INPUT/OUTPUT PROCESSOR (IOP)
Model 8000 and 8001

IPU CONSOLE IOP

Reference Manual

April 1983

Publication Order Number: 301-000170-000

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 Reference Manual, Publication Order Number **301-000170-000**, was printed July, 1980.

Publication Order Number **301-000170-001** (Change 1) was printed December, 1982.

Publication Order Number **301-000170-002** (Change 2) was printed April 1983. The updated manual contains the following pages:

	* Change Number		* Change Number
Title page	2	4-1	1
Copyright page	0	4-2 through 4-4	0
iii/iv Change 2	2	4-5	1
iii/iv Change 1	1	4-6 through 4-23	0
v through x	2	4-24	2
1-1 through 1-5/1-6	2	4-25 through 4-39	0
2-1/2-2	1	4-40	2
3-1 through 3-10	0	4-41 through 4-58	0
3-11 and 3-12	2	5-1	1
3-13/3-14	0	5-2 through 5-3/5-4	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) (Models 8000 and 8001) Reference Manual, Publication Order Number **301-000170-000,** was printed July, 1980.

Publication Order Number **301-000170-001 (Change 1)** was printed December, 1982. The updated manual contains the following pages:

	* Change Number		* Change Number
Title page	1	4-1	1
Copyright page	0	4-2 through 4-4	0
iii through xii	1	4-5	1
1-1 through 1-4	0	4-6 through 4-58	0
2-1/2-2	1	5-1	1
3-1 through 3-14	0	5-2 through 5-4	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.

CONTENTS

Chapter Page

Chapter Page

ILLUSTRATIONS

TABLES

CHAPTER 1

GENERAL INFORMATION

1.1 Introduction

This User Reference Manual has been designed to familiarize the operator/user with the configuration conditions, operational characteristics, software programming considerations, and other special aspects of the Model 8000, 8001 Input/Output Processor (IOP) and the IPU Console IOP. They are designed and manufactured by Gould Inc., S.E.L. Computer Systems Division, Fort Lauderdale, Florida.

The primary supporting documents for this user reference manual are the IOP Technical Manual, publication number 303-000170, the IOP User Interface Manual, publication number 310-000890, and the IOP and I/O Expansion Chassis Drawings Manual, publication number 304-000170.

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

Chapter 1 - General Information
Chapter 2 - Configuration Information
Chapter 3 - Operation
Chapter 4 - Software Programming
Chapter 5 - IOP Real-Time Option Module (RTOM)

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 multipurpose 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.

1.2 Product Description

The IOP is an I/O multiplexing channel, which also includes the functions of a real-time option module (RTOM), system control panel (SCP), and 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 MPB slot of a split backplane or an I/O expansion chassis. Using this type of board connection, high packing density can be achieved.

The IOP is implemented using 2901 slice processors. This processor and its tri-state bus allows shared processing, for all of the functions and power to control the four interfaces resident on the IOP.

The SelBUS interface is a dual bidirectional interface used for memory and CPU communications. It utilizes two physical addresses on the SelBUS and allows for simultaneous data and command transfers from memory and the CPU respectively. 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.

Both the master and slave console ports are identical and have RS-232C and current loop interfaces. Data is formatted and outputted to the cathode ray tube (CRT) terminal in a serial manner. Switching between SCP and operator console modes is achieved using specific commands that are entered into the operator console, via the CRT terminal. Anytime that the system halts, the CRT terminal automatically switches to the SCP mode, the current program status word (PSW) and machine states are outputted. Outputting data in the serial format allows the serial ports to be used for CRT or hard copy SR43 device.

The slave monitor/initial program load (IPL) interface supports a full-duplex RS-232C modem interface. This port may be used locally or through a modem control, monitor, or IPL the system. This action allows maximum flexibility for the master or slave user to exercise the system. Since the IOP may also be the system control panel, macrodiagnostics or software programs may also be loaded and executed using this port. The slave port is activated by the master port so that the slave port 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 slave user. The IPU Console IOP does not support IPL.

The IOP MPB interface provides a type of first-in, first-out (FIFO) buffering of the data, allowing the IOP microengine time to perform RTOM or SCP functions during a high-speed (1.5-2.0M bytes/sec) burst mode transfer. The MPB protocol allows easy implementatiion 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 are allowed on any one MPB. Multiple IOPs may be used to expand the number of controller devices. Grab controller and IOP to IOP communications are not allowed on the MPB.

1.3 Functional Description

The IOP implements both the 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 port, is designated for the support of a master device such as a CRT terminal, teletypewriter (TTY), or full-duplex modem.

The other port, the slave port, 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 and operator console functions from a slave I/O device.

In the operator console mode, the displays on each of the CRT terminals (master and slave), are independent and under software control. In the SCP mode, both the master and slave RT terminals display SCP data, unless the slave CRT terminal is deactivated by the system control panel command (PRIP). Activation of the slave CRT terminal causes both the master and slave CRT terminals to display identical information. However, only one mode or the other can be activated at any one time. Automatic switching, form the SCP mode to the operator console mode, occurs when the run command is entered into the SCP. CPU transitions from run to halt cause the CRT terminal to be switched to the SCP mode and current data outputted to the display.

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

If one operator begins to input a message and the other operator inputs a character, the two characters appear sequentially on the screen. Thus the command will be rejectec as being incorrectly inputted.

The master and slave ports use a universal synchronous/asynchronous receiver/transmitter (USART) integrated circuit to handle the asynchronous formatting.

The master and slave ports have a choice of RS-232C or current loop interface. Both ports support full-duplex modem controls.

The ports have a set of jumpers to determine operating characteristics. These jumpers are read during the powerup system reset routine and used to initialize the USART. The master and slave ports each have four jumpers dedicated to the selection of baud rate. Note that if only one port is in use, the baud rate for the other port should be set at the same or higher baud rate. This action is necessary, when the IOP is in the SCP mode, because the firmware will attempt to echo (send information) to both ports. A low baud rate on the unused port causes the active port to slow down to the slower baud rate.

An eight-bit American Standard Code for Information Interchange (ASCII) code, with no parity (most-significant bit equal to zero) can be jumper selected into the USART, thus allowing eight-bit binary data to be inputted through the ports. A seven-bit ASCII code, with even parity, allows mechanical TTY terminals to be used.

The SCP, implemented on the IOP, supports the IPL function. The two commands relating to the IPL function are as follows:

IPL	IPL device is at default address
IPL=XXXX	IPL device is at address XXXX

The IPL device may be connected to the SelBUS locally on the MPB, connected to the master terminal port of the IOP, connected to the slave port, or connected at a distance over a modem line to one of the two ports.

Input of one of the IPL commands, given above, causes the IOP firmware to notify the CPU to initiate its IPL routine. This routine sets up an IOCD at memory address 0000 with the command chain and suppress incorrect length flags set. The IOCD calls for a binary read of 120 bytes, beginning at memory address 0000. The CPU performs an initial load RAM with a dummy interrupt level, to communicate with the SCP. A command is then sent to the addressed device to start I/O. The initial records read from the IPL device, writes a predetermined program status doubleword (PSD), beginning at memory address 0000 followed by the necessary number of chained IOCD's 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 intiate the IPL firmware sequence. The IPU Console IOP does not support IPL.

1.4 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

CHAPTER 2

CONFIGURATION INFORMATION

2.1 Software

The real-time option module (RTOM) and system control panel (SCP) functions are implemented using the SelBUS protocol that is defined in the IOP Technical Manual, publication number 303-000170. The multipurpose bus (MPB) uses a subset of extended I/O protocol as it is defined in Chapter 3 of this manual. The serial ports on the IOP are treated as subaddresses FC-FF of the physical address of the IOP. FC and FE are input addresses. FD and FF are output addresses.

2.2 Hardware

A jumper is provided on the Input/Ouput Processor (IOP) to allow selection of the SCP function. The SCP cathode ray tube (CRT) ports may be selected to operate from 110 to 19.2K baud. The IOP contains two physical addresses on the SelBUS (an even and an odd pair).

The MPB is a medium speed asynchronous bus. The MPB protocol allows easy implementation, and low cost bus interfaces, for each of the 16 possible controllers. Each of the controllers may operate up to 16 separate devices; however, a total of 124 devices are allowed on any one MPB. Multiple IOPs may be used to expand the number of devices that are to be controlled.

The MPB is designed so that it may be contained in one computer cabinet. From 1 to 16 controllers can be connected to the MPB. The controllers that are used in a particular system can be distributed in any manner desired along the length of a controller backplane. The number of devices that can be connected to a controller is limited to 16.

The IOP is at one end of the MPB in a slot adjacent to the system's primary central processing unit (CPU), if the system contains more than one CPU. Note that the IOP must be located next to the system CPU if the IOP is to be used as the SCP. Termination resistors are provided on the MPB terminator board at the opposite end of the bus.

Maintaining the integrity of these requirements allows the MPB to vary in length using the necessary controller backplanes.

The MPB interconnect cable is a 50-pin flat cable consisting of 16 data lines, one parity line, three handshake lines, six control lines, and two clock lines.

CHAPTER 3

OPERATION

3.1 General Information

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

The local-remote ports will be known as master-slave ports. Both ports support a full-duplex modem control and two wire-20 mA current loop interface.

The master port is designated for the support of a master device such as a cathode ray tube (CRT) terminal, teletypewriter (TTY), or full-duplex modem. The master port can be used to select the slave port when it is required.

The slave port 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 and operator console functions from a slave I/O device.

In the operator console mode the displays on each of the CRT terminals, (master and slave), are independent and under software control. In the SCP mode, both the master and slave CRT terminals display SCP data, unless the slave CRT terminal is deactivated by the secondary panel (SECP) command. Activation of the slave CRT terminal causes both the master and slave CRT terminals to display identical 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. CPU transitions from run to halt cause the CRT terminal to be switched to the SCP mode and current data outputted to the display.

3.1.1 System Control Panel Mode

On powerup, baud rate jumpers are read and the appropriate baud rate selected. The ports come up in the SCP mode with a data terminal ready (DTR) indication. The DTR line remains on even if the ports are switched to the operator console mode, as long as a disconnect command has not been executed. Care should be exercised, with certain operator console modems, especially when a ring-in indication is expected to occur. These particular modems give a busy signal indication to an attempted ring-in condition whenever there is a DTR indication. Thus, the operator console port would never receive a ring-in indication and attention status would never be reported.

The system comes up with the SECP enabled (the slave port is also activated as a SCP). This action facilitates using the turnkey panel functions via the CRT terminal. Thus, reset and initial program load (IPL) commands can be executed after powerup.

In the SCP mode, the operator at the master port can input a primary panel (PRIP) command, thus preventing the slave port from being active and inputting commands. The SECP command returns the slave port to the SCP mode. This action allows the slave port to be active in the SCP mode. Note that the SECP and PRIP commands do not affect the operator console mode's operation.

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

If one operator begins to input a message and the other operator inputs a character, the two characters appear sequentially on the screen. This action serves as an indication that the master operator wishes to take control.

3.1.2 Operator Console Mode

To change from the SCP mode to the operator console mode it is necessary for one of the CRT terminal operators to input @@C [CR]. Upon receipt of the CR command the firmware moves the cursor, on both CRT terminals, to the extreme left margin of the next line.

To return to the SCP mode, enter @@P[CR]. When the SCP mode is selected, // is used as the prompt so that a command can be entered on the CRT terminal. Operator console mode prompt messages are controlled by software.

When the run command is entered while the IOP is in the SCP mode the display is switched to the operator console mode where milestone messages may be outputted.

As the characters are inputted to the CRT terminal, operating in the operator console mode, they are sent to memory and the byte count is decremented. Software must determine when the message is complete so that the information can be decoded.

To exit the operator console mode and return to the SCP mode it is necessary for one of the CRT terminal operators to input @@P[CR]. SCP information is outputted and the cursors on both CRT terminals move to the extreme left. Operator console mode messages, that were currently on the screen, are left in the display. A run-halt transition also causes an automatic switch to the SCP mode. This action causes the program status word (PSW) to also be displayed.

If the CRT terminals are in the SCP mode and software has a message to output to the operator console, the firmware will accept an I/O command to an inactive subchannel, but will not perform the operation until it is in the operator console mode. Any subsequent commands to the subchannel will receive busy status.

3.2 System Control Panel Controls and Indicator

There is no panel lock capability and no power switch associated with the SCP.

The SCP display of state indicators consists of one line, followed by the SCP prompt (//) at the extreme left of the line. The cursor is positioned immediately after the prompt so that operator commands can be inputted. This SCP display occurs in response to certain operator commands, system resets, or run-halt transitions. An alternative display, consisting of eight eight-bit hexadecimal character numbers, is outputted in response to an operator request to read the general purpose central processing unit (CPU) base registers. This output is also followed by the SCP prompt on the next line.

In response to an operator's request to read the general purpose registers (GPRs) or the base registers in the CPU, the SCP outputs one line with all eight registers shown. The display consists of eight eight-digit displays, side-by-side, beginning with register 0 at the left, through register 7 at the right. (See Figure 3-1). Note that NNNNNNNN is an eight-digit numeral field.

```
NNNNNNNN   NNNNNNNN   NNNNNNNN   NNNNNNNN   NNNNNNNN   NNNNNNNN   NNNNNNNN   NNNNNNNN
```

Figure 3-1. State Indicator Output-Read GPR Or Base Registers

For some commands the state indicators are not outputted. When this situation occurs, only the SCP prompt appears on the next line following the input command. The appearance of the prompt // always indicates to the operator, that the previously inputted command has been executed by the SCP. In the event of an error condition, such as a misspelling of a command by the operator, or the inputting of certain commands while the system is in the run mode, the words OPERATOR ERROR are outputted on the line following the command, and the SCP prompt appears on the next line. This indication signifies the readiness of the SCP to accept a new command. Note that the command that was in error will not be executed.

The display parameters are not dynamic (i.e. the indicated states are only true at the time they are outputted). However, an operator can readout current state indicators while the machine is in the run mode by inputting a command that will provide the states (i.e., a memory read command).

If a memory read is indicated by the operator command, and a nonpresent memory indication is sent to the SCP, a NON PRES MEM message appears after the command with the prompt appearing on the next line. If a SelBUS error has occurred in the course of executing an operator command, the words SELBUS ERROR appear followed by the prompt on the next line.

As commands are inputted and executed, they are outputted and scrolled on the screen.

The format for the state indicators is shown in Figure 3-2.

Note that in Figure 3-2 all the alternatives for each field of the line are given. Only one of these fields will appear at any one time.

```
MA  NNNNNNNN  MD    NNNNNNNN  OVR  INT  WAIT  AS  ST  RUN
PSW           INST                            IS  PE  HALT
PSD           EA                              RS
              CS                              WS
```

Figure 3-2. State Indicator Format

The state indicators are paired with other state indicators as shown in Figure 3-3.

MEMORY ADDRESS (MA)	-	MEMORY DATA (MD)
MEMORY ADDRESS (MA)	-	CONTROL SWITCHES (CS)
PROGRAM STATUS WORD (PSW)	-	INSTRUCTION (INST)
PROGRAM STATUS DOUBLEWORD (PSD)	-	(BLANK)

Figure 3-3. Status Indicator Pairs

State indicators that are not present when the line is outputted are left blank. The information in the left half of the line, including the two number fields, reflects the information last obtained in response to certain operator commands. For example, if the operator inputs a clock override command, the corresponding change is shown in the appropriate space of the state indicators line; however, the number fields display the previous information that was in these fields. Table 3-1 defines the state indicators.

Table 3-1
State Indicators

AS	Address stop set (all three stops set)
CS	Control switch settings is adjacent display (NNNNNNNN)
EA	Effective address in adjacent display (NNNNNNNN)
HALT	CPU in halt mode
INST	Instruction in adjacent display (NNNNNN)
INT	Interrupt active
IS	Instruction stop set
MA	Memory address in adjacent display (NNNNNNNN)
MD	Memory data in adjacent display (NNNNNNNN)
OVR	Clock override on
PE	Parity error from memory
PSD	Program status doubleword in adjacent two displays (NNNNNNNN NNNNNNNN)
PSW	Program status word 1 in adjacent display (NNNNNNNN)
RS	Operand read stop set
RUN	CPU in run mode
ST	Stop (one of the address stop criteria has been met)
WAIT	Instruction execution not in progress
WS	Operand write stop set

Note: Setting of any address stop causes the disable CACHE line to be set on the turnkey panel cable. Resetting the address stop causes the disable CACHE line.

The IOP binary data pattern is shown in the following figures. The state indicators give various number fields and indications. The output pattern of the state indicators is shown in Figure 3-4.

Following the final CR LF indication, either a slash slash (reverse slash, ASCII '2F', for the SCP prompt) or a software message may occur if the system switches to the operator console mode.

The GPR registers display the contents of the general purpose registers. The output pattern of the GPR registers is shown in Figure 3-5.

The GPR readout display is shown in Figure 3-6.

The operator error message is given in response to an incorrectly entered SCP command. The output pattern for the operator error message is shown in Figure 3-7.

The output on the CRT is shown in Figure 3-8.

```
[CR]  [LF]  [NUL]   [NUL]   [NUL]
[SP]  MA            or,
PSW                 or,
PSD                 or,
[SP]  [SP]  [SP]    [SP]
[SP]
NNNNNNNN            or,
[SP]  [SP]  [SP]    [SP]    [SP]  [SP]  [SP]  [SP]
[SP]
[SP]  [SP]  MD              or,
INST                or,
[SP]  [SP]  EA              or,
[SP]  [SP]  CS              or,
[SP]  [SP]  [SP]
NNNNNNNN            or,
[SP]  [SP]  [SP]    [SP]    [SP]  [SP]  [SP]  [SP]
[SP]  [SP]  [SP]
OVR                 or,
[SP]  [SP]  [SP]
[SP]  [SP]  [SP]
INT                 or,
[SP]  [SP]  [SP]
[SP]  [SP]  [SP]
WAIT                or,
[SP]  [SP]  [SP]    [SP]
[SP]  [SP]  [SP]
AS                  or,
IS                  or,
RS                  or,
WS                  or,
[SP]  [SP]
[SP]  [SP]  [SP]
ST                  or,
PE                  or,
[SP]  [SP]
[SP]  [SP]  [SP]
[SP]  RUN           or,
HALT
[CR]  [LF]  [NUL]   [NUL]   [NUL]   [NUL]
```

Notes: [SP] = Space, [LF] = Line Feed, [CR] = Carriage returns, NNNNNNNN = an eight-digit numeral field, [NUL] = a Nul character (needed to give time for a [CR] to occur when operating the mechanical Teletypewriter (TTY) carriages).

Figure 3-4. State Indicator Output Pattern

```
RRRRRRRR
[SP]
RRRRRRRR
[SP]
RRRRRRRR
[SP]
RRRRRRRR
[SP]
RRRRRRRR
[SP]
RRRRRRRR
[SP]
RRRRRRRR
[SP]
RRRRRRRR
[SP]
[CR]  [LF]  [NUL]  [NUL]  [NUL]  [NUL]
```

Figure 3-5. GPR Output Pattern

```
RRRRRRRR   RRRRRRRR   RRRRRRRR   RRRRRRRR   RRRRRRRR   RRRRRRRR   RRRRRRRR   RRRRRRRR
NOTE:THAT  RRRRRRRR  DENOTES REGISTER CONTENTS.
```

Figure 3-6. GPR Output Display

```
[CR]  [LF]  [NUL]  [NUL]  [NUL]  [NUL]
OPERATOR
[SP]
ERROR
[CR]  [LF]  [NUL]  [NUL]  [NUL]  [NUL]
//
```

Figure 3-7. Operator Error Output Pattern

```
OPERATOR ERROR
//
```

Figure 3-8. Operator Error Message Format

The SCP prompt message indicates that the SCP is ready to receive a new command. The output on the CRT is shown in Figure 3-9.

```
[CR]  [LF]  [NUL]  [NUL]  [NUL]  [NUL]
//
```

Figure 3-9. SCP Prompt Format

3.3 System Control Panel Command

Table 3-2 contains the alphabetical list of commands for the SCP. Note that a carriage return (CR) is required after each command. Lock on and lock off commands are not supported by the CRT panel.

The following commands will not be executed and an operator error indication displayed if the CPU is not in the halt mode.

```
CRMA=XXXX
CRMD=XXXXXXXXXXXX
CLE
EA
EXEC
GPR
GPRA=XXXXXXXX
IPL
PL=XXXX
MAV=XXXXXX
PC=XXXXXX
PSD
PSD=XXXXXXXX
PSW
PSW=XXXXXXXX
STEP
RST
```

Note that XXXXXXXX indicates a hexadecimal number field.

When the SCPs are active, they operate in the echo mode. In the echo mode, each character inputted from the CRT is stored in a buffer and echoed (sent) to both terminals. Upon receiving a CR command, the firmware decodes the input characters and executes the command. The display is updated after certain commands are executed. After all the commands are executed and applicable characters displayed, the SCP prompt (//) is outputted. This action indicates, to the operator(s), that this particular command has been executed. If the particular command cannot be executed, an abbreviated error message is outputted on the line following the command.

During the execution of some commands, all states are checked and the display updated. The cursor appears at the extreme left margin to the right of the prompt message (//).

The memory must be cleared (set to zero) before an IPL operation can be performed. To clear memory the clear (CLE) command is used. This command causes zeros to be written into all of the memory locations. After the powerup procedure is completed the CLE command also clears any parity errors. When the memory has been cleared, the // indication appears on the CRT terminal.

Note that in all of the SCP commands, XXXXXX indicates a hexidecimal number field. These number fields are right justified. For example, if an eight-digit field is required and the operator only inputs five characters before the carriage return, these five characters are considered to be the least-significant characters. The most-significant characters are assumed to be zeros. If the operator inputs more than the number of digits required for a particular command, only the last digits that fit into the command field are used.

3.3.1 Attention (@@A [CR])

Inputting @@A[CR] while the system is in the operator console mode causes an attention interrupt to occur. Inputting @@A[CR] while the system is in the SCP mode causes an operator error indication to occur with no interrupt.

3.3.2 Enter Console Mode (@@C [CR])

The @@C[CR] command causes an immediate switch to the operator console mode.

3.3.3 Enter Panel Mode (@@P [CR])

When in the operator console mode the @@P[CR] command causes the ports to switch to the SCP mode and output the SCP prompt //.

3.3.4 Address Stop Commands

The address stop commands (AS[CR], AS=XXXXXXXX[CR], IS[CR], IS=XXXXXXXX[CR], RS [CR], RS=XXXXXXXX[CR], WS [CR], and WS=XXXXXXXX [CR]) set and reset the address stop conditions. An AS, IS, RS, or WS output indicates that a specific address stop condition has been set up. These conditions correspond to address stop, instruction stop, operand read stop, and operand write stop respectively, An instruction read stop command causes the CPU to halt and output ST only when an instruction fetch occurs at an address that was previously loaded into the address compare register. Operand read and write stop commands cause the CPU to halt if a read or write occurs at the address previously loaded into the address compare register. ST is outputted to the CRT with every CPU halt condition. All address stop conditions cause the CPU to halt and output ST to the CRT terminal.

The IS [CR] command only relates to an address stop on an instruction fetch operation. The RS [CR] command only relates to an address stop on a read operand operation. The WS [CR] command relates to an address stop on a write operand operation. The AS [CR] command relates to all of the above conditions.

To set an address stop condition, the operator must decide which type of address stop command to use. The appropriate command is then inputted with XXXXXXXX being the actual program stop address. Inputting AS, IS, RS, or WS, followed immediately by a [CR], clears the address stop command. Memory address C bits are ignored and truncated by the SCP. Note that when an address match occurs a stop condition also occurs. When this condition occurs, the address stop condition is not cleared. If that address should occur again, on the address bus, the CPU would halt. Also note that it is not possible to set up a general address stop condition (AS=XXXXXXXX[CR]) and then clear only a particular stop condition (i.e., IS[CR]) without clearing all address stops. Then an address stop condition has been set up, either AS, RS, IS or WS appears in the CRT display field. The actual stop condition is indicated by ST in the CRT display.

3.3.5 Clear Memory (CLE[CR])

The CLE[CR] command clears memory, starting at address 0000 and continuing until the end of the first contiguous block of memory addresses. The end of this operation is signalled by the SCP prompt (//). The purpose of this command is to clear parity errors so that the IPL operation can be performed.

Table 3-2.
Alphabetical List of SCP Commands

@@A	Attention (when in console mode)
@@C	Enter console mode
@@P	Enter panel mode (when in console mode)
AS[CR]	Clear address stop
AS=XXXXXXXX[CR]	Set address stop
CLE[CR]	Clear memory
[CONTROL][H]	Deletes last character inputted
CRMA=XXXX[CR]	Address CRAM
CRMD=XXXXXXXXXXXX[CR]	Load CRAM with XXXXXXXXXXXX
=XXXXXXXXXXXX[CR]	*** Load CRAM with data and increment address
CS[CR]	Read control switches
CS=XXXXXXXX[CR]	Set control switches
EA[CR]	Read effective address
EXEC[CR]	Execute CRAM
GPR[CR]	Read general purpose registers
GPRA=XXXXXXXX[CR]	Write general purpose register A
HALT[CR]	Halt
IPL[CR]	IPL from default address
IPL=XXXX[CR]	IPL from XXXX
IS[CR]	Clear instruction stop
IS=XXXXXXXX[CR]	Set instruction stop
[LF]	Repeat command (except RST)
MA=XXXXXX[CR]	Read physical memory address location
=[CR]	*** Increment and read memory address
MAV=XXXXXX[CR]	Read virtual memory address location
=[CR]	*** Increment and read memory address
MD=XXXXXXXX[CR]	Write memory data
=XXXXXXX[CR]	Increment and write memory data
=[CR]	*** Increment and write previous data
MSG[CR]	Message
OVR[CR]	Toggle clock override
PC=XXXXXX[CR]	Load program counter
PRIP[CR]	Set primary panel (master terminal only)
PSD[CR]	Read program status doubleword (PSD1 and PSD2)
PSD=XXXXXXXX[CR]	Write program status word (PSD2)
PSW[CR]	Read program status word (PSD1)
PSW=XXXXXXXX[CR]	Write program status word (PSD1)
RS[CR]	Clear read operand stop
RS=XXXXXXXX[CR]	Set read operand stop
RST[CR]	Reset
RUN[CR]	Run
SECP[CR]	Set secondary panel (master and slave terminals)
[SHIFT][DEL]	Deletes entire command line (Hazeltine terminal only)
STEP[CR]	Instruction step
=[CR]	*** Instruction step
WS[CR]	Clear write operand stop
WS=XXXXXXXX[CR]	Set write operand stop

NOTE

[CR] denotes carriage return following the command.
*** denotes a continuation of the current command.

3.3.6 Control RAM (CRAM) Memory Address (CRMA=XXXX[CR])

This command preaddresses the CRAM for a subsequent write into a CRAM location. The SCP prompt is the only output.

3.3.7 CRAM Memory Write (CRMD=XXXXXXXXXXXX[CR]; =XXXXXXXXXXXX[CR])

This command writes XXXXXXXXXXX into the CRAM memory address previously selected by the CRAM command. Note that there must be no intervening commands if the CRAM memory write command is to be executed. The SCP prompt is the only output.

By executing this command the CRAM address is incremented so that the next data word may be inputted. Thus after the first CRMD=XXXXXXXXXXXX[CR] command has been inputted, =XXXXXXXXXXXX[CR] may be used for subsequent CRAM loads to subsequent addresses.

3.3.8 Read Control Switches (CS[CR]; CS=XXXXXXXX[CR])

These commands read from and write into the dedicated control switches memory location. The memory address and contents of the control switches are displayed after each operation.

3.3.9 Rub Out (ASCII 08)

The backspace character (ASCII 08) is used when the system is in the SCP mode to move the cursor back one space and delete the last inputted character.

The combination CONTROL and H keys accomplishes the backspace and delete operation on the Hazeltine and KSR-43 terminals.

3.3.10 Effective Address (EA[CR])

The EA[CR] command requests the effective address of the instruction currently addressed by the program counter (PC). The effective address is outputted in the second number field of the display line following EA.

3.3.11 Execute CRAM (EXEC[CR])

The EXEC command causes the microprogram counter to be loaded with the first address of CRAM so that whatever program that was loaded into it can be executed.

3.3.12 General Purpose Registers (GPR[CR]; GPRA=XXXXXXXX[CR])

These commands read the GPRs as a group and write into them individually. The first GPR command causes a read of the eight GPRs. Their contents are outputted in order, on a single line following the command (starting with GPR 0, on the left, and going through register 7). An eight-digit hexadecimal output is obtained from each GPR. To write into GPR A (A being 0 through 7), the operator must input GPRA=XXXXXXXX[CR] The prompt (//) is the only output.

3.3.13 HALT CR

Inputting HALT CR while the CPU is in the run mode causes the CPU to enter the halt mode. In the halt mode, the CPU no longer executes instructions from memory. Instead, the CPU is placed in a microroutine that monitors selected SCP support functions. If the CPU is already in the halt mode and the halt command is inputted, the CPU remains in the halt mode. After the halt command is inputted, the CPU should enact a series of bus transfers to the SCP, to pass the information used to output a line indicating the current PSW, the instruction (at the PC address), and the current machine states.

3.3.14 Initial Program Load (IPL CR ; IPL=XXXX CR)

Inputting IPL=XXXX CR when the CPU is in the halt mode causes it to go into the IPL mode. The IPL mode initiates a microprogram loading sequence so that data can be read from device number XXXX. Inputting only IPL CR causes the CPU to use a default address for the IPL device. After the first IPL, from address XXXX is completed, an input of IPL CR causes another IPL from the previously specified address (XXXX). This function is not supported by the IPU Console IOP.

3.3.15 Line Feed LF

Inputting LF , after a command has been executed by the SCP, causes the SCP to reexecute the same command unless the command was a reset.

3.3.16 Memory Address Read (MA=XXXXXX CR ; CR)

The MA=XXXXXX CR command causes a read of memory address XXXXXX, with the address and data displayed on the line following the command. After a memory address read command is executed, a CR causes the memory address to be incremented, and the next location to be read. The MA=XXXXXX CR instruction is used to set up the address so that a subsequent memory write operation can be initiated. Memory address C bits are ignored and truncated by the SCP.

3.3.17 Memory Address Read Virtual (MAV=XXXXXX CR ; CR)

The MAV=XXXXXX CR command requests the physical address of a logical address from the CPU. After this address is obtained, a memory read operation of that location is performed. The physical address and the data is displayed on the line following the command. If one-to-one mapping is used, the physical address and the logical address are the same. After the memory address read virtual command is inputted into the CPU, a CR causes the logical address to be incremented, and a read, of the corresponding physical address to be performed. Memory address C bits are ignored and truncated by the SCP.

3.3.18 Memory Write Data (MD=XXXXXXXX CR ; =XXXXXXXX CR ; CR)

These commands write data into the memory address previously set up by the memory read commands. Note that in these MD commands the physical address must be used. Data XXXXXXXX is written into the specified address and the contents are read back and displayed. After a memory write operation is executed, it is only necessary to input =XXXXXXXX CR to write new data into the next memory address. The =XXXXXXXX CR command is actually an increment and write operation. If the same data is to be written into the next location only a single CR is required.

3.3.19 Message (MSGE CR)

The MSGE CR command places the ports into the interoperator message mode, allowing the operators to communicate with each other via the master and slave ports. Either operator may enter a message (note that the characters inputted from one port also appear at the other port). Single carriage returns are permitted in the character stream. To exit the MSGE mode, two successive CRs are required.

3.3.20 Clock Override (OVR CR)

The OVR CR command toggles the clock override state. Activating the clock override state causes a blocking of interrupts from the real time clock or interval timer. If the OVR command activates the clock override state, OVR is displayed on the CRT. If it clears the clock override state, OVR is removed from the display. The IPU Console IOP does not support OVR.

3.3.21 Load Program Counter (PC=XXXXXX CR)

This command fetches the PSW and replaces the PC portion (six bits) with the new PC value. The new PSW is then sent back to the CPU. Only the SCP prompt is outputted to the CRT.

3.3.22 Primary and Secondary Panel (PRIP CR and SECP CR)

These commands deactivate or activate the secondary panel at the slave port. At system reset, the ports come up in the SCP and secondary panel modes (either port is a viable SCP). To disallow the slave port from being used as an SCP, it is necessary to input PRIP CR to either port. In the primary panel mode, the master port is activated as a SCP. The slave port is disabled when the ports are in the SCP mode. Note that the PRIP CR command doesn't affect the operation of the slave port operating in the operator console mode.

3.3.23 Program Status Doubleword (PSD CR ; PSD=XXXXXXXX CR)

These commands read from and write into the second word of the PSD. The PSD CR command causes a read of both words. These words are then outputted in the two number fields of the display line. Note that the PC value outputted in the first word of the PSD is a virtual address. The PSD=XXXXXXXX CR command writes the new PC value into word two of the PSD. Only the SCP prompt is outputted to the CRT.

3.3.24 Program Status Word (PSW CR ; PSW=XXXXXXXX CR)

These commands reads from and write into the PSW (word one of the PSD). During a PSW read operation, the CPU is requested to furnish the physical address of the PC portion of the PSW. This address is then used to fetch the next executable instruction. A display of the PSW, together with the instruction, is then sent to the CRT. Note that the physical address does not appear in the PC part of the PSW; however, the virtual address does. These conditions will be the same for one-to-one mapping. The PSW write command requests the CPU to change the PSW to XXXXXXXX. A write PC command also exists if the operator only wants to change the contents of the PC. Only the SCP prompt is outputted during PSW write commands.

3.3.25 Reset (RST[CR])

The RST[CR] command sends a reset signal to the CPU causing a system reset to occur. After the reset has occurred, the CPU sends the SCP PSW information which is then outputted by the SCP. This action indicates that the CPU is on line and operational. Inputting RST[CR] when the system is in the halt mode initializes all appropriate logic in all of the SelBUS devices.

3.3.26 RUN[CR]

The RUN[CR] command requests that the CPU be placed into the run mode if it is currently in the halt mode. The RUN[CR] command also causes the ports to be placed into the operator console mode after a display line, with all zeros in the number fields, has been outputted. If the CPU halts again, after the run command is issued, the new PSW is outputted to the CRT. If the halt comes before the SCP has had time to print the display line for the run command, the display line is aborted and the PSW line is outputted.

Inputting RUN[CR] to an active port with the CPU in the halt mode causes the CPU to enter the run mode and begin executing instructions from the location specified in the PSW. If the CPU is already in the run mode (RUN is displayed on the CRT) it continues to operate in the run mode. After a run command is inputted via the SCP, a display line is outputted to the SCP. This display line contains all zeros in the number fields. The current state indicators are also displayed.

3.3.27 [SHIFT][DEL] (ASCII 7F)

This ASCII character causes the entire last line displayed on the CRT terminal to be deleted, and a new SCP prompt to be outputted. This action corresponds to keying shift delete on the Hazeltine terminal. The operator should use the delete key on the KSR-43 terminal.

3.3.28 STEP[CR]

The STEP[CR] command causes the CPU to execute the next sequential instruction. After this action occurs the system switches to the operator console mode.

CHAPTER 4

SOFTWARE PROGRAMMING

4.1 Input/Output Processing (IOP) Programming

The IOP is programmed using extended I/O protocol. Extended I/O is the collective name for several separate, but interrelated, specifications. The overall object of these specifications is to provide a flexible and economical, but standardized, mechanism for attaching and controlling the different input/output devices of a 32 SERIES computer system.

The objectives of extended I/O are to:

1. Provide a common software perspective towards various I/O devices through standardized I/O instructions, IOP commands, and status presentations.

2. Define standardized IOP programs which allow an IOP or integrated IOP/controller to perform:

   Command chaining
   Data chaining
   Program controlled interrupt
   Transfer in channel operations

3. Define an IOP which executes standardized programs and is connected to I/O device controllers in a conventional way.

4. Allow I/O commands to directly address up to 16-million bytes of main memory.

The extended I/O specification defines the following:

1. 32 SERIES I/O instructions including format, action, and condition codes.

2. I/O commands executed by an IOP to form an IOP program.

3. Interrupts and status sent to the central processing unit (CPU) by the I/O system.

4. Operation of the IOP.

5. CPU to IOP communications over the SelBUS.

Extended I/O does not attempt to standardize the following:

1. The interface (cabling, electrical, timing) between the device controller and its devices.

2. The way in which the general classes of I/O commands (read, write, control, and sense) are customized to the particular characteristics of specified devices.

4.1.1 Software Environment

The extended I/O specification was conceived with careful considerations to software usage. The following assumptions were made about the software for extended I/O.

1. Interrupt driven I/O.
2. Software priority dispatching.
3. Device dependent routines in data management and error recovery only.
4. The software interrupt control instruction usage is minimal or nonexistent.
5. The start I/O instruction is the predominant I/O instruction and all other I/O instructions are low usage.

In order to fully understand the software usage assumption, an understanding of the intended software structure is required. Figure 4-1 shows the intended software structure.

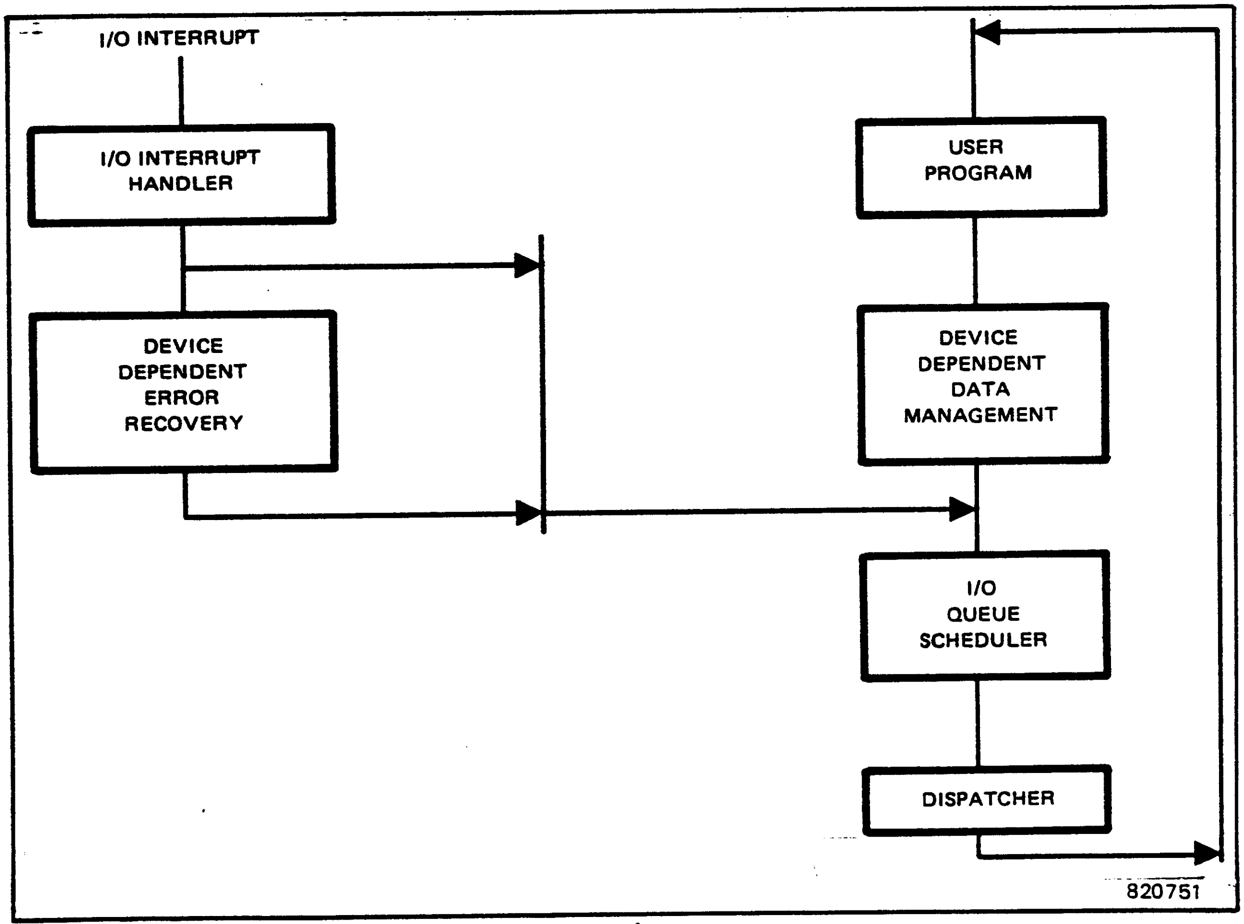


Figure 4-1. Extended I/O Software Structure

The user program, in most cases, never issues an I/O request directly to the I/O device. The use requests deal primarily with high level requests, such as reading or writing a record.

Generally, the user program's primary responsibility is to compute. Therefore, the users program does not concern itself with the mechanics needed to read or write the required information to or from the I/O device.

Device dependent data management is the interface between the user program and the operating system. Data management transforms the high level user request into a form acceptable by the operating system and the I/O device. In addition, data management also validates and correctly positions the media and causes the data buffers to be emptied and the media removed at program termination.

The I/O queue scheduler places all I/O requests into a queue on a prioritized basis. The queue scheduler attempts to optimize I/O device activity.

The dispatcher controls the scheduling of all user program execution. When multiple user programs require execution, the dispatcher establishes a priority so that the highest user receives control. The duration that the user program is given control is dictated by the user program or an external event called an interrupt. Interrupts can pertain to I/O, clock, or real-time requests.

The I/O interrupt handler receives control when an interrupt occurs. The facilities shown in the status words provide the capability to rapidly determine whether control must be transferred to the device dependent error recovery circuit or to the I/O queue scheduler.

The device dependent error recovery circuit performs analysis of the error information and attempts to retry a failing command in order to determine whether the error is recoverable or permanent. Unique I/O device characteristics prevent the common analysis and recovery attempt in a single common process.

The utilization of the IOP requires close cooperation between the software programmer and the IOP designer. Special requirements may appear unachievable with extended I/O unless the users are familiar with the intent of this specification. This specification in no way restricts the interpretation of the input/output command doubleword (IOCD) by the IOP. The IOCD data address is not solely intended as data transfered to an I/O device. It can also be considered as control or parameter information to the IOP and/or controller and its interpretation is the responsibility of the IOP.

Another example is the presentation of status. The requirements for this specification are that at least one type of I/O termination interrupt and status information be sent to the software. The number of other interrupts or the information sent in no way implies that the users of IOPs are restricted to a single interrupt per I/O initiation. The only basic restriction is the interpretation of the interrupt by the CPU firmware for post program controlled or I/O termination interrupts. These interrupts determine the number of status words that must be exchanged.

4.2 Input/Output System Component

I/O operations involve the transfer of information between an I/O device and main memory of a computer system. Figure 4-2 shows the basic components of the computer system.

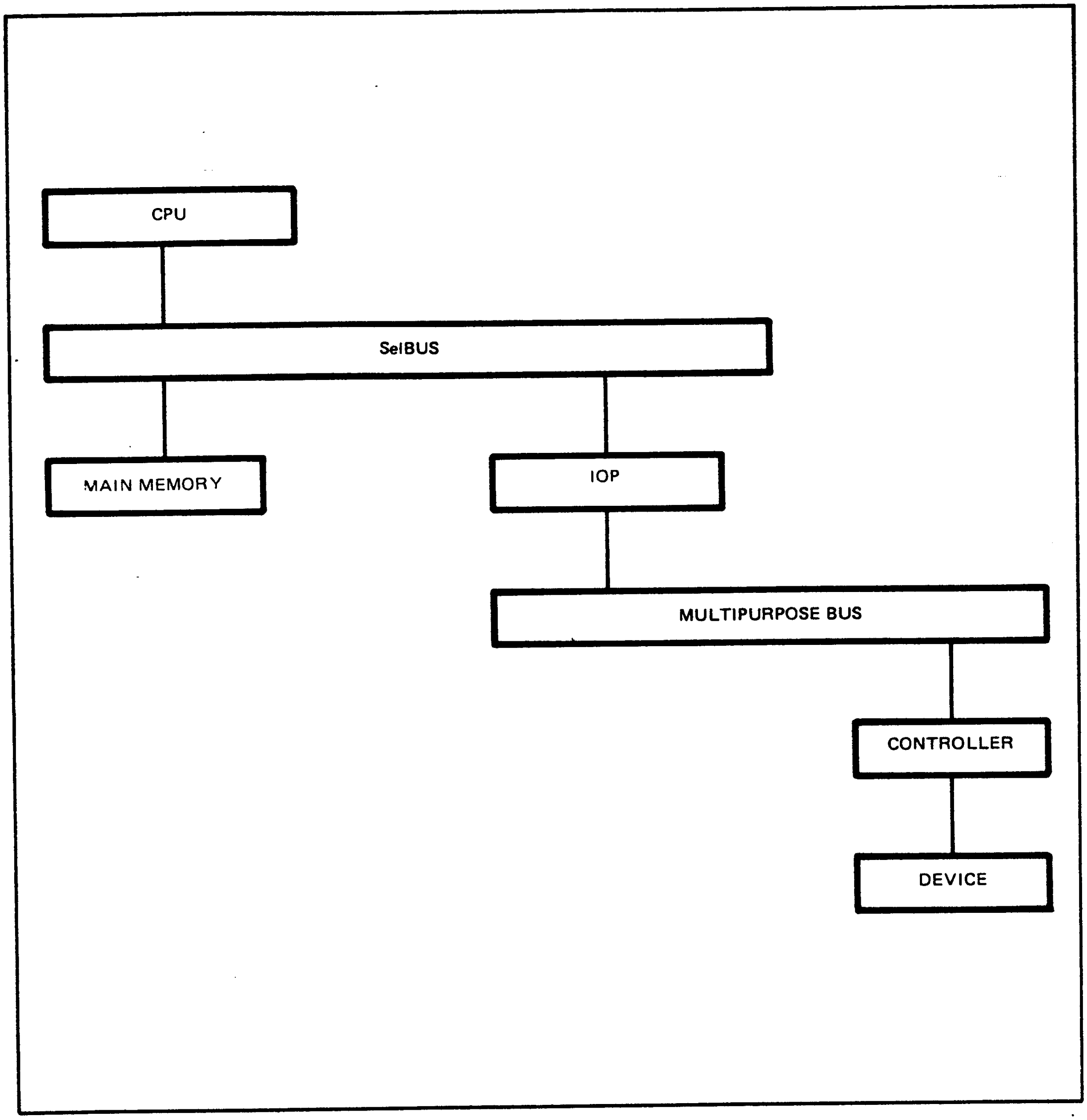


Figure 4-2. Basic Components of a Computer System

The I/O devices include card readers, line printers, discs, magnetic tapes, and telecommunication equipment. The functions of an I/O device are regulated by the controller. The controller, which is attached to an IOP, provides the logical and buffering capabilities necessary to operate an I/O device.

The IOP's function is to schedule the requests for main memory between a number of controllers. The IOP also connects the controller to the CPU to initiate or terminate an I/O operation.

The IOP controls up to 16 independent controllers located some distance from the IOP.

4.2.1 Input/Output Processors

There is currently a capability for 16 IOPs associated with the 32 SERIES computer system. Each IOP provides an independent interface to the SelBUS and is associated with a single I/O interrupt level. Dedicated memory locations are also associated with each IOP and provide main memory locations to transmit or receive control information required to initiate or terminate an I/O operation. The control information is as follows:

1. Service interrupt vector (SIV)
2. Input/output command list address (IOCLA)
3. Status address service interrupt control area (SICA)
4. New program status doubleword (PSD)
5. Old PSD

4.2.1.1 Input/Output Memory Addressing

Memory addresses are transferred to the IOP when a start I/O instruction is executed by the CPU. Before the execution of the I/O instruction, the software must store the address of the first IOCD. This IOCD is placed into the word indicated by adding 16 decimal to the contents of the SIV. This word is the IOCLA.

The memory addressing method selected for extended I/O is real addressing. Real addressing is the capability to directly address any memory location within the 16M byte maximum capacity of the 32 SERIES computer system without any address translation. This method of addressing differs from the addressing method normally used by the software programmer. The software programmer relies on a hardware address conversion to transform the 20-bit logical address into a 24-bit real address in order to address memory locations greater than 512K byte.

4.2.1.2 Extended I/O Instructions

Extended I/O includes the usage of the following instructions:

Start I/O (SIO)
Test I/O (TIO)
Halt I/O (HIO)
Enable channel interrupt (ECI)
Disable channel interrupt (DCI)
Activate channel interrupt (ACI)
Deactivate channel interrupt (DACI)
Reset channel (RSCHNL)
Stop I/O (STPIO)
Reset controller (RSCTL)

For all extended I/O instructions the constant in bits 16 through 31 of the instruction, plus the contents of the general register indicated by bits 6 through 8 of the instruction (if bits 6 through 8 are nonzero), specify the logical IOP and subaddress. The IOP ignors the subaddress for operations that pertain only to the IOP.

4.2.1.2.1 Start I/O

The SIO instruction initiates an I/O operation.

4.2.1.2.2 Test I/O

The TIO instruction interrogates the current state of the IOP and may be used to clear pending interrupt status for the next device with status pending in FIFO queue. The TIO instruction can be used as an alternate to successive interrupts when status for more than one device is available at the time of the interrupt.

4.2.1.2.3 Halt I/O

The HIO instruction immediately terminates a subchannel and device operation. Pending status if any, will be returned. In any case, a device clear indication is sent to the device resulting in status being generated.

4.2.1.2.4 Enable Channel Interrupt

The ECI instruction allows the IOP to request interrupts from the CPU.

4.2.1.2.5 Disable Channel Interrupt

The DCI instruction prohibits the IOP from requesting an interrupt. Pending status conditions can only be cleared by the execution of a SIO or TIO instruction if the IOP is disabled.

4.2.1.2.6 Activate Channel Interrupt

The ACI instruction causes the IOP to actively contend for interrupt priority except that the IOP never requests an interrupt. This instruction has no effect on pending status conditions except that they can only be cleared by a SIO or TIO instruction.

4.2.1.2.7 Deactivate Channel Interrupt

The DCI instruction causes the IOP to suspend contention for interrupt priority. If an interrupt request was queued, the IOP may now request an interrupt if it is enabled.

4.2.1.2.8 Reset Channel

The RSCHNL instruction resets all activity in the IOP associated wih its I/O controllers. All requesting and pending conditions are cleared. All controllers are reset by this instruction. RTOM and panel functions are not affected.

4.2.1.2.9 Stop I/O

The STPIO instruction terminates the operation in the controller after the completion of the current IOCD. The termination is orderly. The IOP suppresses command and data chaining.

4.2.1.2.10 Reset Controller

The RSCTL instruction immediately resets a specific controller regardless of its previous condition. Pending status is cleared for all devices and an interrupt is not generated.

4.2.1.3 I/O Initiation

An I/O operation is initiated by a SIO instruction. If the specified IOP/subchannel is not busy, the SIO is accepted (and the CPU continues to the next sequential instruction). The IOP/controller asynchronously processes the I/O request specified by the instruction.

4.2.1.4 Input/Output Command List Address

Successful execution of the SIO instruction causes the CPU to transmit the IOCLA to the IOP. The IOCLA is found in main memory at locations specified by the service interrupt vector plus 16. The format of the IOCLA is shown in Figure 4-3. Each of the 16 IOPs has a corresponding service interrupt vector. The real IOCLA is passed to the IOP on the data bus (DBUS).

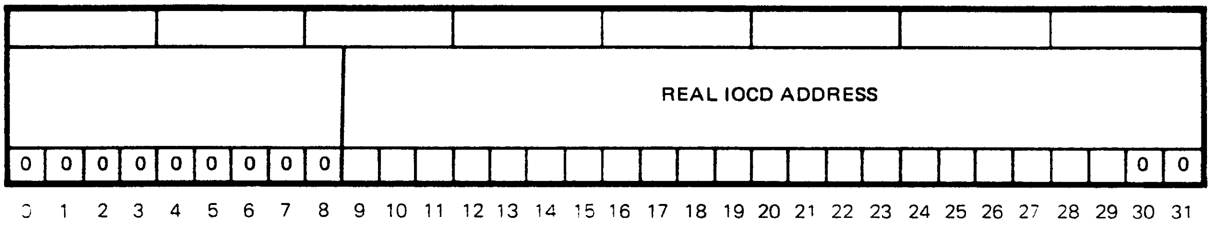


Figure 4-3. IOCLA Format

4.2.1.5 Input/Output Command Doubleword

The IOCLA instruction specifies the address of the IOCL that is to be executed by the IOP. An IOCL consists of one or more IOCDs. Each IOCD must be aligned on a fullword boundary. The IOCD format is shown in Figure 4-4.

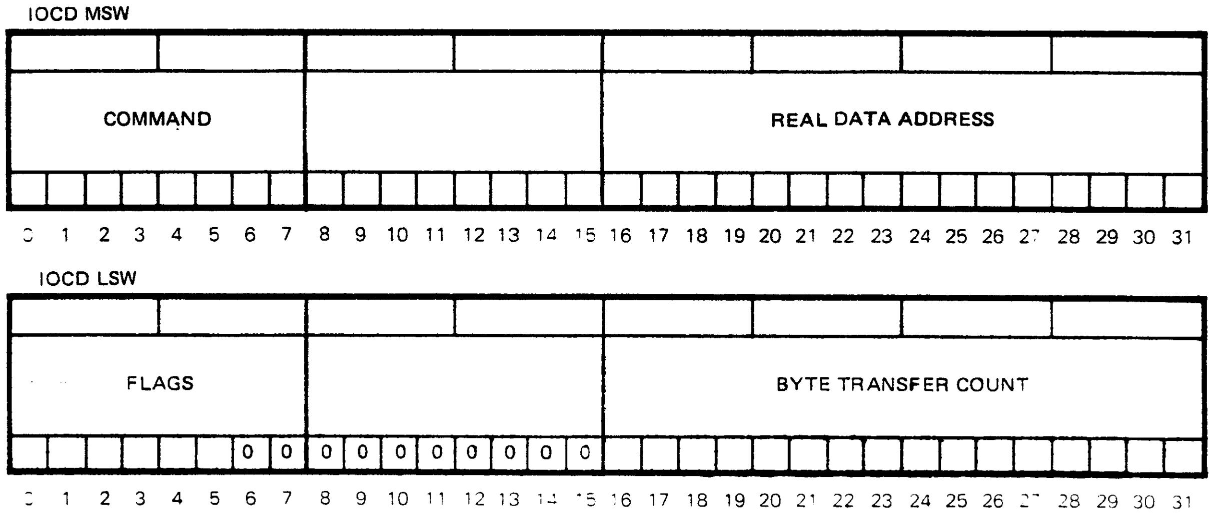


Figure 4-4. IOCD Format

The command field specifies one of seven commands:

Write
Read
Read backward
Control
Sense
Transfer in channel
Channel control

If more than one IOCD is specified, the IOCDs are fetched sequentially, except when a transfer in channel (TIC) command is specified.

The real data address specifies the starting address of the data area. The real data address is a 24-bit address. The IOP internally aligns the information transferred to or from main memory. While any starting address is allowable, more efficient system operation results if the software programmer aligns the data area to start at a fullword boundary (bits 30 and 31 of the address being zero).

The byte transfer count specifies the number of bytes that are to be transferred to or from main memory. Since the IOP may transfer data to/from memory one, two, or four bytes at a time, it accomodates counts which are not a multiple of its natural transfer width.

4.2.2 I/O Commands

4.2.2.1 Execution of I/O Commands

The IOP can execute seven commands: write, read, read backward, control, sense, transfer in channel, and channel control. Each command except transfer in channel and channel control, initiates a corresponding I/O operation. The term I/O operation refers to activity initiated by a command in the controller/device. The operation in the device lasts from the initiation of the SIO instruction until the device end time.

4.2.2.1.1 Write

The write command causes a write (output) operation to the selected I/O device from the specified main memory address.

4.2.2.1.2 Read

The read command causes a read (input) operation from the selected I/O device to the specified main memory address.

4.2.2.1.3 Read Backward

The read backward command causes a read (input) operation from the selected I/O device to the specified main memory address in decending order.

4.2.2.1.4 Control

The control commands cause control information to be passed to the selected I/O device. The control command may provide a data address and byte count for additional control information that may be stored in main memory. Control commands with no associated data to be transmitted to the device may have a byte transfer count of zero or the suppress incorrect length flag might be set.

Control information is device dependent and may instruct a magnetic tape to rewind or a printer to space a certain number of lines.

4.2.2.1.5 Sense

The sense command causes the storing of controller/device information to the specified location of main memory. One or more bytes of information are transferred depending upon the device. The sense information provides additional device dependent information not provided in the status flags.

4.2.2.1.6 Transfer In Channel

The TIC command specifies the address of the next IOCD that is to be executed. The TIC command allows the programmer to change the sequence of the IOCDs executed. A TIC command may not specify another TIC command as the next IOCD.

4.2.2.1.7 Channel Control

The channel control command causes the transfer of information to or from a specific location in main memory. One or more bytes of information is transmitted or received from the IOP. Channel control sends the required information to initialize the IOP.

4.2.3 I/O Termination

An I/O operation terminates when the IOP, controller and/or device indicates the end of an operation. All I/O operations accepted by the IOP always terminate with at least one termination status command being sent to the software.

An I/O operation can also fail to be accepted by the IOP during I/O initiation. Conditions that prevent I/O initiation are

1. IOP or subchannel busy
2. IOP not operational or nonexistent
3. I/O operation not defined for IOP

I/O initiation failures are reported to the software by setting the condition codes and, where appicable, storing the status.

4.2.4 I/O Status Words

The status words are maintained and stored by the IOP. The address of the status words is transmitted to the CPU when an interrupt is acknowledged or when another I/O instruction is executed. An interrupt or TIO instruction causes the storing of the next queued device status from a FIFO queue. A SIO or HIO instruction will always cause the storing of any pending status for the device specified in the SIO or HIO instruction. The status buffer address is sent to the IOP with a channel control command during IOP initialization. The status words contain information relating to the execution of the last IOCD or from any asynchronous condition requiring software notification (i.e., tape loaded, disc pack mounted). The status words have the format shown in Figure 4-5.

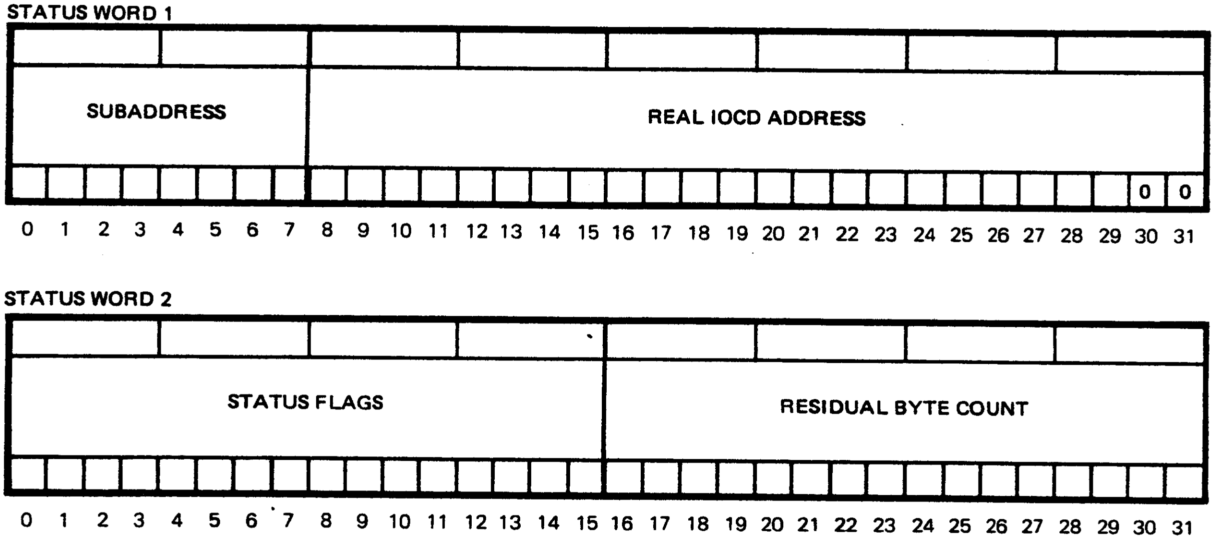


Figure 4-5. I/O Status Words Format

The status flags contain termination information pertaining to both the IOP and controller.

The address of the status is stored in main memory and can be located by adding 20 to the contents of the SIV.

4.2.5 I/O Interrupts

I/O interrupts can be caused by the termination of an I/O operation, by operator intervention at the I/O device, or when a post program controlled interrupt is requested by an IOCD. The associated I/O interrupt causes the status address and the current PSD to be stored into the SICA specified by the SIV. (It also causes the new PSD that is loaded from the SICA).

An I/O interrupt can be initiated by the IOP. If an IOP has multiple I/O interrupt requests pending, it establishes a priority sequence for them before initiating an I/O interrupt request to the CPU. This priority sequence is maintained when the IOP stores the status and reports the status address to the CPU.

4.2.6 Blocking of Data

Data recording by an I/O device may be divided into blocks. A block of data is the amount of information recorded between adjacent starting and stopping points of the device. The starting and stopping of the device may be mechanical or electronic. The length of the block depends on the external element, such as a card, a printed line, information recorded between gaps on a magnetic tape or disc.

Except for the disc and other complex controllers, the maximum amount of information that can be transferred in one I/O transfer operation is one block. An I/O operation is terminated when the associated main memory area is completely filled or the end of the block is reached (for single operations). For some operations, such as on magnetic tape, block lengths are not defined and the amount of information transfered is controlled only by the IOCD. Disc controllers may transfer more than one block of information in response to a single IOCD.

4.2.7 Input/Output Command List Address

The IOCLA specifies the real address of the first IOCLA associated with a SIO instruction. The information is transferred to the IOP with the contents of the IOP unaffected by the I/O operation. After the I/O instruction is executed the software is free to modify the IOCLA. Only the SIO instruction causes the IOCLA to be transferred to the IOP.

4.3 I/O Operations

Transfer of information to and from main memory, other than to or from the CPU, is referred to as an I/O operation. I/O devices perform the I/O operation using controllers attached to the SelBUS via an IOP.

The following discussion describes the control of an I/O device by the controller, IOP, and CPU. This section also provides detailed information about the I/O system.

4.3.1 I/O Devices

I/O devices provide a means of communication between the computer system and an external media. Examples of external media are punched cards, printed forms, magnetic tape, and discs.

I/O devices include: card readers, line printers, magnetic tapes, direct access devices (discs), and communication devices.

Most I/O devices deal directly with the external media. Other I/O devices consist of purely electronic equipment and do not directly control the media. A communications device handles the exchange of information over transmission lines.

An I/O device is attached to a controller; the controller in turn, is connected to the IOP.

4.3.1.1 Controllers

The controller provides the logical capabilities necessary to operate and control an I/O device.

All communications between the IOP and controllers occur over the MPB. The controller receives control information from the IOP, controls timing, provides data buffers, validates the command and converts the I/O command to a form acceptable by the I/O devices.

Controllers also contain the necessary electronics to interface to the IOP and the I/O device.

Each controller has the capabiity to recognize the addresses(es) of the device(s) to which it is attached.

4.3.1.1.1 Types of Controller

There are three types of controllers: a single device, a multiplexing, and a multidevice controller (MDC). The single device controller is dedicated to a single device. The multiplexing controller services several devices in such a way as to allow complete concurrent operation of all its devices. The MDC services several devices; however, it is only able to service one device at a time. Such a controller appears busy when service is simultaneously requested of a second device.

4.3.1.2 Input/Output Processor

The IOP is an intermediary between the I/O device controller and CPU/memory. Executing IOP programs allows the CPU to proceed concurrently with I/O operations.

The IOP interfaces on one side to the SelBUS and thus indirectly to the CPU and main memory. On the other side, the IOP interfaces to the MPB to which device controllers are attached.

The IOP receives requests over the SelBUS from the CPU as a result of I/O instructions. The IOP initiates CPU interrupt/status transfers back across the SelBUS to indicate I/O completion or exceptional conditions.

4.3.1.3 Subchannels

The resources of the IOP required to support IOP program execution on behalf of a single device are referred to as a subchannel. The subchannel information, IOCD address, byte transfer count, data address, status, etc., is accessed using the eight-bit device or subaddress number. During SelBUS transactions toward the IOP, the subaddress becomes the low order eight lines of the 24-SelBUS address lines. During I/O MPB transactions, toward the IOP, the subaddress (device address) is the source address of the transaction.

A subchannel is busy from the time a start I/0 (SIO) instruction is initiated, until a device end status condition is received from the controller.

4.3.2 System Operation

I/O operations are initiated and controlled by information with two types of formats: instruction and commands. Instructions are decoded by the CPU and are part of the CPU program. Commands are decoded and executed by the IOP and I/O device controller and initiate the I/O operation. One or more commands arranged for sequential execution from an IOCL. Both instructions and commands are fetched from main memory and are common to all types of I/O devices, although modifier bits in the command may specify device-dependent conditions for the execution of a data transfer operation at the device.

The CPU program initiates I/O operations via a SIO instruction. This instruction, plus the contents of the general register specified by the instruction (if nonzero), identifies the logical IOP and starting subaddress. The logical IOP address is used to vector to the appropriate device entry in local storage (scratchpad). The device entry contains the information required to convert the logical IOP address to a real IOP address. The interrupt level in the device is used to vector to the main memory location which contains the address of the service interrupt control area (SICA) associated with the IOP.

The address of the IOCL can be located by adding 16 decimal to the contents of the service interrupt vector (SIV) location. This address is called the input/output command list address (IOCLA).

A 24-bit address word is created with the interrupt level and IOP address specified in the device entry. The subaddress is specified by the instruction plus the contents of the general register.

The CPU transfers the requested transaction via the software. The IOP may or may not accept the transfer by the CPU.

If the I/O operation was successfully initiated, the IOP uses the real IOCD address, specified in the SIO data word, to fetch the first IOCD.

The command may specify an operation that does not require any data transfers with the I/O device. The controller or device may signal the end of the operation immediately upon receipt of the command.

The command may specify an operation that involves data transfer to one area of main memory, specified in a single IOCD, or to a number of noncontiguous main memory locations. In the latter case a command list is used for the operation and each IOCD specifies a noncontiguous main memory location. The IOCDs are coupled by data chaining.

Data chaining is specified by a flag in the IOCD and causes the IOP to fetch the next IOCD when the byte transfer count in the current IOCD reaches zero. The I/O device is not aware that a new IOCD has been fetched. Provisions are made to notify the software, using a post program controlled interrupt (PPCI), that the operation has progressed to a particular IOCD in the command list.

Termination of an I/O operation is normally indicated by two conditions: channel end and device end. The channel end condition indicates that the I/O device has received or provided all information associated with the operation and no longer requires the IOP or controller. The device end condition indicates that the I/O device has terminated execution of the operation. The device end condition can occur concurrently with channel end or at any time in the future.

Operations that keep the controller busy after reporting channel end may, under certain conditions, cause a third condition. This condition called controller end, may occur after the channel end to indicate that the controller has become available for initiation of another operation.

A software I/O interrupt is used to identify a terminated I/O operation condition. If the interrupts are disabled, the execution of another I/O instruction is required. In either case, these conditions cause the storing of the status words. The status words contain the information concerning the execution of the I/O operation. At the time the channel end condition is generated, the IOP sends the software the subaddress and the address of the next IOCD. It also provides its residule byte count, thus indicating the extent of main memory used. The IOP, controller, and device can provide indications of unusual conditions with channel end. Controller end and device end conditions can also be accompanied by error indications from the device.

Facilities are provided for the software to initiate execution of a chain of I/O operations with a single SIO instruction. When chaining flags in the current IOCD specify command chaining, and no unusual conditions have been detected in the operation, the receipt of the device end signal causes the IOP to fetch a new IOCD and initiate a new command to the device. A chained command is initiated by the IOP with the same set of signals, over the IOP to the controller interface as the first command specified by the SIO instruction. The ending conditions occuring at the termination of an operation when the IOCD specifies command chaining are not reported to software when another operation is initiated by command chaining. Therefore, the IOP continues execution of the current IOCD. If, however, an unusual condition occurs, the ending conditions cause suppression of command chaining and terminates the command list.

Conditions that initiate an I/O interruption are asynchronous to the activity in the CPU. More than one condition can occur at the same time. The interrupt priority logic in the CPU and IOP establish a priority among the conditions so that only one interrupt is processed at a time. These conditions must be preserved in the IOP, subchannel, controller, or I/O device until they are accepted by the CPU.

Execution of an I/O operation or chain of operations involve up to four levels of participation. These four levels are

1. The CPU is busy until the SIO data word is accepted by the IOP.
2. The IOP remains busy until acceptance of the I/O operation from the CPU.
3. The controller may remain busy after the channel end condition is reported and may generate the controller end when it becomes free.
4. The I/O Device is busy from the initiation of the first command until the device end condition associated with the last command is accepted and cleared by the CPU.

A pending device end condition causes the I/O device to appear busy, but it does not affect the status of any other part of the system.

4.4 I/O Instruction Formats

The following discussion defines the formats used for various types of controlling information. These formats apply to all I/O operations. The formats are independent of the type of I/O device, its speed, or mode of operation, and they include provisions for unique device dependent functions to an I/O device.

4.4.1 Initial Program Load

The initial program load (IPL) operation initializes main memory with a user program. The IPL operation depends on an operator stimulus (i.e., panel IPL pushbutton). The presumed state of the CPU upon IPL initiation is as follows:

1. The CPU is halted.
2. The SYSTEM RESET pushbutton is depressed.
3. The operator has specified a physical IPL device address.
4. The IPL pushbutton is depressed.

The IPL operation excludes the initial configuration load (ICL) instruction. The ICL instruction becomes a software initiated function upon completion of the IPL. Software capabilities for configuration load are provided by two new macroinstructions: read and write scratchpad.

The CPU firmware for the extended I/O devices generates a single IOCD. This IOCD is located in main memory locations 0 through 7. The contents of the IOCD are as follows:

COMMAND								DATA ADDRESS																							
0	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	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

FLAGS																BYTE TRANSFER COUNT															
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	1	1	1	1	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 contain a read command (hexadecimal 2) that can be designed by the controller to be an IPL read or a read command suitable for core image reads.

Bits 8 through 31 are the address of the data that is to be read.

Bits 32 through 37 contain flag bits that specify command chain and cuppress incorrect length.

Bits 48 through 63 contain a byte transfer count of 10 decimal.

The IPL that is read is sent to the appropriate IOP. The 120 bytes read into memory are in the following format:

0 (hexadecimal) New IPL program status doubleword (PSD)
8 (hexadecimal) IOCD1
10 (hexadecimal) IOCD2

The new PSD contains the PSD image that is loaded upon receipt of the IPL termination interrupt. IOCD1 and IOCD2 contain the necessary commands to sustain the program load. Note that the initial IPL IOCD had specified command chaining and thus the next IOCD to be executed is IOCD1. The IOCD list from the device can be initialized to read as many records as the program requires.

While the IOP continues executing the commnd list, the CPU recursively tests byte locations 4 through 7 for their contents to change from hexadecimal 60000078. The non-hexadecimal 60000078 conditon indicates that the IOP has stored the IPLs new PSD. The CPU implies an enable channel interrupt command to the IOP when an IPL command is issued. This action indicates that the IOP can store status. The testing of locations 4 through 7 is required because the IOP may store the final status for the IPL I/O operation into locations 0 through 7 if a subsequent IOCD is not an initialize channel command. The initialize channel command establishes a new status location.

When the I/O interrupt occurs, the CPU transfers control to the software that is indicated by the software via locations 0 through 7.

4.4.2 Extended I/O Instructions

All extended I/O instructions are in the following format:

OP CODE	R	SUB OP	A/C	CONSTANT
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 operation (OP) code, bits 0 through 5, and the augment code (A/C), bits 13 through 15, must contain ones. The register (R) field, bits 6 through 8, if nonzero, specifies the general register whose contents are added to the constant field (bits 16 through 31) to form the logical IOP and subaddress. If R is specified as zero, only the constant field is used. The format of the computed logical IOP and subaddress is shown below:

	LOGICAL IOP	SUBADDRESS
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

The subaddress is ignored by the IOP if the operation does not apply to a controller or device.

The sub-op field (bits 9 through 12) specifies the type of operation that is to be performed.

The extended I/O instruction must be executed while the SYSTEMS 32 SERIES computer is operating in the privileged mode. This operating condition is obtainable by executing a mode set instruction.

Bits 9-12	Sub-Op
0 0 0 0 - X'0'	Unassigned
0 0 0 1 - X'1'	Unassigned
0 0 1 0 - X'2'	Start I/O (SIO)
0 0 1 1 - X'3'	Test I/O (TIO)
0 1 0 0 - X'4'	Stop I/O (STPIO)
0 1 0 1 - X'5'	Reset channel (RSCHNL)
0 1 1 0 - X'6'	Halt I/O (HIO)
0 1 1 1 - X'7'	Not supported

Bits 9-14	Sub-Op
1 0 0 0 - X'8'	Reset controller (RSCTL)
1 0 0 1 - X'9'	Not supported
1 0 1 0 - X'A'	Unassigned
1 0 1 1 - X'B'	Not supported
1 1 0 0 - X'C'	Enable channel interrupt (ECI)
1 1 0 1 - X'D'	Disable channel interrupt (DIC)
1 1 1 0 - X'E'	Activate channel interrupt (ACI)
1 1 1 1 - X'F'	Deactivate channel interrupt (DACI)

If the execution of an extended I/O instruction is requested in the nonprivileged mode, a privileged violation trap occurs.

Condition codes are set for the execution of all extended I/O instructions. These codes indicate the successful or unsuccessful initiation of an I/O instruction. The Condition Codes can be set by the CPU for IOP busy and inoperable or undefined IOP, or by the information passed by the IOP. The assignments for the condition codes are as follows:

Condition Codes				Assignment
CC1	CC2	CC3	CC4	
0	0	0	0	Request activate, will echo status not sent by the IOP
0	0	0	1	IOP busy
0	0	1	0	IOP inoperable or undefined
0	0	1	1	Subchannel busy
0	1	0	0	Status stored
0	1	0	1	Unsupported transaction
0	1	1	0	Unassigned
0	1	1	1	Unassigned
1	0	0	0	Request accepted/queued, no echo status
1	0	0	1	Unassigned
1	0	1	0	Unassigned
1	0	1	1	Unassigned
1	1	0	0	Unassigned
1	1	0	1	Unassigned
1	1	1	0	Unassigned
1	1	1	1	Unassigned

Although 16 encoded conditions are possible, only the assigned patterns occur.

4.4.2.1 Start I/O

The SIO instruction begins an execution or returns the appropriate condition codes and status if I/O execution was not accomplshed.

OP CODE						R			SIO				A/C			CONSTANT															
1	1	1	1	1	1				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

Bits 0 through 5 (octal 77) specify the operation.

Bits 6 through 8, when nonzero, specify the general register whose contents are added to the constant to form the logical IOP and subaddress.

Bits 9 through 12 (hexadecimal 2) specify the operation as a SIO.

Bits 16 through 31 specify a contant that is added to the contents of R to form the logical IOP and subaddress. If R is zero, only the constant is used to specify the logical IOP and subaddress.

The SIO instruction can fail to be accepted by the IOP if any of the following conditions exist:

1. The IOP or Subchannel is busy.
2. The IOP is not operational or not assigned.

An SIO, accepted by the IOP (condition code or 1000), does not always indicate that the I/O operation has been successfully initiated.

Pending status conditions are cleared by the IOP if they are reported to the SIO as status stored conditions (condition code 0100). This condition implies that the new IOCLA has been accepted.

4.4.2.2 Test I/O

The TIO instruction tests the controller's state so that it can return the appropriate condition codes and status reflecting the state of the next queued device status.

OP CODE						R			TIO				A/C			CONSTANT															
1	1	1	1	1	1				0	0	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 5 (octal 77) specify the operation code.

Bits 6 through 8, when nonzero, specify the general register whose contents are added to the constant to form the logical IOP and subaddress.

Bits 9 through 12 (hexadecimal 3) specify the operation as a TIO.

Bits 13 through 15 (octal 7) specify the augment code.

Bits 16 through 31 specify the constant that is added to the contents of R to form the logical IOP and subaddress. If R is zero only the constant is used to specify the logical IOP and subadderss. This field is ignored for the TIO instruction.

4.4.2.3 Stop I/O

The STPIO instruction terminates the current I/O operation after the completion of the current IOCD. The STPIO instruction applies only to the addressed subchannel. Its only function is to suppress command and data chain flags in the current IOCD.

OP CODE						R			STPIO				A/C			CONSTANT															
1	1	1	1	1	1				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

Bits 0 through 5 (octal 77) specify the operation code.

Bits 6 through 8, when nonzero, specify the general register whose contents are added to the constant to form the logical IOP and subaddress.

Bits 9 through 12 (hexadecimal 4) specify the operation as STPIO.

Bits 13 through 15 (octal 7) specify the augment code.

Bits 16 through 31 specify a constant that is added to the contents of R to form the logical IOP and subaddress. If R is zero, only the constant is used to specify the logical IOP and subaddress.

4.4.2.4 Halt I/O

The HIO instruction causes an immediate, but orderly, termination in the controller. The device end condition notifies the software of the actual termination in the controller. The controller then indicates its availability for new requests. If the HIO instruction caused the generation of status relating to the terminated I/O operation, the device end condition, for the termination of the I/O operation, is the only device end condition generated. A subchannel busy indication is reported via condition codes until the controller returns a device end indication.

OP CODE						R			HIO				A/C			CONSTANT															
1	1	1	1	1	1				0	1	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

Bits 0 through 5 (octal 77) specify the operation code.

Bits 6 through 8, when nonzero, specify the general register whose contents are added to the constant to form the logical IOP and subaddress.

Bits 9 through 12 (hexadecimal 6) specify the operation as a HIO.

Bits 13 through 15 (octal 7) specify the augment code.

Bits 16 through 31 specify a constant that is added to the contents of R to form the logical IOP and subaddress. If R is zero, only the constant is used to specify the logical IOP and subaddress.

4.4.2.5 Reset Channel

The RSCHNL instruction ceases and resets all activity on the addressed IOP and returns the IOP to the idle state. The IOP also resets all subchannels. Any requesting or active interrupt level is reset. This command has no effect on RTOM or panel functions.

OP CODE	R	RSCHL	A/C	CONSTANT
1 1 1 1 1 1		0 1 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

Bits 0 through 5 (octal 77) specify the operation code.

Bits 6 through 8, when nonzero, specify the general register whose contents are added to the constant to form the logical IOP and subaddress.

Bits 9 through 12 (hexadecimal 5) specify the operation as a RSCHNL.

Bits 13 through 15 (octal 7) specify the augment code.

Bits 16 through 31 specify a constant that is added to the contents of R to form the logical IOP and subaddress. If R is zero, only the constant is used to specify the logical IOP and subaddress.

4.4.2.6 Reset Controller

The RSCTL instruction completely resets the addressed controller. The subchannel is also cleared and all pending and generatedstatus conditions are cleared.

OP CODE	R	RSCTL	A/C	CONSTANT
1 1 1 1 1 1		1 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 5 (octal 77) specify the operation code.

Bits 6 through 8, when nonzero, specify the general register whose contents are added to the constant to form the logical IOP and subaddress.

Bits 9 through 12 (hexadecimal 8) specify the operation as a RSCTL.

Bits 13 through 15 (octal 7) specify the augment code.

Bits 15 through 31 specify a constant that is added to the contents of R to form the logical IOP and subaddress. If R is zero, only the constant is used to specify the logical IOP and subaddress.

4.4.2.7 Enable Channel Interrupt

The ECI instruction enables the addressed IOP so that it can request interrupts from the CPU.

OP CODE	R	ECI	A/C	CONSTANT
1 1 1 1 1 1		1 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

Bits 0 through 5 (octal 77) specify the operation code.

Bits 6 through 8, when nonzero, specify the general register whose contents are added to the constant to form the logical IOP and subaddress.

Bits 9 through 12 (hexadecimal C) specify the operation as ECI.

Bits 13 through 15 (octal 7) specify the augment code.

Bits 16 through 31 specify a constant that is added to the contents of R to form the logical IOP and subaddress. If R is zero, only the constant is used to specify the logical IOP and subaddress.

4.4.2.8 Disable Channel Interrupt

The DCI instruction disables the addressed IOP from requesting interrupts from the CPU.

OP CODE	R	DCI	A/C	CONSTANT
1 1 1 1 1 1		1 1 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

Bits 0 through 5 (octal 77) specify the operation code.

Bits 6 through 8, when nonzero, specify general register whose contents are added to the constant to form the logical IOP and subaddress.

Bits 9 through 12 (hexadecimal D) specify the operation as a DCI.

Bits 13 through 15 (octal 7) specify the augment code.

Bits 16 through 31 specify a constant that is added to the contents of R to form the logical IOP and subaddress. If R is zero, only the constant is used to specify the logical IOP and subaddress.

4.4.2.9 Activate Channel Interrupt

The ACI instruction causes the addressed IOP to begin actively contending with other interrupt levels. This action causes a blocking of its level and all lower priority levels from requesting an interrupt. If a request for interrupt is currently pending in the IOP, the requested interrupt is removed; however, the interrupt level remains in contention.

OP CODE	R	ACI	A/C	CONSTANT
1 1 1 1 1 1		1 1 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

Bits 0 through 05 (octal 77) specify the operation code.

Bits 6 through 8, when nonzero, specify the general register whose contents are added to the constant to form the logical IOP and subaddress.

Bits 9 through 12 (hexadecimal E) specify the operation as an ACI.

Bits 13 through 15 (octal 7) specify the augment code.

Bits 16 through 31 specify a constant that is added to the contents of R to form the logical IOP and subaddress. If R is zero, only the constant is used to specify the logical IOP and subaddress.

4.4.2.10 Deactivate Channel Interrupt

The DACI instruction removes the addressed IOPs interrupt level from contention. If a request for an interrupt instruction is currently queued, the DACI instruction causes the queued request to actively request for an enabled IOP.

OP CODE	R	DACI	A/C	CONSTANT
1 1 1 1 1 1		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 5 (octal 77) specify the operation code.

Bits 6 through 8, when nonzero, specify the general register whose contents are added to the constant to form the logical IOP and subaddress.

Bits 9 through 12 (hexadecimal F) specify the operation as a DACI.

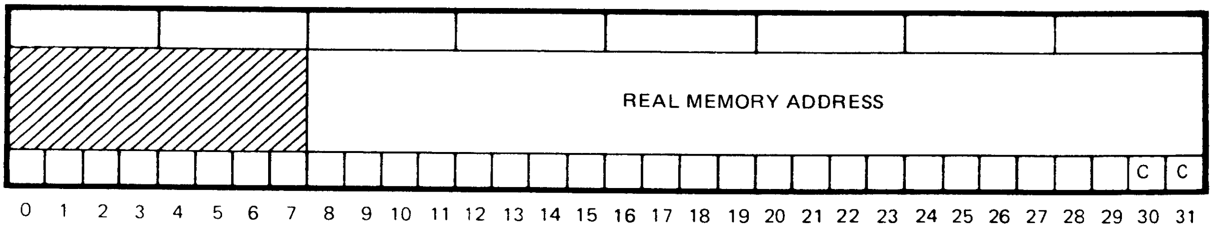
Bits 13 through 15 (octal 7) specify the augment code.

Bits 16 through 31 specify a constant that is added to the contents of R to form the logical IOP and subaddress. If R is zero, only the contant is used to specify the logical IOP and subaddress.

The deactivated capability is also provided by the branch and reset interrupt (BRI) instruction. The execution of the DACI and BRI instructions imply that the IOP is currently active.

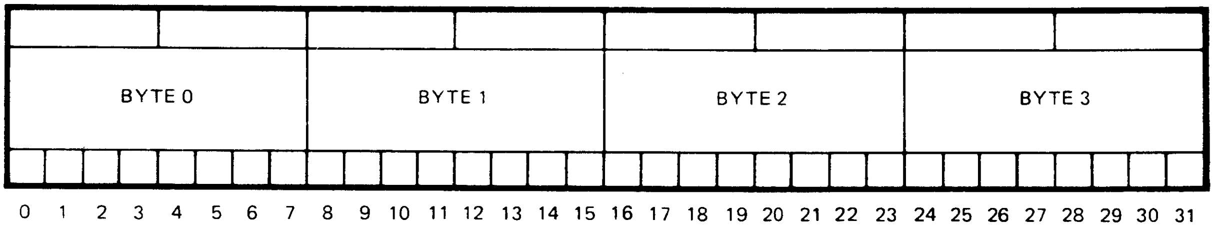
4.4.3 Memory Addressing

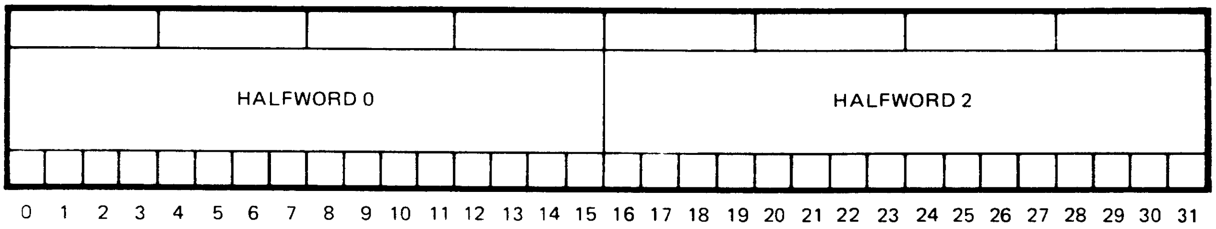
All memory addresses sent to the IOP are real addesses. Real addressing provides the capability to directly address any location within a 16M byte memory system. The format of the real memory address is shown below:

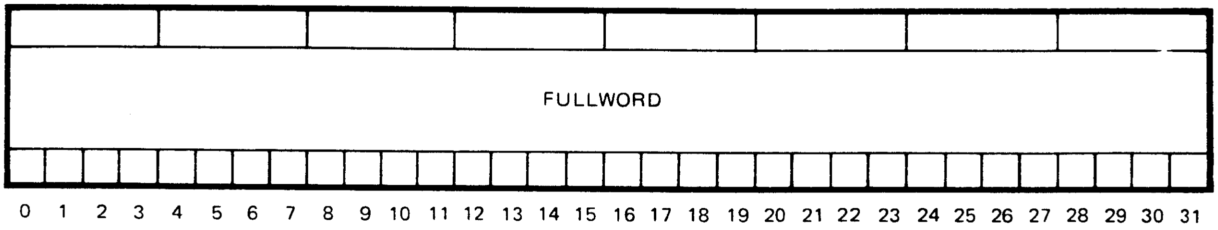


Bits 8 through 29 specify the fullword (32 bits) location in main memory.

Bits 30 through 31 (the C bits) specify the location within the fullword as shown below. Note that the real memory address must be positioned on the proper boundry when class F devices are used. For example, a disc is a word device; therefore, the IOCD must specify aword boundary if the disc is being used.





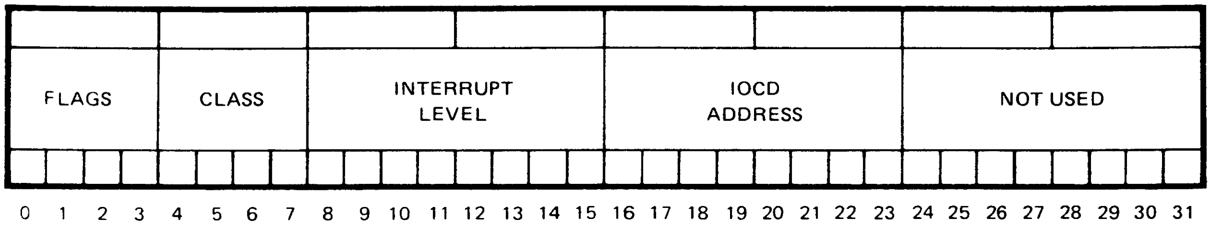


830590

4.4.4 Device Entry

The logical IOP address, specified in the I/O instruction, is converted to an actual IOP and/or controller address. The logical IOP address is capable of specifing 128 logical IOPs (0 through 127). An appropriate entry is placed in local store (scratchpad) for each of the 128 local IOPs. This entry is initialized at IPL, and is called a device entry.

The device entry contains the information required to associate the device with an interrupt level and a physical IOP and controller address. The format of the device entry is shown below:



830591

Bits 0 through 3 are not used with extended I/O devices.

Bits 4 through 7 identify the firmware classification that resides in the IOP or controller. A hexadecimal E indicates nonextended I/O and hexadecimal F indicates extended I/O. The IPU Console IOP uses class 6 and 7 for its I/O modes and requires the IOCD address to contain bit 22 and not bit 21.

Bits 8 through 15 identify the I/O interrupt level of the entry.

Bits 16 through 23 identify either the actual IOP and controller address, for the integrated IOP/controller, or only the IOP.

4.4.5 I/O Command Doubleword

The IOCD specifies the command being executed. For commands initiating I/O operations, the IOCD designates the main memory area associated with the operation and action to be taken whenever transfer to or from the area is completed. More than one IOCD can be associated with a single SIO instruction. The IOP refers to each IOCD only once to store the pertinent information in the IOP.

The first SIO instruction causes the first IOCD to be fetched. Each additional sequential IOCD is fetched as the operation progresses to the point where an additional IOCD is needed. Fetching of the IOCD does not affect the contents of the location in main memory.

4.4.5.1 Chaining

When the IOP has performed the transfer of information specified by the IOCD, it can continue the activities initiated by the SIO instruction by fetching the next IOCD. The fetching of the next IOCD is referred to as chaining. The IOCDs belonging to such a sequence are said to be chained.

Chaining only occurs between IOCDs located in sequential addresses where the next IOCD is obtained by adding eight to the address of the current IOCD. Two chained Ds in noncontiguous main memory locations can be coupled, for chaining purposes, with a TIC IOCD instruction. All IOCDs in the chain refer to the I/O device specified in the SIO instruction.

Two types of chaining are provided: data and command. Data and command chaining are specified by flags in the IOCD. The flags specify the action that is to be performed when the current byte transfer count is reduced to zero and the receipt of ending status from the device. The flags are not disturbed by a TIC command.

Figure 4-6 shows the actions taken at zero count of the device end command.

CD	CC	SIL	CNT=O,NOT EOB		CNT=0,DE	CNT/=0,DE
0	0	0	STOP,END,IL		END	END,IL
0	0	1	STOP,END		END	END
0	1	0	STOP,END,IL		CCC	END,IL
0	1	1	STOP,CCC		CCC	CCC
1	0	0	CCD	DNA	END,IL	
1	0	1	CCD	DNA	END,IL	
1	1	0	CCD	DNA	END,IL	
1	1	1	CCD	DNA	END,IL	

CD CHAIN DATA
CC COMMAND CHAIN
SIL SUPPRESS INCORRECT LENGTH
CNT=0 BYTE TRANSFER COUNT REDUCED TO ZERO
NOTEOB NOT END OF BLOCK AT BEVICE
DE DEVICE END
CNT/=0 BYTE TRANSFER COUNT NOT REUCED TO ZERO
STOP DEVICE ORDERES TO STOP DATA TRANSFER
IL INCORRECT LENGTH
END OPERATION ERMINATED
CCC CONTINUE, COMMAND CHAIN
CCD CONTINUE CHAIN DATA
DNA DOES NOT APPLY, THIS CONDITION CANNOT EXIST UNLESS A BYTE TRANSFER CONT OF ZERO IS SPECIFIED IN THE IOCD. BY THE TIME THE IOP END INDICATION IS RECEIVED BY THE IOP, IT HAS ALREADY FETCHED THE NEXT IOCD.

NOTE

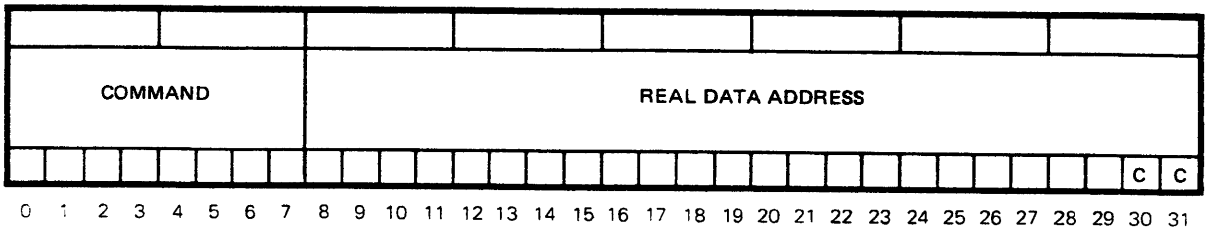
If data chaining is specified, the SIL flag of the IOCD is not effected.

Figure 4-6 Device End Command Actions

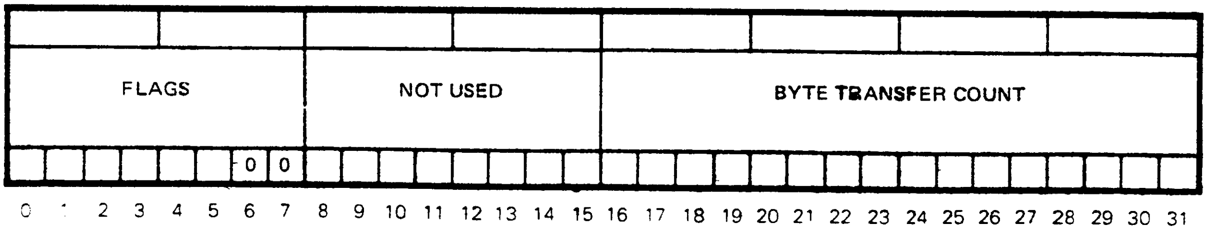
4.4.5.2 IOCD Format

The IOCD is located in paired words on word boundaries and is in the following format:

IOCD MSW



IOCD LSW



Bits 0 through 7 of IOCD word 1 (command) specify to the IOP and controller, the operation to be performed. The IOP distinguishes between the following operations:

Output (write or control)
Input forward (read or sense)
Input backward (read backward)
Branching (transfer in channel)

Bit assignments in the command are as follows:

X X X X 0 0 0 0	Channel control
M M M M 0 1 0 0	Sense
X X X X 1 0 0 0	Transfer in channel
M M M M 1 1 0 0	Read backward
M M M M M M 0 1	Write
M M M M M M 1 0	Read
M M M M M M 1 1	Control

Note: X = Don't care, M = Modifier bits

Bits 8 through 31 of IOCD word 1 specify the real data address in main memory.

Bits 0 through 7 of IOCD word 2 specify additional control information required at the termination of the current IOCD. The individual flags are as follows:

Data chain: Bit 0 (LSW) specifies data chaining and the fact that the real data address, flags and byte transfer count of the next IOCD, is to be used to continue the operation when the current byte transfer count is reduced to zero.

Command chain: Bit 1 (LSW) specifies that, upon normal termination of the current command, the next IOCD is to be initiated.

Suppress incorrect length: Bit 2 (LSW) specifies when both the command chain and suppress incorrect length flags are set. The command chaining procedure progresses regardless of the presence of the incorrect length condition.

Skip: Bit 3 (LSW) specifies that no data is to be transferread to main memory during the read, read backward, or sense operation.

Post program controlled interrupt: Bit 4 (LSW) causes the IOP to interrupt the CPU when the byte transfer count is zero, if data chaining is specified or at the device end if command chaining is specified.

Bits 16 through 31 of IOCD word 2 (byte transfer count) specify the number of transfers that are required by the IOCD. This requirement is dictated by each device. A byte transfer count of zero, appearing in any IOCD, (except TIC) is invalid and the I/O operation is terminated.

4.4.5.2.1 Data Chain

During data chaining the new IOCD, fetched by the IOP, defines a new real data address and byte transfer count for the old operation. Execution of the operation at the controller/device is not affected and the controller/device is unaware that a new IOCD has been fetched. Data chaining only occurs when all data, specified by the current IOCD, is transferred to or from the device. This action causes the operation to continue using the data area and count specified in the new IOCD. The contents of the command field are ignored unless a TIC command is specified. Data chaining increases the possibility of overrun conditions and degrades IOP performance; therefore, discretion should be used when data chaining is requested.

Data chaining occurs immediately after the last byte of data is transferred to or from the controller/device, as specified by the current IOCD. When the last byte is placed in main storage or accepted by the controller/device, the new IOCD controls the operation and replaces the pertinent information in the IOP. If the controller/device sends an channel end command, after exhausting the count of the current IOCD but before tansferring any data to or from the memory area designated by the new IOCD, the status associated with the termination pertains to the new IOCD.

If errors are detected in the new IOCD an error indication is generated and the device is signaled to terminate the operation. The termination indication occurs when the device attempts to transfer data designated by the new IOCD. If the controller/device signals the channel end condition before transferring any data (designated by the new IOCD) a IOP program check is indicated in the status associated with the termination. Unless the new IOCD address is invalid or programming errors are detected in any intervening TIC command, the contents of the status pertains to the new IOCD.

A data address referring to a nonexistant area causes an error indication only after the controller/device has attempted to transfer data to or from the invalid location.

4.4.5.2.2 Command Chain

During a command chaining operation the new IOCD, fetched by the IOP, specifies a new I/O operation. The IOP fetches the new IOCD and initiates the new operation upon receipt of the device end signal for the current operation. When command chaining occurs the completion of the current operation does not cause an I/O interruption or status to be stored. The count indicating the amount of data transferred during the current operation is not available to the software. For data transfer operations, the new IOCD always applies to the next block of data at the device. Command chaining occurs and the new operation is initiated only if no unusual conditions are detected in the current operation.

The IOP initiates a new I/O operation with the command chaining flag upon receipt of a status byte. The status byte must contain only the following bit combinations:

1. Device end
2. Device end and status modifier
3. Device end and channel end
4. Device end, channel end, and status modifier

For the device end or device end and status modifier bit combination, channel end status must be generated before device end status. Also, all other status bits must be off. If the attention, unit check, unit exception, incorrect length (unless the suppress incorrect length flag is set), or the IOP program check condition is present, the new I/O operation is terminated and the status, associated with the current I/O operation causes an interrupt to occur. The next IOCD is not fetched. The incorrect length condition does not suppress command chaining if the suppress incorrect length (SIL) flag is set in the current IOCD. If data chaining is specified and an incorrect length condition exists, the SIL flag is prohibited.

Exceptions to sequential chaining of IOCDs occur when the controller/device presents the status modifier condition with a device end command. When command chaining is specified and no unusual conditiond are detected, the combination of the status modifier and device end combination cause the IOP to fetch and chain to an IOCD whose main memory address is 16 higher than that of the current IOCD.

If both command and data chaining are specified, the first IOCD associated with the operation specifies the operation as executed.

4.4.5.2.3 Suppress Incorrect Length

The SIL condition occurs when both the command chain and SIL flags are set. The command chaining procedure continues whether or not an incorrect length condition exists.

4.4.5.2.4 Skip

The skip command suppresses main memory writes during an I/O operation. This command is only defined for read, read backward, and sense operations, and is controlled by the skip flag in the IOCD. When the skip flag is set, skipping occurs. When the flag is reset, normal operation occurs. The setting of the skip flag is ignored in all other operations. Skipping affects the information handled by the IOP. The operation at the device proceeds normally and the information is transferred to the IOP. The IOP updates the count but does not send the information to main memory. If command or data chaining is specified, a new IOCD is obtained when the count reaches zero. For data chaining, normal operation is resumed if the new IOCD is not set. Checking for invalid data addresses does not occur during the skipping procedure.

4.4.5.2.5 Post Program Controlled Interrupt (PPCI)

The PPCI function allows the software to interrupt the I/O during the execution of an I/O operation. The function is controlled by the PPCI flag in the IOCD. The flag can be set in any IOCD in the list. The PPCI, when requested by the IOP, decreases IOP throughput. The true effect on IOP performance cannot be determined because of several factors, such as interrupt priority of the PPCI, the number of higher priority interrupts active or pending, and CPU/IOP communication overhead. Data overruns are possible if data chaining is also specified and the transfer rate of the device is high enough to cause the data loss. The IOP cannot service a data request when executing the acknowledge interrupt sequence.

Whenever the IOP completes the processing of an IOCD, with the PPCI bit set, it attempts to interrupt the CPU. The PPI interrupt occurs when the byte transfer count in the current IOCD reaches zero (if data chaining is specified) or at device end (if command chaining is specified). No predictable time relationship exists between the PPCI interrupt and the I/O transfer.

If chaining occurs before the PPCI interrupt is serviced, the PPCI interrupt condition is carried over into the new IOCD. This carry-over condition occurs in both data and command chaining and, in either case, the condition is propagated through a TIC command. PPCI interrupt conditions are no stacked; therefore, if another interrupt condition occurs on the same subchannel, before the previous PPCI interrupt has been acknowledged; only the last interrupt is reported.

Normally the status words are stored before the I/O operation is terminated. The status words contain only subaddresses, real IOCD addresses, and the PPCI bit in the status flags. All other fields contain zeros. The real IOCD address indicates the current IOCD position plus eight. This IOCD may not be the one that originally requested the PPCI.

If the status words are not stored by a TIO instruction or an interrupt until after the termination of the I/O operation, they contain an ending status (controller status and residual byte count) in addition to the PPCI bit. The presence of any controller status indicates the termination of the I/O operation.

4.4.5.3 Command Instructions

Figure 4-7 lists the command code for the seven command instructions. The figure also indicates the allowable flags that may be set in the IOCD. The flags are ignored for commands that are not defined.

COMMAND	BITS	FLAGS				
WRITE	MMMM MM01	CD	CC	SIL		PPCI
READ	MMMM MM10	CD	CC	SIL	SKIP	PPCI
READ BACKWARD	MMMM 1100	CD	CC	SIL	SKIP	PPCI
CONTROL	MMMM MM11	CD	CC	SIL		PPCI
SENSE	MMMM 0100	CD	CC	SIL		PPCI
TRANSFER IN CHANNEL	000 1000					
CHANNEL CONTROL	MMMM 0000		CC	SIL		

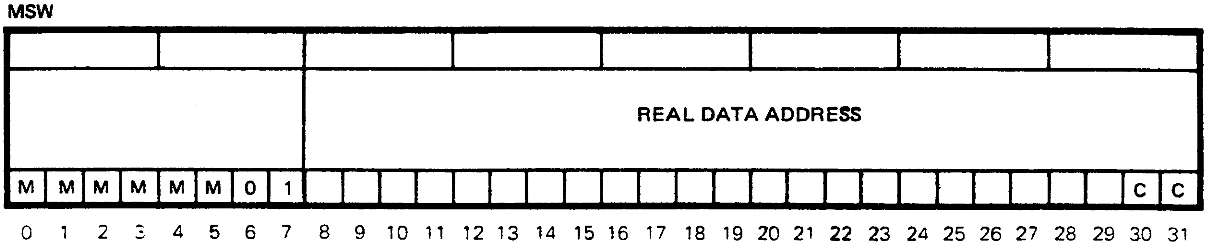
CD	CHAIN DATA
CC	CHAIN COMMAND
SIL	SUPPRESS INCORRECT LENGTH
SKIP	SUPPRESS TRANSFER TO MEMORY
PPCI	POST PROGRAM CONTROLLER INTERRUPT
M	MODIFIER BITS

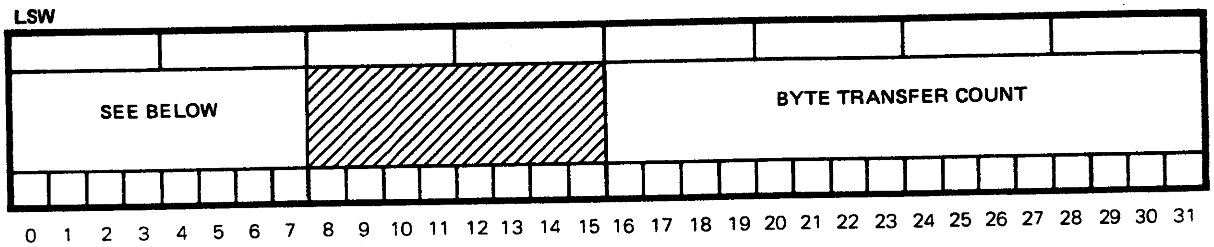
Figure 4-7. Command Code for the Seven Command Instructions

All flags have individual significance. The SIL flag is ignored on immediate operations or operations which do not require additional information from main memory when using real addresses. In these cases, the incorrect length condition is suppressed regardless of the setting of the SIL flag. Each command is described in the following discussion. Included in the discussion is an illustration of the IOCD.

4.4.5.3.1 Write

The write instruction places the controller/device in the output mode and sets up the IOP/subchannel to transfer data from main memory to the device. Data in memory is fetched in ascending order, starting with the address specified by the real data address in the IOCD.



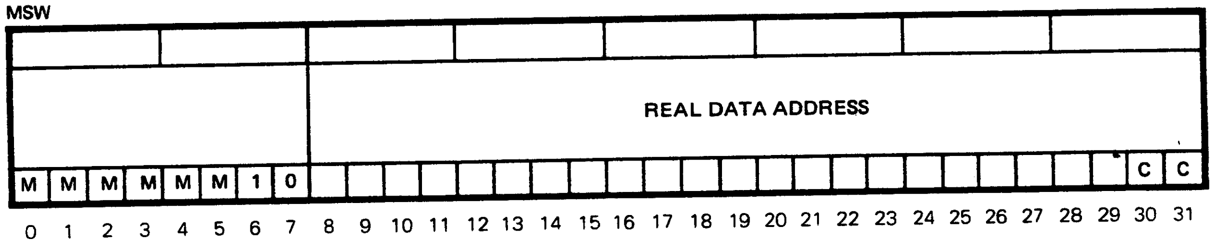


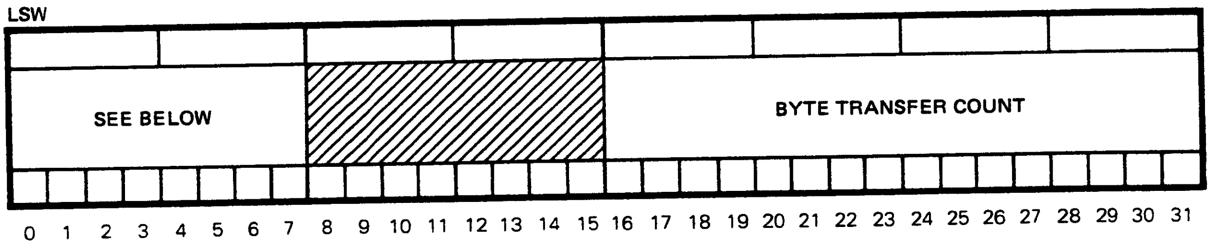
LSW	Bit 0	=	CD
	Bit 1	=	CC
	Bit 2	=	SIL
	Bit 3	=	Not used
	Bit 4	=	PPCI
	Bit 5	=	Not used
	Bits 6-7	=	0

Even though the CD and PPCI functions can be specified in the IOCD, their successful execution is device dependent. The Device data rate may preclude successful execution of these functions.

4.4.5.3.2 Read

The read instruction places the controller/device in the input mode and sets-up the IOP/subchannel to transfer data to memory from the device. Data is stored in memory in ascending order, starting with the address specified by the real data address in the IOCD.





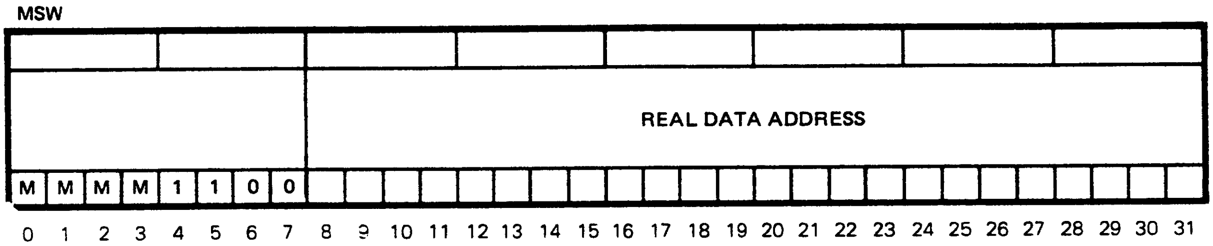
LSW	Bit 0	=	CD
	Bit 1	=	CC
	Bit 2	=	SIL
	Bit 3	=	SKIP
	Bit 4	=	PPCI
	Bit 5	=	Not used
	Bits 6-7	=	Not used

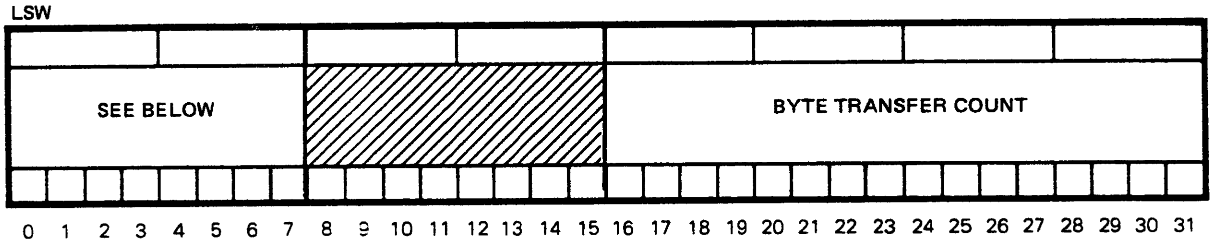
The basic read command (hexadecimal 02) is used by the IPL procedure and, therefore, it must be defined as the command that transfers data in a mode suitable for transferring instructions in executable form.

Even though the CD and PPCI functions can be specified in the IOCD, their successful execution is device dependent. Device data rates may preclude successful execution of these functions.

4.4.5.3.3 Read Backward

The read backward instruction places the controller/device in the input mode and sets-up the IOP to transfer data from the device to main memory. The devices capable of this operation transfer data in a sequence opposite to that written. Data is stored in memory in decending order, starting with the address specified by the real data address in the IOCD.





LSW	Bit 0	=	CD
	Bit 1	=	CC
	Bit 2	=	SIL
	Bit 3	=	SKIP
	Bit 4	=	PPCI
	Bit 5	=	Not used
	Bits 6-7	=	0

Even though the CD and PPCI functions can be specified in the IOCD, their successful execution is device dependent. Device data rates may preclude successful execution of these functions.

4.4.5.3.4 Control

The control instruction places the controller/device in the output mode and sets-up the IOP/subchannel to transfer data from main memory to the device. The controller/device interprets the data as control information. The control information, if required, is fetched from memory in ascending order, starting with the address specified by the real data address in the IOCD.

MSW

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
								REAL DATA ADDRESS																							
M	M	M	M	M	M	1	1																							C	C

LSW

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
SEE BELOW								▨								BYTE TRANSFER COUNT															

LSW Bit 0	=	CD
Bit 1	=	CC
Bit 2	=	SIL
Bit 3	=	Not used
Bit 4	=	PPCI
Bit 5	=	Not used
Bits 6-7	=	0

Some control functions may not require additional information from main memory and, in such cases, are classified as an immediate operation. A byte transfer count of zero should accompany commands with no data transfers.

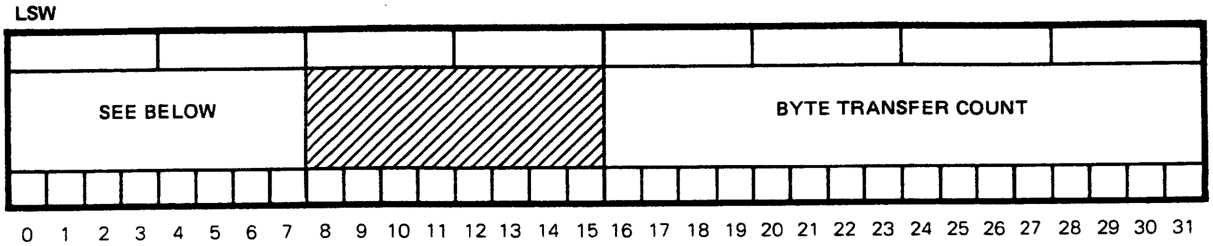
The basic control command (hexadecimal 3) is defined as a no operation (NOP). The controller responds with an immediate channel and device end with no action at the device.

4.4.5.3.5 Sense

The sense instruction places the controller/device in the input mode and sets-up the IOP/subchannel to transfer data from the controller/device to main memory. The data is placed in memory in ascending order, starting with the address specified by the real data address in the IOCD.

MSW

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
								REAL DATA ADDRESS																							
M	M	M	M	0	1	0	0																							C	C



LSW	Bit 0	=	CD
	Bit 1	=	CC
	Bit 2	=	SIL
	Bit 3	=	SKIP
	Bit 4	=	PPCI
	Bit 5	=	Not used
	Bits 6-7	=	0

Data transferred by the sense operation provides one or more bytes of information concerning both unusual conditions detected in the last operation and device status.

The sense information provided by the sense command is more detailed than that supplied by the status words status flags. The sense command may also provide descriptive information about the device that does not constitute an error condition (unit check indication). However, it is up to the particular device handler to issue a sense command regardless of the state of the unit check to retrieve his information. The following describes the assignment of the first six bits of information.

Bit	Description
0	Command reject
1	Intervention required
2	Bus out check
3	Equipment check
4	Data check
5	Overrun

Command reject: The controller/device has detected a programming error in the IOCD. The command portion of the IOCD was not supported by the controller/device, or an invalid IOCD sequence was executed.

Intervention required: The last operation was not executed because a condition at the I/O device requires operator intervention. This condition could indicate a card or line printer jam, out of paper, or not ready state.

Bus out check: The controller has received a transaction, over the I/O MPB, with invalid parity.

Equipment check: The controller/device has detected an I/O device malfunction during the last operation.

Data check: The controller/device has detected a data error. A data check identifies errors associated with the recording medium during data tranfer. These errors include conditions such as invalid card code or detecting invalid parity on data recorded on magnetic tape.

During a read operation a data check may indicate that incorrect data was placed in memory. The IOP/controller forces the correct parity when data is written in main memory. During a write operation this condition indicates that incorrect data was recorded at the device. Unless the operation is a type where errors preclude meaningful continuation, data errors do not cause the operation to terminate prematurely. The data transfer continues until the end of block command.

Overrun: The IOP has failed to respond in time to a device data service request. Overruns occur when the IOP fails to maintain the transfer rate required by the device. The overrun condition can also occur when the device receives the new command too late during command chaining. The data transfer continues until the end of block command.

Data from the controller and device can contain one or more bytes of information. The amount of information transferred is device dependent and, therefore, the appropriate device publication should be consulted.

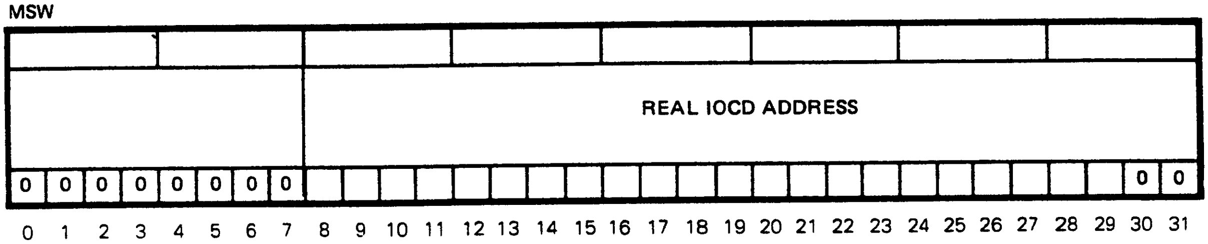
The basic sense command (hexadecimal 4) requests device status and any unusual conditions detected during the last operation. This command cannot cause command reject, intervention required, data check, or the overrun bits to be turned on. If the controller detects an equipment malfunction, the equipment check bit is set and a unit check is reported using an channel end command.

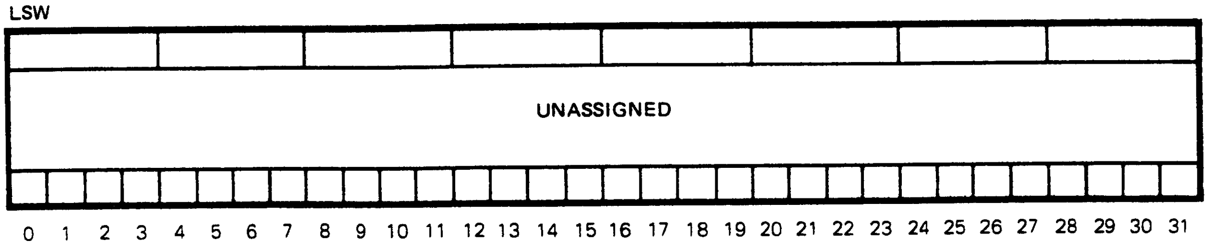
Devices that can provide diagnostic sense information or can be instructed to perform other special functions may define modifier bits for the control of these functions. Any remaining sense commands may be considered invalid, thus causing a status unit check. If a status unit check is not initiated, the basic sense command may be executed.

The sense information, pertaining to the last I/O operation, is reset following a SIO command to the controller, unless the SIO command specifies a sense or NOP command to that particular controller.

4.4.5.3.6 Transfer In Channel

The TIC instruction fetches the next IOCD from main memory at the location designated by the real IOCD address specified in the TIC IOCD. The TIC command is executed by the IOP. The controller/device is unaware of this command. The purpose of the TIC command is to provide chaining between IOCDs not located on adjacent two-word ascending addresses. The TIC command occurs during both data and command chaining. The chaining flags from the previous IOCD propagates through this command.



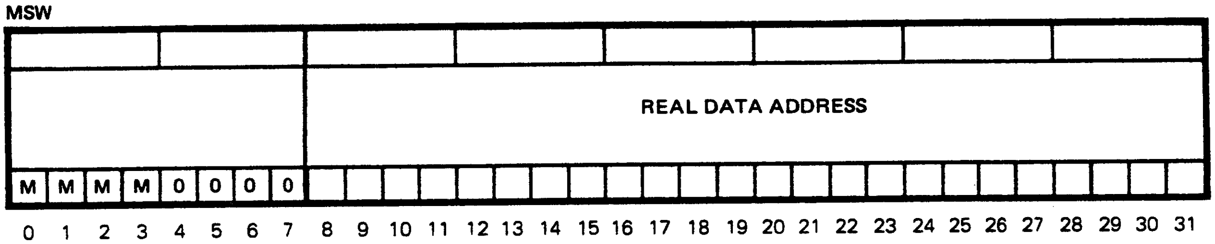


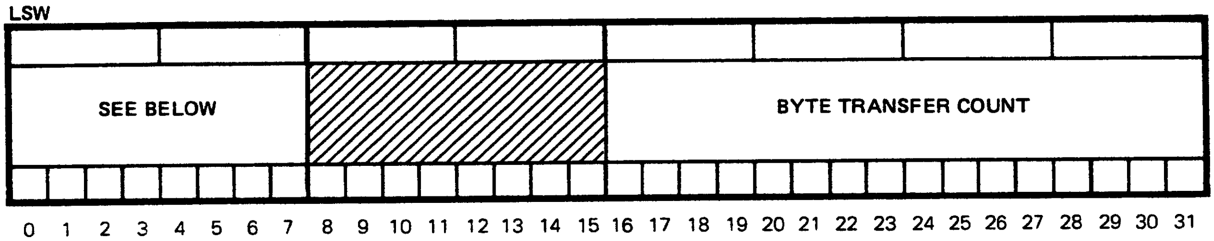
The first IOCD designated by the IOCLA can be a TIC, but a TIC IOCD cannot specify an address of another TIC IOCD. The real address is invalid if any C bits or a zero are specified (the real address must address a word). For all of the cases specified above the IOP program check indicator is set in the status word and the operation is terminated.

Although the LSW of the IOCD is unused, it must be present to correctly displace the next IOCD by eight.

4.4.5.3.7 Channel Control

The channel control instruction allows the passing of IOP initialization information to the IOP. This command is never transferred to the controller; however, it is decoded and executed entirely within the IOP.





LSW	Bit 0	=	Not used
	Bit 1	=	CC
	Bit 2	=	SIL
	Bits 3-7	=	Not used

The basic channel control (modifier bits all zero) is interpreted as a channel memory allocation (CMA). A CMA instruction assigns main memory locations that are used by the IOP for status management and storing. The memory allocated must reside in a single 512-word page or any other 512-page increment. The starting location is defined by the real data address. The byte transfer count specifies the number of bytes that are to be allocated for the specific IOP; therefore, it cannot exceed 512 decimal.

The software loses the capability to use the memory allocated to the IOP and must consider the memory allocation to only be read, since it is write protected.

4.5 Termination of I/O Operations

When the operation or sequence of operations initiated by a start I/O (SIO) command is terminated, the IOP and controller/device generate status conditions. The status conditions, along with the subaddress, IOCD address and a count, indicating the extent of the operation sequence, are sent to the memory as status words, at the time of the interrupt.

4.5.1 Type of Termination

An I/O operation at the subchannel normally lasts until the controller initiates a device end command. The device end condition can be started during the sequence initiating the operation or later into the operation. When the IOP detects equipment malfunctions or a master reset is initiated, it disconnects the controller/device without receiving a device end command. The software can force a controller/device to be disconnected prematurely using a reset channel (RSCHNL), reset controller (RSCTL), halt I/O (HIO), or stop I/O (STPIO) command.

4.5.1.1 Termination At Operation Initiation

After the addressed IOP and subchannel are verified and are in a state that a SIO command can be executed, certain tests are performed on the validity of the information specified by the software. This testing occurs during both the execution of SIO and command chaining commands.

A data transfer operation is initiated at the subchannel, controller, and device only when no IOP programming or equipment errors are detected by the IOP and when the controller responds with no unusual conditions. When an unusual condition causes a command to be rejected during initiation of an I/O operation, either by a SIO or command chaining command, an interrupt condition is generated. With this condition pending, the new operation is not started, and the subchannel is not available until the condition is cleared, either by the interrupt being serviced or by an SIO or TIO command causing the status to be stored.

4.5.1.2 Immediate Operation

An I/O operation reporting a device end condition without data transfer is called an immediate operation. The controller responds immediately after command validation with a device end instruction indicating to the subchannel that no data transfer is required. If command chaining is not specified, the condition generates an interrupt. When the device end condition is cleared, the subchannel becomes available.

When command chaining is specified, only the channel end/device end condition for the last IOCD generates an interrupt.

Whenever an immediate termination of an I/O operation is reported, no data is transferred to or from the device. The logical data address is not checked for validity.

Since zero is valid for control commands, any IOCD specifying an immediate operation should contain a zero byte transfer count. When an immediate operation is executed, the incorrect length status is not indicated to the software and command chaining, when specified, continues.

4.5.1.3 Termination Of Data Transfers

When the controller and device accept the command the subchannel is set-up for data transfer. The subchannel is said to be busy during this time period. Unless the IOP detects an equipment malfunction or a RSCHNL, RSCTL, or HIO command is issued, the busy state continues until the IOP receives the channel end condition from the controller. Unless command chaining has been specified and is not suppressed, due to receipt of a STPIO command or other unusual condition, a channel end condition causes the data transfer, at the subchannel, to be terminated and an interrupt generated. The status flags, in the associated status words, indicate a channel end and any other unusual conditions. The controller/device can signal a channel end at any time after initiation of the operation. This signal may occur before any data has been transferred.

The controller/device normally controls the timing of the channel end condition for operations not involving data transfers. The duration of the data transfer may be variable and can be controlled by the device or IOP.

Excluding equipment errors (RSCTL, RSCHNL, and HIO commands) the IOP signals the device to terminate data transfers when any of the following conditions occur:

1. The memory area becomes filled or emptied.
2. An IOP program check condition is detected.
3. A chaining check condition is detected.

The first condition occurs when the IOP has reduced the byte transfer count, in the last IOCD associated with the operation, to zero. A byte transfer count of zero indicates that the IOP has transferred all information specified by the IOCD. The other conditions are due to errors and cause premature termination of the data transfer. In either case, the termination is generated in response to a data service request from the device. This condition causes te data transfer to cease. If the controller/device has no fixed block size specified for the operation (such as writing on magnetic tape), it terminates the operation and generate the channel end condition until the end of the block is reached. This condition exists whether or not the device has been previously notified to terminate the data transfer.

When command chaining occurs, the subchannel is in the working state from the time the first operation is initiated until the controller signals the device end condition for the last operation in the chain.

Any unusual conditions cause command chaining to be suppressed and an interrupt to be generated. The unusual conditions can be detected by either the IOP or the controller/device. The controller/device can provide the unusual end conditions with an channel end, controller end, or device end condition. If the IOP is aware of the unusual conditions by the time the channel end condition for the operation is received, the chain is terminated. This termination is initiated as if the operation, during which the condition occurred, was the last operation of the chain. The device end condition subsequently is processed as an interrupt condition. When the controller/device signals a unit check or unit exception with controller end or device end, the subchannel terminates the working state upon receipt of those signals from the controller. The channel end condition, in this case, is made unavailable to the software.

4.5.1.4 Termination By an RSCHNL Instruction

The reset channel instruction causes the addressed IOP to reset immediately. No interruption occurs and the IOP resets all subchannels. Controllers and devices attached to the IOP, through the MPB, are unaffected. The reset command is immediate and the IOP becomes quiescent in a noninitialized state.

When a reset channel command is issued to a IOP operating in the burst mode, the data transfer portion of the operation is immediately terminated and the controller/device is disconnected from the IOP.

4.5.1.5 Termination By an RSCTL Instruction

The reset controller instruction causes the current operation, at the addressed subchannel and controller, to be immediately terminated. No interrupt occurs from the addressed subchannel and controller.

The reset controller instruction is issued to a subchannel and controller operating in the burst mode, the data transfer is immediately terminated and the controller/device is disconnected from the IOP.

4.5.1.6 Termination By an HIO Instruction

The HIO instruction causes the current operation, at the addresed subchannel and controller, to be terminated. The HIO instruction is always transferred to the controller using an IOP initiated request instruction. A device end interrupt is generated by the controller and if any other conditions are owed, they are delivered with the device end command.

4.5.1.7 Termination By an STPIO Instruction

The STPIO instruction causes the IOP to suppress the command and data chaining flags in the addressed subchannel. The current operation at the subchannel terminates normally and the STPIO instruction is not transmitted to the controller/device.

4.5.1.8 Termination Due To Equipment Malfunction

When an IOP equipment malfunction is detected or invalid signals are received over the I/O device interface, the software is alerted of the termination. An I/O interrupt and the associated status words, indicating an IOP control check or interface control check condition, is use to alert the software.

4.5.2 I/O Interrupts

An I/O interrupt alerts the software to changes in state that occur at the IOP, controller, or device. These conditions are caused by the IOP program or by an external event at the device.

4.5.2.1 Interrupt Conditions

The conditions causing requests for I/O interrupt to be generated are called I/O interrupt conditions. An I/O interrupt condition alerts the software only once and is cleared when the interrupt is acknowledged by the CPU. Alternately, an I/O interrupt condition can be cleared by a TIO or SIO instruction.

The controller/device attempts to initiate a request to the IOP for an interruption whenever it detects any of the following conditions:

Channel end
Controller end
Device end
Attention

The IOP/controller/device may also create an interrupt condition caused by the following conditions:

Unit check
Unit exception
Device busy
IOP program check

When an operation is terminated because an unusual condition is detected in the IOP during a command initiation sequence (initial or chaining), the interrupt condition may remain pending within the IOP or the IOP may create an interrupt condition at the controller/device. An interrupt condition is created at the controller/device in response to the presentation of a status request command from the IOP. The interrupt condition at the controller/device may or may not be associated with the controller/device status field in the status words. If unusual conditions are detected by the controller/device (unit check, unit exception or busy) the controller/device status field of the status words identify the condition. With an IOP program check, the condition can be either created by the IOP or the controller, but it is preserved in the subchannel and appears in the IOP status field of the associated status words.

An interrupt condition, caused by the controller/device can be accompanied by IOP status and other controller/device conditions. More than one interrupt condition associated with the same device can be cleared simultaneously (i.e., when an channel end condition is not cleared at the device at the time the device end command is generated. Both interrupt conditions may be indicated concurrently in the controller/device status portion of the status words and cleared at the device.

When the IOP detects a post program controlled interrupt (PPCI) flag in the IOCD and the condition is met by the subchannel, it requests an interrupt without communicating with and/or receiving any status from the controller/device. The PPCI interrupt is not accompanied with any other status indicators.

4.5.2.2 Status Management

The status words are maintainedand managed by the IOP in a main memory area which is allocated for this purpose by the software. All IOPs must be initialized before any I/O operation is sent to any of the IOPs subchannels.

The channel memory allocation (CMA) command is used by the software to reserve the required memory area. The allocation must at least be on a doubleword boundary. Status entries are posted by order of termination.

During an acknowledge interrupt command, or the execution of an I/O instruction, the address of the status words is transferred to the CPU. The acknowledge interrupt command or the delivery of a status address, during the execution of an I/O operation, indicated to the IOP that the previously specified status is now available.

4.5.2.2.1 Status Address

The address of the status information passed to the CPU, during an acknowledge interrupt command or from the execution of an I/O operation, can be located using the service interrupt vector (SIV) locations associated with the IOP.

The initial assignments for the SIV are shown below:

Interrupt Level	SIV
14	140
15	144
16	148
17	14C
18	150
19	154
1A	158
1B	15C
1C	160
1D	164
1E	168
1F	16C
20	170
21	174
22	178
23	17C

The SIV contents are assigned during initial program load (IPL) and can be dynamically reassigned by the software; therefore, the initial assignments are only applicable if the software has not made any reassignments. Software must dynamically relocate the pointers for the IPU Console IOP to avoid conflicts.

Each SIV contains the address of a service interrupt control area (SICA). The ICA is used by the acknowledge interrupt, SIO, and other I/O instructions if the status stored is indicated by condition codes. The format and contents of the SICA are:

0	OLD PSD (WORD 1)
4	OLD PSD (WORD 2)
8	NEW PSD (WORD 1)
12	NEW PSD (WORD 2)
16	IOCLA
20	STATUS ADDRESS

Displacement 20 (decimal) of the SICA contains the address of the status words presented by the IOP. This address is changed when another acknowledge interrupt command or the execution of an I/O instruction indicates that the status is stored.

The relationship between the interrupt level, service interrupt location, and parameter list is shown in Figure 4-8.

4.5.2.2.2 Status Words

The status words send the software status of an I/O device or the conditions under which an I/O operation is terminated. The status words are formed, or parts of them replaced, before the I/O interrupt process and during execution of an I/O instruction (SIO, TIO, or HIO). The status words are placed in main memory by the IOP. They can be located using the associated SICA which, in turn, contains the main memory address of the current status doubleword.

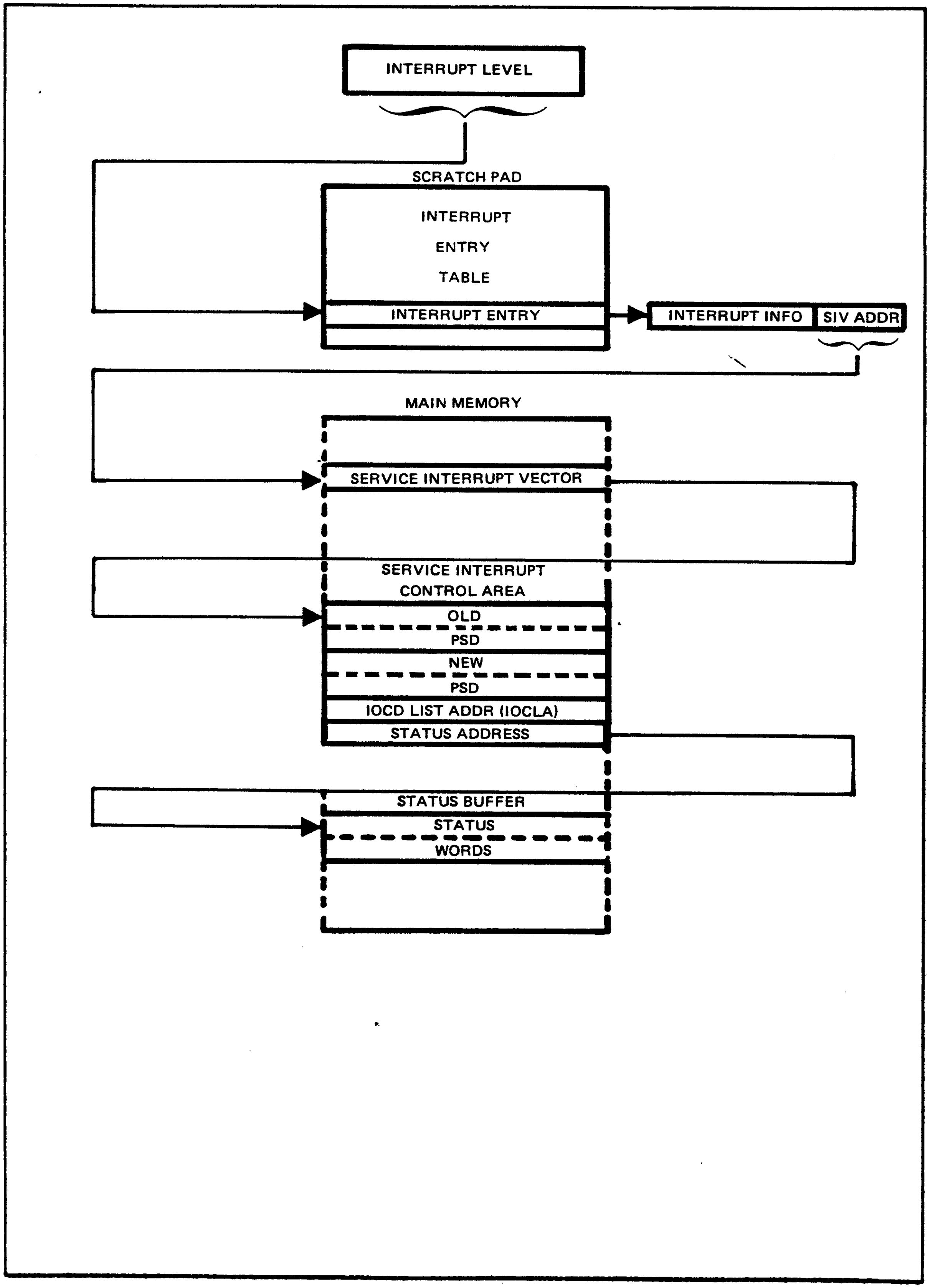


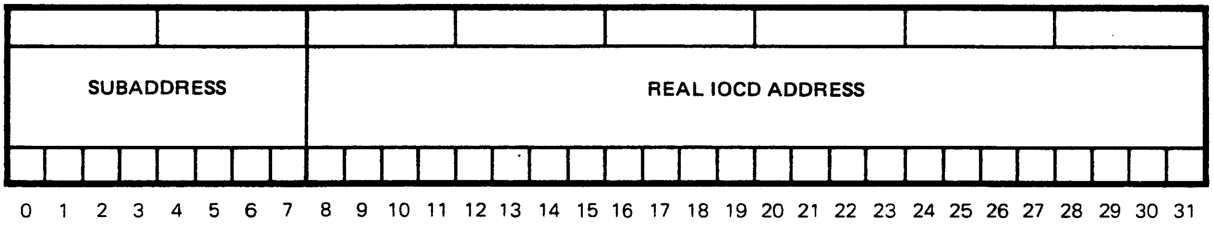
Figure 4-8. Interrupt Level, Service Interrupt Location, and Parameter List Relationships

The status words are available to the software until the next I/O interrupt, at the same level, occurs or another I/O instruction causes the contents to be dequeued, whichever occurs first.

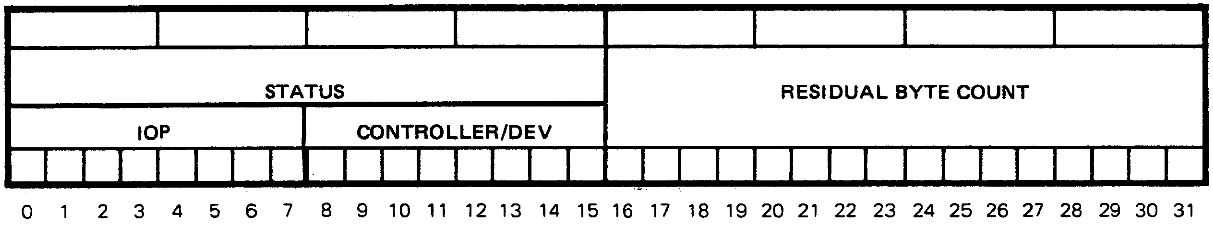
When the status words are stored, the I/O Device is identified by the subaddress field in the status word. The information stored by the execution of an I/O instruction pertains to the I/O device addressed by the subaddress. It may also apply to another device if pending status conditions prevent the start of a CPU initiated request.

The Status Words have the following format:

MSW



LSW



The fields in the status words have the following meaning:

The subaddress (bits 0 through 7) specifies the controller and device address of the device sending the status.

The real IOCD address (bits 8 through 31) specifies the address of the last IOCD executed plus 8.

The status (bits 0 through 15 of LSW) identifies the conditions in the IOP, controller, and device that caused the storing of the status words. Bits 0 through 7 of the LSW identify IOP subchannel conditions and bits 8 through 15 of the LSW identify the controller/device conditions. Each of the 16 bits represent one condition. These types of conditions are as follows:

Bit	
0	Echo
1	PPCI
2	Incorrect length
3	IOP program check
4	IOP data check
5	IOP control check
6	Interface check
7	Chaining check
8	Busy
9	Status modifier

Bit	
10	Controller end
11	Attention
12	Channel end
13	Device end
14	Unit check
15	Unit exception

The residual byte count (bits 16 through 31) specifies the count of the last IOCD used.

4.5.2.3 IOP Status Conditions

The following conditions are detected and indicated by the IOP. The conditions can only occur while the subchannel is involved with the execution of an I/O operation, except for the conditions caused by equipment malfunction.

4.5.2.3.1 Echo

The echo bit is not used in the IOP. Its meaning in extended I/O operations is briefly discussed below. The echo bit is set only if the addressed subchannel was busy with a previously initiated request. It indicates that this status presentation may only apply to the accepted transaction and not the transaction that caused the subchannel busy indication. In other cases the echo condition can clear other conditions that become pending after the acceptance of the transaction.

The rules for setting the echo bit are

1. Echo with pending status indicates that status became pending after the acceptance of the instruction.

2. Echo by itself indicates that the subchannel is idle.

3. Echo with other busy status bits indicates that the controller or device contains a currently executing I/O operation.

4. Echo with other status bits indicates that an asynchronous condition has occurred after the acceptance of the transaction.

4.5.2.3.2 Post Program Controlled Interrupt (PPCI)

The PPCI is generated when the IOCD PPCI flag is set and the byte transfer count in the current IOCD reaches zero (if data chaining is specified) or the device end condition is generated (if command chaining is specified). No predictable relationship exists between the PPCI condition and the PPCI interrupt occurring in the CPU.

The PPCI is not stacked with other interrupts for the same subchannel address; therefore, if another PPCI condition occurs when the prior PPCI condition is pending, the last PPCI is executed. The IOP continues normal processing; however, it must be sensitive to the acknowledged interrupt signal sequence from the CPU.

4.5.2.3.3 Incorrect Length

The incorrect length condition occurs when the byte transfer count, specified in the IOCD, is not equal to the number of bytes transferred to or from the device. The incorrect length condition is indicated for any of the following operations:

1. Long block on input. During a read, read backward, or sense operation the device requested one or more bytes after the IOP had reduced the IOCD byte transfer count to zero. The extra bytes are not written to memory. The residual byte count of the status is set to zero.

2. Long block on output. During a write or control operation the device requested one or more bytes after the IOP has reduced the IOCD byte transfer count to zero. The extra bytes are not written to the device. The residual byte count of the status is zero.
3. Short block on input. The number of bytes transferred during a read, read backward, or sense operation was insufficient to reduce the IOP IOCD byte transfer count to zero. The status count does not contain a zero; therefore, it reflects the number of bytes that were not tranferred.

4. Short block on output. The device terminated a write or control operation before all the information was transferred. The status count is not zero.

The incorrect length condition does not inhibit command chaining if the suppress incorrect length flag is set in the IOCD or a control command is executed.

4.5.2.3.4 Program Check

A program check occurs in the IOP when there is an IOP programming error. The following conditions can cause a program check:

1. Invalid IOCD address specification. This condition is set when the address, contained in a transfer in channel (TIC) IOCD or SIO data word, failed to be within word boundaries.

2. Invalid IOCD address. The IOP attempted to fetch an IOCD but received a no response or timeout indication.

3. Invalid command. The command in the IOCD contained all zeros in the four least-significant bits and the IOP did not support the control facility.

4. Invalid count. The IOCD contained a zero byte transfer count and was not a TIC command.

5. Invalid data address. The IOP attempted to read or write a byte in memory and received a no response or timeout indication.

6. Invalid sequence. The first IOCD in the command list specified a TIC or a TIC was attempted to transfer to another TIC command.

4.5.2.3.5 IOP Data Check

The IOP data check indicates that the IOP has detected a parity error in the information transferred to it from main memory.

This condition causes command chaining to be suppressed; however, the data transfer command continues for a normal completion. The interrupt is generated when the device sends a channel end command.

4.5.2.3.6 IOP Control Check

The IOP control check occurs when any equipment malfunction affects the IOP controls. This condition includes parity errors in the IOCD read from memory. Chaining is suppressed and the operation is terminated.

4.5.2.3.7 Interface Control Check

The interface control check occurs when the IOP to controller communication malfunctions. Conditions may include parity errors in the data tranasmitted to the IOP from controllers or invalid event sequences during IOP to controller communication. chaining is suppressed and the operation is terminated.

4.5.2.3.8 Chaining Check

Chaining check is to be defined.

4.5.2.4 Controller/Device Status Conditions

The following conditions are detected by the I/O device or controller and are sent to the IOP over the I/O interface. The timing and cause of these conditions are device dependent; therefore, the individual device publications should be consulted.

Due to device state changes (not ready to ready) the handling of conditions not associated with I/O operations depends on the type of device and conditions specified in the device publications.

4.5.2.4.1 Busy Status

Busy status indicates that the device or controller cannot execute the command or I/O instruction because it is executing a previously initiated operation. If the busy condition applies to the controller, the status modifier bit is also set.

4.5.2.4.2 Status Modifier Status

Status modifier status indicates that the controller cannot provide current status because it is busy or the normal sequence of the IOCDs is to be modified.

When status modifier status is stored, in the response to an I/O instruction, and all other status bits are off, the controller cannot execute the instruction and it has not provided current status. Any pending interrupt conditions present in the controller or device are not cleared and the status word stored, in memory, contains all zeros except the Status modifier bit.

When the status modifier bit appears with the busy bit, it indicates that the condition pertains to the controller and the addressed device. The controller appears busy when executing operations which preclude the acceptance and execution of any command or

I/O instruction, or contain a pending interrupt and status for a device other than the device addressed.

The busy state occurs for operations which the controller remains busy after providing an channel end indication.

The presence of status modifier and device end status indicate that the normal command sequence must be modified. The handling of this set of bits by the IOP depends on the operation. If command chaining is specified in the current IOCD and no unusual conditions are present, status modifier and device end status cause the IOP to fetch and chain to the IOCD whose main memory address is 16 greater than that of the current IOCD. If command chaining is not specified or if an unusual condition exists, the IOP takes no action, and status is saved in the status words.

4.5.2.4.3 Controller End Status

Controller end status indicates that the controller has become available for use in another operation. Controller end status is only provided by a controller shared by multiple I/O devices and only when one or both of the following conditions have occurred:

1. The software previously caused the controller to be interrogated while the controller was in the busy state. The controller is in the busy state when a command or I/O instruction has been issued to a device on the controller and the controller had responded with a busy status modifier in the status data.
2. The controller detected an unusual condition during the operation after an channel end command was executed, but before a device end command. The unusual condition accompanies the controller end status.

If the controller remains busy after a channel end condition, but was not interrogated by the software, controller end status is not generated. The controller does not provide controller end status if the instruction was accepted.

When a temporary controller busy state occurs, controller end status is included with the busy and status modifier status even though the controller has not been freed. The busy condition is considered temporary if its duration doesn't exceed 50 microseconds.

Controller end status can be generated with a channel end or device end condition. It can also be generated between the two conditions. When a controller end status condition is generated, using an interrupt in the absence of any other condition, the controller can be identified by the subaddress provided in the status. Pending controller end status causes the controller to appear busy for the initiation of any new operations.

4.5.2.4.4 Attention Status

Attention status indicates that the device has detected a device dependent asynchronous condition. The device can generate attention status only when no operation is in process at the subchannel, controller, and device. The attention status accompanies a device end status upon completion of an operation. It can also be presented during the initiation of a new oeration. Attention status accompanying a device end condition causes status chaining to be suppressed. Attention status can also be generated by itself when no operation is in progress at the subchannel, controller, or device.

4.5.2.4.5 Channel End Status

Channel end status is caused by the completion of the I/O operation involving data transfer or control information between the controller/device and the IOP. This status condition indicates that the subchannel has become available for use in another operation.

Each I/O operation causes channel end status to be generated. When command chaining occurs only the channel end status condition of the last operation of the chain is reported to the software. Channel end status is unavailable to the software when a chain of commands is prematurely terminated because of an unusual condition, indicated with control unit end or device end status information or during the initiation of a command.

The timing of channel end status is device dependent. For operations such as writing on magnetic tape or disc, channel end status occurs when the block has been written. When reading from magnetic tape or disc, channel end status occurs when the gap is reached. On buffered devices, such as the line printer, channel end status occurs when the data has been transferred to the buffer. During control operations, channel end status is generated when the control information has been transferred to the device. For operations of short duration, channel end status may be delayed until the completion of the operation. Operations that cause no data to be transferred can generate an channel end status during the initiation sequence of the command.

Channel end status pending in the controller causes the controller to appear busy for the initiation of the new operation.

4.5.2.4.6 Device End Status

Device end status is caused by the completion of an I/O operation at the device. On some devices, device end status is caused by manually changing the device from a not ready to a ready state. Device end status indicates that the device has become available for another operation.

Each I/O operation causes a device end status to be generated. When command chaining occurs, only the device end status of the last operation of the chain is reported to the software, unless an unusual status is detected during the initiation of a command. If this condition occurs, the command is terminated without device end status.

Device end status associated with an I/O operation is generated either simultaneously with the channel end status or at a later time. With data transfer operations on devices such as magnetic tape and discs, device end and channel end status are generated simultaneously. On buffered devices, such as the line printer, device end status occurs at the completion of the mechanical motion. For control operations, device end status is generated at the completion of the operation of the device. The operation may be completed at the time the channel end status is generated or at a later time.

When command chaining is specified in the subchannel, receipt of device end status, in the absence of any unusual status, causes the IOP to initiate a new I/O operation.

4.5.2.4.7 Unit Check Status

Unit check status indicates that the controller or device has detected an unusual condition that is detailed by the information available to a sense command. Unit check status may indicate that a IOP programming error has occurred, an equipment error has occurred, or a not ready state of the device has affected the execution of the command. The unit check status provides a summary indication of the conditions identified in the sense data.

An error condition causes unit check status only when it occurs during the execution of a command or during some activity associated with an I/O operation. Unless the error condition pertains to the activity initiated by a command and is of immediate significance to the software, unit check status does not cause the software to be alerted after the device end condition was cleared. A malfunction may cause the device to enter a not ready state.

A unit check condition indicates the existence of the not ready state precluding the satisfactory execution of the command. The unit check status bit, in the absence of the device end, channel end, or controller end conditions, indicates that the command was aborted before any action was attempted.

Unless the command is designed to cause a unit check indication, unit check status is not reported if the command is properly executed, even though the device entered a not ready state during the operation or as a result of the operation. Similarly, unit check status is not reported if the command can be executed with the device not ready. Selection of the device in the not ready state does not cause a unit check status indication when the sense command is issued, and whenever a controller interruption is pending for the addressed device.

Errors detected during the execution of a command are reported by a unit check status condition accompanied by channel end, device end, or controller end status, depending upon when the condition was detected. Any errors associated with an operation, but detected after the device end status condition has been cleared, are indicated by generating unit check and attention status.

Termination of an operation with a unit check indication causes command chaining to be suppressed.

4.5.2.4.8 Unit Exception Status

Unit exception status is generated when the controller or device detects a condition that usually does not occur. Unit exception status includes conditions such as bottom of form for the line printer and recognition of the end of file mark on magnetic tape. It has only one meaning for any particular command and type of device.

Unit exception status can only be generated when the device is executing an I/O operation, or when the device is involved with some activity associated with an I/O operation and the condition is of immediate significance o the software.

Unit exception status is normally accompanied by a channel end, device end, or controller end status, depending upon when the status was detected. By generating the unit exception and attention status conditions a device may request an initial program load (IPL) from itself.

Termination of a operation with unit exception status causes command chaining to be suppressed.

4.6 Operator Console Programming

Programming of the operator console communications interface is accomplished using the operator console software. Four subchannel addresses, along with their particular

IOCL, are required by the operator console. Two receive (RX) and two transmit (TX) subchannels are required to support the two full-duplex communication lines.

4.6.1 Subchannel Allocation

Subchannel allocation is as follows:

Subaddress	Function	Port Allocation
FC	RX LINE 0	MASTER PORT RECEIVE
FD	TX LINE 0	MASTER PORT TRANSMIT
FE	RX NE 1	SLAVE PORT RECEIVE
FF	TX LINE 1	SLAVE PORT TRANSMIT

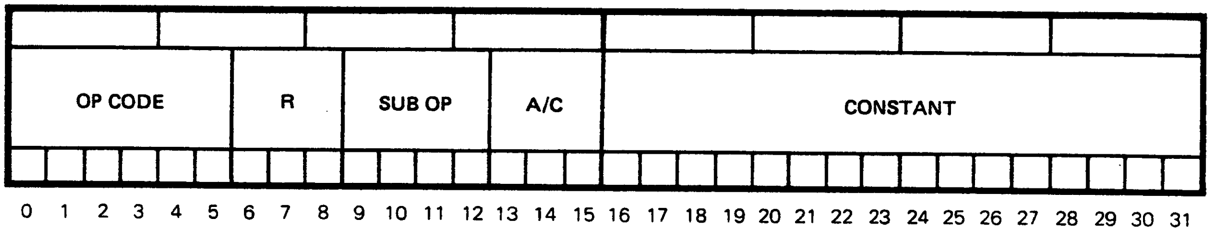
The vehicle of communications between the SYSTEMS 32 SERIES computer system and the operator console interface is the IOCL. The IOCL contains an order (read, write, etc.), flags (data chain, command chain), a byte transfer count, and the data buffer address.

The software initiates an I/O transfer by implanting an IOCL address in the dedicated SI plus 16 location and issuing a start I/O (SIO) command.

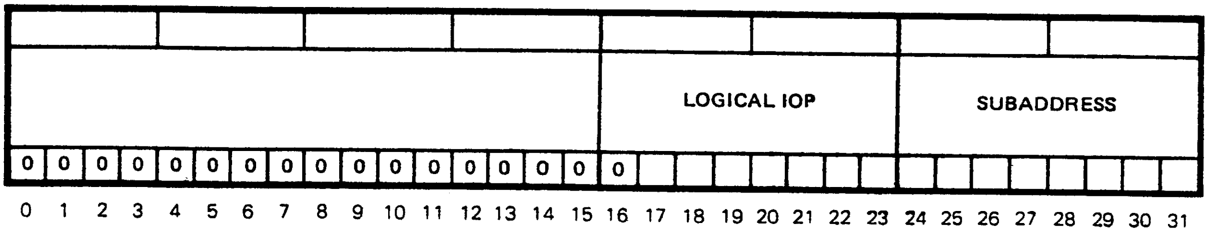
The CPU program initiates I/O operations with a SIO instruction. This instruction, plus the contents of the general register specified by the instruction (if nonzero), identifies the logical IOP and starting subaddress. The logical IOP address is used to vector to the appropriate device entry in local storage (scratchpad). The device entry contains the information required to convert the logical address to a real IOP address. The interrupt level, in the device, is used to vector to the main memory location containing the address of the service interrupt control area (SICA) associated with the IOP.

4.6.2 Instruction Format

All extended I/O instructions are in the following format:



The operation (Op) code (bits 0 through 5) and the augment code (A/C) field (bits 13 through 15) must contain ones. The R field (bits 6 through 8), if nonzero, specifies the general register whose contents are added to the constant field (bits 16 through 31) to form the logical IOP and subaddress. If R is specified as zero, only the constant field is used. The format of the computed logical IOP and subaddress is shown below:



The Sub-Op field (bits 9 to 12) assignments are as follows:

Bits 9 through 12	Sub-Op
0 0 0 0 - X'0'	Unassigned
0 0 0 1 - X'1'	Unassigned
0 0 1 0 - X'2'	Start I/O (SIO)
0 0 1 1 - X'3'	Test I/O (TIO)
0 1 0 0 - X'4'	Stop I/O (STPIO)
0 1 0 1 - X'5'	Reset channel (RCHNL)
0 1 1 0 - X'6'	Halt I/O (HIO)
0 1 1 1 - X'7'	Not supported
1 0 0 0 - X'8'	Not supported
1 0 0 1 - X'9'	Not supported
1 0 1 0 - X'A'	Unassigned
1 0 1 1 - X'B'	Not supported
1 1 0 0 - X'C	Enable channel interrupt (ECI)
1 1 0 1 - X'D'	Disable channel interrupt (DCI)
1 1 1 0 - X'E'	Activate channel interrupt (ACI)
1 1 1 1 - X'F'	Deactivate channel interrupt (DACI)

4.6.3 Condition Codes

The condition codes (CCs) are set for the execution of all extended I/O instructions. These codes indicate the successful or unsuccessful initiation of an I/O instruction. The condition codes can be set by the CPU, for IOP busy and inoperable or undefined IOP, or by the information passed directly from the IOP. The assignments for the condition codes are as follows:

CONDITION CODES				ASSIGNMENT
CC1	CC2	CC3	CC4	
0	0	0	0	Request activate, will echo status
0	0	0	1	IOP busy
0	0	1	0	IOP inoperable or undefined
0	0	1	1	Subchannel busy
0	1	0	0	Status stored
0	1	0	1	Unsupported transaction
0	1	1	0	Unassigned
0	1	1	1	Unassigned
1	0	0	0	Request accepted/queued, no echo status
1	0	0	1	Unassigned
1	0	1	0	Unassigned
1	0	1	1	Unassigned
1	1	0	0	Unassigned
1	1	0	1	Unassigned
1	1	1	0	Unassigned
1	1	1	1	Unassigned

Although 16 encoded conditions are possible, only the assigned patterns occur.

4.6.4 Initialize Channel (ICH) Instruction

Before execution of any instructions to start or stop I/O, an instruction to initialize the IOP must be executed. This action is accomplished using the initialization buffer. This buffer contains the address of the CPU status buffer. The CPU status buffer must be word boundaried and 256-words deep. The memory allocated must reside in a single 512-word page.

An attempt to execute any other instruction (other than reset channel) before executing an initialize IOP instruction results in the instruction being ignored.

4.6.5 RSCHNL Instruction

The RSCHNL instruction causes the addressed IOP to immediately reset. No interrupt is issued and the IOP resets all of its subchannels. The operator console subaddresses are completely reset; however, controllers and devices attached to the IOP, through the I/O MPB, are unaffected. The reset channel command is immediate, causing the IOP to become quiescent in a noninitialized state. The data terminal ready (DTR) flag is reset and no service interrupt (SI) signal is generated. The IOP then returns to the system control panel (SCP) mode.

4.6.6 RSCTL Instruction

The RSCTL instruction causes the current operation at the addressed subchannel, to be immediately terminated. No interrupt is issued from the addressd subchannel. The DTR flag is reset and no SI is generated.

4.6.7 HIO Instruction

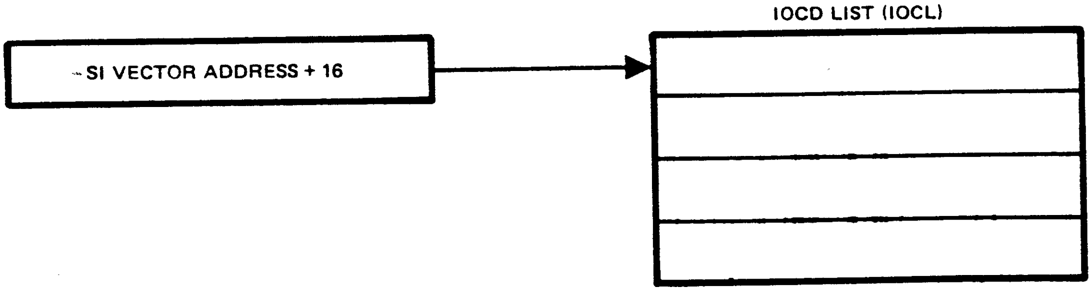
The HIO instruction causes the current operation, at the addressed subchannel, to be terminated. A device end interrupt is generated by the IOP and current status is outputted with the device end status. Termination by a HIO command is immediate. The state of the modem control lines is maintained.

4.6.8 STPIO Instruction

The STPIO instruction causes the IOP to suppress the command and data chaining flags in the addressed subchannel. The current operation at the subchannel terminates normally. The state of the modem control lines is maintained.

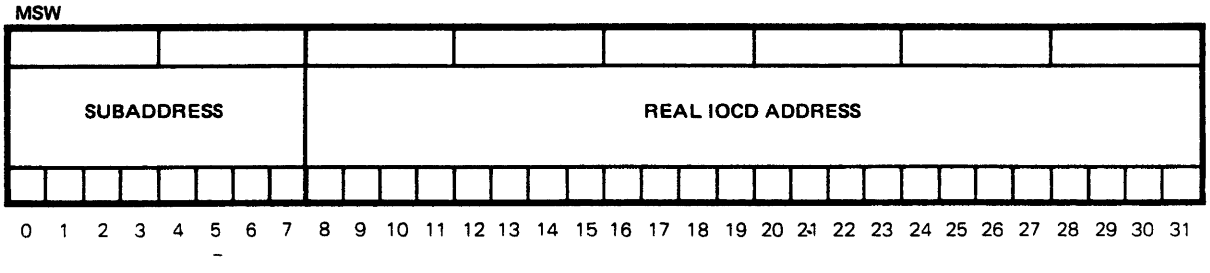
4.6.9 SIO Instruction

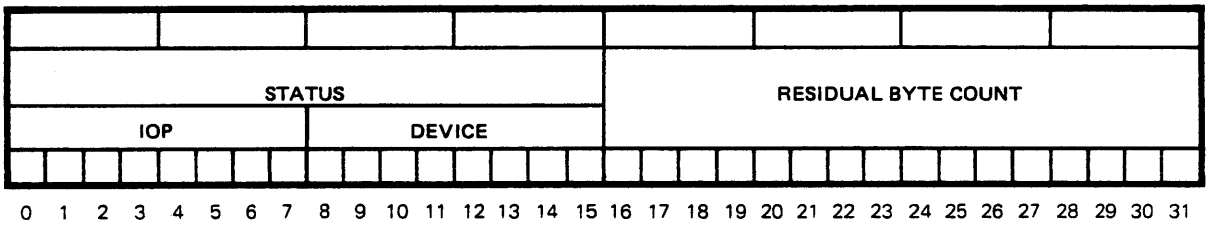
The I/O operation is started by storing the address of the IOCD or IOCD list into the SI plus 16 location and executing the SIO instruction.



4.6.10 CPU Status Words

During an acknowledge interrupt command, or the execution of an I/O instruction, the address of the status words are transferred to the CPU. The acknowledge interrupt command or the delivery of a status address, during the execution of an I/O operation, indicates to the IOP, that the previously specified status is now available. This status defines which subchannel caused the interrupt and the reason for the interrupt. The format for the status words is as follows:





The status word field definitions are listed below:

Subaddress: Specifies the communications subchannel presenting the status.

Real IOCD address Specifies the address of the IOCD executed plus eight.

Residual byte count Specifies the residual byte count of the last IOCD used.

Status Bit 0 = Echo
1 = Post program controlled interrupt (PPCI)
2 = Incorrect length
3 = IOP program check
4 = IOP data check
5 = IOP control check
6 = Not used
7 = Not used
8 = Busy
9 = Not used
10 = Not used
11 = Attention
12 = Channel end
13 = Device end
14 = Unit check
15 = Not used

4.6.10.1 Echo Status

The echo status bit is set only if the addressed subchannel was busy with a previously initiated request. It indicates that this status presentation may only apply to the accepted transaction and not the transaction that caused the subchannel busy indication.

4.6.10.2 Post Programmed Controlled Interrupt (PPCI)

PPCI status is generated when the IOCD PPCI flag is set and the byte transfer count is reduced to zero (if data chaining is specified) or the device end condition is generated (if command chaining is specified). No predictable relationship exists between the PPCI condition and the PPCI interrupt occurring in the CPU.

4.6.10.3 Incorrect Length Status

The incorrect length condition occurs when the byte transfer count, specified in the IOCD, is not equal to the number of bytes transferred to or from the device.

The incorrect length condition does not inhibit command chaining if the suppress incorrect length (SIL) indicator is set in the IOCD.

4.6.10.4 IOP Program Check

An IOP program check occurs in the IOP on an IOP programming error. Programming error conditions are in the following discussion.

4.6.10.4.1 Invalid IOCD Address Specification

The invalid IOCD address specification status bit is set when the address contained in an IOP IOCD or SIO data word failed to be on word boundaries.

4.6.10.4.2 Invalid IOCD Address

An invalid IOCD address condition indicated the IOP attempted to fetch an IOCD but received a nonpresent memory indication.

4.6.10.4.3 Invalid Command

An invalid command indicates that the command in the IOCD is not supported by the IOP.

4.6.10.4.4 Invalid Count

An invalid count indicates that the IOCD contained a zero byte transfer count and was not a TIC command.

4.6.10.4.5 Invalid Data Address

An invalid data address occurs when the IOP attempted to read or write a byte in memory and received a nonpresent memory indication.

4.6.10.5 IOP Data Check Status

An IOP data check indicates that the IOP has detected a parity error in the information transferred to it from main memory. The data transfer proceeds to a normal completion.

4.6.10.6 IOP Control Check Status

The IOP control check occurs when any equipment malfunction affects the IOP controls. This procedure also includes a check for parity errors in the IOCD read from memory. When this condition occurs, chaining is suppressed and the operation is terminated.

4.6.10.7 Busy Status

A busy condition indicates that the IOP cannot execute the SIO instruction because it is executing a previously initiated SIO instruction for the specified subaddress or the subchannel is operating in the SCP mode.

4.6.10.8 Attention Status

The attention command indicates that the device has detected a ring tone from the modem. If an I/O instruction is in progress it is aborted and attention status is reported.

4.6.10.9 Channel End Status

An channel end condition is caused by the completion of a portion of the I/O operation involving data or control information being transferred beteen the device and the IOP. This condition indicates that the subchannel has become available for another operation.

4.6.10.10 Device End Status

A device end inication is caused by the completion of a device I/O operation. This status condition indicates that the device has become available for use for another operation.

4.6.10.11 Unit Check Status

A unit check indicates that the device has detected an unusual condition that is detailed by the information available to a sense command. The unit check provides a summary indication of the condition identified in the sense data.

An error condition causes the unit check indication only when it occurs during the execution of a command or during some activity associated with an I/O operation. Errors detected during the execution of a command are reported by a unit check accompanied by an channel end and/or device end command, depending on when the condition was detected. If an unusual condition was detected, during the initiation of a command, the command is terminated without a device end status.

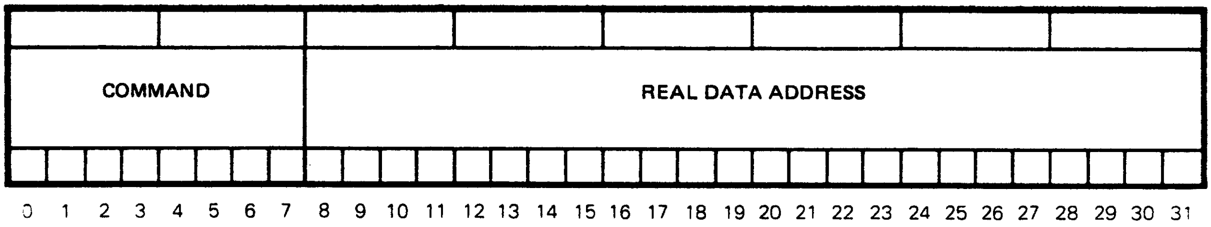
The unit check indication is caused by the following conditions:

1. Loss of a data set ready (DSR) indication.
2. Loss of a data carrier detected (DCD) indication.
3. An attempt, by software, to transfer data before a DSR indication is received.

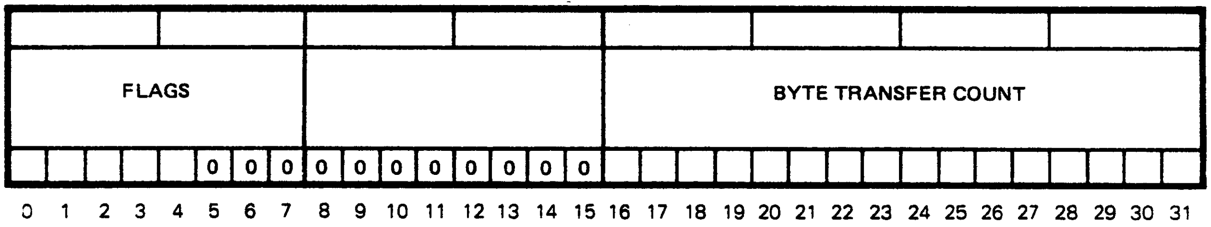
4.6.11 Basic I/O Command Doubleword

The format for the basic I/O command doubleword is shown below:

IOCD MSW



IOCD LSW



The I/O command doubleword field assignment are defined below:

Command field

00000000	=	Initialize IOP
00000100	=	Sense
00001000	=	TIC
XXXXXX01	=	Transmit (write) *
XXXXXX10	=	Receive (read) **
00001010	=	Receive with Echo
MMMMMM11	=	Control ***

* Note: X = Don't care

** Note: Any bit pattern ending in 10 results in a read operation except 00001010, which results in a read with echo operation.

*** Note: M = modifier bit. There are three control commands associated with the IOCD. The bit patterns for these control commands are shown below:

00011111	=	Connect line
00100011	=	Disconnect line
00000011	=	No operation (NOP)

Flags field:

Bit 0 = Data chain
Bit 1 = Command chain
Bit 2 = SIL
Bit 3 = Suppress transfer to memory
Bit 4 = PPCI

Real Data Address field:

Bits 8 through 31 (IOCD MSW) specify the location, in main memory, that is to be read from or written into.

Byte Transfer Count field:

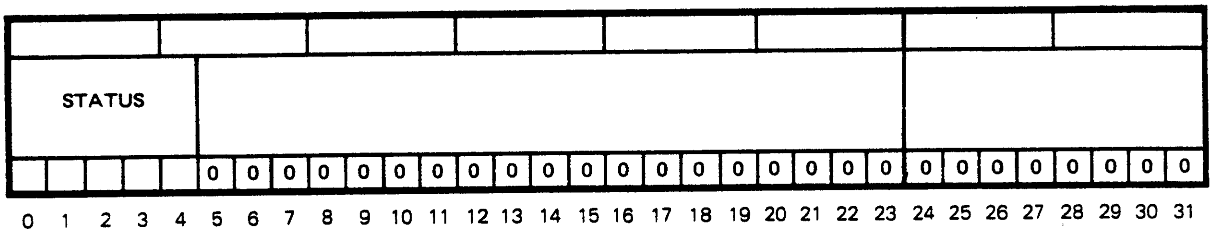
Bits 16 through 31 (IOCD LSW) specify the number of bytes that are designated by the IOCD.

4.6.12 Operator Console Commands

The operator console commands are defined below:

4.6.12.1 Sense

The sense operation places the programmable communications interface (PCI) circuit in the input mode and sets-up the IOP/subchannel to transfer data from the device to main memory. The data is placed in memory at the address specified by the real data address in the IOCD. The data format for the sense command is shwon below. Note that the byte transfer count in the IOCD must equal one for the sense command to be properly executed.



Sense data bit assignments are defined below:

Bit	0 through 3	=	Not used
	4	=	Data set ready
	5	=	Data carrier detected
	6 through 31	=	Not used

4.6.12.1.1 Data Set Ready

The DSR bit indicates that the local modem is connected to a communication line. If the DSR drops, during a transmission, a unit check is reported in the status information.

4.6.12.1.2 Data Carrier Detected

The DCD bit indicates that the local modem is receiving a carrier. If the DCD drops during a transmission, a unit check is reported in the status information.

4.6.12.2 Transfer In Channel (TIC)

The TIC command specifies the address the next IOCD to be executed. The TIC command allows the programmer to change the sequence of the IOCDs executed. The IOCLA cannot specify a TIC as the first IOCD in a command list nor can a TIC specify another TIC command.

4.6.12.3 Transmit (Write)

The transmit command causes a write operation, to the selected I/O device, from the specified main memory address. The write command is only issued over the transmit subchannel.

4.6.12.4 Receive (Read)

The receive command causes a read operation, from the selected I/O device, to the specified main memory address. The read command is only issued over the receive subchannel.

4.6.12.5 Receive With Echo

The receive with echo command causes a read, with cathode ray tube (CRT) terminal echo, operation from the selected I/O device, to the specified main memory address. The receive with echo command is only issued over the receive subchannel.

4.6.12.6 Connect Line

The connect line command, issued over the transmit or receive subchannel, causes the controller to set the DTR flag. This action is the only method of setting the DTR flag. The DTR flag determines the state of the DTR whenever the operator console mode is entered. Note that the IOCD byte transfer count must equal zero for the connect line command to be properly executed.

4.6.12.7 Disconnect Line

The disconnect line command, issued over the transmit or receive subchannel, causes the DTR flag to be dropped. Note that the IOCD byte transfer count must equal zero for the disconnect line command to be properly executed.

4.6.12.8 No Operation

The no operation (NOP) command, issued over the transmit or receive subchannel, causes no operation to be executed. Note that the IOCD byte transfer count must equal zero for the NOP command to be properly executed.

4.6.12.9 Flags

Flags associated with the above commands are defined below in the following discussion.

4.6.12.9.1 Data Chain Flag

During data chaining the new IOCD, fetched by the IOP, defines a new real data address and byte transfer count for the old I/O operation. Data chaining only occurs when all data, specified by the current IOCD, has been transferred to or from the device. This action causes the operation to continue using the data area and byte transfer count specified in the IOCD.

4.6.12.9.2 Command Chaining Flag

During command chaining, the new IOCD, fetched by the IOP, specifies a new I/O operation. The IOP fetches the new IOCD and initiates the new operation on receipt of the device end signal from the current operation.

4.6.12.9.3 Suppress Incorrect Length Flag

If any incorrect length condition occurs it is ignored, except in reporting status. The chaining procedure then continues.

4.6.12.9.4 Suppress Transfer To Memory (Skip) Flag

The skip command suppresses main memory writes during an I/O operation. This command is only defined for read, and sense operations, and is controlled by the skip flag in the IOCD. When the skip flag is set, skipping occurs. When the flag is reset, normal operation occurs. The setting of the skip flag is ignored in all other operations. Skipping affects the information handled by the IOP. The operation at the device proceeds normally and the information is transferred to the IOP. The IOP updates the byte transfer count but does not send the information to main memory. If command or data channing is specified, a new IOCD is obtained when the count reaches zero. For data chaining, normal operation is resumed if the skip flag in the new IOCD is not set. checking for invalid data addresses does not occur during skipping.

4.6.12.9.5 Post Program Controllerd Interrupt Flag

The PPCI function allows the software to interrupt the I/O device during the execution of an I/O operation. The function is controlled by the PPCI flag in the IOCD. The flag can be set in any IOCD in the list. The PPCI, when requested by the software, decreases IOP throughput. The true effect on IOP performance cannot be determined because of several factors, such as interrupt priority of the PPCI, the number of higher priority interrupts active or pending, and CPU/IOP communication overhead. Data overruns are possible if data chaining is also specified and the transfer rate of the device is high enough to cause the data loss. The IOP cannot service a data request when executing the acknowledge interrupt sequence.

Whenever the IOP completes the processing of an IOCD, with the PPCI bit set, it attempts to interrupt the CPU. The PPCI interrupt occurs when the byte transfer count in the current IOCD reaches zero (if data chaining is specified) or at device end (if command chaining is specified). No predictable time relationship exists between the PPCI interrupt and the I/O transfer.

CHAPTER 5

IOP REAL-TIME OPTION MODULE (RTOM)

5.1 General Description

The IOP RTOM receives external interrupt lines, integates them into the 32 SERIES Computer system discipline, and properly interfaces them to the SelBUS. In addition to providing access to four external interrupt levels, the IOP contains an interval timer and real-time clock (RTC) providing for internally generated timed interrupts. Interrupt level priorities may be dynamically assigned and intermixed in any order with interrupt levels from the I/O group. However, the IOP has a fixed internal priority. By using techniques employing subaddressing of random access memory (RAM) to achieve noncontiguous priority assignments, manual interrupt priority switches are eliminated. Program control of the interrupt levles enables the user to selectively command any interrupt level to a desired condition. The RTOM also provides each levels current condition (polling or not polling) to an external interrupt source.

5.2 Functional Description

The RTOM provides 16 priority levels and a RTC for the 32/27 SERIES computer. Figure 5-1 is a simplified block diagram of a RTOM function. When the RTC is used to provide a system interrupt, it uses 1 of the 16 available RTOM levels. Each of the 16 interrupt levels can be individually enabled, disabled, activated, deactivated, or requested under program control.

The RTC, which may be only connected to 1 of the 16 interrupt levels on the RTOM, operates from 60 or 120-Hz power and produces interrupts at an interval of 1/60 or 1/120 second. In 50-Hz installations, the RTC reference signal is either 50 or 100 Hz. The specific frequency of the reference signal is defined by the AC distribution panel.

The interval timer, which uses one of the 16 interrupt levels provided by the RTOM, is a 16-bit hardware and 16-bit firmware counter that may be programmed and/or read via the command device (CD) instruction. The CD instruction can control the interval timer thus allowing it to:

1. Select one of three counting rates

2. Select either single or multiple interrupts for a single count value

3. Enable or disable the counter

4. Load the initial count value

5. Transfer the contents of the counter to the CPU general purpose register 0 (GPR 0).

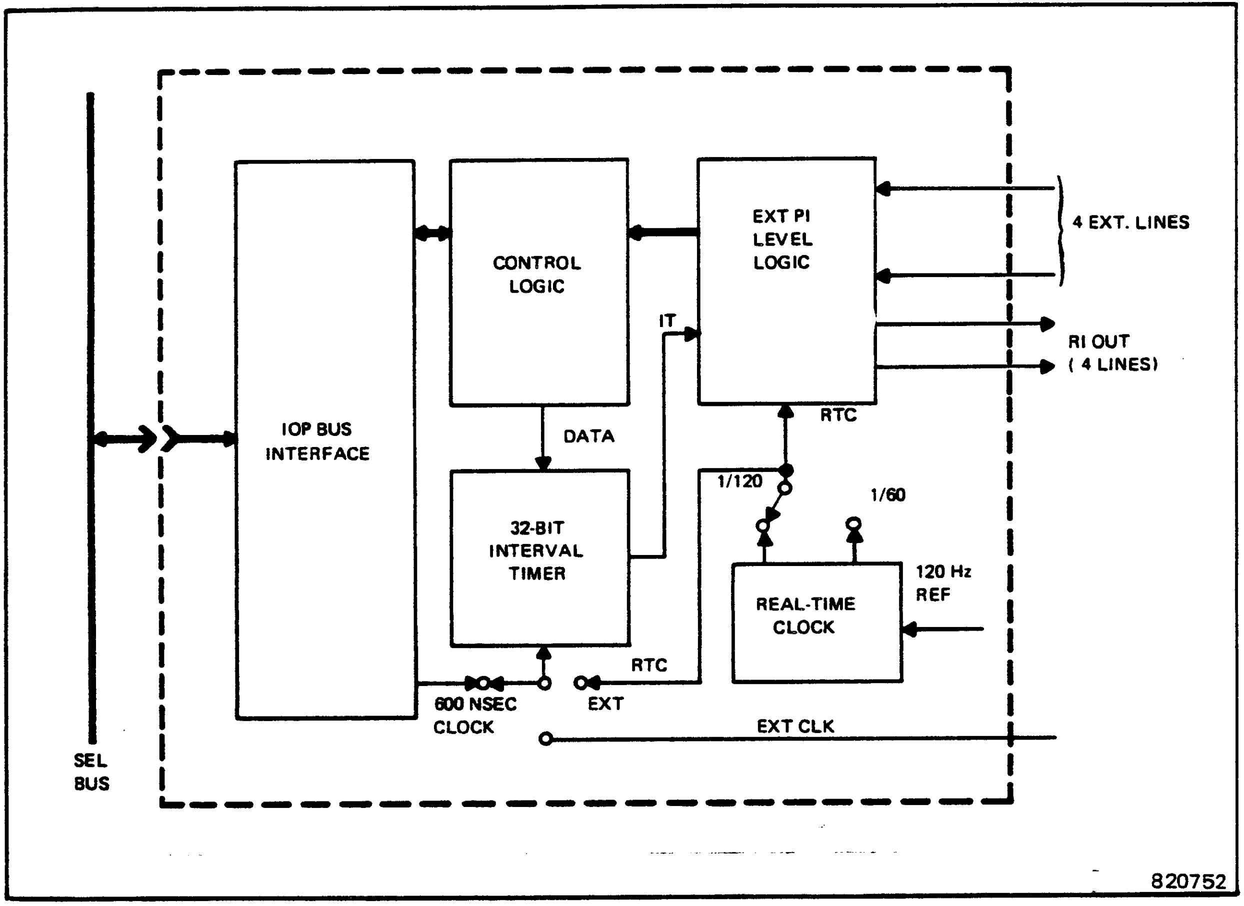


Figure 5-1. Simplified Block Diagram-RTOM Function

The RTOM performs the following functions:

1. Provides a media for integrating external levels into the basc interrupt system.

2. Performs the required tasks for all external interrupt levels that an I/O controller performs for a single I/O interrupt level.

3. Provides an interface to the SelBUS and external stimulus input lines permitting bidirectional communications between the CPU and external interrupt levels.

4. Provides a convenient method of incorporating a program controlled RTC into the system.

5. Permits stimulas inputs from hardware or software to initiate interrupt requests to the CPU.

6. Provide a means to achieve intersystem interrupt capability by making the outputs and external request inputs of each 'request' storage element available at the card edge. Connection to another RTOM, in a separate system, may be made by a 20-foot (maximum) ribbon cable.

The RTOM specifications and leading particulars are shown in Table 5-1.

Table 5-1
RTOM Specifications and Leading Particulars

Characteristic	Specification
External Interrupts (Input)	Low true
Number of lines	4

Logic level input	One=+0.8V (request interrupt)
	Zero=+2.4V (no request)

Signal driver	75452B or 75453B (recommended)
Driving end terminator	220 OHMS to +5 vdc 330 OHMS to ground (recommended)
Request pulse width	Greater than 300 ns Less than or equal to 1 ms
External interrupts (output)	Low true
Number of lines	4

Logic level outputs	One=+0.8V (external request)
	Zero=2.4V (no request)

Signal driver	75452
Receiving end terminator	220 ohm to +5 vdc 330 ohm to ground
Request pulse width	450 ns
Real-time clock	60/120 Hz (60-Hz installations) 50/100 Hz (60-Hz installations)
External clock	
Signal driver	75452B or 75453B (recommended)
Driving end terminator	220 ohm to +5 vdc 300 ohm to ground
Pulse width	Nominal = symetrical min. 600 ns
Pulse repetition rate	10 kHz
Interval timer	32 bit
Resolution	600 ns

Gould Inc., Computer Systems Division
6901 W. Sunrise Blvd.
P. O. Box 409148
Fort Lauderdale, FL 33340-9148
Telephone (305) 587-2900

GOULD
Electronics

Users Group Membership Application

USER ORGANIZATION: ____________________

REPRESENTATIVE(S): ____________________

____

____

ADDRESS: ____________________

____

TELEX NUMBER: ____________________ PHONE NUMBER: ____________________

NUMBER AND TYPE OF GOULD CSD COMPUTERS: ____________________

____

OPERATING SYSTEM AND REV. LEVEL: ____________________

____

APPLICATIONS (Please Indicate)

1. EDP
   A. Inventory Control
   B. Engineering & Production Data Control
   C. Large Machine Off-Load
   D. Remote Batch Terminal
   E. Other

4. Industrial Automation
   A. Continuous Process Control Op.
   B. Production Scheduling & Control
   C. Process Planning
   D. Numerical Control
   E. Other

7. Simulation
   A. Flight Simulators
   B. Power Plant Simulators
   C. Electronic Warfare
   D. Other

2. Communications
   A. Telephone System Monitoring
   B. Front End Processors
   C. Message Switching
   D. Other

5. Laboratory and Computational
   A. Seismic
   B. Scientific Calculation
   C. Experiment Monitoring
   D. Mathematical Modeling
   E. Signal Processing
   F. Other

8. Other

3. Design & Drafting
   A. Electrical
   B. Mechanical
   C. Architectural
   D. Cartography
   E. Image Processing
   F. Other

6. Energy Monitoring & Control
   A. Power Generation
   B. Power Distribution
   C. Environmental Control
   D. Meter Monitoring
   E. Other

Please return to:

Users Group Representative

Date:__________

Gould Inc., Computer Systems Division Users Group. . .

The purpose of the Gould CSD Users Group is to help create better User/User and User/Gould CSD communications.

There is no fee to join the Users Group. Simply complete the Membership Application on the reverse side and mail to the Users Group Representative. You will automatically receive Users Group Newsletters, Referral Guide and other pertinent Users Group activity information.

Fold and Staple for Mailing

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

(Detach Here)

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 947 FT. LAUDERDALE, FL

POSTAGE WILL BE PAID BY ADDRESSEE

GOULD INC., COMPUTER SYSTEMS DIVISION
ATTENTION: USERS GROUP REPRESENTATIVE
6901 W. SUNRISE BLVD.
P. O. BOX 409148
FT. LAUDERDALE FL 33340-9970

Fold and Staple for Mailing

GOULD
Electronics