The LENATFCNT signal is true whenever the clock in output buffer (HCKINOB) or write to input buffer (HWTTOIB) control signals are used during the three-wire handshake protocol sequence.

The end message (LENDMSG) signal is generated by the transfer counter on the last byte transferred from either the input or output FIFOs.

A parallel poll order can set the send end or identify (LENDEOI) and attention (ATN) lines while the IOP is in the receive mode.

When the IOP is in the transmit mode, the end or identify (EOI) signal is sent by the IOP output EOI (LCOEOI) line. The EOI line can be set by the force EOI (LFORCEEOI) signal coming from the IOP. The LSENDEOI signal can also set the EOI line as a result of an LENDMSG signal.

When the IOP is in the receive mode, the EOI line sends the data bus EOI (LDBEOI) signal to the IOP.

The LCOEOI signal is generated when either the LSENDEOI or EOI order (from the IOP) is sent, indicating that the output FIFO is ready to accept data.

3.2.4.6 Control Panel/Interval Timer Logic

The circuitry for the control panel/interval timer logic is listed below:

1. UART/interval timer selection circuit
2. Interval timer
3. Interval timer rate selection circuit

3.2.4.6.1 UART/Interval Timer Selection Circuit

The IOP performs both SCP and teletypewriter (TTY) functions. Universal Asynchronous Receiver/Transmitter (UART) 0 is the master UART that controls the CRT (SCP and operator console functions). A separate UART, with its driver and receiver, controls a control console slave CRT for down-line loading or remotely controlling a down-line program load from some variable source. The UARTs and drivers have full duplex modem interface so that data can be sent over a full duplex asynchronous modem.

The data sent to the octal latch come from the buffered IOP DBUS (LDB00B through LDB07B). The octal latch issues certain commands for the sequence of events necessary to select, load, or control the UART or interval timer. The control data are clocked into the control latch by the LLDCONT signal (P1A-03) from the IOP.

Proper bit arrangement on the buffered DBUS allows the required true signals at the output of the octal latch. These output signals are then used to select either the interval timer, UART0, or UART1, or to determine whether to perform a read or write operation. They also select the registers associated with the selected UARTs or interval timer.

The IOP destination decode load control (LLDCONT) signal is active when data bits are to be intepreted as commands for controlling the UARTs and interval timer. The LLDCONT signal allows time to latch its output information from the buffered DBUS. At other times, this signal is inactive and information on the buffered DBUS is considered data.

When the UARTs are selected, the LREAD signal is sent directly to the read/write port of the UARTs, placing them in either the read or write mode.

When the interval timer is selected, the LREAD signal is sent to the interval timer and UARTs. Note that the interval timer and UARTs are selected via enable lines to each device. The READ signal is then used to place the interval timer in either the read or write mode.

The HREAD signal is also sent to the UART/interval timer driver/receiver interface circuit to allow data to be sent out to the MPB during the execution of a read command.

The sequence clock (HSEQCLK) signal is delayed by 225 nsec from the time the LLDCONT signal goes false and is used to clock data to or from the interval timer or UARTs.

3.2.4.6.2 Interval Timer

The RTOM/interval timer circuitry contains interrupt storage capabilities. Control functions are provided by the IOP microengine. The interval timer is 16-bits wide, is loaded by the IOP, and can be interrogated at any time by the IOP. The most significant 16 bits of the RTOM/interval timer are provided in firmware by the IOP microengine. Response to an interrupt occurs within 50 microseconds of the external interrupt.

The interval timer is a programmable interval timer/counter integrated circuit (IC). There are three independent counters and a control word register on this IC. The DI board presently utilizes only one of the counters.

The LLDCONT control signal at P1A-03 is a destination decode signal from the IOP. The LLDCONT signal clocks in the control signals from the IOP and is used to read/write or select counters and the control register of the interval timer. The bit assignments for the LLDCONT signal are shown below:

Bit 4 - Register select 0 (HREGSEL0)
Bit 5 - Register select 1 (HREGSEL1)
Bit 7 - Read/write

The read/write control signals are clocked in from the IOP to either read from or write into the interval timer. The read/write data consist of the values on the UART/interval timer bus (HUT00 through HUT07). With both read/write lines in the high state, a no operation (NOP) results in the third state mode of the interval timer. Therefore, the CS NOT input is permanently enabled.

The HREGSEL0 and HREGSEL1 control signals address either the control register or one of the three available counters.

The HUT00 through HUT07 lines form an eight-bit, three-state, buffered data bus, which interfaces with UART0 and UART1.

Data are input to the interval timer from the IOP via a destination decode circuit. This circuit latches the input data into a register. Data is read from the register using the firmware-controlled read/write line.

Data is read from the interval timer/counter and placed into a register via the read clock enable (HRDCLKENA) command. This command is initiated 225 nsec after the LLDCONT destination decode signal from the IOP, as shown in Figure 3-4. Data is then read from the register and sent to the IOP via another destination decode read device interface (LRDDI) signal on P1A-46. This action places the data from HUT00 through HUT07 onto the IOP DBUS (LDB00 through LDB07).

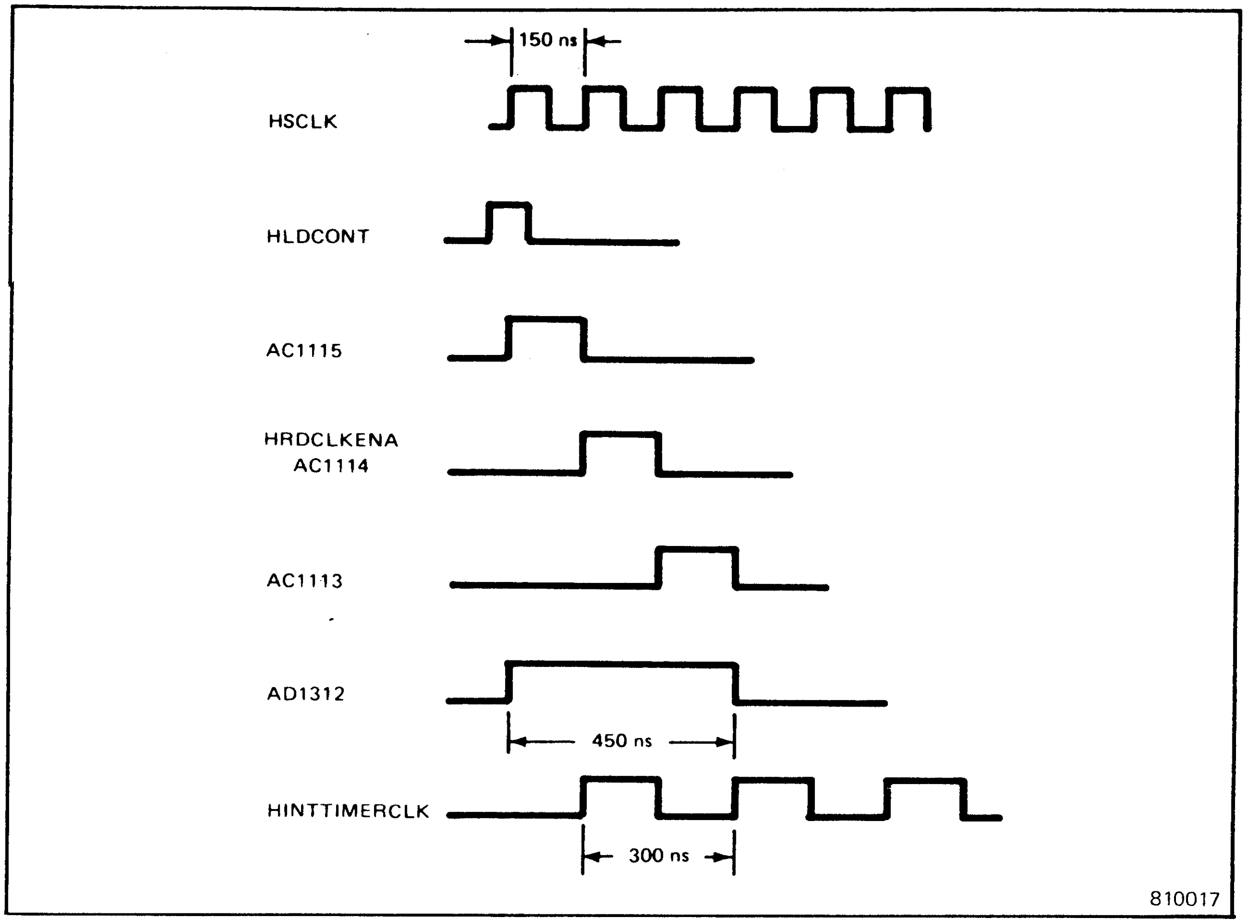


Figure 3-4. UART/Interval Timer Timing Diagram

3.2.4.6.3 Interval Timer Rate Selection Circuit

The multiplexer selects one of the following inputs, depending upon the value of the rate select (HRATESEL00 and HRATESEL01) signals. Note that HRATESEL00 is the MSB and HRATESEL01 is the LSB.

A value of zero (00) selects an external source. A value of two (10) or three (11) selects a continuously running clock that is 300 nsec low and 300 nsec high, producing the clock signal used by the interval timer. A value of one (01) selects the real time clock (60 or 120 Hz) present on the SelBUS backplane.

3.2.4.7 Ports 0 and 1

The IOP uses the speed and power of its microengine for handling two full duplex asynchronous communication lines capable of transferring data at up to 19.2K bits per second. The communication lines interface through a programmable communications interface (PCI) circuit to the interface logic of the IOP. Both lines are only operated in the full duplex mode. Interfacing circuitry for a BELL 103 or other full duplex RS232C modem is provided on the DI board. The baud rate of the on-board baud rate generator is jumper-selectable.

The IOP-DI uses two separate UARTs and a common data bus (HUT00 through HUT07). The UARTs communicate (in a parallel mode) with the IOP and support serial data transfers on both the current loop and a full duplex RS232C modem interface. Using two UARTs enables the capability of including both a master and control console slave port in the IOPDI.

The DI board uses a 2651 programmable communications interface (PCI) IC. It also contains a 5.0688 MHz oscillator. The 5.0688 oscillator is used to generate the UART clock signals.

The PCI is a universal asynchronous data communications controller chip designed primarily for microcomputer systems. The PCI directly interfaces to the IOP microengine, is used in an interrupt-driven system environment, accepts programmed instructions from the IOP microengine, and supports a full duplex asynchronous communication interface.

The PCI serializes parallel data characters, received from the IOP microengine, for transmission. Simultaneously, it can receive serial data and convert them into parallel data characters for input to the IOP microengine.

The PCI contains a baud rate generator that can be programmed to generate internal transmit or receive clock signals. Sixteen different baud rates can be selected under jumper control.

Software support for the PCI circuit is provided by the operator console software, using the extended I/O format, commands, and status. This approach minimizes the number of SelBUS protocols and allows software programs or macrodiagnostics to be loaded from a remote location. However, the receive subchannels are required to operate continuously.

Drivers and receivers are used to support the RS232C interface. These drivers and receivers require a ± 12v power supply. The 12 volt supply voltage is obtained by using a diode voltage-dividing circuit and ± 15 volts from the SelBUS.

The DI board sends the IOP transmitter empty, receiver ready, and transmitter ready signals for both of the UARTs. This task is accomplished via a logic latch that is used to enable the IOP and determine the UART status.

3.2.4.8 External Interrupts

The external interrupt logic interfaces four external interrupts (LEXDINT00 through LEXDINT03) to the IOP. The logic consists of four flip-flops that latch the external interrupt signals until the IOP can service them. The IOP is notified of an external interrupt by the external interrupt on (HEXINTON) signal at PIC-34. Upon seeing this line active, the IOP will, when ready, search the interrupt levels and service the highest interrupt level. It then will clear all the interrupts. If another interrupt is pending, HEXINTON will again be active and the interrupt search procedure will continue. The external interrupts are applied to connector J3.

There is a set of four polling lines coming from the IOP via the drivers to a socket on the DI board. These lines are used by the IOP to indicate to the external devices that polling is in progress.

There are four lines from the IOP to the external devices that enable the IOP to request an interrupt from the external device. These lines are applied to the external device on connector J3.

3.2.4.9 System Control Panel Interface Circuit

When using the IOP DI board as the system control panel it is connected to the DI board associated with the CPU or straight into the CPU on the PIC connector. The system control panel connection to the CPU board is paralleled to jack 8 on the IOP DI board. The system control panel cable, connected to jack 8 on the IOP DI board, allows the generation of the system control panel key functions from the CPU control circuit to the IOP.

The signals normally coming from the CPU are sent to a latching circuit. Some of these signals are sent to the IOP for testing purposes. Two of the signals - reset over (LRSTOVR) and reset switches (LRSTSW) - control the latches that are contained on the IOP DI board.

The disable CACHE signal (HDISCACHE) is inverted and sent to the CPU. Its purpose is to disable the CACHE memory when the address stop compare command is enabled by the SCP. This feature is available for future product enhancement.

The other signals to the system control panel interface circuit perform the same functions that their counterparts perform when using the control panel keys. This action is needed to support future product enhancement.

The system control panel uses signal switch indicator suffixes so that the signals are set apart from the control panel key signals. The suffix ORD (order) allows the clock override, run, halt, attention, and initial program load (IPL) signals generated by the CPU to be distinguished from the same functions generated by the system control panel switches.

3.2.5 Operator Console Interface Description

The IOP uses its speed and power to handle two asynchronous communications lines. These lines interface directly to logic on the IOP interface board. This structure provides the most efficient and economical solution for interfacing with the operator console. Register buffering is provided at both the input and output ports of the PCI. These registers, controlled by the hardware, act as an integral part of the PCI, providing an additional holding register for transmit and receive operations.

PCI status is collected by the firmware and posted through the holding register. The holding registers allow the IOP microengine to load or read the PCI data or status during one system clock. Sequencer hardware transfers the information to or from the PCI chip and generates the required timing signals.

As data is received, it is sent to an input buffer allocated by the operator console software. All character comparisons must be performed by the software. Transmission status is reported to the status buffer when the residual byte count is reduced to zero. The status entry then flags the operator console software to process the received message and test the status entry for transmission errors. If a modem error has occurred, status bits (containing the current modem status) are input to the software. The console subchannel commands can be data and command chained. They, therefore do not require any CPU intervention after each transmission. The operator console interface is used to connect the operator console to either two full-duplex RS232C or two current loop communication lines. Electrical interfaces to BELL 103 (300 baud) or any other full-duplex EIA RS232C modem are also provided.

3.2.5.1 Operator Console Interface Characteristics

The operator console interface characteristics are as follows:

Bit rate: internally jumper selectable from 110 to 19.2K bits/second, as shown in Table 3-3. The jumpers are located at A-17 (X1) on Model 8000 and H-23 (X5) on Model 8001 and IPU Console IOP. Jumpers 8 (most significant bit) through 5 (least significant bit) set the baud rate for the master port and jumpers 4 (most significant bit) through 1 (least significant bit) set the baud rate for the control console slave port. Note that when the system is operating in the SCP mode, the same information is successively written to each port and therefore output at the slower rate. Therefore, when operating with only the master port connected, set both ports to operate at the same baud rate. The ports operate independently in the operator console mode; therefore, the outputs reflect the different baud rates.

Table 3-3.
Baud Rate Selection

Bit Pattern (MSB) (LSB)	Baud Rate	Parity Gen/Check	Stop Bits	Word Length
0000	110	Even	2	7 bits
0001	300	Even	1	7 bits
0010	110	None	2	8 bits
0011	134.5	None	1	8 Bits
0100	150	None	1	8 bits
0101	300	None	1	8 bits
0110	600	None	1	8 bits
0111	1,200	None	1	8 bits
1000	1,800	None	1	8 bits
1001	2,000	None	1	8 bits
1010	2,400	None	1	8 bits
1011	3,600	None	1	8 bits
1100	4,800	None	1	8 bits
1101	7,200	None	1	8 bits
1110	9,600	None	1	8 bits
1111	19,200	None	1	8 bits

Link capabilities: two full-duplex lines.

Interface: Electronic Industries Association (EIA) RS232C or current loop interface.

Clock generation: baud rate clock is internally selectable. The baud rate selection circuitry provides for the selection of the I/O port's baud rate. Control circuitry provides control for the loading and unloading of data to/from the universal asynchronous receivers/transmitters (UARTs) and interval timer.

3.3 SCP SelBUS Transfer Protocol

The SelBUS transfers used by the IOP in communicating with the CPU (when acting as the SCP) are identical to those used with the 32/70 series CPU. Note that these transfers are not Extended I/O.

3.3.1 Types Of SelBUS Transfers

The microprogram uses the following types of SelBUS transfers:

1. The advanced read status transfer (ARSTX), originated by the CPU as a prerequest for a status transfer.

2. The read status transfer (RSTX), originated by the CPU as the actual request for status transfer.

3. The data return transfer (DRT), originated either by the IOP firmware in response to an ARSTX/RSTX pair or by the memory as a response to the MRT.

4. The write data or order transfer (WDOT), originated by the CPU as a response to a data read operation.

5. The memory write transfer (MWT), originated by the microprogram to write a data word into memory.

6. The memory read transfer (MRT), originated by the IOP firmware as a request for a memory read operation.

Note that in all SelBUS transfers, the physical address and subaddress of the SCP must be zero.

3.3.1.1 ARSTX/RSTX Transfer

The ARSTX and RSTX formats are shown in Figure 3-5. The CPU originates these transfers to communicate with the SCP firmware. In most cases, the SCP firmware actually requests the ARSTX and RSTX by generating an SCP attention signal. The CPU responds with the ARSTX and RSTX to identify the cause of the attention signal. The microprogram responds with a DRT that informs the CPU of the task to be accomplished.

Function Codes for the ARSTX/RSTX formats are listed below:

0001 - Send function
0010 - Send data
0110 - Send IPL data (device address and subaddress)
1001 - Send function for request of control switches memory address

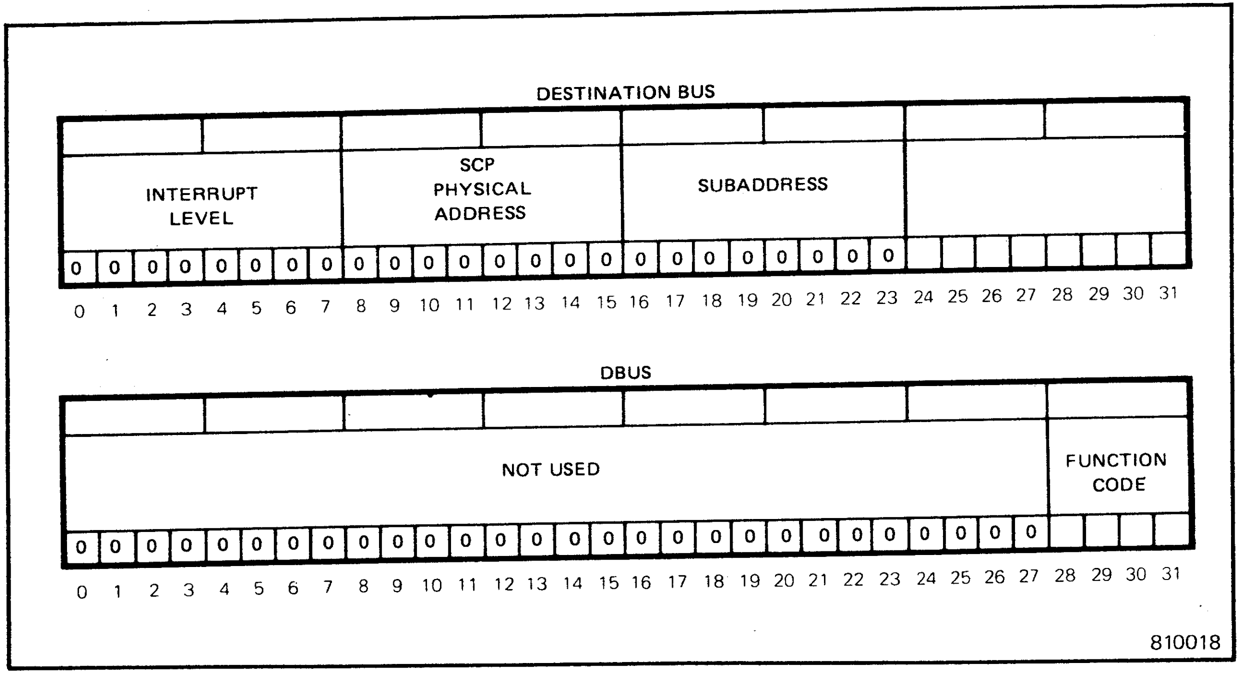


Figure 3-5 ARXTX/RSTX Formats

In some cases, the CPU can issue an ARSTX/RSTX pair without having previously received an attention signal from the firmware. This action occurs when the CPU is requesting that the SCP perform some operation.

The ARSTX is always the first transfer of the pair. The ARSTX informs the firmware to prepare to execute the specific function. When the microprogram is ready to comply, it returns a ready signal to the CPU. The CPU responds with the RSTX transfer, causing the microprogram to execute the function and transfer the requested information to the CPU in a DRT that completes the ARSTX/RSTX operation.

The ARSTX/RSTX is capable of specifying four different operations by configuring the SelBUS DBUS as follows:

1. Data bit 28 specifies that the microprogram must request the control switch dedicated address from the CPU.

2. Data bit 29 specifies that the microprogram must send the IPL I/O device address to the CPU.

3. Data bit 30 specifies that the microprogram must send the data word for a CPU write register operation.

4. Data bit 31 specifies that the microprogram must send the function code or the operation to be performed by the CPU.

The following paragraphs discuss the four variations of the ARSTX/RSTX in more detail and indicate the circumstances under which the variations are used.

The request control switches address variation of the ARSTX/RSTX is not preceded by an SCP attention signal. It occurs during the CPU IPL sequence and causes the microprogram to request the control switches dedicated memory address from the CPU. The control switches address request is made in the form of a DRT that is returned to the CPU following the RSTX. Format A of the RSTX DRT (shown in Figure 3-6) illustrates this procedure.

The send IPL data variation of the ARSTX/RSTX is not preceded by an attention signal from the firmware. This variation occurs during the CPU IPL sequence and causes the microprogram to transfer the IPL I/O controller and device physical address and subaddress to the CPU in a DRT following the RSTX. Format B of the RSTX DRT (shown in Figure 3-7) illustrates this procedure.

Function Codes for RSTX DRT Format B are listed below:

0000 - Read source (see note 1)
0001 - Write destination (see notes 2 and 3)
0010 - Instruction step

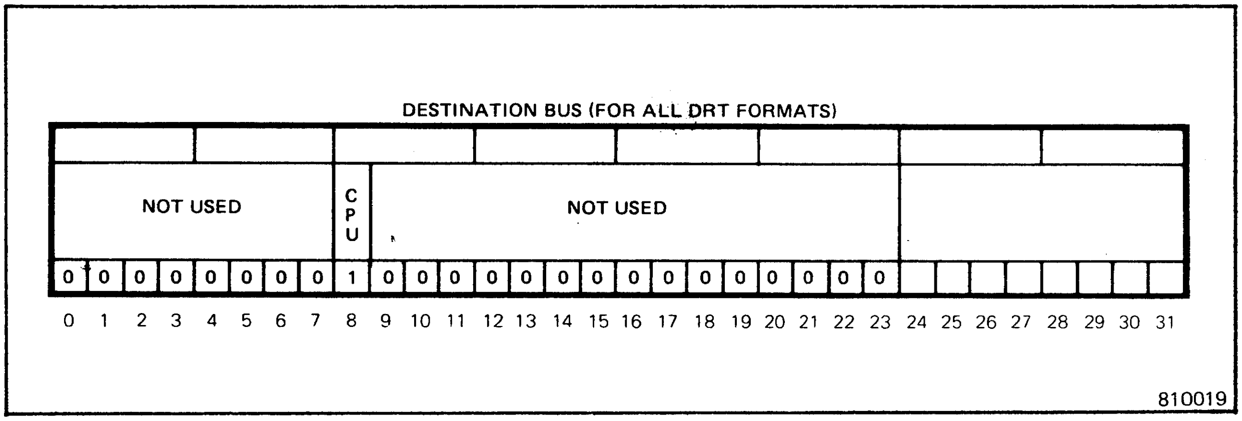


Figure 3-6 RSTX DRT Format A

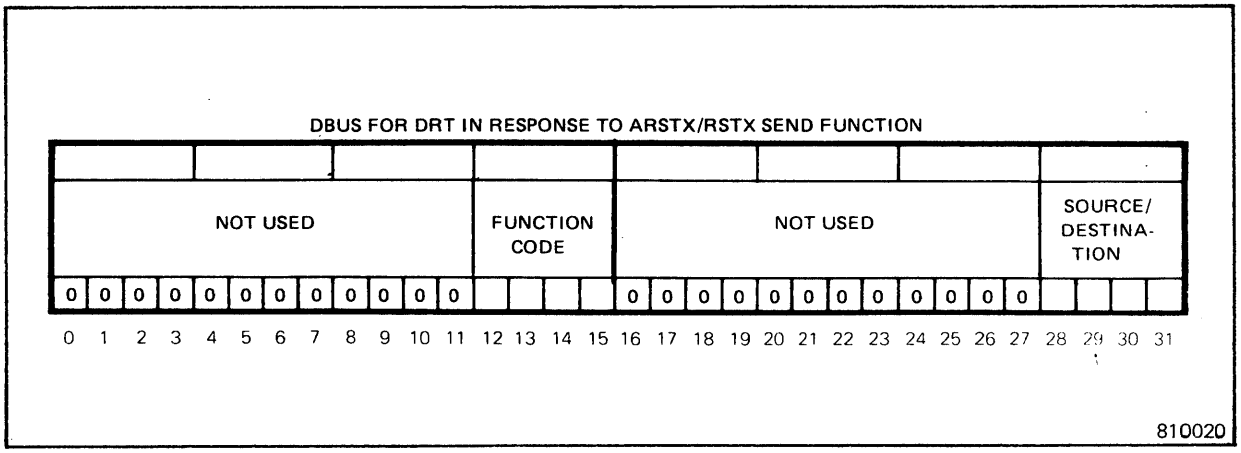


Figure 3-7. RSTX DRT Format B

Source/Destination Codes for RSTX DRT Format B are listed below:

0000 - Register 0 - General purpose register (GPR)
0001 - Register 1 (GPR)
0010 - Register 2 (GPR)
0011 - Register 3 (GPR)
0100 - Register 4 (GPR)
0101 - Register 5 (GPR)
0110 - Register 6 (GPR)
0111 - Register 7 (GPR)
1000 - Program status word (PSW)
1001 - Effective address (see note 2)
1010 - Request control switches dedicated memory address (see note 2)

Note:

1. For the read source function codes, the CPU responds with a WDOT containing the requested data.

2. Destination codes that refer to the effective address cannot be used with a write function code.

3. Following the DRT - write destination transfer - the CPU responds with an ARSTX/RSTX - send DBUS transfer - to obtain the data to be stored by the write destination function.

The CPU IPL firmware sequence is initiated by inputting an IPL command on the CRT, which is acting as an SCP.

The send function variation of the ARSTX/RSTX is preceded by an attention signal. The send function variation is used by the CPU to identify the cause of the attention signal. In response, the microprogram sends a function code of the operation to be performed and, if the operation is a read or write, the source or destination code of the operation to be performed. These codes are returned in the DRT SelBUS transfer as shown in Figure 3-6.

The send data variation of the ARSTX/RSTX is not preceded by an attention signal from the firmware. However, the send data variation must be preceded by an ARSTX/RSTX send function with a write function code DRT response. In this case, the send data variation causes the microprogram to transfer the data to be written in the CPU register (specified by the previous DRT) in a DRT following the RSTX. Format C of the RSTX DRT (shown in Figure 3-8) illustrates this procedure.

3.3.1.2 Data Return Transfer

A DRT can be either generated or received by the SCP microprogram. The only DRT received is from the memory in response to an MRT. In this case, the DRT contains the data word read from memory.

The only DRTs generated by the SCP firmware are in response to an ARSTX/RSTX pair from the CPU. In these cases, the DRT either contains information requested by the CPU or the code of an operation to be performed by the CPU.

When the ARSTX/RSTX pair specifies a request control switches address operation, the DRT contains a function code for a CPU read operation and a source code of the control switches dedicated memory address (see Figure 3-6). The CPU responds to the DRT with a WDOT containing the control switches dedicated memory address.

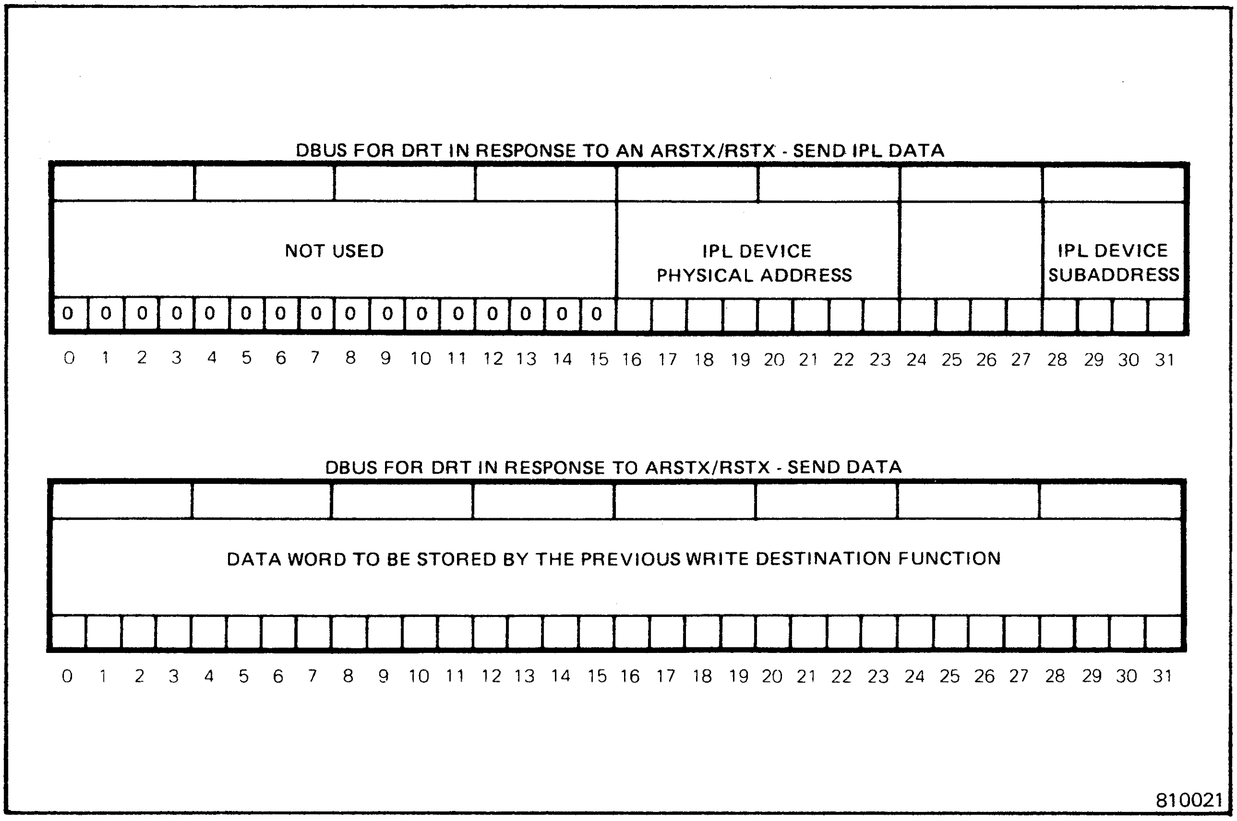


Figure 3-8. RSTX DRT Format C

When the ARSTX/RSTX pair specifies a send IPL data command, the DRT contains the IPL I/O controller and device physical and subaddress (see Figure 3-6).

When the ARSTX/RSTX pair specifies a send function operation, the DRT contains a function code for the operation to be performed by the CPU and a source/destination code of the function to be performed. The function code can specify a CPU read, write, or instruction step operation. The source/destination code specifies the CPU register that is to be read from or written into (see Figure 3-6).

If the function code specifies a read operation, the CPU reads the contents of the register and returns a WDOT containing the data. If the function code specifies a write function, the CPU generates an ARSTX/RSTX pair to send data.

When the ARSTX/RSTX pair specifies a send data operation, the DRT contains the data to be loaded into the CPU register specified by a previous ARSTX/RSTX send function transfer sequence (see Figure 3-8).

3.3.1.3 Write Data or Order Transfer

The WDOT is used by the CPU to send data to the SCP. The data were requested by a previous DRT read function response to an ARSTX/RSTX send data function. The WDOT contains the requested data on the SelBUS DBUS, as shown in Figure 3-9.

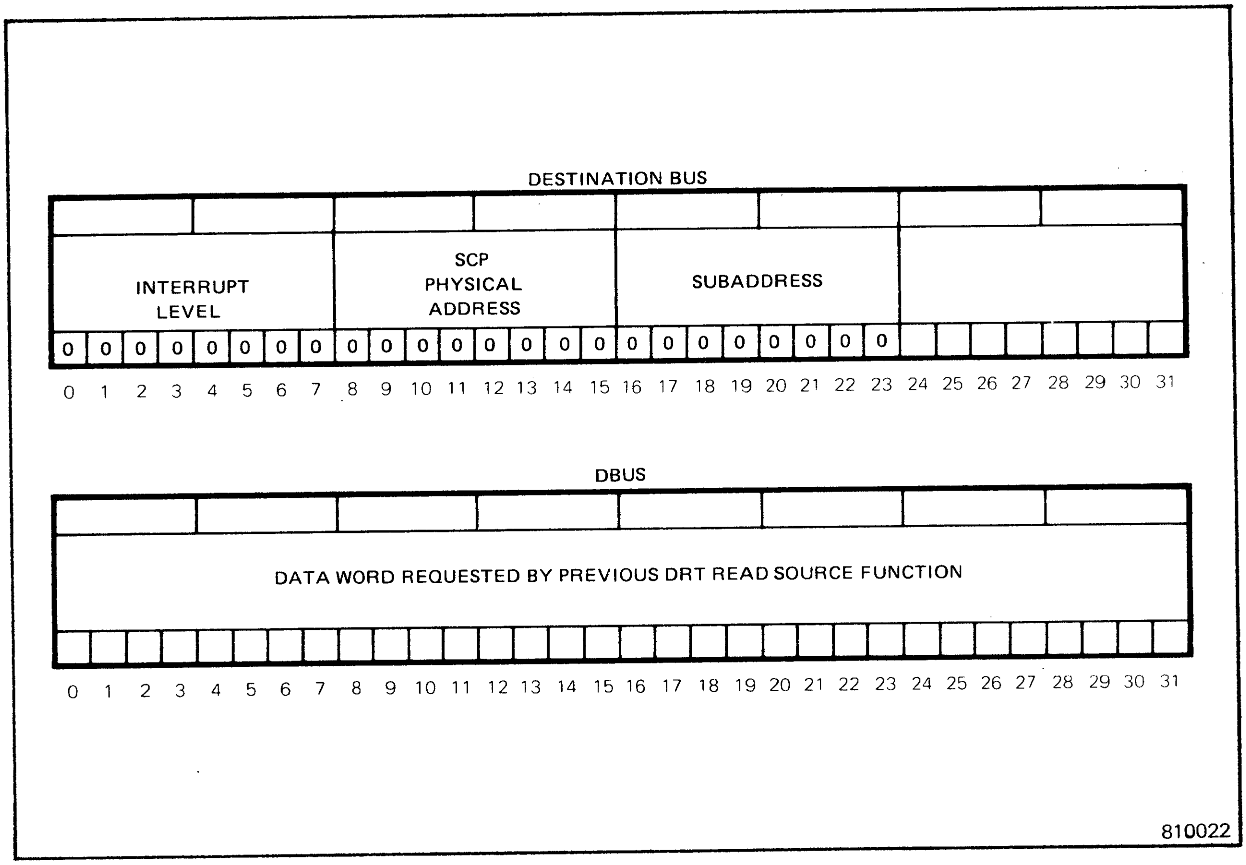
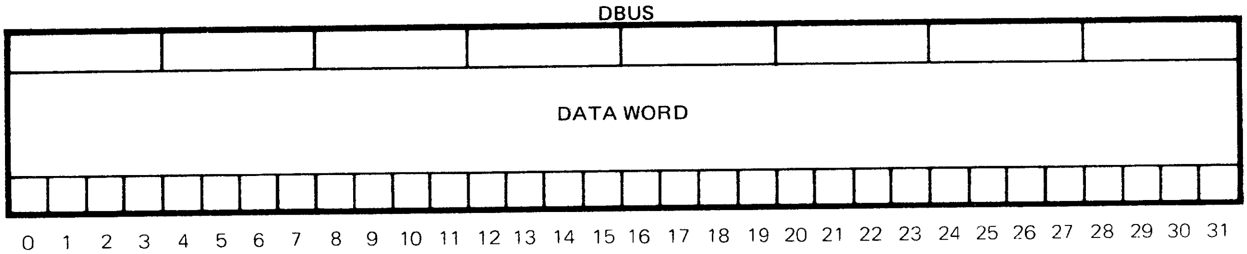


Figure 3-9 WDOT SelBUS

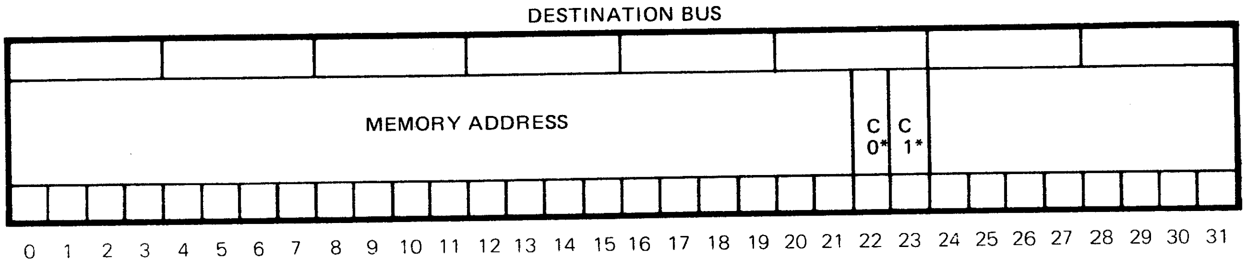
3.3.1.4 MWT SelBUS Transfer

The format for an MWT SelBUS transfer is shown below:



*For 'C' bits, see note 2.

Bits 00 through 23 provide the memory address for the write memory function.



Bits 00 through 31 provide the data word that is to be stored at the address specified by the destination bus.

Note:

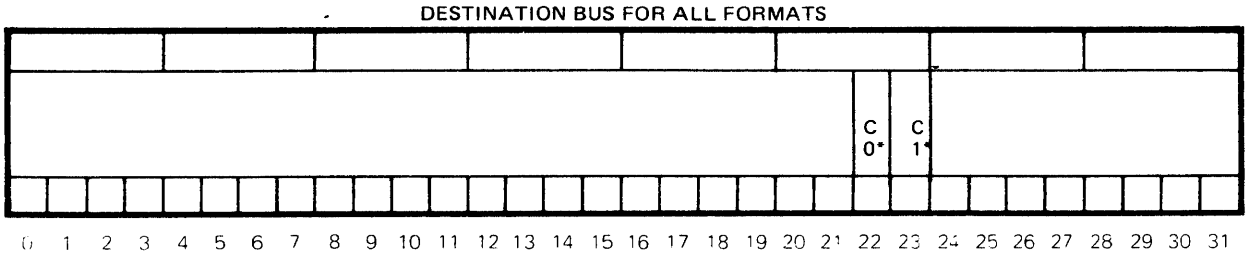
1. The MWT can be originated by either the IOP or the CPU. The destination of the transfer is the memory bus controller (MBC).
2. The MWT can be used to specify a word, halfword, or byte write function. In any of these cases, the data to be stored in memory are returned on the DBUS, right-justified, in a DRT transfer. The 'F' bit (the tag bus LDTF signal) and the destination bus 'C' bits specify one of the modes, as shown in Table 3-4.
3. Table 3-5 contains the SelBUS input/output transfer identification. Refer to Table 3-5 for the SelBUS tag signal configurations that identify a transfer as a memory read transfer (MRT).

Table 3-4
MWT/MRT Formats

F Bit	C Bits Destination Bus Bits		
(LDTF Signal)	22	23	Transfer Function
0 (high)	0	0	Fullword transfer
0 (high	0	1	Halfword transfer (left half)
0 (high)	1	1	Halfword transfer (right half)
1 (low)	0	1	Byte transfer to byte 0 (bits 00-07)
1 (low)	0	1	Byte transfer to byte 1 (bits 08-15)
1 (low)	1	0	Byte transfer to byte 2 (bits 16-23)
1 (low)	1	1	Byte transfer to byte 3 (bits 24-31)

3.3.1.5 MRT SelBUS Transfer

The format for an MRT SelBUS transfer is shown below:



*For 'C' bits, see note 4.

Table 3-5
SelBUS Input/Output Transfer Identification

Tag Bus Signals						
LTX	LMEM	LCNT0	LCNT1	LRD	LERROR	Input Transfer Operation
L	H	H	H	H	H	WDOT
L	H	H	L	H	H	ICT
L	H	H	L	L	H	RSTX
L	H	L	H	H	H	ARSTX
L	H	L	H	L	H	AICT
L	H	L	L	H	H	DRT (read data from memory)
H	H	H	H	H	L	Error transfer (previous DRT from memory) contained a parity error
						Output Transfer Operation
L	L	H	H	H	L	MWT
L	L	H	H	L	H	MRT
L	H	L	L	H	H	DRT (status information to the CPU)

Bits 00 through 23 provide the memory address of the location to be read.

DBUS FOR A CPU - ORIGINATED MRT

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
NOT USED																								CPU	NOT USED						ECK
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	

Bit 24, when equal to one, specifies that the CPU is the source of the MRT and the data read from the addressed location are to be returned to the CPU.

Bit 31, when equal to zero, specifies that an operand is to be read from memory. Bit 31, when equal to one, specifies that an instruction is to be read from memory.

DBUS FOR I/O CONTROLLER - ORIGINATED MRT

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
NOT USED																									PHYSICAL ADDRESS						
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							

Bits 25 through 31 provide the physical address of the I/O controller that originated the MRT. They also specify that the data read from the addressed memory location are to be sent to this I/O controller address.

Note:

1. The MRT can be originated by either an IOP or a CPU. The destination of the transfer is the MBC.

2. The MBC responds to the MRT with a DRT containing the contents of the memory location addressed by the MRT. The destination of the DRT is the module that originated the MRT.

3. Refer to Table 3-3 for the SelBUS tag signal configurations that identify the transfer as an MRT.

4. The MRT can be used to specify a word, halfword, or byte read function. In any of these cases, the data read from memory is returned on the DBUS, right-justified in a DRT transfer. The 'F' bit (the tag bus LDTF signal) and the destination bus 'C' bits specify one of the modes, as shown in Table 3-2.

3.4 Firmware Discussion

3.4.1 Interrupts

Major asynchronous events are processed on a microinterrupt level, assuring fast response and prioritizing processing according to predefined importance. Microinterrupts are defined for the following events:

System Reset (Pseudointerrupt)

0 Multipurpose bus (MPB) time-out
1 MPB data service
2 Central processing unit (CPU) interrupt control
3 Not defined
4 CPU data in
5 Not defined
6 Real-time clock request
7 Interval timer request
8 External interrupt
9 MPB data request
10 End or identify (EOI)
11 Service request (SRQ)
12 Not defined
13 Run/halt transition
14 Not defined
15 Not defined

3.4.1.1 System Reset

A system reset detected via the SelBUS interface causes the input/output processor (IOP) to initialize all registers and some random access memory (RAM). This initialization procedure is accomplished under firmware control. In addition to initializing the IOP, the firmware also issues appropriate interface clear (IFC) commands to all attached controllers, initializes the interval timer and real-time clock, and allows the SCP to perform its usual system reset procedures.

3.4.1.2 Multipurpose Bus Time-out

An MPB time-out occurs when a controller has the MPB for more than 537 microseconds. This request supersedes all other data and bus request signal processing. An EOI signal is forced and data transfer requests are terminated on the MPB.

3.4.1.3 MPB Data Service

Data transfers requiring immediate service to or from the multipurpose bus can be processed via this interrupt level. The MPB data service interrupt level is not supported by existing IOP firmware.

3.4.1.4 CPU Interrupt Control

Since a CPU interrupt request can be generated by many levels of IOP processing, this level exists to process requests in an orderly manner. Any lower IOP processing level may request this level to access RAM and hardware registers associated with interrupt processing.

3.4.1.5 CPU Data In

The CPU data in event is detected and captured by the SelBUS interface hardware. Firmware recognition of that event is accomplished by a microinterrupt. The contents of the captured data from the CPU always represent an IOP command. The types of IOP commands received from the CPU are listed below:

1. Advanced read status transfer (ARSTX)
2. Read status transfer (RSTX)
3. Write data or order transfer (WDOT)
4. Advance interrupt control transfer (AICT)
5. Interrupt control transfer (ICT)

3.4.1.6 Real-Time Clock Request

A real-time clock request can be generated by the IOP hardware if it is not blocked by clock override. This signal causes a microinterrupt.

3.4.1.7 Interval Timer Request

The interval timer is under the control of the IOP firmware. Decrementing the interval timer to zero causes a microinterrupt if enabled.

3.4.1.8 End or Identify

An end or identify signal has been detected on the MPB causing the EOI interrupt to occur. The current burst is also terminated.

3.4.1.9 MPB Data Request

In order to allow windows for other processing during burst mode operations, burst data transfers are processed using a microinterrupt.

3.4.1.10 Multipurpose Bus Service Request

A MPB service request (SRQ) can be generated by any controller needing to transfer data to/from the IOP or by any controller needing to transfer status to the IOP. The MPB SRQ causes a microinterrupt and supports data transfers for the burst mode. Burst mode transfers are driven by the burst SRQ interrupts.

The procedures required to identify the controller and device requiring service are as follows:

1. A parallel poll to determine the controller pair that generated the SRQ.
2. A serial poll to determine which of the paired controllers generated the SRQ. This poll is accomplished by reading the addressed controllers state register. In addition to SRQ pending, this state register also contains the device address of the active device on the controller.
3. A status ready indication in the state register implies that a status transfer is to be read from the controller. Otherwise, a data transfer is requested by the controller.

3.4.1.11 Run/Halt Transition

When a run/halt condition is detected, a microinterrupt occurs, thus allowing associated system control panel (SCP) firmware processing.

3.4.2 Processing

The IOP firmware is interrupt-driven. Initialization of the firmware is accomplished by a SYSTEM RESET interrupt (a pseudointerrupt that forces the program counter to zero). After IOP initialization, the firmware enters a background loop. The CPU, or any device controlled by the IOP and requiring processing, can have that processing performed by triggering the appropriate interrupt. All IOP firmware processing falls into one of the categories defined in the following discussion.

3.4.2.1 Initialization Firmware

Initialization firmware responds to the SYSTEM RESET interrupt and performs two major functions: initializing the RAM storage and issuing the appropriate commands and orders to all IOP-controlled devices.

After initialization is complete, the IOP state has no interrupts active, no IOP devices defined, and no current activity associated with any devices that the IOP controls.

3.4.2.2 Control Firmware

Control firmware is defined as firmware that recognizes an IOP-controlled device requires attention. The major functions performed by this section are:

1. Recognizing, capturing, and initial decoding of CPU input data.
2. Recognizing a real-time clock request and calling the CPU interrupt handler to stack or initiate the interrupt.

3. Recognizing an interval timer request, restarting the interval timer if commanded last with auto-restart, and calling the CPU interrupt handler to stack or initiate the interrupt.

4. Background processing, which includes support for the SCP and operator console firmware. The no interrupt command is active for this processing.

3.4.2.3 CPU Communications Firmware

The CPU communications firmware performs the initial processing of CPU data in, with the exception of SCP and operator console commands. Initial decode by the control firmware yields transfer type and subaddress. The requested action in the CPU transfer can be an IOP, controller, or interval timer command. If the request is either an IOP or an interval timer command, processing is completed in the CPU communications firmware. If the command is a controller command, processing is initiated by the CPU communications firmware. Controller commands are interpreted by this section of the firmware. If the device is able to accept the command, the appropriate commands are sent to the controller. Information pertaining to the activity of each device is recorded in the RAM at that time.

3.4.2.3.1 Destination Bus Contents

In all cases, except for the interval timer, the destination bus contains the associated interrupt level, the IOP address, and the IOP subaddress. For ARSTX/RSTX functions, the interval timer has a function code in place of the interrupt level on the destination bus. The IOP controls the SCP, real-time option module (RTOM) function, operator consoles, and 16 multiplexed controllers. The interrupt levels, the IOP addresses, and the subaddresses associated with each are shown in Table 3-6. The interrupt level is contained in bits 8 through 16, the IOP address in bits 16 through 23, and the subaddress in bits 24 through 31 of the destination bus contents.

In addition to interrupt level, physical address, and subaddress, bus tag signals are associated with the destination bus contents. These signals show the type of transaction associated with the current CPU transfer. Types of transfers from the CPU are WDOT, AICT, ICT, ARSTX, and RSTX. These signals must be tested before the contents of the data bus (DBUS) can be interpreted.

3.4.2.3.2 DBUS Contents

In order to properly interpret the BUS contents, the type of transfer and IOP address must be known. The IOP address is necessary to identify the class of device. The type of transfer is necessary to identify the function to be performed.

The IOP controls three classes of devices: extended I/O (includes the operator console), RTOM, and SCP.

The extended I/O DBUS contents and their interpretations are shown in Table 3-7.

Table 3-6
Destination Bus Contents

Device	Interrupt Level	IOP Address	Subchannel Address
SCP	None	0	0
Console in	Assigned	Any even address ≠ 0	FC
Console out	Assigned	Any even address ≠ 0	FD
RTM con in	Assigned	Any even address ≠ 0	FE
RTM con out	Assigned	Any even address ≠ 0	FF
IOP	Assigned	Any even address ≠ 0	00-FB
RTOM#1			
Software #10	Assigned	Any odd address ≠ 1	F
Software #9	Assigned	Any odd address ≠ 1	E
Software #8	Assigned	Any odd address ≠ 1	D
Software #7	Assigned	Any odd address ≠ 1	C
Software #6	Assigned	Any odd address ≠ 1	B
Software #5	Assigned	Any odd address ≠ 1	A
Software #4	Assigned	Any odd address ≠ 1	9
Software #3	Assigned	Any odd address ≠ 1	8
R-T clock	Assigned	Any odd address ≠ 1	7
Software #2	Assigned	Any odd address ≠ 1	6
Software #1	Assigned	Any odd address ≠ 1	5
Interval TMR	Assigned	Any odd address ≠ 1	4
External #4	Assigned	Any odd address ≠ 1	3
External #3	Assigned	Any odd address ≠ 1	2
External #2	Assigned	Any odd address ≠ 1	1
External #1	Assigned	Any odd address ≠ 1	0

Table 3-7
Extended I/O DBUS Contents

Transfer Type	Primary Decode	Secondary Decode	Function Requested
Bits	Bits	Bits	
ARSTX/ RSTX	0-7	8-19	
	0		Identify I/O protcol
	2	IOCD Addr	Start I/O (SIO)
	1		Unassigned
	3	IOCD Addr	Initial program load (IPL)
	4		Test I/O (TIO)
	5		Stop I/O (STPIO)
	6		Halt I/O (HIO)
	7		Reset controller (RSCTL)
	8	Chan Addr	Load RAM
	9		N/U
	A		Unassigned
	B		N/U
	C		Enable channel interrupt (ECI)
	D		Disable channel interrupt (DCI)
	E		Activate channel interrupt (ACI)
	F		Unassigned
AICT/ ICT	28-31	None	
	1		Acknowledge and deactivate
	2		Acknowledge
	8		Deactivate
WDOT	0-7	None	
	40		Reset channel (RSCHNL)

The SCP DBUS contents and their interpretations are shown in Table 3-8.

Table 3-8
SCP DBUS Contents

Transfer Type	Primary Decode	Secondary Decode	Function Requested
	Bits	Bits	
ARSTX/ RSTX	28-31	None	
	1		Send function
	2		Send data
	6		Send IPL data
	9		Send control switch request
WDOT			Expected data (no decode)

3.4.2.4 CPU Interrupt Firmware

The RTOM, IOP, and operator console device interrupts are coordinated by this section of the IOP firmware. The IOP controls 17 levels of CPU interrupts, 16 associated with the RTOM and one associated with the IOP. The operator console TTY, although not a multipurpose DBUS device, uses the IOP interrupt. There is no CPU interrupt associated with the SCP. Before a specific level can be activated or requested, a comparison must be made to ensure that the level is currently enabled and is the highest within the IOP requesting interrupt support. If the level is not enabled or not the highest, the activity must be queued for later processing. If the level is enabled and the highest in the IOP, a check must be made to ensure that no other level is active before polling starts. If another level is currently active within the IOP, that level must be queued and its polling stopped. At that time, polling can commence for the highest requesting level.

Deactivation of any interrupt level requires checking the pending and enabled queue for a waiting activity. If another level is to start polling, that polling must commence with the deactivation of the current interrupt level.

3.4.2.5 Multipurpose Bus Data Transfer Firmware

This section of the firmware performs burst data transfers, along with command and data chain processing. To accomplish this processing in an efficient manner, the firmware uses three microinterrupts. The first of these microinterrupts is the MPB SRQ. Initial burst and status available sequences from any controller are recognized via the SRQ interrupt. The second of these microinterrupts is used for subsequent data transfer requests. This interrupt is called the burst service request (level 11).

Burst mode transactions, once identified as such, utilize dedicated SelBUS interface registers to access memory. These registers use a unique physical address for memory operations. The RTOM address is used since the RTOM does not reference memory. In addition to the dedicated memory address register, a special purpose transfer counter is used. This counter can be decremented by the MPB interface, loaded, or read by an order from the IOP microengine. The memory address counter (MAC) may also be loaded or incremented by an order from the IOP microengine. This additional hardware is required to maintain a high transfer rate (1.5 M byte/sec) on the MPB and to allow the processing of other IOP functions. The third microinterrupt is used for ending burst transactions. Burst transactions end when an EOI signal accompanies a data transfer on the MPB. The EOI signal causes the EOI interrupt to be enabled.

CHAPTER 4

OPERATION AND PROGRAMMING

4.1 CPU and IOP Communications

SelBUS communication protocol between the CPU and IOP can be initiated in the following ways:

1. System initialization procedure
2. Extended I/O macroinstruction
3. IOP interrupt request

4.1.1 System Initialization Procedure

System initialization starts when the computer operator depresses the initial program load (IPL) pushbutton on the system control panel (SCP). The IPL sequence begins loading the software program into main memory. The IPU Console IOP does not support IPL.

4.1.2 Extended I/O Macroinstructions

Extended I/O macroinstructions request specific actions from the IOP. These actions were defined earlier.

4.1.3 IOP Interrupt Requests

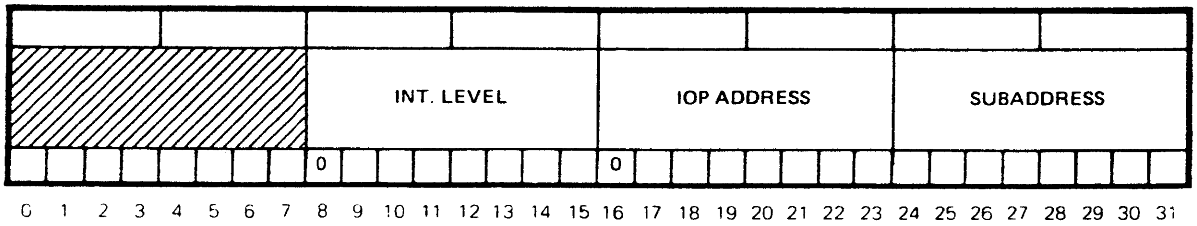
IOP interrupt requests initiate an acknowledgement of the requests before transferring control to the appropriate software interrupt handler. The type of acknowledgement initiated by the CPU is determined by the mode used for the IOP and the way the IOP is implemented.

4.1.4 CPU to IOP Transfer Protocol

The CPU to IOP transfer protocol is the same as the current SelBUS protocol. There are five types of transfers:

1. Advance request for status transfer (ARSTX)
2. Request for status transfer (RSTX)
3. Write data or order transfer (WDOT)
4. Advance interrupt control transfer (AICT)
5. Interrupt control transfer (ICT)

The advance transfers indicate to the IOP that the CPU is requesting information or services. Those services are identified by a code in a data word. The information on the destination bus for all CPU to IOP communications is in the following format:



For all ARSTX and AICT communications, the IOP obtains the required information or prepares to initiate the desired service, then responds with a ready indication. The CPU, upon receipt of the ready indication, issues the RSTX or ICT to request the information and/or service.

4.2 IOP or Subchannel Busy Indications

The IOP, in most cases, operates asynchronously with the CPU. However, there are times when the IOP may wish to delay or postpone execution of a transfer request. Typically, there are three types of busy indications that the IOP reports:

1. Retry
2. Busy
3. Subchannel busy status

4.2.1 Retry

The retry condition is a hardware signal implemented in the IOP interface logic. This signal is under the direct control of the IOP's microprogram. The retry signal indicates to the CPU that the IOP cannot accept any CPU transfer requests for a period of not more than 20 microseconds. The retry signal applies to all CPU bus transfers: advance read status transfer (ARSTX), read status transfer (RSTX), interrupt control transfer (ICT), and write data or order transfer (WDOT). When the retry command is true, the CPU attempts to retransmit the transfer request every microsecond until the transfer request is finally accepted by the IOP. The CPU does not attempt to do any unrelated processing during this interval; therefore, descretion should be exercised when this signal is used. If the IOP does not accept the transfer request within a 20-microsecond period, the CPU terminates the macroinstruction with a condition code. This action indicates that the IOP is busy.

4.2.2 Busy

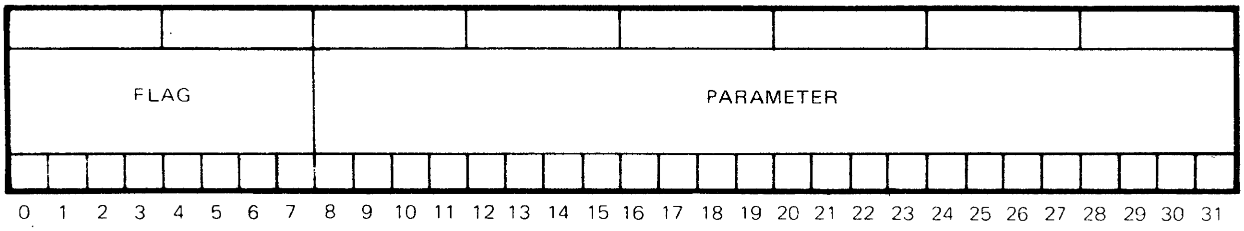
The busy status is a hardware signal implemented in the IOP interface logic, under the direct control of the IOP's microprogram. This signal indicates to the CPU that the IOP cannot accept any CPU transfer requests for at least 20 microseconds. This signal is significant to the ARSTX and AICT CPU bus transfers. When busy status is true, the CPU immediately terminates the macroinstruction with a condition code. This action

4.2.3 Subchannel Busy Status

The subchannel busy status is reported to the CPU when the IOP's data return transfer (DRT) data response to an RSTX contains a hexadecimal one in bits 0 through 3. This response indicates that the addressed controller/device is executing a previously requested I/O function and cannot accept another function.

4.3 ARSTX/RSTX Requests

There are 13 types of advance read status transfer (ARSTX) and read status transfer (RSTX) requests. Each request is identified by bits in the data word that are transferred to the IOP on the data bus (DBUS). Condition codes are returned to the CPU as data return transfer (DRT) data for all RSTX function requests. The function request word format is shown below:



Flag (bits 0 through 3) - Reserved for future use

Flag (bits 4 through 7)

0 - Identify I/O protocol
1 - Unassigned
2 - Start I/O (SIO)
3 - Initial program load (IPL)
4 - Test I/O (TIO)
5 - Stop I/O (STPIO)
6 - Halt I/O (HIO)
7 - Reset channel (RSCTL)
8 - Load random access memory (RAM)
9 - Not used
A - Unassigned
B - Not used
C - Enable channel interrupt (ECI)
D - Disable channel interrupt (DACI)
E - Activate channel interrupt (ACI)
F - Unassigned

Parameter (bits 8 through 31)

Parameter requirements are transfer-type-dependent, based on the specification of flag bits 0 through 7.

4.3.1 RSTX Data Return Transfer

The IOP returns a data word to the CPU after decoding the RSTX transfer. The data word contains information relating to the acceptance or rejection of the RSTX. The format for the data word, excluding the identify I/O protocol, is shown below:

<table>
<tr><td colspan="4">CC</td><td colspan="4"></td><td colspan="24">STATUS ADDRESS</td></tr>
<tr><td></td><td></td><td></td><td></td><td>0</td><td>0</td><td>0</td><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
</table>

CC bits 0 through 3 contain the information that is passed to the software in the form of condition codes. The relationship of the data bits to the CCs are as follows:

Bit 0 = CC1
Bit 1 = CC2
Bit 2 = CC3
Bit 3 = CC4

The CPU does not interrogate the CCs returned to the IOP. The CC information is sent directly to the software for interrogation. The current assigned values for the CCs are:

0 - Accepted by the IOP, will echo status
1 - IOP busy*
2 - IOP undefined or inoperable**
3 - Subchannel busy
4 - Status stored
5 - Unsupported transaction
6 - Reserved
7 - Reserved
8 - Accepted by the IOP, no echo status
9 - Reserved
A - Reserved
B - Reserved
C - Reserved
D - Reserved
E - Reserved
F - Reserved

* Indicates that either the CPU or IOP can set this state.
** Indicates that this assignment is reserved for the CPU.

The status address (bits 8 through 31) indicates the address of the status words that the IOP wishes to transmit. The status address is always transferred to the software when it is sent by the IOP.

The 'accepted by the IOP, will echo status' condition is not used by the IOP.

The IOP busy condition is not used by the IOP.

The IOP undefined or inoperable condition indicates that the IOP does not respond to an ARSTX request. This condition can only be generated by the CPU.

The subchannel busy condition indicates that the IOP is capable of limited interrogation of the RSTX request and that the addressed subchannel is already busy from a previous initiation.

The status stored condition indicates that status conditions are present and the IOP is prevented from continuing its processing. The status address may not be related to the addressed subchannel/controller/device. This condition only applies to the SIO, TIO, and HIO instructions.

The unsupported transaction condition indicates that the IOP does not have the feature required or implied by the bus transaction.

The 'accepted by the IOP, no echo status' condition indicates that the IOP is capable of limited interrogation or that the addressed subchannel is idle. No echo status is associated with the accepted transaction.

4.3.2 Identify I/O Protocol

The identify I/O protocol bus transfer format is shown below:

FLAG								PARAMETER																							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

The identify I/O protocol bus transfer indicates to the IOP that the CPU requires an I/O protocol identity. All extended I/O IOPs and integrated IOP/controllers must respond with the following format:

UNASSIGNED																PROTOCOL				TYPE				MODEL							
																1	1	1	1												
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Bits 0 through 15 are unassigned and, therefore, are ignored.

Bits 16 through 19 identify the protocol as being extended I/O (hexadecimal F).

Bit 20 identifies the IOP memory type as control read only memory (CROM) and is equal to zero.

Bits 21 through 23 identify the IOP type as follows:

000-001	=	Not used
010	=	Blocked IOP
011	=	Not used
100-111	=	Unassigned

Bits 24 through 27 identify the model number and are currently all zeros.

Bits 28 through 31 are unassigned and are, therefore, ignored.

4.3.3 SIO

The SIO bus transfer sequence attempts to initiate an I/O operation. The information sent to the IOP on the DBUS is in the following format:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
FLAG								PARAMETER REAL IOCD ADDRESS																							
0	0	0	0	0	0	1	0																								

The acceptance of the SIO ARSTX/RSTX does not preclude the IOP from terminating the request before initiation. When the I/O fails to be initiated, the status sent never contains channel end or device end.

4.3.4 IPL

The IPL bus transfer sequence attempts to initiate an IPL I/O operation. The information sent to the IOP on the DBUS is in the following format:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
FLAG								PARAMETER REAL IOCD ADDRESS																							
0	0	0	0	0	0	1	1																								

4.3.5 TIO

The TIO bus transfer sequence attempts to dequeue any pending status. The information sent to the IOP on the DBUS is in the following format:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
FLAG								PARAMETER																							
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.6 STPIO

The STPIO bus transfer sequence requests termination of an addressed device I/O operation. The command and data chaining flags for the currently executing IOCD are suppressed in the subchannel. The current I/O operation continues until a normal termination occurs. The information sent to the IOP on the DBUS is in the following format:

FLAG								PARAMETER																							
0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

4.3.7 HIO

The HIO bus transfer sequence requests an immediate termination of an I/O operation to the addressed subchannel and controller. A termination request is transmitted to the controller and the termination is orderly and immediate. A device end condition sent from the controller indicates to the CPU that the termination has occurred and the subchannel and controller are available. The controller determines whether the device end condition is immediate or deferred. The information sent to the IOP is in the following format:

FLAG								PARAMETER																							
0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

4.3.8 RSCTL

The RSCTL bus transfer sequence is an immediate reset request for the controller. No status conditions are generated. The information sent to the IOP is in the following format:

FLAG								PARAMETER																							
0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

4.3.9 Load Random Access Memory (RAM)

The load RAM bus transfer sequence requests the assignment of an interrupt level to the addressed IOP. Normally, this transaction occurs only once if the IOP is not the IPL IOP.

Future requirements may require a dynamic reassignment of this bus interrupt level. Therefore, the IOP must always be aware of this transaction. The information sent to the IOP is in the following format:

FLAG																								IOP ADDRESS							
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

4.3.10 ECI

The ECI bus transfer sequence requests that the IOP allow the normal I/O interrupts to occur between the IOP and CPU. The information sent to the IOP is in the following format:

FLAG								PARAMETER																							
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

4.3.11 DCI

The DCI bus transfer sequence requests that the IOP stop the normal I/O interrupts that occur between the IOP and CPU. The information sent to the IOP is in the following format:

FLAG								PARAMETER																							
0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

4.3.12 ACI

The ACI bus transfer sequence requests that the IOP block all lower priorities from interrupting the CPU. The information sent to the IOP is in the following format:

FLAG								PARAMETER																							
0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

4.4 Advance Read Status Transfer (ARSTX)/Read Status Transfer (RSTX) Timing

The IOP's response timing must be considered by the IOP designer. The protocol and response timing for each bus transfer request is shown in Figure 4-1.

4.5 Advance Interrupt Control Transfer (AICT)/Interrupt Control Transfer (ICT) Requests

There are three types of AICT and ICT requests. Each request is identified by bits in the data word that are transferred to the IOP on the DBUS. The IOP's retry and busy conditions apply to both the AICT and ICT requests. A 38.4 microsecond time-out is provided for these functions. No provision is made to return an IOP busy indication for these transfer requests, since they deal entirely with the software context switch.

All AICT/ICT communications cannot be deferred by the IOP. The AICT/ICT request word is in the following format:

UNASSIGNED																												CODE			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0														
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Bits 0 through 17 are unassigned and currently contain zeros.

Bits 28 through 31 (code) identify the function request required by the CPU as follows:

Bit	28	=	Deactivate interrupt
Bit	29	=	Not assigned
Bit	30	=	Acknowledge and activate interrupt
Bit	31	=	Acknowledge and deactivate interrupt

Note that bits 28 through 31 are bit-significant and are mutually exclusive.

4.5.1 Acknowledge and Deactivate Interrupt

The acknowledge and deactive interrupt bus transfer sequence indicates to the IOP that its interrupt request has been sensed by the CPU and must be removed by the IOP. The information sent to the IOP is in the following format:

																												CODE			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

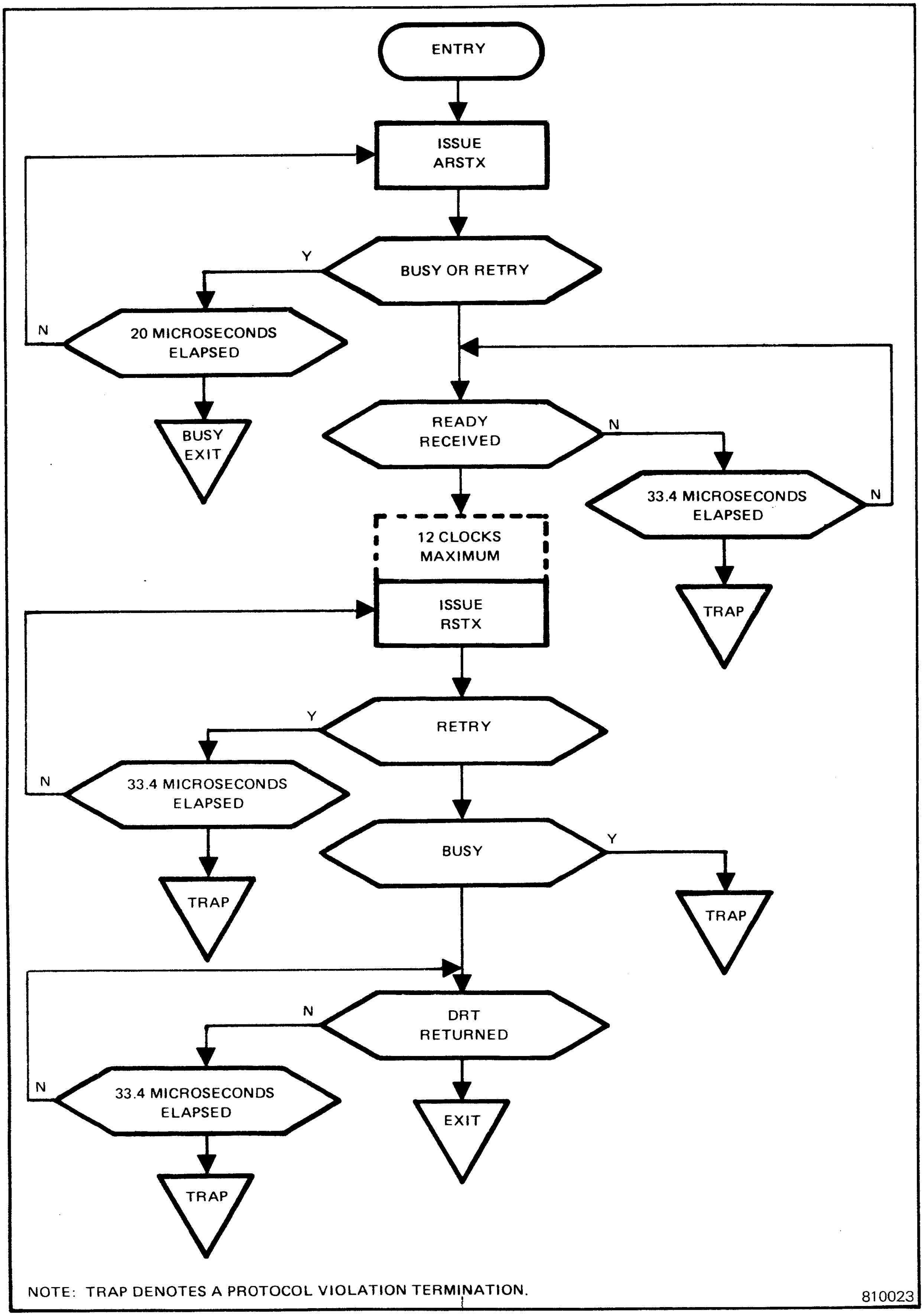


Figure 4-1. ARSTX/RSTX Bus Transfer Request Protocol and Response Timing

The IOP returns a data word to identify the location in main memory of the stored status words. The data word returned from the IOP is in the following format:

▨								STATUS ADDRESS																							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

The acknowledge and deactivate condition also dequeues the status area previously sent to the CPU by the IOP, using an interrupt or an I/O instruction to indicate that status is stored. The software programmer should exercise caution when allowing interrupts to occur before the completion of the interrogation of status in the memory location maintained by the IOP.

4.5.2 Acknowledge and Activate Interrupt

The acknowledge and activate bus transfer sequence indicates to the IOP that its interrupt request has been sensed by the CPU and must be advanced to an active state by the IOP. A subsequent deactivate interrupt bus transfer sequence is initiated by the software to remove the active state. The information sent to the IOP is in the following format:

																												CODE			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

The IOP returns a data word identifying the location in main memory of the stored status words. The data word returned from the IOP is in the following format:

▨								STATUS ADDRESS																							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

The acknowledge and activate interrupt condition also dequeues the status area previously sent to the CPU by the IOP, using an interrupt or an I/O instruction to indicate that status is stored.

The software programmer should exercise caution to allow interrupts before the completion of the interrogation of status in the memory locations maintained by the IOP.

The post program controlled interrupt (PPCI) for this type of IOP also requires a deactivate interrupt command to reset the activate state.

4.5.3 Deactivate Interrupt

The deactivate interrupt bus transfer indicates to the IOP that the software no longer requires the IOP to enqueue its interrupt processing from other interrupt levels. The IOP resets its own active state. The information sent to the IOP is in the following format:

																												CODE			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

The IOP returns a data word indicating that the active state has been reset. The data are not interrogated by the CPU.

This bus transaction is initiated by the software by the execution of a deactivate interrupt or a branch and reset interrupt (BRI) instruction, and only if the IOP is defined as a blocking type IOP.

4.6 Advance Interrupt Control Transfer (AICT)/Interrupt Control Transfer (ICT) Timing

The protocol and response timing for each bus transfer request is shown in Figure 4-2.

4.7 Write Data or Order Transfer (WDOT) Bus Transfer Sequence

There is only one type of WDOT bus transfer sequence used to reset an IOP. The MPB interface cannot block this transfer. The retry or busy condition from the IOP is not interrograted by the CPU. The information sent to the IOP is in the following format:

FUNCTION																															
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

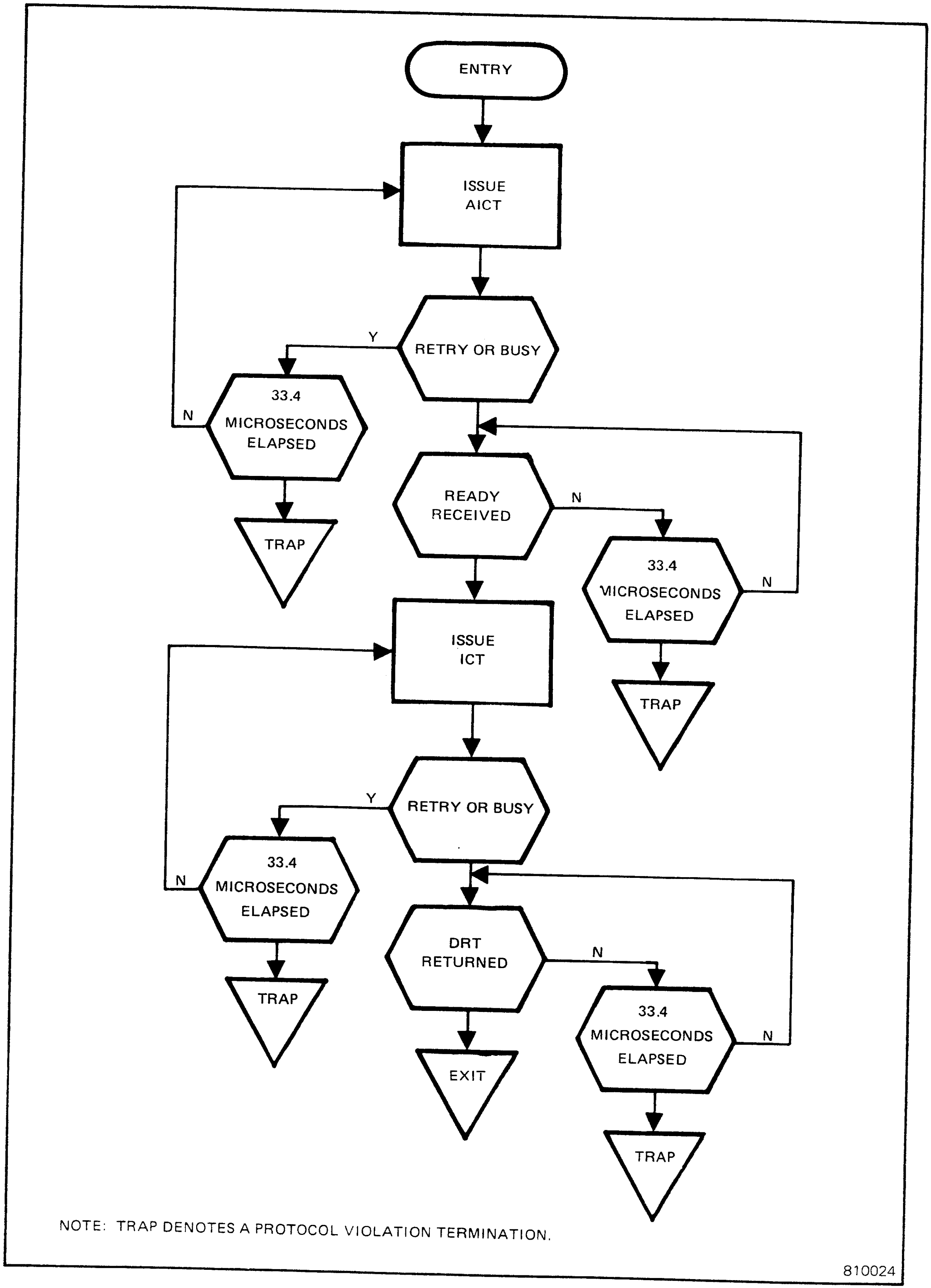
Bits 0 through 7 specify the function of the WDOT. The only code assigned to the WDOT is the reset channel function.

4.7.1 Reset Channel

The reset channel bus transfer sequence requests that the IOP reset all of the following:

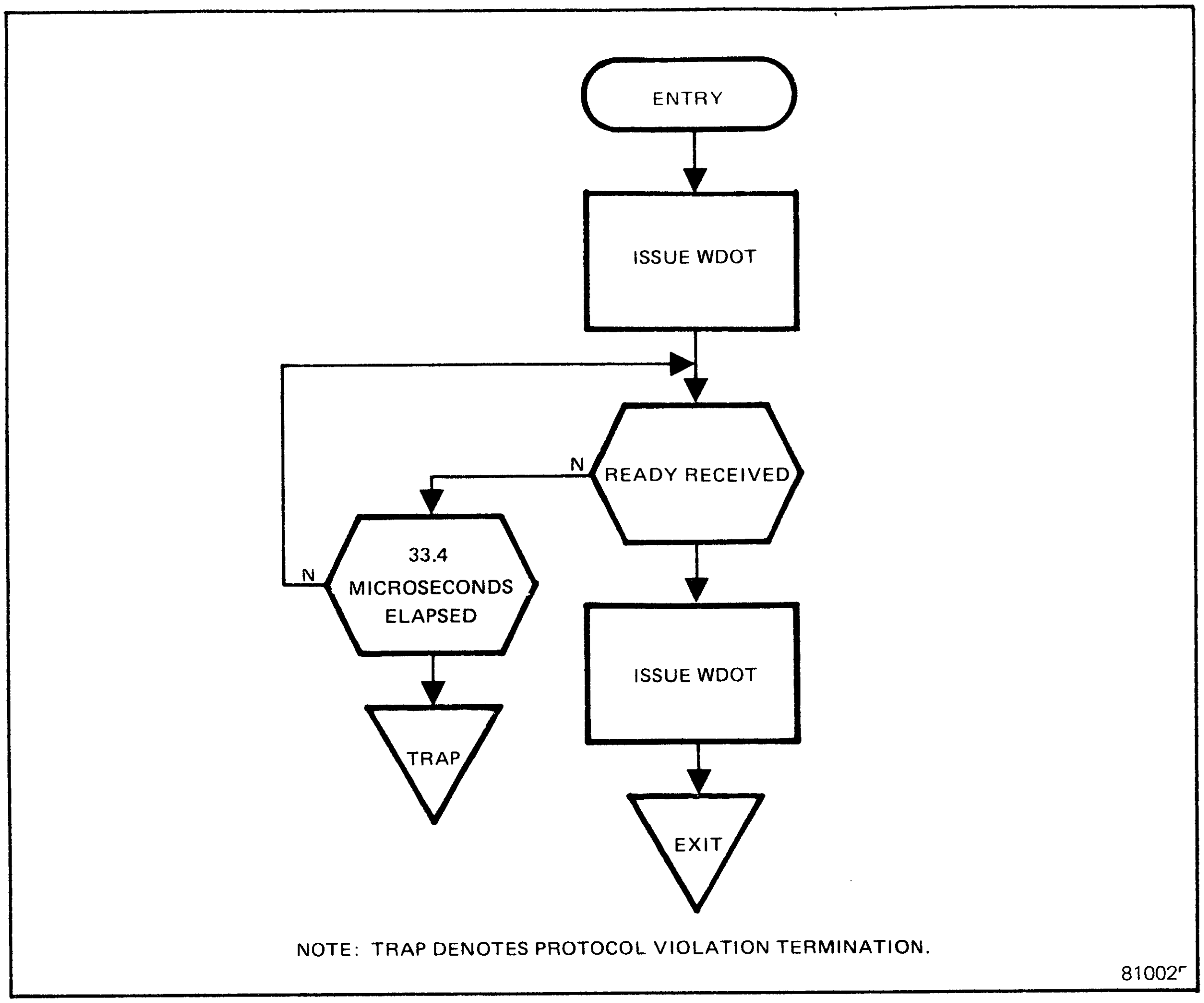
1. Subchannels
2. IOPs main memory allocation
3. Interrupt level
4. Interrupt enable
5. All internal registers and indicators

The IOP then becomes idle in a noninitialized state. A load random access memory (RAM) bus transfer sequence and a channel memory allocation (CMA) command are issued in order to initialize the IOP before software utilization. No data are transferred to the CPU by the IOP during the bus transfer sequence.



NOTE: TRAP DENOTES A PROTOCOL VIOLATION TERMINATION.

Figure 4-2. AICT/ICT Bus Transfer Request Protocol and Response Timing



NOTE: TRAP DENOTES PROTOCOL VIOLATION TERMINATION.

Figure 4-3. WDOT Protocol and Response Timing

4.7.2 WDOT Transfer Timing

The timing of the IOP's response must be considered by the IOP designer. The protocol and response timing for each transfer request is shown in Figure 4-3.

4.8 CPU and IOP Real-Time Option Module (RTOM) Communications

The SelBUS communication protocol between the CPU and the IOP RTOM/interval timer is contained in this section of the manual.

4.8.1 Real-Time Clock Select Jumpers

The RTOM function has jumpers that allow maintenance or operator personnel the option of selecting the real-time clock with its 60-Hz or 120-Hz rate or an external device interrupt. Tables 4-1 and 4-1A show the RTOM and real-time clock jumper assignments.

Table 4-1
IOP Jumper Assignments for Model 8000

To Assign	Remove Jumper	Add Jumper	Logic Sheet Location	
Real-time clock to an external clock	E16-6 to E16-11 E16-7 to E16-10	E16-5 to E16-12	Sheet 13	X6
Real-time clock to 120 Hz (assuming 120 Hz at P1B-108)	E16-5 to E16-12 E16-7 to E16-10	E16-6 to E16-11	Sheet 13	
Real-time clock to 60 Hz (assuming 120 Hz at P1B-108)	E16-5 to E16-12 E16-6 to E16-11	E16-7 to E16-10	Sheet 13	
4.8 ms*	-------	C14-1 to C14-16	Sheet 11	X4
9.6 ms*	-------	C14-8 to C14-9	Sheet 11	
19.2 ms*	-------	C14-7 to C14-10	Sheet 11	
38.4 ms*	-------	C14-6 to C14-11	Sheet 11	
76.8 ms*	-------	C14-5 to C14-12	Sheet 11	
600 ns*	-------	C14-4 to C14-13	Sheet 11	
1.2 ms*	-------	C14-3 to C14-14	Sheet 11	
2.4 ms*	-------	C14-2 to C14-15	Sheet 11	

*To select the interval timer count rate, install only one jumper.

4.8.2 Interval Timer Selection Jumpers

The IOP has jumpers that allow operation or maintenance personnel the option of selecting the interval timer count zero interrupt levels and counter frequency rate. The selection of the counter frequency rate is under software control. IOP jumper assignments for the interval timer select jumpers are also shown in Tables 4-1 and 4-1A.

4.8.3 Microdiagnostic Test Configuration Jumpers

The microdiagnostic test configuration jumpers allow the selection of either the operational or test modes of operation. These jumper assignments are shown in Tables 4-2 and 4-2A.

Table 4-1A
IOP Jumper Assignments for Model 8001 and 8002

To Assign	Remove Jumper	Add Jumper	Logic Sheet Location	
Real-time clock to an external clock	F7-6 to F7-11 F7-4 to F7-13	F7-5 to E7-12	Sheet 13	X4
Real-time clock to 120 Hz (assuming 120 Hz at P1B-108)	F7-5 to F7-12 F7-4 to F7-13	F7-6 to F7-11	Sheet 13	
Real-time clock to 60 Hz (assuming 120 Hz at P1B-108)	F7-5 to F7-12 F7-6 to F7-11	F7-4 to F7-13	Sheet 13	
4.8 ms*	-------	K23-1 to K23-16	Sheet 11	X6
9.6 ms*	-------	K23-8 to K23-9	Sheet 11	
19.2 ms*	-------	K23-7 to K23-10	Sheet 11	
38.4 ms*	-------	K23-6 to K23-11	Sheet 11	
76.8 ms*	-------	K23-5 to K23-12	Sheet 11	
600 ns*	-------	K23-4 to K23-13	Sheet 11	
1.2 ms*	-------	K23-3 to K23-14	Sheet 11	
2.4 ms*	-------	K23-2 to K23-15	Sheet 11	

*To select the interval timer count rate, install only one jumper.

Table 4-2
Microdiagnostic Test Configuration Jumper Assignments for Model 8000

Mode of Operation	Remove Jumper	Add Jumper	
Operational	E16-4 to E16-13 E16-3 to E16-14	E16-2 to E16-15	X6
Test	E16-2 to E16-15	E16-4 to E16-13 E16-3 to E16-14	

Table 4-2A
Microdiagnostic Test Configuration Jumper Assignments for Model 8001 and 8002

Mode of Operation	Remove Jumper	Add Jumper
Operational	F7-8 to F7-9 F7-2 to F7-15	F7-1 to F7-16
Test	F7-1 to F7-16	X4 F7-8 to F7-9 F7-2 to F7-15

4.8.4 Real-Time Option Module (RTOM) Programming

The Gould S.E.L. 32/27 Single Slot Central Processing Unit Reference Manual, publication number 301-000400, provides a complete explanation of each instruction used by the CPU; therefore, only those instructions that apply to the RTOM are discussed in the following paragraphs.

4.8.5 Interrupt System

Interrupt requests may be generated from the following sources:

1. An external event, real time clock, interval timer
2. CPU request for RTOM interrupt
3. The completion of an I/O operation within the IOP

4.8.6 Program Control of Interrupts

The following instructions are provided for program interrupt control:

-RTOM-

1. Enable interrupt (EI)
2. Disable interrupt (DI)
3. Request interrupt (RI)
4. Activate interrupt (AI)
5. Deactivate interrupt (DAI)

-Extended I/O-

1. Enable channel interrupt (ECI)
2. Disable channel interrupt (DCI)
3. Activate channel interrupt (ACI)
4. Deactivate channel interrupt (DACI)

The RTOM instructions contain a level number field that permits any level to be operated on by the execution of these instructions. The extended I/O instructions contain the IOP number.

4.9 SelBUS System Control Panel Transfers

The SelBUS can handle three types of transfers between the CPU and the SCP. These transfers are listed below:

1. Read status transfer (ARSTX/RSTX)
2. Data return transfer (DRT)
3. Write data or order transfer (WDOT)

The advance transfers indicate to the SCP that the CPU is requesting information or services. Those services are identified by a code in the data word. Figure 4-4 shows the format for the SCP read status transfers. Note that the SelBUS ARSTX/RSTX transfers are from the CPU and are sent to the addressed I/O controller.

A data return transfer is sent to the CPU after the ARSTX/RSTX transfer is decoded. The DRT contains information relating to the acceptance or rejection of the RSTX. Figure 4-5 shows the format of the SCP DRT.

The CPU WDOT transfer is sent to the SCP in response to an ARSTX, RSTX, or DRT sequence for a read CPU source function. Figure 4-6 shows the format of the SCP WDOT.

The CPU interrupt control transfer (AICT/ICT) is sent to the I/O controller or RTOM during interrupt control instructions, branch and reset interrupt instructions, and interrupt sequences. The I/O controllers or RTOM responds to the AICT/ICT transfer with a DRT. Note that the contents of the DRT are not used by the AICT/ICT transfer instruction. Figure 4-7 shows the format for the interrupt control transfers.

The advance transfers indicate, to the IOP, that the CPU will be requesting information or services identified by the code in the data word. The information on the destination bus, for all CPU-to-IOP communications, is shown in Figure 4-8.

There are 11 types of Class F ARSTX/RSTX requests associated with the IOP. Each function is identified by bits in the data word transferred to the IOP on the data bus. The ARSTX and RSTX data bus format for Class F transfers is shown in Figure 4-9.

The IOP returns a data word to the CPU upon decoding of the RSTX transfer. The data word contains information relating to the acceptance or rejection of the RSTX. The format for the data word (excluding the Identify I/O Protocol) is shown in Figure 4-10.

The Identify I/O Protocol bus transfer indicates, to the IOP, that the CPU required I/O Protocol identity. All Class F I/O IOPs and integrated IOP/controllers must respond as shown in Figure 4-11.

The AICT/ICT SelBUS transfers are used by the CPU for three Class F I/O functions: Acknowledge and Activate Interrupt, Acknowledge and Deactivate Interrupt, and Deactivate Interrupt. The data bus format for the AICT/ICT transfer pair is shown in Figure 4-12.

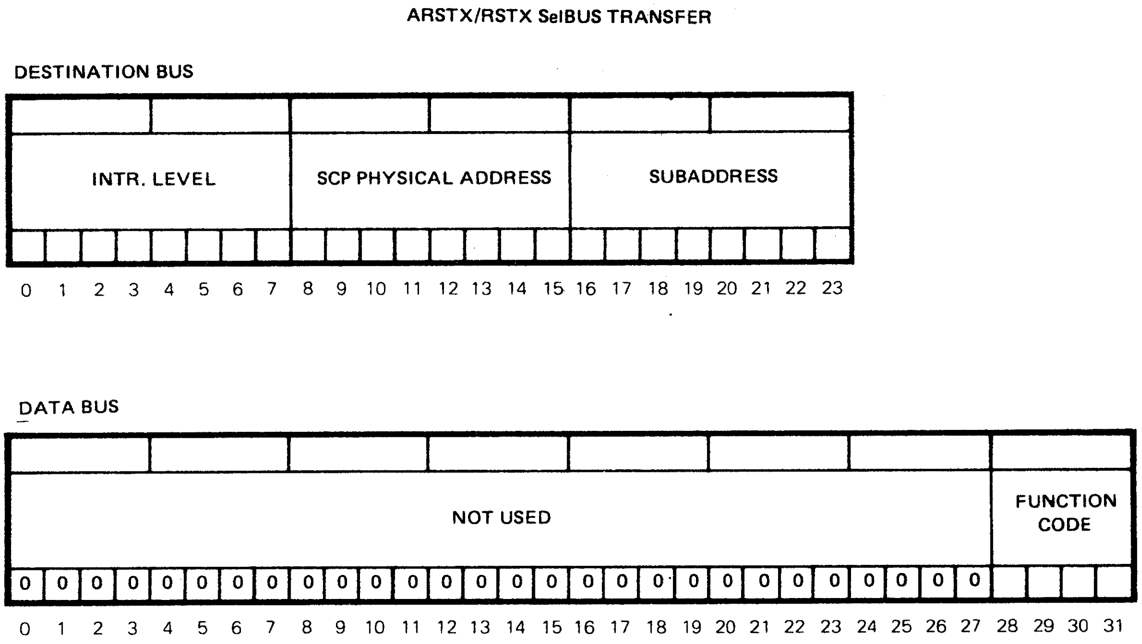
The Acknowledge and Activate Interrupt function of the AICT/ICT transfer pair indicates, to the IOP, that its interrupt request has been sensed by the CPU and that the IOP must advance its interrupt to an active state. A subsequent Deactivate Interrupt bus transfer sequence is then initiated by software to remove the active state. The IOP returns a data word to the CPU that identifies the main memory location where the IOP's status words are stored. This action also dequeues any status areas previously specified by the IOP. The format for the data word returned by the IOP is shown is Figure 4-13.

The Acknowledge and Deactivate Interrupt function of the AICT/ICT transfer pair indicates, to the IOP, that its interrupt request has been sensed by the CPU and that the IOP must now remove its interrupt request. The IOP returns a data word that identifies the main memory location where the IOP's status words are stored. This action also dequeues any status areas previously specified by the IOP. The format for the data word returned by the IOP is also shown in Figure 4-13.

The Deactivate Interrupt function of the AICT/ICT transfer pair indicates, to the IOP, that the software no longer requires the IOP to maintain its interrupt in an active state. The IOP resets the active state and returns a data word. The data word is not examined by the CPU or saved in main memory. The format for this data word is also shown in Figure 4-13.

The WDOT/WDOT SelBUS transfer sequence has only one function for Class F I/O operations: IOP reset. The IOP must accept this transfer, since the CPU does not recognize a busy or retry response signal.

The reset IOP WDOT/WDOT requests the IOP to return to an idle, noninitialized condition by resetting its interrupt level, interrupt enable line, internal registers, subchannels and main memory allocation. No data is returned to the CPU in response to this transfer sequence. The format of the data word sent to the IOP for a Class F WDOT/WDOT sequence is shown in Figure 4-14.



INTERRUPT LEVEL

INTERRUPT LEVEL IS NOT USED WITH THE SCP AND BITS 0-7 MUST EQUAL 0

PHYSICAL ADDRESS

BITS 8-15 MUST EQUAL 0 FOR THE SCP

SUBADDRESS

BITS 16-23 ARE NOT USED WITH THE SCP AND MUST EQUAL 0

FUNCTION CODE					
BITS					DEFINITION
27	28	29	30	31	
0	0	0	0	1	SEND FUNCTION
0	0	0	1	0	SEND DATA
0	0	1	1	0	SEND IPL DATA (IPL DEVICE PHYSICAL AND SUBADDRESSES)
0	1	0	0	1	SEND FUNCTION FOR SCP REQUEST OF CONTROL SWITCHES MEMORY ADDRESS.
1	0	0	0	1	ADDRESS STOP ACKNOWLEDGE (SET PANEL STOP LIGHT)

810026

Figure 4-4. System Control Panel Read Status Transfer Format

DRT SelBUS TRANSFER

DESTINATION BUS (FOR ALL DRT FORMATS)

NOT USED								1	NOT USED														
0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

CPU ADDRESS BIT

FORMAT A

DATA BUS FOR DRT IN RESPONSE TO A ARSTX/RSTX SEND FUNCTION

NOT USED												FUNCTION CODE				NOT USED											* BASE	SOURCE/ DEST.			
0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0					
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

* BIT 27 = 1 IS BASE REGISTER READ/WRITE

FUNCTION CODE

BITS				DEFINITION
12	13	14	15	
0	0	0	0	READ SOURCE (SEE NOTE 1)
0	0	0	1	WRITE DESTINATION (SEE NOTES 2 AND 3)
0	0	1	0	INSTRUCTION STEP
0	0	1	1	RESERVED FOR FUTURE
0	1	0	0	ENHANCEMENT
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

SOURCE/DESTINATION CODE

BITS				DEFINITION
28	29	30	31	
0	0	0	0	REGISTER 0 (GPR) /B0
0	0	0	1	REGISTER 1 (GPR) /B1
0	0	1	0	REGISTER 2 (GPR) /B2
0	0	1	1	REGISTER 3 (GPR)
0	1	0	0	REGISTER 4 (GPR)
0	1	0	1	REGISTER 5 (GPR)
0	1	1	0	REGISTER 6 (GPR)
0	1	1	1	REGISTER 7 (GPR)
1	0	0	0	PROGRAM STATUS WORD (PSW) (PSD1)
1	0	1	1	PROGRAM STATUS WORD 2 (PSD2)
1	1	0	0	CONVERT LOGICAL TO PHYSICAL ADDRESS (USED ONLY WITH A WRITE DESTINATION CODE INSTRUCTION)

NOTES: 1. FOR READ SOURCE FUNCTION CODES, THE CPU RESPONDS WITH A WDOT TRANSFER CONTAINING THE REQUESTED DATA, REFER TO FIGURE 4-12 FOR WDOT FORMAT.

2. DESTINATION CODES THAT REFER TO THE EFFECTIVE ADDRESS OR REQUEST CONTROL SWITCHES ADDRESS CANNOT BE USED WITH A WRITE FUNCTION CODE.

3. FOLLOWING THE DRT-WRITE DESTINATION TRANSFER, THE CPU RESPONDS WITH AN ARSTX/RSTX - SEND DATA BUS TRANSFER TO OBTAIN THE DATA TO BE STORED BY THE WRITE DESTINATION FUNCTION.

810027-1

Figure 4-5. System Control Panel Data Return Transfer Format (Sheet 1 of 2)

FORMAT B

DRT SelBUS TRANSFER

DATA BUS FOR DRT IN RESPONSE TO AN ARSTX/RSTX SEND IPL DATA

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
NOT USED																PHYSICAL ADDRESS												SUBADDRESS			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									0	0	0	0				

PHYSICAL ADDRESS

BITS 16-23 PROVIDE THE ADDRESS OF THE IPL DEVICE I/O CONTROLLER.

SUBADDRESS

BITS 28-31 PROVIDE THE SUBADDRESS OF THE IPL DEVICE.

FORMAT C

DATA BUS FOR DRT IN RESPONSE TO AN ARSTX/RSTX SEND DATA

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
DATA WORD FROM 'A' OF 'B' DISPLAY																															

DATA WORD

BITS 0-31 PROVIDE THE DATA WORD TO BE STORED BY THE PREVIOUS WRITE DESTINATION FUNCTION.

810027-2

Figure 4-5. System Control Panel Data Return Transfer Format (Sheet 2 of 2)

WDOT SelBUS TRANSFER

DESTINATION BUS

INTR. LEVEL								SCP PHYSICAL ADDRESS								SUBADDRESS							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

INTERRUPT LEVEL

THE INTERRUPT LEVEL IS NOT USED WITH THE SCP AND BITS 0-7 MUST BE ZERO.

PHYSICAL ADDRESS

BITS 8-15 MUST BE ZERO FOR THE SCP.

SUBADDRESS

BITS 16-23 ARE NOT USED WITH THE SCP AND MUST BE ZERO.

DATA BUS

REQUESTED DATA																															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

REQUESTED DATA

BITS 0-31 PROVIDE THE DATA WORD REQUESTED BY THE PREVIOUS DRT-READ SOURCE FUNCTION.

810028

Figure 4-6. System Control Panel Write Data or Order Transfer Format

AICT/ICT SelBUS TRANSFER

DESTINATION BUS

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
	INTR. LEVEL								PHYSICAL ADDRESS											SUBADDRESS			
1								0								0	0	0	0				

INTERRUPT LEVEL

BITS 1-7 PROVIDE THE ONES COMPLEMENT OF THE INTERRUPT LEVEL TO BE CONTROLLED.

PHYSICAL ADDRESS

BITS 9-15 PROVIDE THE PHYSICAL ADDRESS OF THE I/O CONTROLLER OR RTOM TO WHICH THE INTERRUPT LEVEL IS ASSOCIATED.

SUBADDRESS

BITS 20-23 PROVIDE THE I/O CONTROLLER OR RTOM SUBADDRESS WITH WHICH THE INTERRUPT LEVEL IS ASSOCIATED.

DATA BUS

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
NOT USED																											FUNCTION CODE				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					

FUNCTION CODE					
BITS					DEFINITION
27	28	29	30	31	
0	0	0	0	1	ENABLE INTERRUPT
0	0	0	1	0	DISABLE INTERRUPT
0	0	1	0	0	ACTIVATE INTERRUPT
0	1	0	0	0	DEACTIVATE INTERRUPT
1	0	0	0	0	REQUEST INTERRUPT

810029

Figure 4-7. Interrupt Control Transfer Format - RTOM

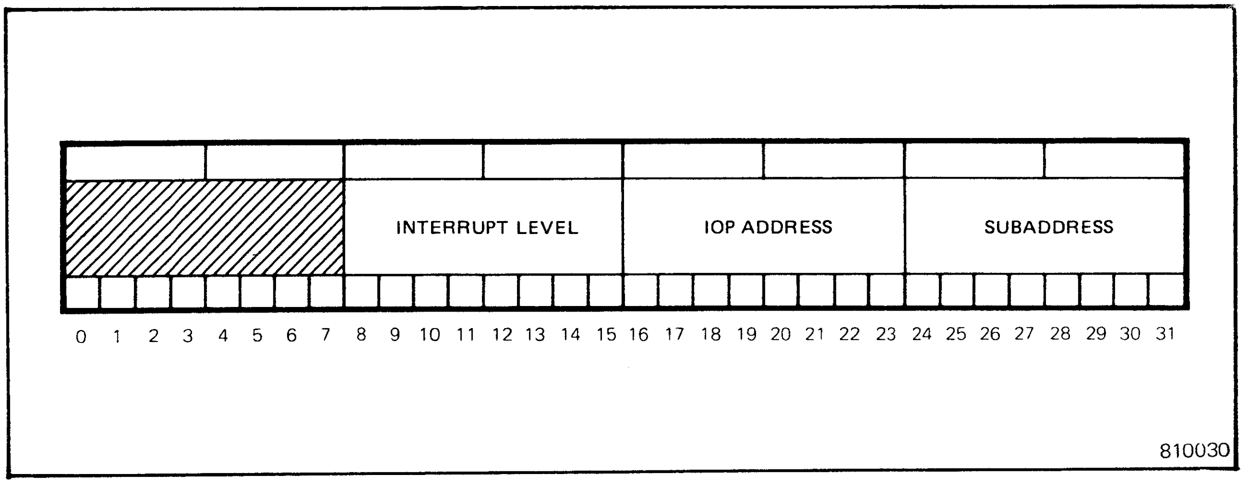
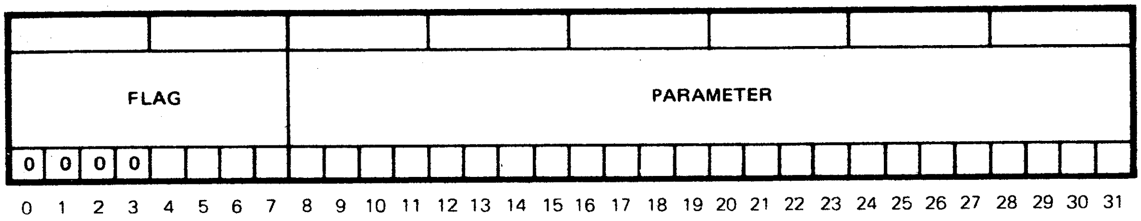


Figure 4-8. CPU to IOP Destination Bus Format (Class F)



NOTE THAT PARAMETER REQUIREMENTS ARE TRANSFER TYPE DEPENDENT BASED UPON THE SPECIFICATION OF FLAG BITS 0 THROUGH 7.

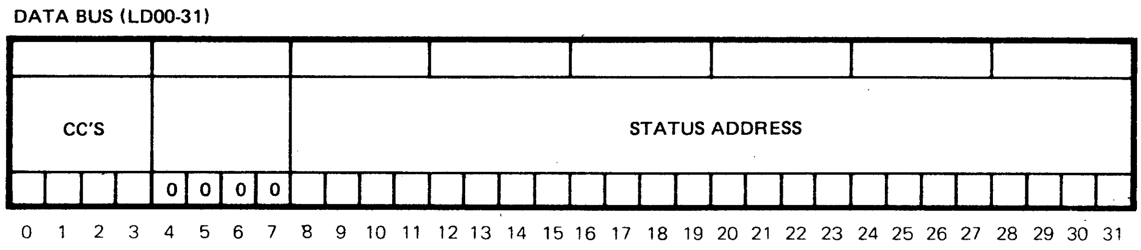
FLAG BITS 0-3

0 - RESERVED FOR FUTURE USE

FLAG BITS 4-7	PARAMETER BITS 8-31
0 - IDENTIFY I/O PROTOCOL	8-28, 30, AND 31 = 0, 29 = 1
1 - UNASSIGNED	
2 - START I/O	REAL IOCD ADDRESS
3 - IPL	REAL IOCD ADDRESS
4 - TEST I/O	ALL ZEROS
5 - STOP I/O	ALL ZEROS
6 - HALT I/O	ALL ZEROS
7 - RESET CONTROLLER	ALL ZEROS
8 - LOAD RAM	8-23 = 0, 24-31 IOP ADDRESS
9 - UNASSIGNED	
A - UNASSIGNED	
B - UNASSIGNED	
C - ENABLE CHANNEL INTERRUPT	ALL ZEROS
D - DISABLE CHANNEL INTERRUPT	ALL ZEROS
E - ACTIVE CHANNEL INTERRUPT	ALL ZEROS
F - UNASSIGNED	

810031

Figure 4-9. ARSTX/RSTX Data Bus Format (Class F)



CC'S (BITS 0-3) WILL BE THE INFORMATION THAT WILL BE PASSED TO THE SOFTWARE IN THE FORM OF CONDITION CODES. THE RELATIONSHIP OF THE DATA BITS TO THE CONDITION CODES ARE:

BIT 0 = CC1

1 = CC2

2 = CC3

3 = CC4

ASSIGNED VALUES FOR THE CC'S ARE.

0 - ACCEPTED BY THE CHANNEL; WILL ECHO STATUS

1 - CHANNEL BUSY

2 - CHANNEL UNDEFINED OR INOPERABLE

3 - SUBCHANNEL BUSY

4 - STATUS STORED

5 - UNSUPPORTED TRANSACTION

6 - RESERVED

7 - RESERVED

8 - ACCEPTED BY THE CHANNEL, NO ECHO STATUS

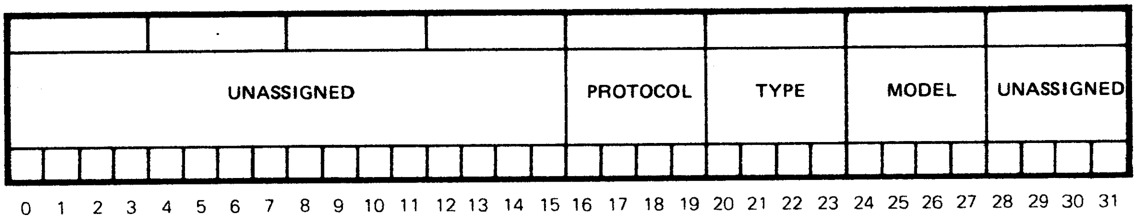
9 - F RESERVED

STATUS ADDRESS (BITS 8-31) INDICATES THE ADDRESS OF THE STATUS WORDS THAT THE CHANNEL WISHES TO TRANSMIT. THE STATUS ADDRESS WILL ALWAYS BE TRANSFERRED TO THE SOFTWARE WHEN PRESENTED BY THE IOP.

810032

Figure 4-10. ARSTX/RSTX Data Return Transfer (Class F)

DATA BUS (LD00-31)



UNASSIGNED - BITS 0-15 ARE IGNORED

PROTOCOL - BITS 16-19 IDENTIFY EXTENDED I/O, HEX 'F'

TYPE - BITS 20-23 IDENTIFY CHANNEL TYPE

BITS 20 IDENTIFIES CHANNEL MEMORY TYPE

0 = CROM

1 = CRAM

BITS 21-23 IDENTIFY CHANNEL TYPE

0 0 0 = INTREGRATED CHANNEL CONTROLLER

0 0 1 = MULTIPLEXER CHANNEL

0 1 0 = BLOCKED MULTIPLEXER CHANNEL

0 1 1 = SELECTOR CHANNEL

1 0 0 = UNASSIGNED

1 0 1 = UNASSIGNED

1 1 0 = UNASSIGNED

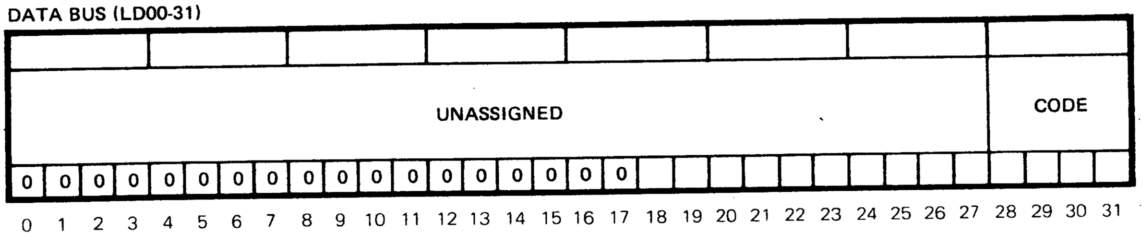
1 1 1 = UNASSIGNED

MODEL - BITS 24-27 IDENTIFY MODEL NUMBER AND ARE CURRENTLY ALL ZEROS

UNASSIGNED - BITS 28-31 ARE IGNORED

810033

Figure 4-11. Identify I/O Protocol - DRT (Class F)



UNASSIGNED: BITS 0-17 CURRENTLY UNUSED AND WILL CONTAIN ZEROS

CODE: BITS 28-31 IDENTIFIES THE FUNCTION REQUEST REQUIRED BY THE CPU

BIT 28 - DEACTIVATE INTERRUPT

BIT 29 - NOT ASSIGNED

BIT 30 - ACKNOWLEDGE AND ACTIVATE INTERRUPT

BIT 31 - ACKNOWLEDGE AND DEACTIVATE INTERRUPT

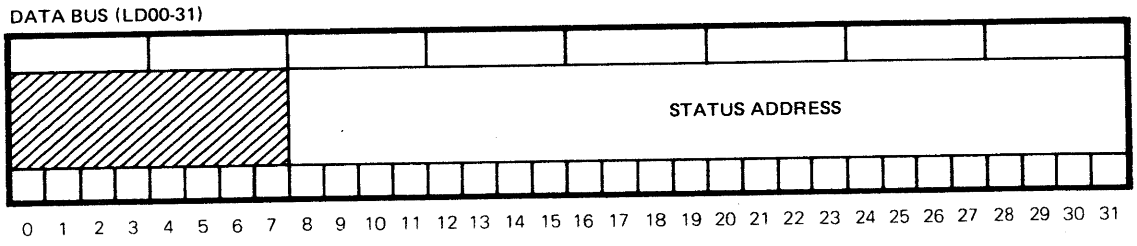
NOTE

BITS 28-32 ARE BIT SIGNIFICANT AND
MUTUALLY EXCLUSIVE.

AICT/ICT COMMUNICATIONS CANNOT BE DEFERRED BY THE IOP.

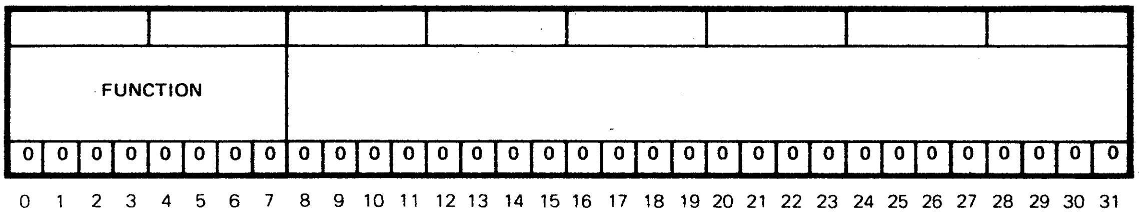
810034

Figure 4-12. AICT/ICT Data Bus Format (Class F)



810035

Figure 4-13. AICT/ICT Data Return Transfer (Class F)



FUNCTION (BITS 0-7): THE ONLY CODE ASSIGNED TO THE WDOT IS A RESET CHANNEL

810036

Figure 4-14. WDOT/WDOT Data Bus Format (Class F)

CHAPTER 5

I/O EXPANSION CHASSIS (OPTIONAL)

5.1 Introduction

This portion of the Technical Manual contains or references information concerning the maintenance and installation of the Model 8910 I/O Expansion Chassis. It is used to provide additional slots and dc power for I/O controller cards when the multipurpose bus of the host system is at capacity. The I/O Expansion Chassis is manufactured by Systems Engineerng Laboratories, Fort Lauderdale, Florida.

NOTE

> The assemblies, backplane, power supply and schematics are contained in the drawings manual 304-328000-000.

Included in the front matter of this manual is a list of related publications which provide a reference to the recommended supporting documentation for the I/O Expansion Chassis.

5.2 Physical Description

The Model 8910 I/O Expansion Chassis consists of the chassis, power supply, backplane, cooling fans and a circuit breaker that also acts as the ON/OFF switch. It provides the power and additional slots needed to expand the multipurpose bus (MPB) of a computer system for the insertion of up to eight I/O cards per chassis. Since it is a separate and self-contained unit it may be attached to the system in the field by the technician either in the same rack or remote rack. The longest cable available is 20 feet therefore racks may be up to approximately 10 feet apart.

NOTE

> Reference to the term "chassis" as used throughout this chapter is synonymous with the Model 8910 I/O Expansion Chassis unless otherwise denoted.

5.2.1 Chassis

The chassis has been designed to mount in a standard 19 inch EIA rack or a customer provided stand or cabinet. The specifications for the chassis are listed in Table 5-1.

Table 5-1
I/O Expansion Chassis Specifications

Characteristics	Specifications
Dimensions:	
Height	12.25 inches (31.12 cm)
Width	19.00 inches (48.26 cm)
Depth	18.25 inches (46.36 cm)
Weight	52.0 lbs (23.4 kilograms)
Mounting:	
Rack	Standard 19 inch EIA rack
Custom	In a cabinet (note: proper ventilation and filtering must be provided in a custom mount situation).
Power Requirements:	
Voltage	115 or 230 vac
Frequency	60 or 50 Hz
Power	7.0 amps at 115 vac 3.5 amps at 230 vac
Operating Environment:	
Temperature	10 to 40° C
Relative Humidity	5% to 95% noncondensing
Multipurpose Bus	dual 4 slots (total of 8)

5.2.2 Power Supply Subassembly

The power supply subassembly contains the power supply, line filter, surge suppressors and a circuit breaker type on/off switch. Special attention must be paid to the fact that the 115 volt and 230 volt subassemblies are different both in wiring and component type. Refer to the drawings manual, drawing number 134-103092-000 sheet 1 for 115 VAC 50/60 Hz and sheet 2 for 230 VAC 50/60 Hz.

5.2.2.1 Power Supply

The power supply is adjusted and tested prior to shipment, and contains no user serviceable parts. All repairs should be performed by SYSTEMS. The power supply contains an input fuse that must be replaced with the exact type and rating as indicated on the label of the power supply. The loss of the input fuse on the power supply indicates that an internal failure has occured. Replacement with the wrong fuse could result in damage to the power supply. To change a blown input fuse, turn off the AC power then change the fuse. Refer to Figure 5-1 for the location of the input fuse.

CAUTION

For continued protection against fire, replace only the same type and rating of fuse.

WARNING

This power supply contains high power capacitors. It is imperative that at least 2 minutes be allowed for the capacitors to drain their voltages off. If this is not done, a HIGH RISK of electrical SHOCK is present!

To change a defective power supply, turn off the main circuit breaker CB1 and disconnect the ac main power line from the power source. Remove the two screws holding the power supply subassembly in place and slide it out. Remove all input and output wires from the power supply, then remove the four mounting screws that secure the unit to the subassembly, they are located on the under side of the subassembly. To install the new unit reverse the removal procedure and then proceed to the next paragraph for adjustments. Do not replace the four power supply mounting screws with longer screws, otherwise internal damage to the power supply will occur.

The power supply voltages and connections should be checked and tightened before the input power can be applied. All adjustments must be made in a loaded condition with a digital meter and a nonconductive type adjusting tool by a qualified technician. The adjustment and test points are shown in Figure 5-1. All dc voltages must be measured on the backplane to insure that the voltages are accurate. Table 5-2 gives the proper voltages.

WARNING

Dangerous voltages are present in this power supply. Handle with care to minimize the hazard of electrical shock. To prevent damage to the power supply, interrupt AC power before disconnecting any power cables!

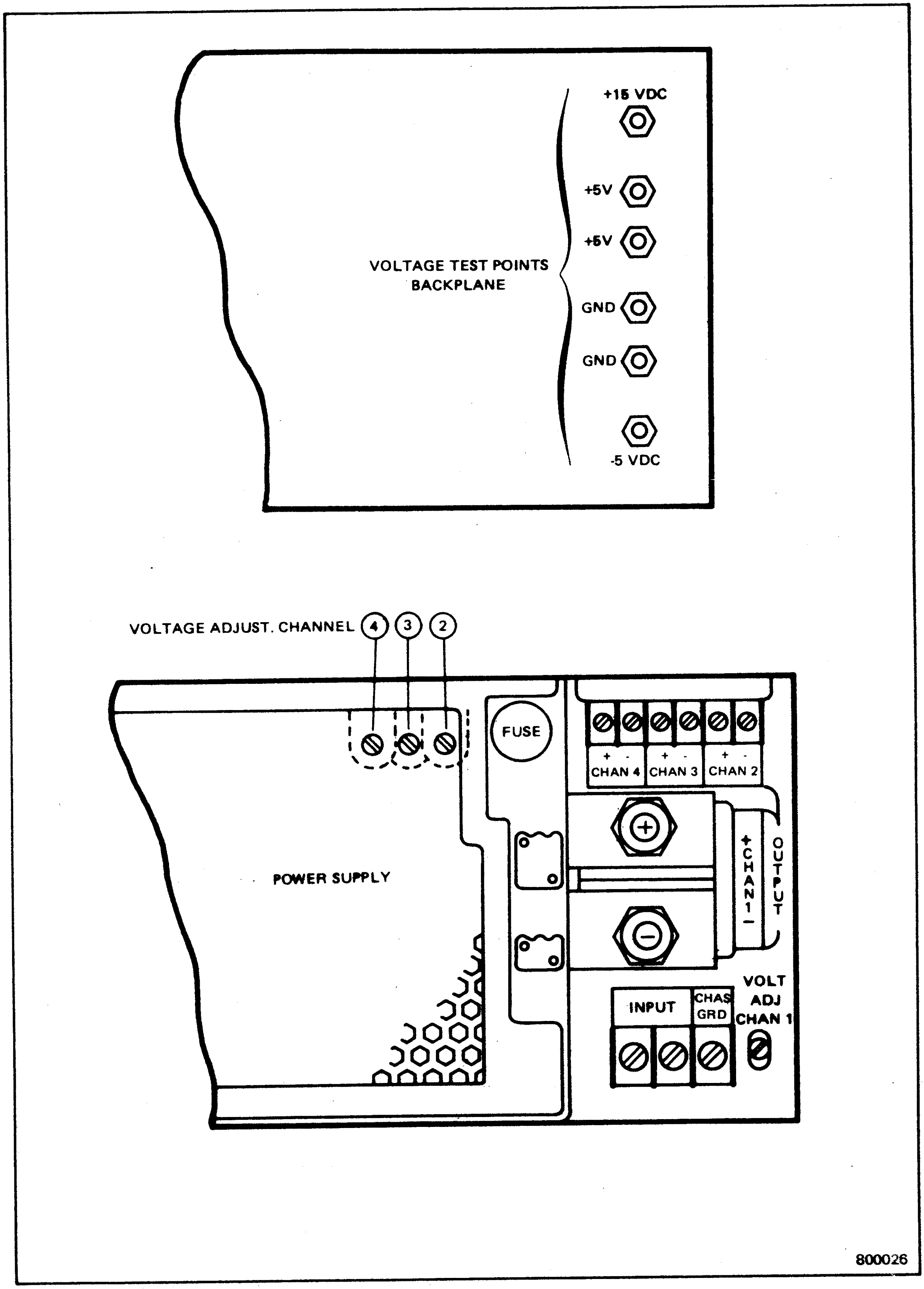


Figure 5-1. Voltage Adjusting and Test Points

Table 5-2
Output Voltages and Amperage

Channel	Specifications
1	+5 vdc @ 40 Amps
2	-5 vdc @ 7 Amps
3	+15 vdc @9 Amps
4	Not Used

5.2.2.2 115 vac 60 Hz (Wiring Diagram 134-103092 Sheet 1)

The power supply subassembly (107-103279-001) is wired for 115 volts ac at 50/60 Hz. Line spike voltages are controlled by a surge suppressor RV1 and a line filter FL1. Overload protection is provided by a single pole circuit breaker CB1. The ac line is connected to the back panel of the chassis on pins 1 and 2 of terminal block TB1 and passes through the chassis, where the ac is distributed to RV1, FL1 and CB1. RV1 is connected across the ac line terminals and limits line spikes. The live leg of the line is connected to CB1 and passes throught the line terminal of FL1, and the neutral leg is connected to the other terminal on the line side of FL1. Distribution of the load side of FL1 goes to the fans (107-103278-001) through TB1 pins 3 and 4, and the input of the power supply.

5.2.2.3 230 vac 50/60 Hz (Wiring Diagram 134-103092 Sheet 2)

The power supply subassembly (107-103279) is wired for 230 volts AC at 50/60 Hz. Line spike voltages are controlled by RV1 and FL1. Overloading of the circuit is protected by a double pole circuit breaker CB1. The ac line is connected to the back panel of the chassis on pins 1 and 2 of TB1, and passes throught the chassis where the ac line is distributed to RV1 and CB1. RV1 is connected across pins 1 and 2 of TB1 and limits line spikes. Unlike the 115 VAC subassembly, the 230 VAC unit connectes both legs of ac to CB1 which in turn supplies the two line terminals of FL1 with power. The load side of FL1 is connected to the fans (107-103278-002) through TB1 pins 3 and 4 and to the input of the power supply.

5.2.2.4 DC Distribution

As shown in drawing number 134-10392-000 of the drawing manual, the dc voltages are fed from the power supply output terminals through terminal block TB2 and connected to the backplane of the chassis. Refer to drawing number 160-103440 of the drawings manual for the actual connecting points to the backplane.

NOTE

Proper regulation of the secondary channel outputs requries that a minimum load be present on the main channel (CHAN 1) output. R1 on sheets 1 and 2 of drawing number 134-103092-000 in the drawings manual shows the power ballast resistor, it has a value of 2 ohms, 50 watts.

5.2.3 Backplane

The I/O expansion chassis backplane is a multilayered printed wiring board, and is fastened on the back of the chassis. connected to the backplane are dc voltage wires, MPB signal cables, I/O interface cables and MPB terminator circuit cards.

Refer to Apppendix B for a listing of the backplane pin allocations. Inside the chassis are eight slots for the insertion of the I/O controller cards, these cards mate with connectors on the backplane. A drawing of the backplane is in the drawings manual, drawing number 160-103440.

5.2.4 Cooling Fans

There are two cooling fans on the left top side of the chassis for the purpose of maintaining a safe operating temperature for the circuit cards and chassis. If the chassis is mounted in Systems standard rack, the air filters in the rack will keep the air flow path clean and clear of dust and papers. If the chassis is mounted in a cabinet or similar device, both ventilation and filtering will have to be supplied by the installing technician. It is of the utmost importance that the chassis, like all chassis, be kept clean to achieve a long life from circuit cards and power supply, therefore filters and air vents should be checked, cleaned and/or changed when dirt is apparent.

To change a defective fan remove the two fastening screws holding the fan subassembly in place, disconnect the ac line connector, then slide out the subassemby (be carefull of the ac supply cable) and change the fan. Pay particular attention to the airflow arrow on the fan, it must point out. When inserting the fan assembly be sure to seat the assembly in the slot provided at the rear of the chassis, if this is not done the fan subassembly will move and possibly cause damage to the fans or surrounding areas.

5.2.5 ON/OFF Switch

The on/off switch is a circuit breaker as well as the power switch. It is mounted on the power supply subassembly, and when the chassis is installed, in SYSTEMS standard rack, it is accessible behind the cabinet door. As previously stated, the circuit breakers are different for 115 and 230 volt operation.

5.2.6 Chassis Swing Door

There are two swing door options, when the chassis is mounted in the rack by SYSTEMS a functional door is supplied. When the chassis is not mounted in a rack by SYSTEMS, a blue decorative air-sealing door is provided. Refer to the kit drawings manual for a detailed drawing of the doors. The drawing number is 118-103335 sheets 1 and 2.

5.3 Installation

For a complete understanding of the installation, refer to the 304-270118 Kit Drawings Manual, drawing number 118-103335 (I/O Expansion Kit) and 118-103337 (I/O Expansion Backplane Kit).

5.3.1 I/O Expansion Kit 118-103335

When the chassis is being installed it is imperative that one of the two doors be in place during operation to insure proper cooling of the chassis and circuit cards.

5.3.2 I/O Expansion Backplane Kit 118-103337

The backplane kit drawings (sheets 1 and 2) illustrate the proper cable installation and routing, it also describes the different cable length options and chassis configurations. The I/O expansion chassis backplane is provided with a split dual four-slot Multipurpose Bus, this provides the capability to expand two computer chassis with four slots each, or one computer chassis with eight slots. To use all eight slots of the chassis, the 2 inch connecting cable (p/n 144-103005-001) must be placed between slots 4 and 5 of the backplane. If more than 8 slots are required, additional chassis may be connected together as shown on sheet 2. A maximum of 16 I/O controller cards may be attached to any one input/output processor.

The multipurpose bus (MPB) terminator provided with the I/O processor kit must be removed from the backplane of the SYSTEMS 32 SERIES computer system. As shown in drawing number 118-103337 the MPB flat cable is connected to the computer backplane where the MPB terminator was removed and the other end of the cable is connected to the MPB of the I/O expansion chassis. Then the MPB terminator should be relocated to the last slot, at the end of the multipurpose bus.

APPENDIX A

IOP MNEMONICS

AAAA	Device address
A/C	Augment code
ACI	Activate channel interrupt
AI	Activate interrupt
AICT	Advance interrupt control transfer
ALU	Arithmetic logic unit
ARSTX	Advance read status transfer
AS	Address stop
ASCII	American Standard Code for Information Interchange
ATN	Attention
BINADD	Binary address
BM	Byte mode
BR	Branch and reset interrupt
BYTEORHW	Byte or halfword
CC	Command chain, condition codes
CCC	Continue command chain
CCCC	Controller address, designated
CCD	Continue chain data
CD	Chain data
CMA	Channel memory allocation
CNT	Count
CPU	Central processing unit
CR	Carriage return
CRAM	Control random access memory (control RAM)
CREG	Control register
CRMA	CRAM address
CRMD	CRAM data
CROM	Control read only memory (control ROM)
CRT	Cathode ray tube
CS	Control switch
CTS	Clear to send
DAC	Data accepted
DACI	Deactivate channel interrupt
DAI	Deactivate interrupt
DAV	Data available
DBUS	Data bus
DCD	Data carrier detected
DCI	Disable channel interrupt
DCR	Device clear
DE	Device end
DI	Device interface
DNA	Does not apply
DRT	Data return transfer
DSPLNUM	Display number

DSR	Data set ready
DSS	Development support system
DTR	Data ready, data terminal ready
EA	Effective address
ECI	Enable channel interrupt
EIA	Electronics industries association
ENP	Enable P
ENRMAOUT	Enable RAM out
ENT	Enable T
EOB	End of block
EOI	End or identify
EOM	End of message
ESC	Execute secondary command
ET	Error transfer
EXEC	Execute
FIFO	First-in-first-out
GPIB	General purpose instrument bus
GPR	General purpose register
HCAD00	High true CROM address 00
HCHNLTIMEOUT	High true channel timeout
HCKINOB	High true clock in output buffer
HCREG00	High true CREG 00
HCROMADDMUXA	High true CROM address MUX A
HCROMADDMUXB	High true CROM address MUX B
HCROM00	High true CROM 00
HDAC	High true data accepted
HDISCACHE	High true disable CACHE
HENACKINOB	High true enable clock in output bus
HENVIX	High true enable interrupt X
HEXINTON	High true external interrupt on
HIFIFOWTENA	High true input FIFO write enable
HIFIRX	High true input FIFO input ready signal X
HIFORX	High true input FIFO output ready signal X
HINHIBIT	High true inhibit
HIO	Halt I/O
HLWRBYTE	High true lower byte
HOBDATAAVAIL	High true output bus data available
HOFIFOWTENA	High true output FIFO write enable
HPARPOLLORD	High true parallel poll level order
HPPROM00	High true parsing PROM 00
HPROM00	High true PROM 00
HRAC00	High true RAM address counter 00
HRAD00	High true RAM address 00
HRDADDRA, B	High true read address A, B
HRDCLKENA	High true read clock enable
HREAD	High true read
HREGSELX	High true register select X
HRESETD	High true reset D
HRDIB	High true read input buffer
HSEQCLK	High true sequence clock
HSKEWCLOCK	High true skew clock
HS00	High true sequence 00
HSTACK00	High true stack 00

HSEQCLK	High true sequence clock
HSKEWCLOCK	High true skew clock
HS00	High true sequence 00
HSTACK00	High true stack 00
HSTADR00	High true stack address 00
HSTKADDRX	High true stack address X
HSTOPCOMM	High true stop command
HS5A, B	High true sequence 5A, 5B
HTESTTRUE	High true test true
HTINFIFO	High true write input FIFO
HTXCHBUS	High true transmit multipurpose bus
HUTXX	High true USART/interval timer data bus XX
HVECTINT	High true vectored interrupt
HWRTADDRA, B	High true write address A, B
HWTFOBLORD	High true write from output buffer level order
HWTOIBORD	High true write to input FIFO order
HWTTOIB	High true write to input buffer
HWTTOOB	High true write to output buffer
H3CLK	High true 3 clock (system clock)
H3CLK00	High true 3 clock 00

I	Input
ICH	Initialize channel
ICL	Initial configuration load
ICT	Interrupt control transfer
ID	Identification
IEEE	Institute of electrical and electronic engineers
IFC	Interface clear
IL	Incorrect length
INST	Instruction
INT	Interrupt
I/O	Input/output
IOCD	Input/output command doubleword
IOCL	Input/output command list
IOCLA	Input/output command list address
IOM	Input/output microprogrammable processor
IOP	Input/output processor
IPL	Initial program load
IPU	Internal processing unit
IS	Instruction stop

LATN	Low true attention
LATTNSW	Low true attention switch
LBLKVECTINT	Low true block vectored interrupt
LBNK00	Low true bank 00
LCIATN	Low true IOP interface attention
LCISTPCOMM	Low true channel interface stop communications
LCKINIB	Low true clock in input buffer
LCKINOB	Low true clock in output buffer
LCLK	Low true clock
LCLKOVR	Low true clock override
LCLRALLEXDINT	Low true clear all external device interrupts
LCLRINT	Low true clear interrupt
LCLRSEQ	Low true clear sequence
LCLRSTKINT	Low true clear stack interrupt

LRAMDATA00	Low true RAM data 00
LRAMIN00	Low true RAM input 00
LRAMWRTUPPR	Low true RAM write upper
LRD	Low true read
LRDDELAY	Low true read delay
LRDDI	Low true read device interface
LRDDIR	Low true read direct
LRDIB	Low true read input buffer
LRDRATESW	Low true read rate switch
LRDUSARTSW	Low true read USART switch
LREAD	Low true read
LRESET	Low true reset
LRESETD	Low true reset D
LRIB	Low true read input buffer
LRILVLXX	Low true request interrupt level
LROMSIM	Low true ROM simulation
LRSTOVR	Low true reset over
LRSTSW	Low true reset switches
LRUN	Low true run
LRUNHALTSW	Low true run/halt switch
LSEND	Low true send
LSENDEOI	Low true send end or identify
LSB	Least significant bit
LSI	Large scale integration
LSN	Least significant nibble
LSOURCEX	Low true source X
LSRQ	Low true service request
LSRSTSW	Low true set/reset switch
LSTKPOP	Low true stack pop
LSTKPUSH	Low true stack push
LSTOPCOMM	Low true stop communications
LSTSC	Low true stop system clock
LSW	Least significant word
LS5A, B	Low true sequence 5A, 5B
LTESTTRUE	Low true test true
LTIMEOUTSTART	Low true timeout start
LUPACK	Low true user panel acknowledge
LUTXX	Low true USART/interval timer bus XX
LVECTINT	Low true vectored interrupt
LVECTINT-1	Low true vectored interrupt minus one
LVI00	Low true vectored interrupt 00
LWAIT	Low true wait
LWRTFILE	Low true write file (gate write)
LWTDELAY	Low true write delay
LWTDIR	Low true write directly
L150NSCLK	Low true 150-nsec clock
L2CLK	Low true 2 clock
L2CLOCK	Low true 2 clock
L5.0688MHZCLK	Low true 5.0688 MHz clock

M	Modifier
MA	Memory address, memory addess read
MAC	Memory address counter
MAR	Memory address register
MAV	Memory address read virtual
MBC	Memory bus controller
MD	Memory data

MDC	Multidevice controller
MIRT	Memory instruction read transfer
MPB	Multipurpose bus
MRLT	Memory read and lock transfer
MRT	Memory read transfer
MSB	Most significant bit
MSGE	Message
MSN	Most significant nibble
MSW	Most significant word
MUX	Multiplexer
MWT	Memory write transfer
NDAC	No data accepted
NNNN	Nibble
NOP	No operation
NOTEOB	Not EOB
NRFD	Not ready for data
O	Output
OFORX	Output FIFO ready line X
OP-CODE	Operation code
ORD	Order
OVR	Override
PC	Program counter
PCI	Programmable communications interface
PE	Parity error
PPCI	Post program controlled interrupt
PPROM	Parsing programmable read only memory
PRIP	Primary panel
PROM	Programmable read only memory
PSD	Program status doubleword
PSW	Program status word
R	Register
RAC	RAM address counter
RAM	Random access memory
RDT	Read data transfer
REGSELX	Register select X
RI	Request interrupt
ROMSIM	Read only memory simulation
RPU	Regional processing unit
RS	Read stop
RSCHNL	Reset channel
RSCTL	Reset controller
RST	Reset
RSTX	Read status transfer
RSV	Request for service
RT	Real time
RTOM	Real-time option module
RTS	Request to send
RX	Receive
SCC	Selected controller clear
SCP	System control panel
SCPI	System control panel interface
SDC	Selected device clear

SECP	Secondary panel, system control panel
SI	Service interrupt
SICA	Service interrupt control area
SIL	Suppress incorrect length
SIO	Start I/O
SIV	Service interrupt vector
SKIP	Skipping
SPD	Serial poll disable
SR	Status ready
SRQ	Service request
ST	Stop
STPIO	Stop I/O
SUB-OP	Suboperation
SWAPORNOT	Swap or not

TA	Transfer acknowledge
TIC	Transfer in channel
TIO	Test I/O
TTL	Transistor/transistor logic
TTY	Teletypewritter
TX	Transmit

UART	Universal asynchronous receiver/transmitter
USART	Universal synchronous/asynchronous receiver/transmitter
UUTXX	USART bus XX

WCS	Writable control storage
WDOT	Write data or order transfer
WRITELWR	Write lower
WRITEUPPER	Write upper
WS	Write stop

APPENDIX B
I/O EXPANSION BACKPLANE

PIN ALLOCATION

PIN	CONNECTOR	SIGNAL	PIN	CONNECTOR	SIGNAL
1	—	GND	40	P1A-38	
2	—	SPARE	41	-39	—
3	P1A-1	—	42	-40	—
4	-2	—	43	-41	—
5	-3	—	44	-42	—
6	-4	—	45	-43	—
7	-5	—	46	-44	—
8	-6	—	47	-45	—
9	-7	—	48	-46	—
10	-8	—	49	-47	—
11	-9	—	50	-48	—
12	-10	—	51	-49	—
13	-11	—	52	-50	—
14	-12	—	53	-51	—
15	-13	—	54	-52	—
16	-14	—	55	-53	—
17	-15	—	56	-54	—
18	-16	—	57	-55	—
19	-17	—	58	-56	—
20	-18	—	59	-57	—
21	-19	—	60	-58	—
22	-20	—	61	-59	—
23	-21	—	62	-60	—
24	-22	—	63	—	GND
25	-23	—	64	—	SPARE
26	-24	—	65	—	GND
27	-25	—	66	—	+5V
28	-25	—	67	—	SPARE
29	-27	—	68	—	SPARE
30	-28	—	69	—	-15V
31	-29	—	70	—	+15V
32	-30	—	71	—	-15V
33	-31	—	72	—	+15V
34	-32	—	73	—	SPARE
35	-33	—	74	—	SPARE
36	-34	—	75	—	GND
37	-35	—	76	—	+5V
38	-36	—	77	—	GND
39	-37	—	78	—	SPARE

I/O EXPANSION BACKPLANE

PIN ALLOCATION

PIN	CONNECTOR	SIGNAL	PIN	CONNECTOR	SIGNAL
79	P1B-1	—	111	P1B-33	—
80	-2	—	112	-34	—
81	-3	—	113	-35	—
82	-4	—	114	-36	—
83	-5	—	115	-37	—
84	-6	—	116	-38	—
85	-7	—	117	-39	—
86	-8	—	118	-40	—
87	-9	—	119	-41	—
88	-10	—	120	-42	—
89	-11	—	121	-43	—
90	-12	—	122	-44	—
91	-13	—	123	-45	—
92	-14	—	124	-46	—
93	-15	—	125	-47	—
94	-16	—	126	-48	—
95	-17	—	127	-49	—
96	-18	—	128	-50	—
97	-19	—	129	-51	—
98	-20	—	130	-52	—
99	-21	—	131	-53	—
100	-22	—	132	-54	—
101	-23	—	133	-55	—
102	-24	—	134	-56	—
103	-25	—	135	-57	—
104	-26	—	136	-58	—
105	-27	—	137	-59	—
106	-28	—	138	-60	—
107	-29	—	139	—	SPARE
108	-30	—	140	—	SPARE
109	-31	—	141	—	GND
110	-32	—	142	—	+5V

I/O EXPANSION BACKPLANE

PIN ALLOCATION

PIN	CONNECTOR	SIGNAL	PIN	CONNECTOR	SIGNAL
143	P1C-1	GND	183	P1C-41	LDI14
144	-2	CLOCK	184	-42	LDI15
145	-3	GND	185	-43	GND
146	-4	LATN	186	-44	LNRFD
147	-5	GND	187	-45	GND
148	-6	LDAV	188	-46	LNDAC
149	-7	GND	189	-47	GND
150	-8	LEOI	190	-48	SPARE
151	-9	GND	191	-49	GND
152	-10	LPARITY	192	-50	+5V
153	-11	GND	193	—	GND
154	-12	LSRQ	194	—	+5V
155	-13	GND	195	—	—
156	-14	LHALFWORD	196	—	—
157	-15	GND	197	P1D-1	—
158	-16	LIFC	198	-2	—
159	-17	GND	199	-3	—
160	-18	5.0688 MH	200	-4	—
161	-19	GND	201	-5	—
162	-20	LDI00	202	-6	—
163	-21	LDI01	203	-7	—
164	-22	GND	204	-8	—
165	-23	LDI02	205	-9	—
166	-24	LDI03	206	-10	—
167	-25	GND	207	-11	—
168	-26	LDI04	208	-12	—
169	-27	LDI05	209	-13	—
170	-28	GND	210	-14	—
171	-29	LDI06	211	-15	—
172	-30	LDI07	212	-16	—
173	-31	GND	213	-17	—
174	-32	LDI08	214	-18	—
175	-33	LDI09	215	—	GND
176	-34	GND	216	—	SPARE
177	-35	LDI10	217	—	GND
178	-36	LDI11	218	—	+5V
179	-37	GND	219	—	GND
180	-38	LDI12	220	—	+5V
181	-39	LDI13	221	—	GND
182	-40	GND	222	—	SPARE

I/O EXPANSION BACKPLANE

PIN ALLOCATION

PIN	CONNECTOR	SIGNAL	PIN	CONNECTOR	SIGNAL
223	P1E-1	—	254	P13-32	—
224	-2	—	255	-33	—
225	-3	—	256	-34	—
226	-4	—	257	-35	—
227	-5	—	258	-36	—
228	-6	—	259	-37	—
229	-7	—	260	-38	—
230	-8	—	261	-39	—
231	-9	—	262	-40	—
232	-10	—	263	-41	—
233	-11	—	264	-42	—
234	-12	—	265	-43	—
235	-13	—	266	-44	—
236	-14	—	267	-45	—
237	-15	—	268	-46	—
238	-16	—	269	-47	—
239	-17	—	270	-48	—
240	-18	—	271	-49	—
241	-19	—	272	-50	—
242	-20	—	273	-51	—
243	-21	—	274	-52	—
244	-22	—	275	-53	—
245	-23	—	276	-54	—
246	-24	—	277	-55	—
247	-25	—	278	-56	—
248	-26	—	279	-57	—
249	-27	—	280	-58	—
250	-28	—	281	-59	—
251	-29	—	282	-60	—
252	-30	—	283	—	GND
253	-31	—	284	—	SPARE

INDEX