SPERRY UNIVAC
90/30 System
Processor

Programmer Reference

SPERRY UNIVAC

UP-8052 Rev. 1 (B)

SEE SEPERATE SECTION

SPERRY UNIVAC
90/30 System
Processor
Programmer Reference

This document contains the latest information available at the time of preparation. Therefore, it may contain descriptions of functions not implemented at manual distribution time. To ensure that you have the latest information regarding levels of implementation and functional availability, please contact your local Sperry Univac representative.

Sperry Univac reserves the right to modify or revise the content of this document. No contractual obligation by Sperry Univac regarding level, scope, or timing of functional implementation is either expressed or implied in this document. It is further understood that in consideration of the receipt or purchase of this document, the recipient or purchaser agrees not to reproduce or copy it by any means whatsoever, nor to permit such action by others, for any purpose without prior written permission from Sperry Univac.

Sperry Univac is a division of the Sperry Rand Corporation.

This document was prepared by Systems Publications using the SPERRY UNIVAC UTS 400 Text Editor. It was printed and distributed by the Customer Information Distribution Center (CIDC), 555 Henderson Rd., King of Prussia, Pa., 19406.

PRINTED IN U.S.A.

PAGE STATUS SUMMARY

REVISIONS IN SEPERATE SECTION →

ISSUE: Update B – UP-8052

Part/Section	Page Number	Update Level
Cover		Orig.
Title Page		B
PSS	1	B
Preface	1, 2	Orig.
Contents	1 thru 7 8	Orig. A
1	1 thru 6 7 8 thru 11	Orig. B Orig.
2	1, 2 3 4 thru 7 8, 9 10 11 12 thru 28 29, 30 31 thru 35 36 37 38	Orig. B Orig. B Orig. B Orig. B Orig. B Orig. B
3	1 2 3 4 5, 6 7 8 9 10 11 12 thru 16 17 18 thru 26 27 28 thru 32 33 34 thru 47 48 49 thru 52 53 54 thru 56 57 58 59 60 thru 64 65 66 thru 70	Orig. B Orig. B Orig. B Orig. B Orig. B Orig. B Orig. B Orig. B Orig. B Orig. B Orig. B Orig.
4	1 thru 16	Orig.
Appendix A	1, 2 3 4	Orig. B Orig.
Appendix B	1 thru 10 11 12	A B A
Appendix C	1 thru 3	Orig.
User Comment Sheet		

Part/Section	Page Number	Update Level

All the technical changes are denoted by an arrow (→) in the margin. A downward pointing arrow (↓) next to a line indicates that technical changes begin at this line and continue until an upward pointing arrow (↑) is found. A horizontal arrow (→) pointing to a line indicates a technical change in only that line. A horizontal arrow located between two consecutive lines indicates technical changes in both lines or deletions.

Preface

This manual is one of a series designed to instruct and guide the programmer in the use of the SPERRY UNIVAC 90/30 System. This particular manual describes the 90/30 central hardware, i.e., the processor, the input/output section, and the main storage.

This manual is divided into the following sections and appendixes:

- SECTION 1. INTRODUCTION

  Briefly describes the components that comprise the central hardware and the optional expansion features that are available.

- SECTION 2. PROCESSOR

  Describes general processor operation, with emphasis placed on the processor registers; data, control word, and instruction formats; interrupts and processor-detected errors; and monitor mode and initial load operation.

- SECTION 3. INPUT/OUTPUT CHANNELS

  Describes the integrated peripheral channel, the integrated disk adapter, and the selector and multiplexer channels.

- SECTION 4. MAIN STORAGE

  Describes general main storage operation, with emphasis placed on addressing, interface with the processor, priorities, boundaries, protection, address relocation, and errors.

- APPENDIX A. IPC BCW COMMAND CODES

  Lists the integrated peripheral channel (IPC) buffer control word (BCW) command codes.

- APPENDIX B. INSTRUCTIONS

  Contains tables that:

  - list the instructions and timing in alphabetic order;
  - list the instructions in operation code order;
  - list the IBM 360/20 and SPERRY UNIVAC 9200/9300 instructions accepted by the 90/30 system; and
  - list the 360/20 and 9200/9300 instructions that result in program exceptions.

- APPENDIX C. ABBREVIATIONS AND ACRONYMS

  An alphabetically arranged list of the abbreviations and acronyms used in this manual.

Complete descriptions of the peripheral subsystems that can be attached to the central hardware are not provided in this manual. They can be found in their appropriate subsystem manuals. The integrated peripheral subsystems are described in the 90/30 processor integrated peripheral channel programmer reference, UP-8041 (current version).

Contents

2. PROCESSOR

FIGURES

TABLES

1. Introduction

1.1. FUNCTIONAL DESCRIPTION

The SPERRY UNIVAC 90/30 central hardware, Figure 1–1, consists of three types of equipment based upon function: processor, input/output (I/O), and main storage.

1.1.1. Processor

This processor is characterized by an arithmetic section. The basis of control for the processor is microprogrammed instructions. The 90/30 machine instructions reside in a separate control storage area and are composed of a set of microcode subroutines that manipulate the logical circuitry of the processor to perform the specified operations.

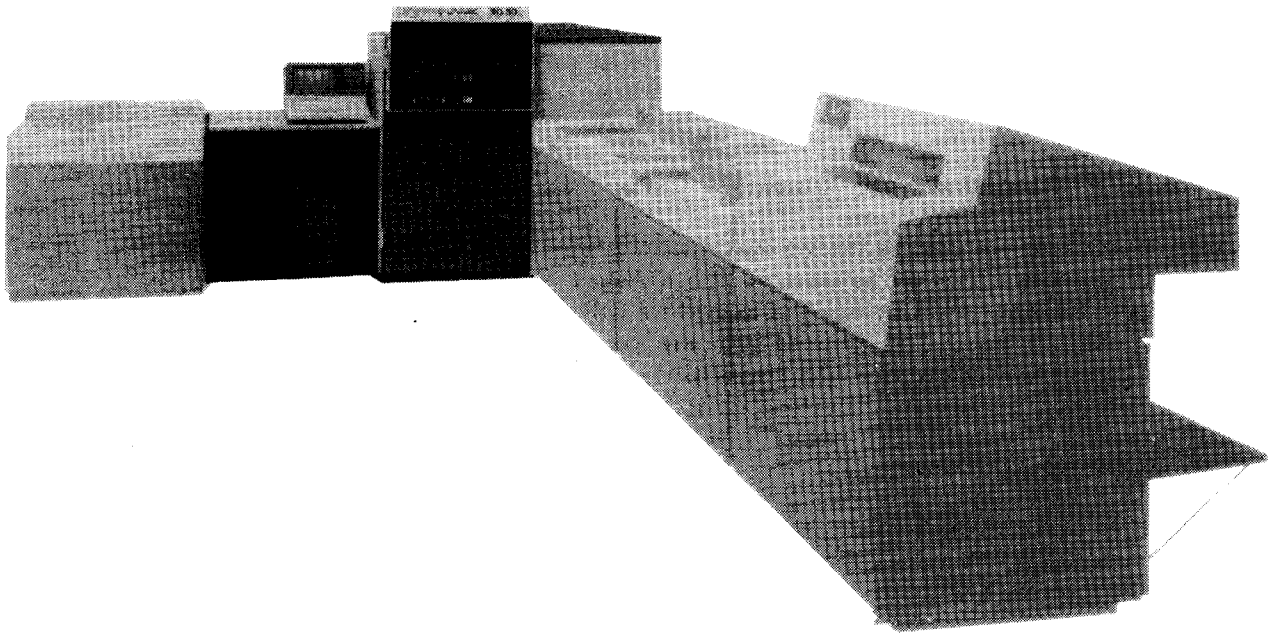


Figure 1—1. SPERRY UNIVAC 90/30 Central Hardware

1.1.1.1. Arithmetic Section

The arithmetic section performs all logical and arithmetic operations utilizing fixed-point and floating-point functions. Logical operations consist of comparing, translating, editing, bit testing, bit manipulation, and shifting. The arithmetic operations require that full-word addresses be located on a word boundary and that half-word addresses be on a word or half-word boundary. Fixed-point binary arithmetic uses the twos complement number representation. Floating-point arithmetic and decimal arithmetic use sign and absolute magnitude number representation.

1.1.1.2. Instruction Repertoire

The instruction repertoire consists of 148 instructions, of which 84 are basic instructions, including 73 nonprivileged and 11 privileged instructions. A total of 44 floating-point instructions is provided. Also included are two privileged instructions for storage protection and five emulation instructions. (Refer to Appendix B).

1.1.1.2.1. Instruction Formats

There are five instruction formats employed: register to register (RR), register to indexed storage (RX), register to storage (RS), storage and immediate operand (SI), and storage to storage (SS). Each format consists of an operation code field and two or more fields that specify the address of the operands. Instructions can be two, four, or six bytes in length (or one, two, or three half words) as dictated by the format.

1.1.1.2.2. Nonprivileged Instruction Set

Nonprivileged instructions are used to process fixed-length binary numbers, floating-point numbers, packed and unpacked decimal numbers, and EBCDIC or ASCII characters. Data can be transferred between main storage and any of the user program registers or from one location in main storage to another. Also included in the nonprivileged instruction set are shifting, branching, and logical functions. If a program attempts to execute a privileged instruction, a program exception interrupt occurs.

1.1.1.2.3. Privileged Instruction Set

Privileged instructions are used by the system software when the processor is in the supervisor state. In this state, all installed instructions can be executed. The privileged instructions include the facility to load and store the contents of low-order main storage and to load the writable section of the microprogram control storage. Additionally, the two instructions provided with the storage protect feature are privileged. The privileged instructions cannot be executed by a user program.

1.1.2. Input/Output Section

The input/output (I/O) section initiates, directs, and monitors the transfer of data between main storage and the peripheral subsystems. After an I/O instruction is initiated, the data is transferred independent of other processor functions; the I/O channels and the processor operate concurrently.

1.1.2.1. Integrated Peripheral Channel

The integrated peripheral channel (IPC) coordinates all information transfers between main storage and the integrated peripheral devices: system console, card reader, card punch, printer, communications adapter (CA), and diskette subsystem. The IPC is a half-duplex channel that transfers commands, data, status, and sense information. I/O activity is initiated by the processor upon issuance of a start I/O (SIO) instruction to the IPC. This instruction results in the transfer of a command to the control logic of a specific peripheral device. The command specifies the type of operation to be performed and is executed on an individual basis. The high transfer rate of the IPC permits simultaneous operation of the integrated peripherals. The control section of each integrated peripheral device is located within the IPC.

1.1.2.2. Integrated Disk Adapter

The integrated disk adapter (IDA) acts as a combination channel and control designed to operate with integrated disk drive units. The IDA is identified as an optional feature even though it may be part of the basic system configuration. This is because the IDA can be deleted from the configuration if another disk subsystem is available on a selector channel.

1.1.2.3. Selector Channel

The selector channel controls the exchange of information among any of eight possible subsystems and the processor and main storage. The selector channel operates in burst mode; that is, one of the eight possible subsystems retains control of the interface for the duration of its I/O operation. Simultaneously, other subsystems can be executing previously initiated operations that do not involve data transfer over the I/O interface. The processor initiates all I/O operations to the selector channel and the specific subsystem connected to the selector channel. Detailed descriptions of the required I/O operation are provided to the channel in software-generated control words. Once the operation is successfully initiated, the channel maintains control of the data transfers between main storage and the subsystem independent of the processor. Upon completion of the I/O operation, the status of the channel and subsystem is presented to the software by way of the appropriate status words and I/O status tables.

1.1.2.4. Multiplexer Channel

The multiplexer channel is similar in operation to the selector channel except that it operates in multiplex mode. Multiplex mode differs from burst mode in that the channel services several concurrently operating subsystems by assigning the I/O interface to a subsystem only long enough to transfer one or a few bytes of information. The multiplexer channel controls up to eight subsystems. I/O operations are initiated by a start I/O instruction from the processor to a selected channel and a selected subsystem. Detailed descriptions of the required I/O operation are provided to the multiplexer channel by software-generated control words. Again, when the operation is successfully initiated, the multiplexer channel controls the flow of data between main storage and the subsystem independent of the processor. At the completion of the I/O operation, the status of the multiplexer channel is presented to the software by way of the appropriate status words and I/O status tables.

1.1.2.5. Integrated Multiplexer Channel

The integrated multiplexer channel is located within the processor cabinet and is functionally equivalent to the multiplexer channel (1.1.2.4) housed in the I/O expansion cabinet (Figure 1-2). The integrated multiplexer channel controls the exchange of information between eight peripheral subsystems attached to the I/O interface and the processor and main storage. Operating in multiplex mode only, the channel services several concurrently operating subsystems by assigning the I/O interface to a subsystem only for the time required to transfer a few bytes of information. The channel then services other subsystems in a similar manner, as required, before returning to service the first subsystem again.

The integrated multiplexer channel services a control unit operating in burst mode as though it were a normal multiplex mode transfer of long duration on a single subchannel. However, the integrated multiplexer never forces a control unit into burst mode.

It should be noted that when the integrated multiplexer channel is operating with a control unit in forced burst mode, all other operations are halted in the integrated multiplexer channel. Therefore, requests for service from other subsystems and processor instruction requests to the channel are ignored while the channel is operating in burst mode. If there is no response to an instruction request within 16 milliseconds, a processor check interrupt may result due to processor stall. This operating condition should be understood by the operator and programmer before placing the integrated multiplexer channel in operation with control unit burst mode.

1.1.3. Main Storage

Main storage is the semiconductor type, with 600-nanosecond half-word read/write cycle time. Minimum main storage is 65K bytes, expandable to a maximum of 512K bytes (K=1024 bytes), depending on the processor series.

1.1.3.1. Information Positioning

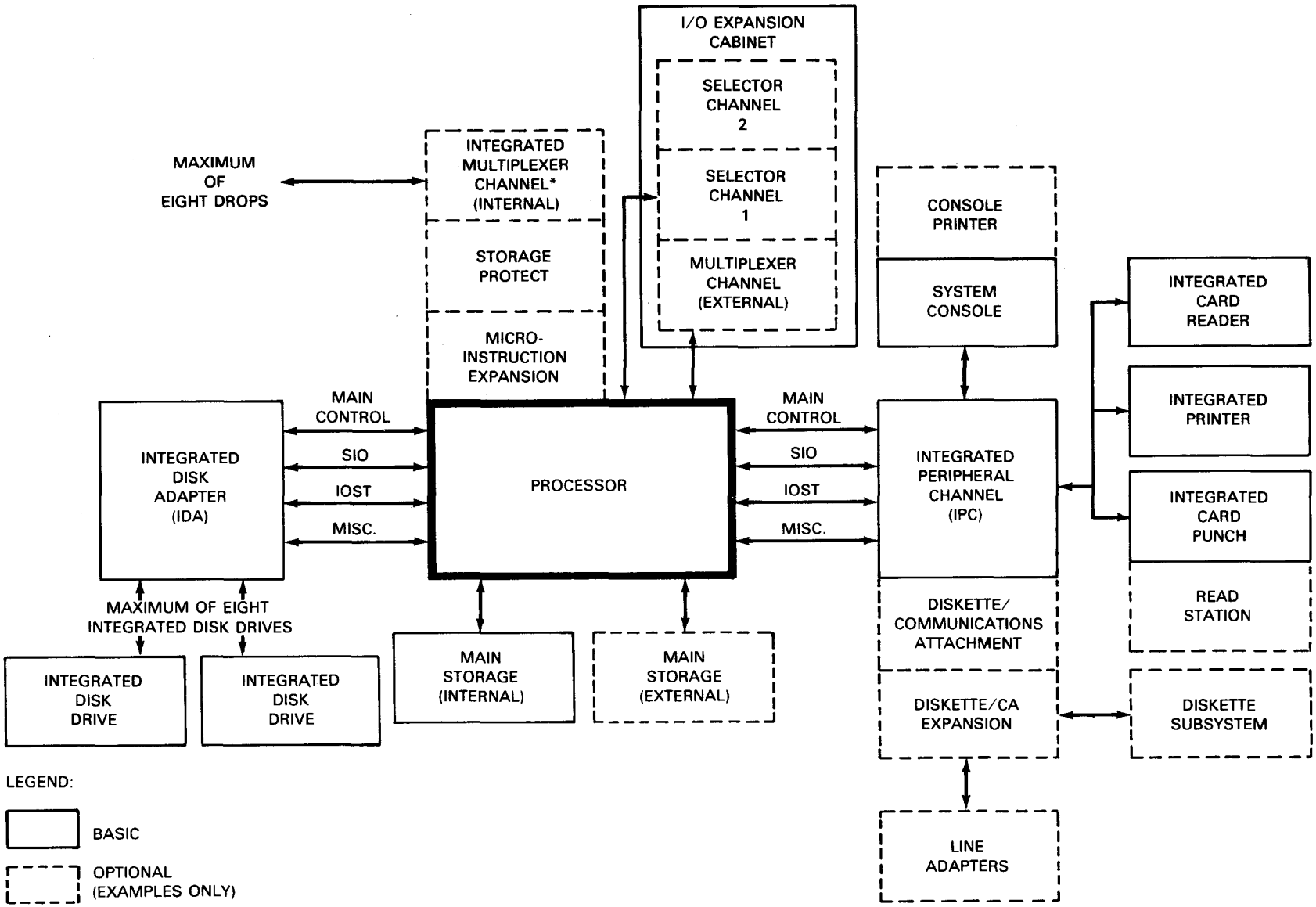
Locations in main storage are addressed consecutively from 0 through a maximum of 524,287 bytes. Bytes may be accessed separately or in groups. A group of bytes is addressed by the leftmost byte of the group. The bits in a byte are numbered from left to right starting with 0.

1.1.3.2. Low-Order Main Storage

The low-order 640 bytes of main storage are reserved for specific operating information. This information is accessed by the hardware and the operating system during execution of supervisor functions. The operating system provides for loading and protecting the information in low-order main storage.

1.1.3.3. Storage Protection

Program integrity in a multiprogram environment is guaranteed by an optional storage protection feature. Eight keys are provided for this purpose. The storage protection facility also uses a 3-bit protection key, which identifies the program or operation that requested access to main storage.



*I/O expansion cabinet is not used when an integrated multiplexer channel is included.

Figure 1—2. Relationship of Processor to I/O Channels, Main Storage, and Peripheral Devices

1.2. CONFIGURATIONS

The minimum and expanded configurations of the 90/30 central hardware are shown in Figure 1-3. The minimum configuration is indicated by the solid-line boxes; the dashed-line boxes indicate examples of the optional expansion features. Available options are listed and briefly described in Table 1-1.

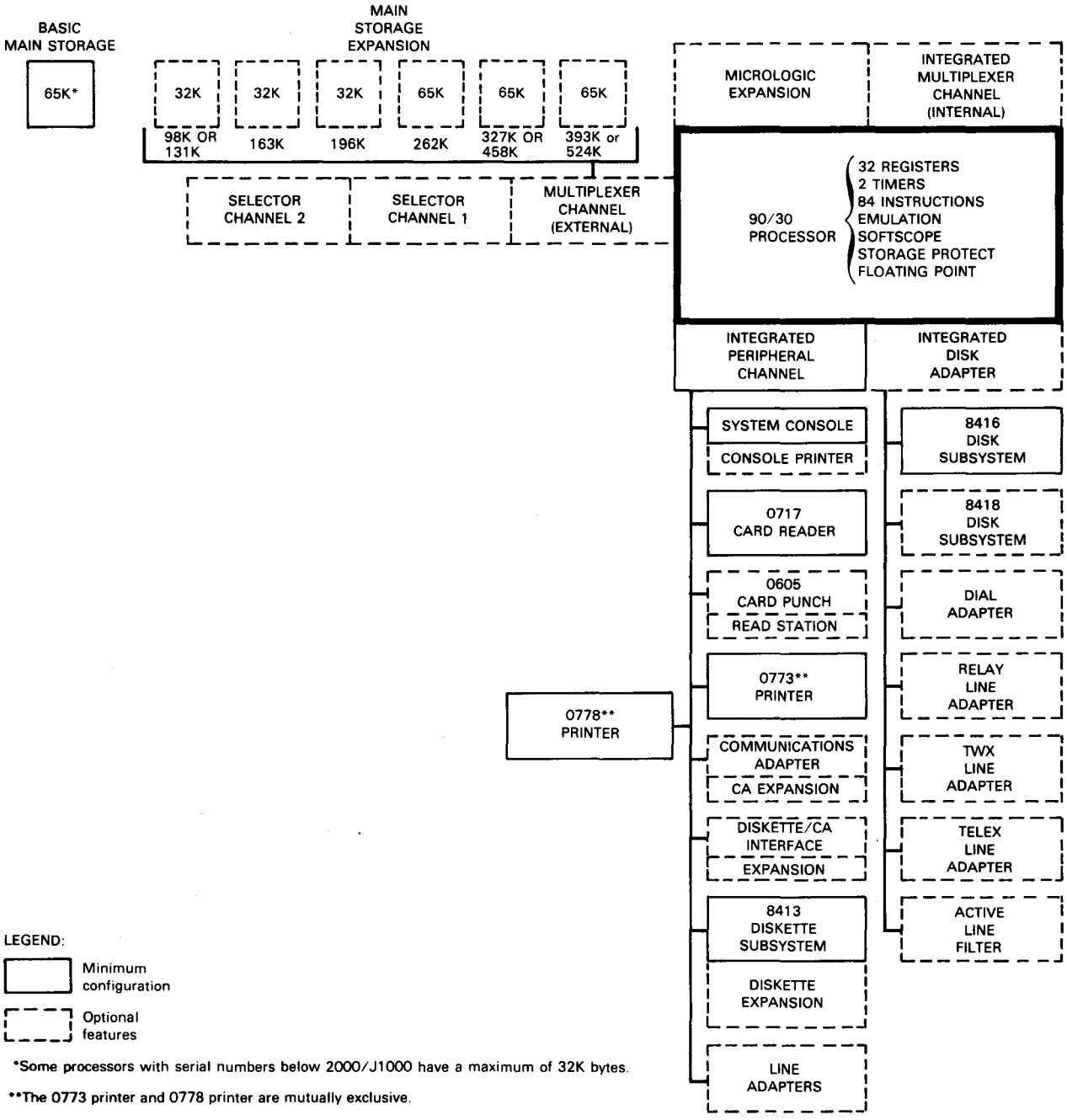


Figure 1—3. 90/30 Configuration of Central Hardware with Integrated Peripheral Subsystems

Table 1—1. 90/30 Central Hardware Optional Expansion Features (Part 1 of 2)

Name	Description
Console printer	Provides a printer capable of printing hard copy output of the data displayed on the system console screen. The printer is connected to the system console by an auxiliary interface. The printer has a 64-character ASCII set with a print rate of 30 characters per second Further information on the console printer is found in the functional description, UP-7939 (current version).
Multiplexer channel (external)	A channel with an 8-subsystem maximum configuration capability, eight active subchannels, and an 83K-byte data transfer rate. Maximum of one to a system
Multiplexer channel (internal)	Same as external multiplexer channel; is not included if an external multiplexer or selector channels are used
Selector channels 1 & 2	A channel with an 8-subsystem maximum configuration capability, one active subchannel, and a nominal transfer rate of 82,516 bytes. Maximum of two to a system
Integrated disk adapter (IDA)	Provides the interface and control between the processor and up to eight 8416 and/or 8418 drives, with a nominal data transfer rate of 625K bytes
Integrated card reader, 0717	A 500-card-per-minute, column-by-column reader; hopper capacity of 2400 cards; stacker capacity of 2000 cards; validity checking; and a read check station. Attaches directly to the IPC by way of the integrated control
Integrated card punch, 0605	An 80-column, 75-to-160 card-per-minute punch; hopper capacity of 700 cards; stacker capacity of 600 cards; validity checking; reject stacker. Contains the mechanism and electronics to control card feeding, punching, and stacker selection. Attaches directly to the IPC by way of the integrated control
Integrated printer, 0773	A 120-print position printer, expandable to 144 print positions, capable of printing 500 lines per minute. The band contains the basic 48-character set. Attaches directly to the IPC by way of the integrated control. Not included when the 0778 printer is used
Integrated printer, 0778	A 120-print position printer, expandable to 136 print positions, capable of printing 300/500 lines per minute. The band contains the basic 48-character set. Attaches directly to IPC by way of integrated control. Not included when the 0773 printer is used
Storage expansion	Provides storage increments that may be added to basic main storage for a maximum of 524K bytes
Micrologic expansion	Provides a repertoire of 64 additional instructions, 4 registers (each 64 bits long), and expanded control storage. Provides micrologic for execution of 44 floating-point instructions in both long and short, normalized and unnormalized formats, and micrologic for the execution of 20 additional nonprivileged instructions
Storage protect	Provides read/write protection on access to main storage and micrologic for two additional privileged instructions (SSK, ISK). Protects up to 524K bytes of internal and external storage
Communications adapter	Provides interfaces and control required to coordinate the transfer of data between this feature and a maximum of 6 full-duplex or 12 half-duplex line adapters (LA)
Communications adapter expansion	Adds a maximum of 6 full-duplex or 12 additional half-duplex line adapters, for a total of 12 full-duplex or 24 half-duplex line adapters
Integrated disk subsystem, 8416	Provides direct access up to 28.9 million bytes of data through a removable disk pack. Average access time is 33 ms; transfer rate is 625K bytes per second

Table 1—1. 90/30 Central Hardware Optional Expansion Features (Part 2 of 2)

Name	Description
Integrated disk subsystem, 8418	Provides direct access up to 57.9 million bytes through a removable disk pack. Average access time is 27 ms; transfer rate is 628K bytes per second
Diskette/communications adapter interface	Permits an 8413 diskette subsystem, a communications adapter, or a diskette/communications adapter expansion feature to be included with the central hardware
Diskette/communications adapter interface expansion	Connects the 8413 diskette subsystem and communications adapter (CA) when both are used or connects a CA and CA expansion
Synchronous line adapters	Provides full-duplex or half-duplex interface to synchronous data sets
Asynchronous line adapters	Provides full-duplex or half-duplex interface to asynchronous data sets
Dial adapter	Provides an interface for rotary or touch-tone automatic dialing units
Relay line adapter	provides an asynchronous full-duplex interface compatible with 20 to 70 mA neutral or 10 to 40 mA polar telegraph lines
TWX line adapter	Provides an interface to U.S.A. TWX network
Telex line adapter	Provides an interface to U.S.A. Western Union telex network
Active line indicator	Provides a display panel to display line activity on up to 12 communications lines
Diskette subsystem, 8413	Provides two floppy-disk read/write diskette drives that use IBM compatible diskettes. Each diskette contains storage of 1898 records, each having a length of 128 bytes. The maximum storage capacity is 242,844 bytes.
Diskette expansion	Provides 90/30 diskette capability, diskette power supply, and diskette dual expansion capability

1.2.1. Minimum Basic Central Hardware Configuration

The minimum 90/30 central hardware consists of:

- Processor
- 65K bytes of main storage
- Integrated peripheral channel
- System console
- Integrated disk adapter (with two disk drives)

The minimum operational 90/30 system also includes the following integrated subsystems:

- Card reader or diskette subsystem
- Disk subsystem
- Printer

1.2.2. Expanded Configuration

The expanded 90/30 central hardware comprises all of the items listed under 1.2.1 and:

- Storage protection
- Micrologic expansion
- Main storage expansion to a maximum of 524K bytes
- Console printer
- Selector channel 1
- Selector channel 2
- Multiplexer channel (internal)
- Multiplexer channel (external)
- Communications adapter
- Communications adapter expansion
- Integrated card punch with read station
- Diskette/communications adapter interface
- Diskette/communications adapter interface expansion
- Line adapters

1.3. CHARACTERISTICS

The 90/30 processor is a microprogrammed computer having a maximum repertoire of 148 instructions. Main storage is of the semiconductor type and is modularly expandable (Figure 1-3). The characteristics, capabilities, and optional features available for the processor, main storage, and I/O channels are as follows:

- Processor and Storage
  - Interval timer
  - Six interrupt levels

- Hardware-assisted dynamic storage relocation
- 32 general registers
- 600-nanosecond cycle time
- SOFTSCOPE maintenance aids
- Microprogrammed native instruction set
- 84 basic instructions (73 nonprivileged and 11 privileged)
- Input/output interrupt status tabling
- Four modes of operation:

  Native (90/30) mode
  9200/9300 mode
  360 model 20 mode
  Monitor mode

- 64 additional instructions by micrologic expansion* (including 44 floating-point and 20 nonprivileged instructions)
- Two storage protection privileged instructions*
- 16K-byte half-word parallel read/write
- Maximum of 524K bytes; expansion increments* of 32K or 65K bytes
- Transfer rate of 600 nanoseconds (access time for a half word)
- Storage protection* – write only or read/write in blocks of 512, 1024, or 2048 bytes
- Dynamic relocation mechanism
- Semiconductor technology

■ Integrated Peripheral Channel (IPC)

- Transfer rate of up to 50K bytes per second
- Up to 32 active subchannels
- Integrated reader control
- Integrated punch control
- Integrated printer control
- Integrated diskette input/output

Optional expansion feature

- Integrated operator console control
- Communications adapter, line adapters, diskette/communications attachment, and communications attachment*
- Data chaining (communications only)
- Communications adapter expansion*

■ Integrated Disk Adapter (IDA)

- Transfer rate of 625K bytes per second
- Modified frequency modulation recording technique
- Minimum of 57.8 million bytes of storage with a total of 173.4 million bytes of online storage using a maximum of eight disk drives

■ Selector Channels*

- Transfer rate of 825K bytes per second
- 9000 Series compatibility interface (compatible control unit interface) for up to eight drops
- Command chaining
- Capability of handling up to eight standard control units
- Each control unit capable of handling up to 16 I/O devices
- Devices serviced in burst mode

■ Multiplexer Channel* or Integrated Multiplexer Channel*

- Transfer rate of 83K bytes per second
- 9000 Series compatibility interface
- Capabilities of handling up to eight standard control units
- Uses up to eight standard subchannels
- Two modes of operation:

  Forced burst mode
  Byte-interface mode

Optional expansion feature

- System Console
  - Monitors operation of system
  - Provides keyboard and a visual display screen
  - Provides interface to connect a console printer to display unit
  - Keyboard/video display:

    64-character capability
    Program-controlled
    Protected format

  - Display screen:

    16 lines, 64 characters per line

  - Console printer*

    64-character capability
    Print rate of 30 characters per second
    Printing from system console buffer
    Printing initialized by either program or operator control unless previously modified

1.4. BASIC DEFINITIONS

1.4.1. Channels and Subchannels

A channel is the functional unit in the processor complex that controls all I/O information transfer with the attached peripheral subsystems; thus, the processor is relieved of this burden. That part of the channel required to sustain a single I/O operation is called a subchannel. A subchannel can control and modify the address, counts, and status information required for a particular I/O operation.

The IPC contains 32 subchannels, which control 5 devices and 24 communications adapters in concurrent operation on the channel. The IDA contains one subchannel which, at a given time, controls one of eight disk drive devices on the channel.

1.4.2. Subsystem, Control Unit, and Device

A subsystem is the complex of hardware units that is attached to one control unit, which in turn is attached to one of the physical selector or multiplexer channels.

A control unit is that part of the subsystem that is directly connected to the I/O channel interface and controls the transfer of data between the device and the I/O interface.

**Optional expansion feature*

A device is that part of the subsystem that handles the medium on which data is recorded or from which data is retrieved. A control unit to which only one device is attached is called a nonshared control unit. A control unit to which more than one device is attached is called a shared control unit.

Since, in many subsystems, the control unit cannot be readily distinguished from the device, the term *device* is used to refer to both. Further, when no confusion is possible (for example, nonshared control units), the terms *device* and *subsystem* are used interchangeably. Thus, the device address field of the start-I/O instruction in reality addresses the subsystem meaning the control unit and the device connected to that control unit. For example, the integrated disk subsystem functions as a shared control unit with up to eight devices (disk drives) connected to a channel containing one subchannel. The channel and control unit functions of this I/O unit are combined in a single system component designated as the integrated disk adapter.

2. Processor

2.1. PROCESSOR REGISTERS

The processor contains two register stacks with a basic usable capacity of 56 words. The A stack contains 16 working registers not program addressable, with each 32 bits in length. The B stack contains two types of registers: 32 general registers and 8 working registers. Four registers, each 64 bits in length, are used for floating-point operations in the B stack when the micrologic expansion feature is installed.

2.1.1. General Registers

The B stack contains 32 general registers. The 32 general registers are addressed in groups of 16 registers as problem or supervisor general registers under control of the PR bit (bit 13 of current PSW; see 2.6). The registers may be used for indexing purposes (except register 0), fixed-point arithmetic, logical operations, and temporary storage. Each register has a capacity of 32 bits. The registers are addressed by various 4-bit fields in the instruction being executed by the processor.

In certain instructions, two adjacent registers hold a double-word operand. In these cases, the register address specified must be an even address, and the register pair consists of the addressed register and the next higher addressed register. The even register contains the most significant half of the double word, and the odd register contains the least significant half.

2.1.2. Working Registers

Twenty-four registers are used by the microprogram for temporary storage of operands and intermediate results when executing instructions. Sixteen working registers are provided in the A stack (Table 2-1), and eight working registers are provided in the B stack (Table 2-2). Each register is 32 bits long and is not program addressable.

2.1.3. Floating-Point Registers

Four floating-point registers are available in the B stack for use with the micrologic expansion feature. Each floating-point register has a length of 64 bits, which accommodates data in either short format (one word) or long format (one double word). A short operand occupies the high-order bit positions of the register. The low-order half of the register is ignored and remains unchanged in short precision floating-point operations.

The floating-point registers are addressed by various 4-bit fields in the instruction being executed by the processor. The operation code of the instruction being executed determines whether a given 4-bit field addresses the floating-point registers or the general registers. The floating-point registers have the addresses 0, 2, 4, and 6.

Table 2-1 lists the processor register stack A allocation, and Table 2-2 lists the processor register stack B allocation, along with both program-specified register addresses and register stack addresses. Table 2-1 is included only for information purposes. There is no program specified register address and no PR bit referenced in the PSW.

Table 2—1. Processor Register Stack A Address Allocation

Working Register Number	Register Stack Address (hardware-generated)
0	000000
1	000001
2	000010
3	000011
4	000100
5	000101
6	000110
7	000111
8	001000
9	001001
10	001010
11	001011
12	001100
13	001101
14	001110
15	001111

Table 2—2. Processor Register Stack B Address Allocation

Register Number	Program-Specified Register Address	PR Bit in PSW	Register Stack Address (hardware-generated)	Identified Register
0	0000	0	000000	(Supervisor) General Register 0
1	0001	0	000001	(Supervisor) General Register 1
2	0010	0	000010	(Supervisor) General Register 2
3	0011	0	000011	(Supervisor) General Register 3
4	0100	0	000100	(Supervisor) General Register 4
5	0101	0	000101	(Supervisor) General Register 5
6	0110	0	000110	(Supervisor) General Register 6
7	0111	0	000111	(Supervisor) General Register 7
8	1000	0	001000	(Supervisor) General Register 8
9	1001	0	001001	(Supervisor) General Register 9
10	1010	0	001010	(Supervisor) General Register 10
11	1011	0	001011	(Supervisor) General Register 11
12	1100	0	001100	(Supervisor) General Register 12
13	1101	0	001101	(Supervisor) General Register 13
14	1110	0	001110	(Supervisor) General Register 14
15	1111	0	001111	(Supervisor) General Register 15
16	0000	1	010000	Problem General Register 0
17	0001	1	010001	Problem General Register 1
18	0010	1	010010	Problem General Register 2
19	0011	1	010011	Problem General Register 3
20	0100	1	010100	Problem General Register 4
21	0101	1	010101	Problem General Register 5
22	0110	1	010110	Problem General Register 6
23	0111	1	010111	Problem General Register 7
24	1000	1	011000	Problem General Register 8
25	1001	1	011001	Problem General Register 9
26	1010	1	011010	Problem General Register 10
27	1011	1	011011	Problem General Register 11
28	1100	1	011100	Problem General Register 12
29	1101	1	011101	Problem General Register 13
30	1110	1	011110	Problem General Register 14
31	1111	1	011111	Problem General Register 15
32	0000	–	100000	Floating-Point Register 0 (high-order)
33	–	–	100001	Floating-Point Register 0 (low-order)
34	0010	–	100010	Floating-Point Register 2 (high-order)
35	–	–	100011	Floating-Point Register 2 (low-order)
36	0100	–	100100	Floating-Point Register 4 (high-order)
37	–	–	100101	Floating-Point Register 4 (low-order)
38	0110	–	100110	Floating-Point Register 6 (high-order)
39	–	–	100111	Floating-Point Register 6 (low-order)
40	–	–	101000	Working Register 0
41	–	–	101001	Working Register 1
42	–	–	101010	Working Register 2
43	–	–	101011	Working Register 3
44	–	–	101100	Working Register 4
45	–	–	101101	Working Register 5
46	–	–	101110	Working Register 6
47	–	–	101111	Working Register 7

2.2. INFORMATION FORMATS

Both data and instructions are transmitted in single 8-bit increments called bytes. Up to two bytes of information may be transferred in parallel between main storage and processor or between main storage and the input/output channels. To ensure data integrity, an odd parity bit accompanies each byte transmitted to or received from main storage or transmitted over the input/output data lines. If a parity error is detected, both the processor and I/O section are capable of causing an interrupt.

Bytes are handled either separately or grouped together in fields. A half word is a field containing two consecutive bytes and is the basic building block for instructions. A word is a field containing four consecutive bytes. A double word is a field containing eight consecutive bytes, or two words. Figure 2–1 shows the various formats, their relationships, and bit numbering conventions.

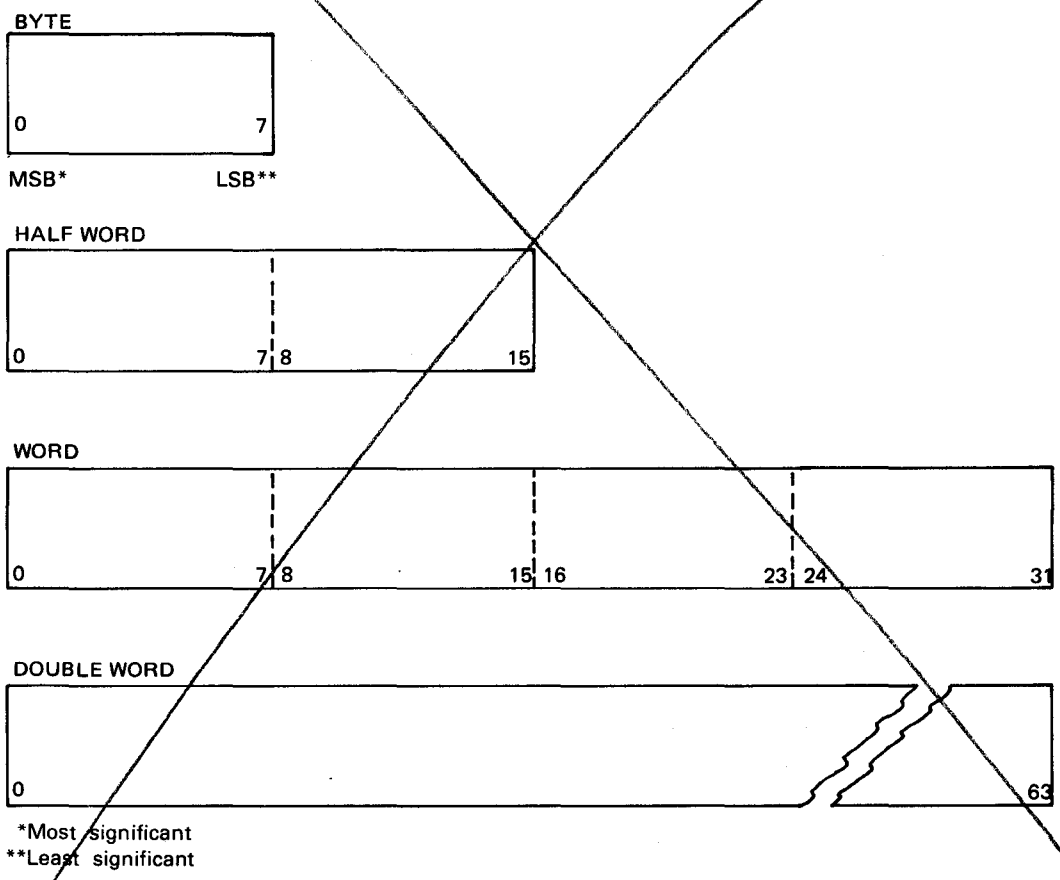


Figure 2—1. Information Formats

2.2.1. Information Addressing

Byte locations in main storage are numbered consecutively starting with 0; each number is considered to be the address of the information byte stored at that location. A group of consecutive bytes, or a field, is referenced by using the address of the leftmost byte of the field. The number of bytes in a field is either implied or explicitly defined by the operation. Variable-length operands may be located with no boundary constraints.

2.2.2. Information Positioning

Fixed-length fields (such as half words, words, and double words) are located in main storage on an integral boundary for that unit of information. An integral boundary is defined as a main storage address for a unit of information that is a multiple of the length of the unit in bytes. For example, the main storage address for a half word (two bytes) is a multiple of 2; for a word (four bytes), a multiple of 4; and for a double word (eight bytes), a multiple of 8.

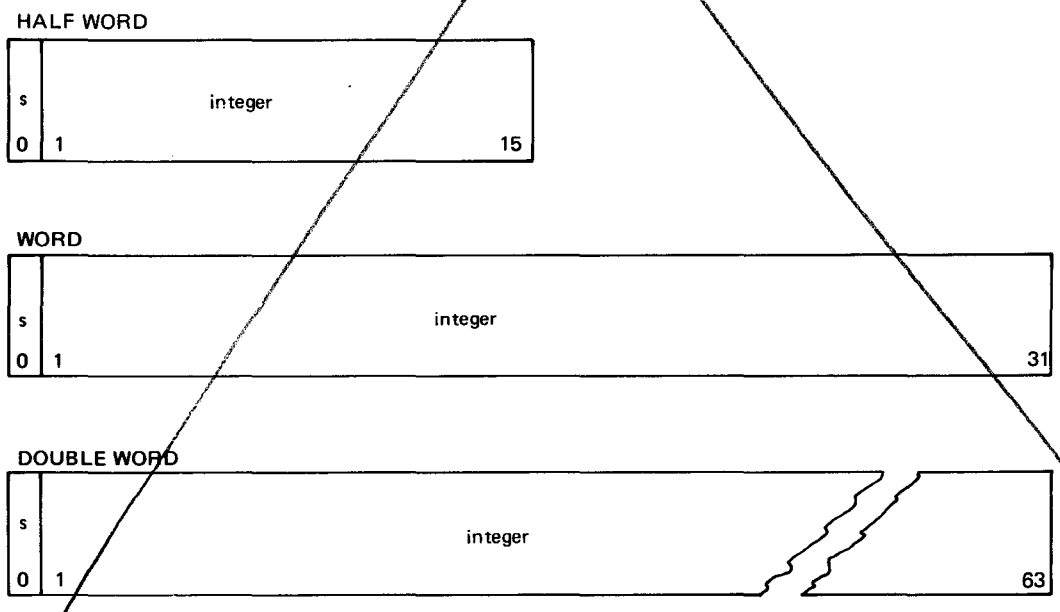
Main storage addresses are expressed in binary form within the processor. Integral boundaries for half words, words, and double words are specified as binary addresses in which one, two, or three of the low-order bits respectively are 0. Variable-length fields are not limited to integral boundaries and may start at any byte address.

2.3. DATA FORMATS

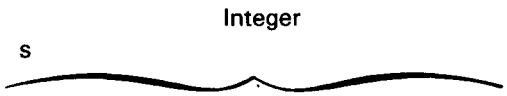
The processor manipulates data in various formats to process fixed-point numbers, decimal numbers, floating-point numbers, and logical information.

2.3.1. Fixed-Point Numbers

Each fixed-point number is represented in one of the three fixed-length formats consisting of a 1-bit sign field followed by an integer field. When the sign bit(s) is 1, the integer represents a negative value; when s is 0, the integer represents a positive value. Negative integers are represented in the twos complement notation. The half-word, word, and double-word formats are as follows:



Examples of fixed-point numbers (in word format) are as follows:



00000000000000000000000000000000	= 0
00000000000000000000000000000001	= +1
11111111111111111111111111111111	= -1
01111111111111111111111111111111	= $+2^{31}-1$ (maximum positive number)
10000000000000000000000000000000	= 2^{31} (maximum negative number)

When held in one of the 16 general registers, a fixed-point number is treated as a 32-bit operand. Certain multiply and divide operations use a 64-bit operand consisting of one sign bit followed by a 63-bit integer field. A 64-bit operand is located in two adjacent general registers and is addressed by referring to the even-numbered register of the even/odd pair. When fixed-point data is located in main storage, it is stored in any one of three formats: half-word, word, or double-word. This data is located on the integral main storage boundary of its associated format.

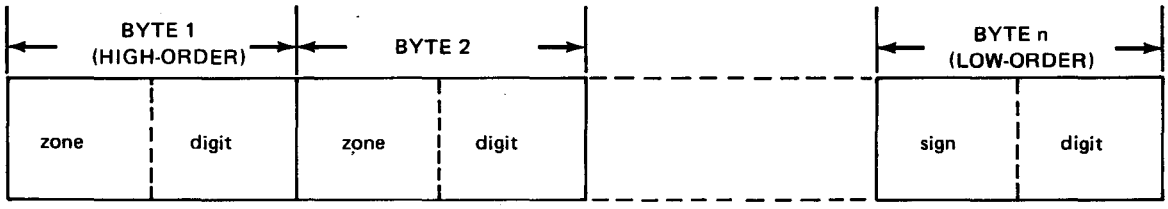
A half word in main storage is sign-extended to a word by propagating the sign through the high-order 16 bits of the word when it is transferred to the processor. It then participates as a word operand in fixed-point arithmetic operations.

2.3.2. Decimal Numbers

Decimal numbers are represented in sign absolute value form in either packed or unpacked formats, in variable lengths, and they are used in decimal arithmetic and logical operations. For decimal arithmetic operations, the numbers are processed in storage, occupy fields that may start at any byte address, and vary from one to sixteen 8-bit bytes in length. For logical operations, the numbers are processed in storage, occupy fields which may start at any byte address, and vary from 1 to 256 bytes in length.

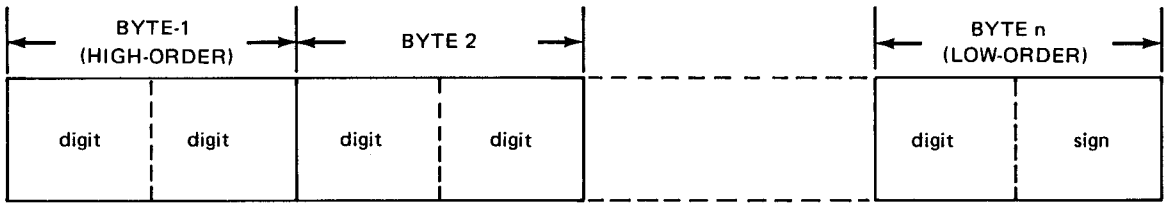
2.3.2.1. Unpacked Decimal Numbers

In unpacked decimal numbers, the low-order four bits of each byte form a decimal numeric, the digit field; the high-order four bits of each byte form a zone field. The sign of the number is in the zone field of the low-order byte.



2.3.2.2. Packed Decimal Numbers

Decimal arithmetic is performed on data in packed format only. In packed decimal numbers, each byte consists of two decimal digits and each number is treated as a signed integer, which can vary in length from 1 to 16 bytes. The sign is in the least significant digit field of the low-order byte. The source field for the edit and edit and mark instructions is represented in packed decimal format, with the exception that no sign is required in the low-order byte.



2.3.2.3. Digit Code Representation

The standard binary code is used for digits 0 through 9. All other code combinations represent a sign convention in either ASCII or EBCDIC and are as follows:

Sign	Binary Code	Hexadecimal Code
+(ASCII/EBCDIC)	1100	C
-(ASCII/EBCDIC)	1101	D

The following externally generated zone codes are recognized as signs during decimal arithmetic operations.

Sign	Binary Code	Hexadecimal Code
+	1010	A
-	1011	B
+	1110	E
+	1111	F

The 0011 code has a dual function: It represents the numeric 3 during arithmetic operations, and it is the processor-generated zone code during the execution of the unpack and edit instructions in the ASCII mode.

The 1111 code is the internal zone code generated during the execution of the unpack and edit instructions in the EBCDIC mode.

Examples of representative decimal numbers as generated by the processor are as follows:

- Unpacked

Zone	Digit	Zone	Digit	Sign	Digit		
1111	0000	1111	0000	1100	0000	=	+000 (EBCDIC mode)
1111	0000	1111	0000	1101	0000	=	-000 (EBCDIC mode)
0011	0100	0011	0010	1100	0011	=	+423 (ASCII mode)
3	4	3	2	+	3		(decimal, zone, sign)

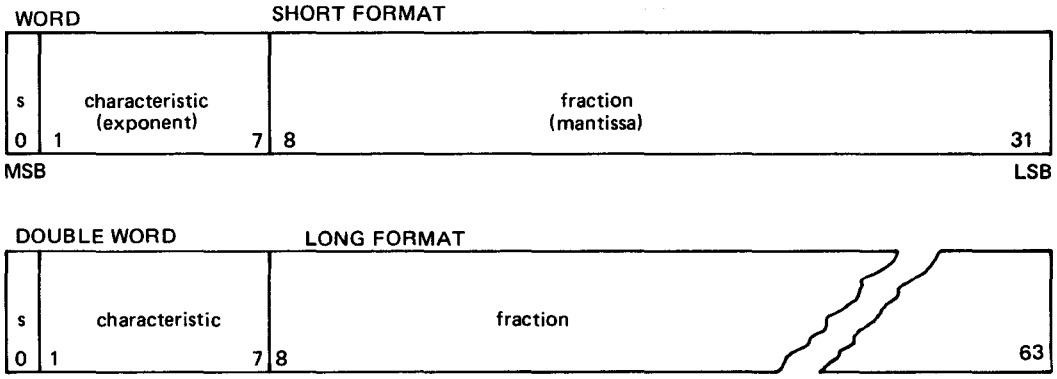
- Packed

Digit	Digit	Digit	Sign		
0000	0000	0000	1100	=	+000 (EBCDIC/ASCII)
0000	0000	0000	1101	=	-000 (EBCDIC/ASCII)
0100	0010	0011	1100	=	+423 (EBCDIC/ASCII)
4	2	3	+		(decimal)

2.3.3. Floating-Point Numbers

Floating-point numbers are represented in signed absolute value form and have a fixed-length format that is either a word (short format) or a double word (long format). Both formats are used in storage or in the floating-point registers. In either format, the s bit (bit 0) is the sign bit of the fraction (mantissa), and bits 1 through 7 are the characteristic exponent. The fraction field consists of bits 8 through 31 in the short format and bits 8 through 63 in the long format, that is, 6 and 14 hexadecimal digits, respectively.

The characteristic is a biased exponent expressed in excess-64 binary notation; the fraction is expressed as a hexadecimal number having the radix point to the left of the high-order fraction digit. The quantity expressed by the full floating-point number is the product of the fraction and the number 16 raised to the power of the characteristic minus 64. The short and long formats are:



Examples of floating-point numbers (in short format) are as follows:

Characteristic Fraction
s

00000000000000000000000000000000 = 0 (true zero)
01000001000100000000000000000000 = $+16^{1} \times 2^{-4} = +1$
11000001000100000000000000000000 = $-16^{1} \times 2^{-4} = -1$
01111111111111111111111111111111 = $+16^{63} \times (2^{-1} + 2^{-2} + \ldots + 2^{-24})$ = (max positive number)
11111111111111111111111111111111 = $-16^{63} \times (2^{-1} + 2^{-2} + \ldots + 2^{-24})$ = (max negative number)

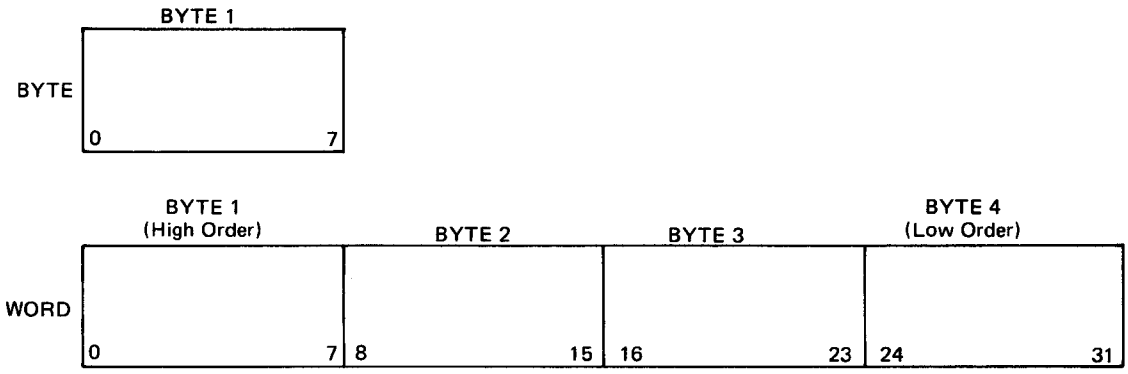
2.3.4. Logical Information

Logical information is processed as fixed- or variable-length data. Operations using this information include moving, comparing, translating, editing, bit testing, bit manipulation, and shifting.

Fixed-length data, consisting of one to four bytes, is processed in the general registers. Data in the general registers normally occupies all 32 bits. Bits are treated uniformly, with no distinction made between the sign and numeric bits. In some operations, only the low-order bits of a register are used, with the remaining 24 bits left unchanged. The load-address instruction loads a 24-bit address into a general register, and the remaining high-order eight bits are cleared to 0.

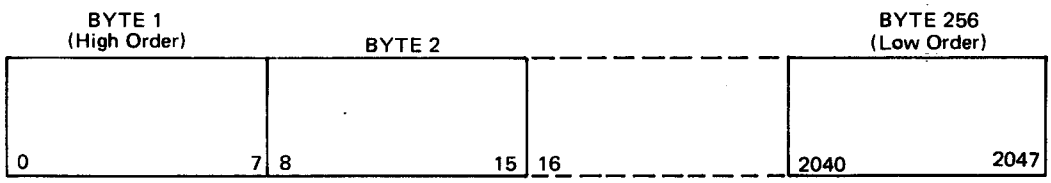
In storage-to-register operations (2.4), the storage data occupies either a word, a half word, or a single byte. The word must be located on an integral boundary and the half word on a word or half-word boundary.

The fixed-length format for logical data is:



Variable-length data can have up to 256 bytes of alphabetic or numeric character codes (alphanumeric data). This information is processed in storage and occupies fields that may start at any byte address The processing order of variable-length data is from the high-order byte to the low-order byte.

The variable-length format for logical data is:



2.4. INSTRUCTION FORMATS

Programmed instructions for processing fixed- or variable-length data can vary in format and length. Generally, the format used is dictated by the operation to be performed and the operand location. Operands may be located in storage, in general registers, or in the instruction itself. The length of an instruction depends upon its function and is either two bytes (one half-word), four bytes (two half-words), or six bytes (three half-words). The instruction formats (Figure 2–2) are:

Format Code	Name
RR	Register-to-register instructions
RX	Register-to-indexed-storage instructions
RS	Register-to-storage instructions
SI	Storage-to-immmediate operand instructions
SS	Storage-to-storage instructions

Each format consists of an operation code (opcode) and two or more fields, which specify, among other things, the addresses of operands. Each field is identified by a letter followed by a subscript numeral. The numeral denotes the operand (1, 2, or 3) to which the field applies. The format code (RR, RX, RS, SI, SS) denotes the general operation to be performed. Blank fields are ignored by the hardware.

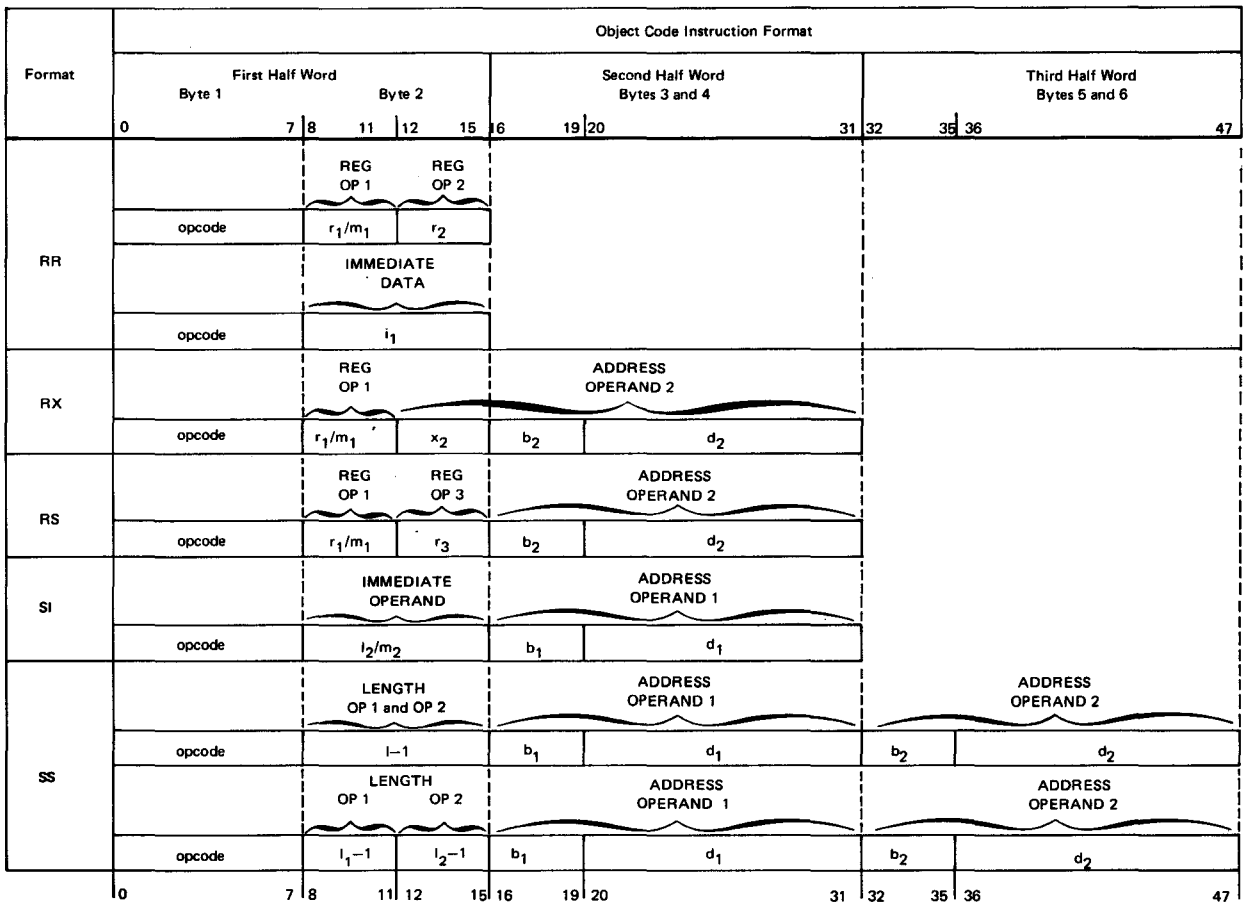


Figure 2—2. Instruction Formats (Part 1 of 2)

Symbol	Definition
opcode	The instruction operation code
$r1/m_1$	The number of a general or floating-point register that contains the first operand. In the branch on condition (BCR) instruction, m identifies a mask. It may specify where the result of an operation is to be stored, if required. In RS format, it may contain a number that represents a boundary for general register use.
r_2	The number of a general or floating-point regsiter that contains the second operand. r_1 and r_2 may specify same address if same operand is used. In set program mask (SPM) instruction, r_2 is not used. In service timer register (STR) instruction, r_2 is used as an opcode modifier.
i_1	An 8-bit byte of immediate data for the supervisor call (SVC) instruction only
x_2	The number of a general register that contains an index value. When x_2 is 0, the index value is 0.
b_2	The number of a general register that contains an index value representing the base address of the second operand. When b_2 is 0, the base address is 0.
d_2	The 12-bit displacement value that, when added to the contents of b_2x_2, represents the address of the second operand. If a load address (LA) instruction is being executed, the final address is stored in a general register rather than being used to address main storage. In RS format and for shift instructions only, it represents the number of bits of shifting to be accomplished. In SS format, it represents the address of the leftmost (high-order) byte of the second operand.
r_3	The number of a general register that contains the third operand or a number that represents a boundary for general-register use. This field is ignored when executing shift instructions.
I_2/m_2	Contains the second or immediate operand, a mask, or a secondary operation code. In the set system mask (SSM) and start I/O (SIO) instruction, I_2 is not used.
b_1	The number of a general regsiter that contains an index value representing the base address of the first operand. When b_1 is 0, the base address is 0.
d_1	The 12-bit displacement value that, when added to the contents of b_1, represents the address of the first operand. In the I/O instructions, d_1 specifies a channel and device number.
I_1	Contains a 4-bit number specifying the number of additional bytes in the first operand field to the right of the leftmost byte.
I_2	Contains a 4-bit number specifying the number of additional bytes in the second operand field to the right of the leftmost byte.
I	For instructions with Dx opcodes, combine I_1 and I_2 to form an 8-bit number that specifies the number of additional bytes to the right to the operand address for both operands. For the translate (TR), translate and test (TRT), and edit (ED) instructions, I applies to the first operand only.

Figure 2—2. Instruction Formats (Part 2 of 2)

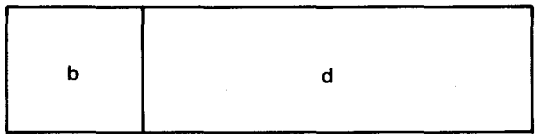
2.4.1. Operand Addressing

Operands may appear in one of three places:

1. contained within the instruction;
2. stored in the operating registers; or
3. located in main storage.

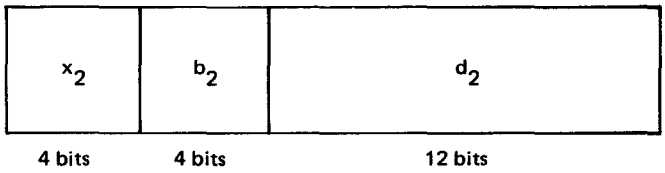
No address is needed for an operand that is part of the instruction (immediate operand). When operands are located in the register set, the register is addressed by using a 1-digit hexadecimal code. However, operands in main storage must be addressed through the rightmost 17 bits of a 24-bit logical address. Formation of this address follows.

Instructions that access operands from main storage have two operand specification fields:



The b field of the instruction designates a general purpose register. The rightmost 17 bits of this register represent the base address. These 17 bits are treated as a positive binary number. The 12 bits of the d field contain the displacement, which is also treated as a positive binary number. The base address and the displacement are added together to obtain the effective address of the operand.

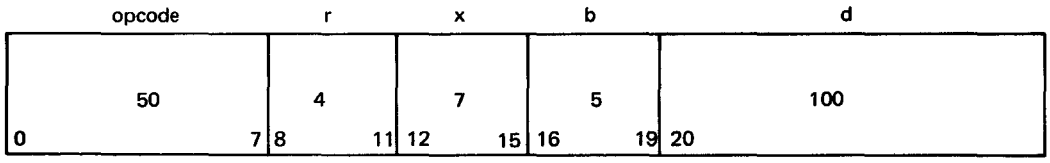
The RX type of instruction contains three fields to specify the operand address:



The x_2 field of the instruction designates one of the general purpose registers containing an index, which is treated as an 18-bit positive binary number and is added to the base address plus the displacement to obtain the effective address. When the b_2 or x_2 fields of an instruction contain 0's, no register is accessed, and the field is not considered in calculating the effective address.

An example of this addressing:

> Store the contents of register 4 (r) in the effective address obtained by adding the contents of base register 5 (b) and the contents of index register 7 (x) to the displacement value 100 (d).



This is a store instruction: Operand 1 is stored in the effective address of operand 2.

Before execution:

- Content of register 4 is 32.
- Content of register 5 is 48.
- Content of register 7 is 52.
- Content of storage location 200 through 203 is 21.

After execution:

- The contents of the registers remain the same.
- Effective address is 48 + 52 + 100 = 200.
- Content of storage location 200 through 203 is 32.

2.5. INSTRUCTION TYPES

There are two general categories of instructions related to a processor state:

1. Supervisor (privileged) instructions
2. Problem (nonprivileged) instructions

Each instruction is identified by an operation code that appears as two hexadecimal digits in the 8-bit opcode field.

2.5.1. Supervisor (Privileged) Instructions

The supervisor instructions, when executed, give rise to special priority processor activity. When the processor is in this state, the PS bit (bit 14) of the program status word (2.6) is 0, and all installed instructions are valid. The processor can be switched from one state to the other by conditioning it with a new program status word whose PS bit is set to 1; thus, the instruction can be placed in the problem state. This conditioning may be accomplished by executing the load PSW instruction. Since this is a privileged instruction, the processor previously must have been in the supervisor state. The switching of states may also occur as a result of an interrupt condition that causes a new program status word to be obtained from storage.

2.5.2. Problem (Nonprivileged) Instructions

The problem instructions are those that are available to the program for normal data processing when the processor is in the problem state (PS bit set to 1). If the program attempts to execute a privileged instruction, a program exception interrupt occurs.

2.6. PROGRAM STATUS WORD FORMAT

The program status word (PSW) is a double word containing an instruction address and various control fields that establish operating modes, specify protection and relocation keys, hold program branching conditions, and assist in analyzing causes of interrupts. A running program is under control of the current PSW that is stored in the processor hardware.

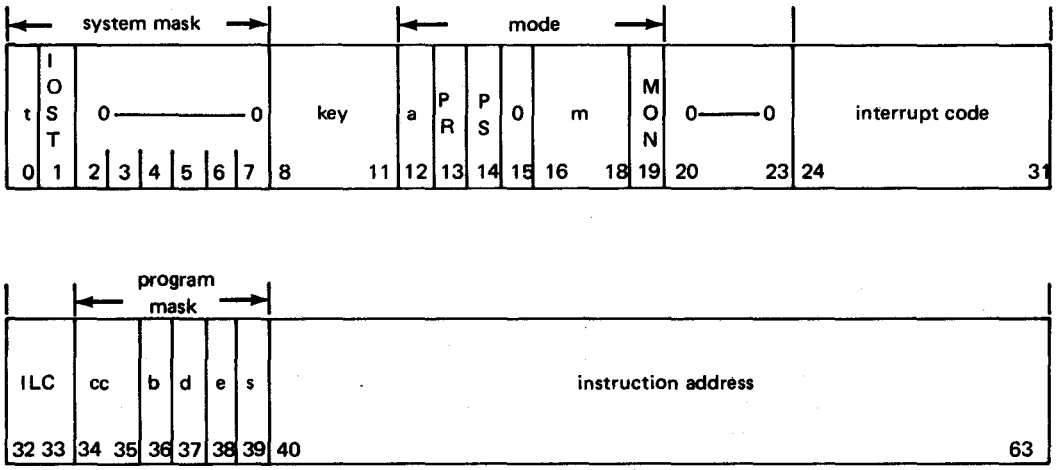
All or part of a PSW may be loaded into the current PSW register by various instructions and interrupts. Program access to the entire current PSW is made only by an interrupt that stores the current PSW in low order main storage. An entire PSW may be loaded into the current PSW register by the processor accepting an interrupt or by executing a load PSW instruction.

The system mask (eight bits) may be loaded separately by the set system mask instructions. The program mask field (six bits) may be loaded by the set program mask instruction. The ILC, cc, and instruction address fields are continuously updated according to the instruction stream being executed. The low order 32 bits of the current PSW may be stored by executing the branch and link instructions.

All 64 bits of the current PSW are cleared to 0 when the RUN switch is pressed after a system reset occurs. The initial program load function tests for device-end status and for error conditions. When device-end is detected, status is tabled via the IOST, and an IOST IIS is entered, overriding the IOST mask bit.

2.6.1. Program Status Word Format and Description

The format and description of the PSW are:



System mask (bits 0–7)

Permits two of six interrupt levels in the processor to be masked. If a mask bit is set to 1 and all other conditions are satisfied, the interrupt corresponding to the mask bit occurs. As long as the mask bit is set to 0, the corresponding interrupt does not occur. See 2.7 for a description of interrupts. The individual bits are defined as follows:

t (bit 0)

Timer level interrupt mask

IOST (bit 1)

I/O status tabler level interrupt mask (channel 7)

Bits 2–7, 15, and 20–23

Unassigned and must be set to 0 by the software.

Key (bits 8–11)

Contains the processor storage protection and relocation key. Bit 8 must always be set to 0. Bits 9–11 correspond to the key. See 4.7 for a complete description of this key.

Mode (bits 12–19)

Establish the various processor operating modes.

a (bit 12)

Defines whether the processor operates in ASCII or EBCDIC mode:

a = 1	ASCII mode
a = 0	EBCDIC mode

Certain processor instructions interpret or generate code-sensitive characters. This mode defines whether the characters are expressed in ASCII or EBCDIC. The unpack, edit, and mask instructions generate code-sensitive zones as follows:

ASCII zone	0011 (binary)
EBCDIC zone	1111 (binary)

The edit instruction detects the following code-sensitive control characters:

	ASCII	EBCDIC
Digit select	1000 0000	0010 0000
Significant start	1000 0001	0010 0001
Field separator	1000 0010	0010 0010

PR (bit 13)

Defines the general register to be used in executing an instruction. The processor contains two sets of 16 general registers: the problem general registers and the supervisor general registers. The selection as to which set of general registers is to be used in executing an instruction is determined by the PR bit setting as follows:

PR = 1	Problem general registers are selected.
PR = 0	Supervisor general registers are selected.

PS (bit 14)

Specifies one of two processor modes of operation:

PS = 1 Problem mode
PS = 0 Supervisor mode

In problem mode, all privileged instructions are invalid and their attempted execution results in a program exception interrupt. In supervisor mode, all installed instructions may be executed.

m (bit 16–18)

Specifies one of three mutually exclusive modes. Undefined codes in this field should be avoided to prevent indeterminate operation. The field is defined as follows:

Mode	Code	Description
Native	000	Specifies 90/30 native mode of operation (Appendix B)
SPERRY UNIVAC 9200/9300	001	Specifies the 9200/9300 compatibility mode, which permits the running of programs written for 9200/9300 systems
IBM 360/20	010	Specifies the 360/20 compatibility mode, which permits the running of programs written for the 360/20 systems

MON (bit 19)

Specifies mode selection as follows:

MON = 1 Monitor mode
MON = 0 Normal execution

The monitor mode is used to generate a monitor level interrupt prior to the execution of every instruction in a running program. Thus, the system software may perform dynamic statistical analysis of programs.

Interrupt code (bits 24–31)

Contains the interrupt code used in software analysis of interrupts. See 2.7 for a complete definition of the contents of this field.

ILC (bits 32, 33)

Contains the instruction length code used after an interrupt occurs to indicate the length, in half words, of a suppressed or terminated instruction. This field, in conjunction with the instruction address field, is used to calculate the beginning address of the address of a suppressed or terminated instruction. For further details, see 2.7.3.

Program mask (bits 34–39)

Contains the condition code (cc) and four individual mask bits for specific program exceptions. See 2.7.6 for further description of this field.

cc (bits 34, 35)

Contains the condition code reflecting the results of an operation. The value of cc may be tested by various branch instructions. The conditions under which cc is set are listed as part of the description of each instruction and are found in the assembler user guide, UP-8061 (current version).

b (bit 36)
When set, permits a fixed-point overflow program exception

d (bit 37)
When set, permits a decimal overflow program exception

e (bit 38)
When set, permits a characteristic (exponent) overflow program exception. The required optional feature must be installed.

s (bit 39)
When set, permits a significant program exception. The required optional feature must be installed.

Instruction address (bits 40–63)
Contains an instruction address. When a new PSW is loaded into the current PSW register, this address (after being relocated) points to the next instruction to be executed. Thereafter, this field of the current PSW is updated as each instruction is executed. When the current PSW is stored, it is pointing to the instruction that would have been executed next had the interrupt or BAL not occurred. Branch instructions replace the instruction address field in the current PSW in the case of a successful branch.

2.7. INTERRUPTS

The interrupt system is the automatic means provided to alert the processor to an exception or unexpected condition, the end of an I/O operation, program errors, hardware errors, and various monitoring operations. Following the detection of such an event, the interrupt system directs the processor to the appropriate program. Also, interrupts make it possible to suspend a task in order to take on a task of higher priority.

2.7.1. Interrupt Initialization Sequence

The PSW specifies much of the system environment for a given program. When an interrupt occurs, this environment is preserved by storing the current PSW into the old PSW location in low-order main storage. When an interrupt occurs, the interrupt code that carries detailed information pertaining to the interrupt is transferred to the old PSW location corresponding to the interrupt level. The new PSW, also held in low-order storage, establishes a new program environment when it is transferred to the current PSW. The current PSW becomes the old PSW when stored. A new PSW corresponding to the interrupt level is then read and transferred to the current PSW register.

After the interrupt has been processed, the system environment can be restored to what it was prior to the interrupt by transferring the old PSW to the current PSW register. This is accomplished by issuing the load PSW instruction. The procedure of switching out one program and switching in another is termed an *interrupt initialization sequence* (IIS). If a particular interrupt level is enabled (the corresponding mask bit in the current PSW is set to 1), an IIS is executed for this level as follows:

1. The hardware generates an interrupt request for this level.

2. Depending on the interrupt type, the processor completes, suppresses, or terminates the instruction currently being executed.

3. The interrupt request for this level is detained until the hardware priority grants this level permission to proceed. Occassionally, the interrupt request may be lost if a higher priority IIS is executed first.

4. If the source of the interrupt request is the processor, then the appropriate interrupt code is placed in bit positions 24 to 31 of the current PSW.

5. If the source of the interrupt request is the I/O status tabler (IOST), the monitor or the timer level (bits 24 to 31 of the old PSW) are cleared to 0.

6. The instruction address (bits 40 to 63 of the current PSW) is converted from absolute to relative. The current PSW then is placed into the old PSW location for this level.

7. The new PSW for this level is read and placed in the current PSW register. Bit positions 32 and 33 (ILC) of the current PSW are set to 0 by the hardware.

8. A new relocation value, determined by the new key in the current PSW, is read from low-order storage and placed in the current relocation register, and the instruction address in the current PSW is converted from relative to absolute.

9. The hardware priority circuit reexamines the state of all interrupt request lines. Two cases result:

   a. An interrupt request line is indicating a pending interrupt and its corresponding mask bit is a 1. In this case, this procedure is repeated beginning with step 3.

   b. No further interrupt requests are pending and enabled. In this case, the processor resumes instruction execution under control of the current PSW.

2.7.2. Interrupt Request and Handling Priority

The system provides six levels of interrupts, each level having an old PSW and a new PSW located in low-order storage. The IOST and interval timer levels also have associated mask bits that control whether a corresponding interrupt request is accepted. The IOST mask bit also controls the equipment check interrupt request. The mask bits appear in bit positions 0 through 7 of the system mask of the current PSW. The monitor interrupt level is controlled by the MON bit (bit 19 in the mode field of the PSW). Also, bits b, d, e, and s (bits 36 through 39 respectively of the current PSW) control the masking of unique interrupt conditions in the program exception level.

No PSW masking is provided for the remainder of the program exception interrupts, the machine check, or the supervisor call levels. Since, by definition, the supervisor call requests a program-controlled interrupt, it need not be masked.

However, a hardware-controlled mask is provided for the two remaining levels, IOST and monitor. The hardware masking operates as follows:

- In exiting from a program exception or a machine check level IIS, the hardware stores the exit condition by setting their respective program masks: the program exception hardware mask (PX) or the machine check hardware mask (MC). The masking is cleared only upon the execution of a load PSW (LPSW) instruction, issued by the corresponding interrupt routine that is used to reinstate the machine environment to its original state prior to entering the respective IIS level. Should both the MC and PX hardware masks be set, the MC mask is cleared by the first LPSW. The PX mask is cleared by the execution of a subsequent LPSW unless the MC mask is again set.

- The PX mask, when set, interprets any further nonmasked program exceptions as machine check types of interrupt.

- The MC mask, when set, causes a nonrecoverable error condition if any processor machine check or any program exception occurs. (See 2.7.4)

- The MC or PX mask, when set, blocks the equipment check or IOST machine check interrupt request. (See 2.7.5.2 and 2.7.5.3.)

For each interrupt level, there are one or more interrupt requests and one or more interrupt flip-flops. When an interrupt flip-flop is set by an interrupt request at a time determined by the processor hardware, an interrupt occurs by the execution of an IIS. Interrupt requests for the IOST and the internal timer can be blocked from setting the corresponding interrupt flip-flop by means of a mask bit in the current PSW. Interrupt requests thus blocked can be held pending until accepted, or they may be rejected and the request dropped. A list of interrupt levels is as follows:

1. Machine check

2. Program exception

3. Supervisor call

4. Timer

5. IOST

6. Monitor

If multiple interrupt requests are present simultaneously and their corresponding mask bits, if any, are set to 1, the appropriate interrupt flip-flops also will be set to 1 simultaneously. A hardware priority circuit then selects which IIS is to be executed. At the conclusion of the IIS, the interrupt flip-flops are cleared, and the interrupt request lines are again sampled. The mask bits placed in the current PSW by the IIS just completed are effective at this time. The priority circuit establishes the priorities shown in Table 2–3. It should be noted that the occurrences of the first three interrupt requests are mutually exclusive.

Since the interrupt request lines are sampled at the end of each IIS, any interrupt requests generated during an IIS are honored at the end of that IIS, subject only to the setting of the current mask bits. The final contents of the current PSW determine which interrupt subroutine is entered first. Therefore, the order of executing these subroutines is the reverse order of executing IISs.

Although a hardware priority exists for resolving simultaneously presented interrupt requests, it is stressed that the order of executing continuous IISs is under control of the system mask bits and interrupt requests that may be generated during the execution of IISs.

Within a given level, an interrupt request can be generated by one or more independent conditions. A hardware priority circuit establishes the condition to be stored in the interrupt code field of the old PSW. In some levels, the specific condition generating the interrupt request changes the interrupt priority of the class itself as shown in Table 2–3. For example, the machine check level has an interrupt priority 1, 4, or 5, depending on the condition generating the interrupt request.

Table 2—3. Interrupt Request Handling Priority

Order of Priority	Interrupt Request
1	Processor machine check
2	Program exception
3	Supervisor call
4	Equipment check (machine check level)
5	IOST machine check (machine check level)
6	Timer
7	IOST
8	Monitor

An IIS takes place only at the end of an instruction execution and before a new instruction is started. Either the processor waits until the end of the instruction before executing the IIS or it ends the instruction early and then executes the IIS. If the processor waits until the end of the current instruction before honoring an interrupt request, the instruction execution is said to be completed. If the processor ends the instruction early in order to honor the interrupt request, the instruction execution is said to be either suppressed or terminated. In the case of suppression, the effect is as if no operation were specified. However, the instruction address and ILC of the current PSW have been updated. Results are not stored and the condition code is not changed. If the execution is terminated, all, part, or none of the result can be stored and, therefore, the result becomes unpredictable. In general, the results should not be used for further computation.

The ILC setting (bits 32 and 33 of PSW) is used to indicate whether an instruction has been completed, suppressed, or terminated. The processor machine check and program exception levels are the only two that may cause an instruction to be suppressed or terminated.

2.7.3. Address Stored in Old PSW

During the handling of certain interrupts, it may be desirable to locate the instruction that was being executed at the conclusion of which the interrupt was honored. The next instruction address (bits 40 through 63 of the old PSW) and the ILC (bits 32 through 33) may be used for this purpose.

A machine check, a program exception, or a supervisor call IIS stores, in the old PSW, the location of the instruction that would have been executed next had the interrupt not occurred. The ILC reflects the length of the current instruction being interrupted as follows:

ILC	PSW Bits		Subtract from old PSW (bits 61-62)
	32	33	
0	0	0	None
1	0	1	One half word
2	1	0	Two half words
3	1	1	Three half words

For the three remaining levels (timer, IOST, and monitor), their respective IISs store, in the old PSW, the location of the instruction that is to be executed next (prior to entering the IIS), and a display of an ILC of 0 is made, since honoring the interrupt occurs after the instruction is completed.

The ILC placed in the program exception old PSW for operating exceptions is always determined by examining the two most significant bits (0 and 1) of the byte that caused the interrupt. The state of the instruction length code is as follows:

ILC	PSW (bits 32-33)	Most Significant Bits of Staticized Byte	Format Implied
1	01	00	RR
2	10	01	RX
2	10	10	RS, SI
3	11	11	SS

An unassigned opcode has no valid instruction length but, for interrupt signalling, it is assumed to have the same ILC as the format group (i.e., RR, RX, etc) that it would belong to if it were designated as a valid opcode.

The instruction that generates the interrupt request can be located by subtracting the ILC from bit positions 61 and 62 of the instruction address in the old PSW. This is true whether the instruction was suppressed or terminated. In the case of a supervisor call (SVC) instruction, since the interrupt is part of the SVC instruction execution, the ILC stored is set to 1.

In cases where the processor machine check or program exception interrupts occur while the current instruction is being fetched (staticized), further fetches and instruction execution are immediately suppressed. The relation between the instruction address and the ILC stored by the respective IIS remains such that, with subtraction of bit positions 31 and 32 of the old PSW (ILC or partial ILC) from bit positions 61 and 62 of the old PSW instruction address, the formulated address then points to the first byte of the instruction being fetched.

2.7.4. Nonrecoverable Errors

It is possible that certain occurrences of multiple hardware faults or single faults in critical areas can make program detection and recovery impossible. In these cases, the action taken by the processor is unpredictable. Upon detection of these faults, the hardware takes the following action:

1. Places system in cycle mode
2. Sends SYSTEM RESET signal to the I/O channels
3. Lights PROCESSOR CHECK indicator on the system maintenance panel. The processor stops at a point as close as possible to where the nonrecoverable error condition was detected.

The following conditions are defined as nonrecoverable errors:

- A control storage addressing exception or a control storage parity check except for the write to control only storage (COS) during the load control storage (LCS) instruction
- The following errors detected during the execution of an IIS, a load PSW instruction, or a set system mask instruction:
  - Address check
  - Storage parity check

- Protection exception
- Addressing exception

- The generation of a processor machine check or a nonmasked program exception interrupt request when the hardware machine check mask (MC) is set
- The presence of a storage hold check indication from internal to external storage
- The I/O status tabler (IOST) detecting a channel time-out or a channel buffer word error during initial load control storage (ILCS) or initial program load (IPL)
- The detection of a power fault (ac, dc, blower, or overtemperature) in the external storage cabinet, communications cabinet, or expansion cabinet
- The generation of any unexpected interrupt request, such as a machine check, or program exception during the initial load control storage (ILCS) or initial program load (IPL)

The PROCESSOR CHECK indicator on the system maintenance panel may be extinguished by pressing the RUN switch. Processor operation, subject to the effect of the fault condition existing, is resumed in the cycle mode.

The action taken upon the detection of a nonrecoverable error can be inhibited by means of the INHIBIT PROCESSOR CHECK switch on the system maintenance panel. (This switch is for maintenance purposes only.) The effect of operating this switch is to block the detection of a nonrecoverable error and to allow the system to operate, while still being subject to the effect of the existing fault condition.

2.7.5. Machine Check Level

The machine check level of interrupt is used to notify a portion of the operating system of the detection of a hardware failure. This detection can result from the use of main storage, certain stall conditions, or an early warning temperature fault detected in the system.

The condition generating the machine check interrupt request can be one of three classes: processor machine check, equipment, and IOST machine check. These three classes have different interrupt request priorities, as shown in Table 2–3.

2.7.5.1. Processor Machine Check Class

The detection of a processor machine check immediately suppresses or terminates the current instruction being executed, and the machine check IIS is entered if the hardware machine check mask (MC) is in the cleared state. The MC mask being set causes a nonrecoverable error.

The interrupt codes for this class are as follows:

Interrupt Code	Interrupt Cause
1110 0110	A control storage write bus check occurs during the execution of the LCS instruction.
1110 0111	A main storage parity check occurs on any processor reference except those that generate a nonrecoverable error.
1110 1000	An address check occurs on any processor reference except those that generate a nonrecoverable error.
1110 1100	A program exception interrupt request occurs and the program exception hardware mask is set.
1110 1111	The processor stall timer expires during processor instruction execution or IIS execution.

2.7.5.2. Equipment Check Class

The equipment check class interrupt request is held pending if the hardware machine check class (MC) is set to 1, the hardware program exception mask (PX) is set to 1, or the IOST system mask (bit 1 of the current PSW) is set to 0. If the IOST system mask is set to a 1 and the MC and PX hardware masks are cleared, the equipment check interrupt request is allowed. When allowed, the interrupt request enters an IIS for the machine check level, based on the priority given in Table 2-3. This class of interrupt is honored only after the instruction is completed. The interrupt code for this class is as follows:

Interrupt Code	Interrupt Cause
1110 0000	Early warning temperature fault detected anywhere in the system

2.7.5.3. IOST Machine Check Class

The IOST machine check class interupt request is held pending whether or not the program exception mask (PX) or hardware machine check mask (MC) is set. This class of interrupt is honored only after the instruction has been executed.

The interrupt codes for this class are as follows:

Interrupt Code	Interrupt Cause
0000 0101	An addressing exception or protection exception occurs when accessing an input/output status table control word (IOSTCW).
0000 1000	An address check or storage parity check occurs on accessing an IOSTCW.
0000 1111	The processor stall timer expires during an IOST tabling sequence.

2.7.6. Program Exception Level

A program exception level interrupt request occurs when the hardware detects improper specification or use of instructions and data. Interrupt requests at this level suppress or terminate the instruction currently being executed. The program exception IIS is then entered if the program exception hardware mask (PX) is not set. In some cases, additional mask bit control exists. For these cases, not only must the PX mask be cleared but the appropriate program mask bits b, d, e, or s (bits 36 through 39, respectively, of the current PSW) must be set to 1. If a program exception interrupt request is generated and the PX mask is set, a machine check interrupt request is generated and the cause of the program exception is lost. For those instructions that examine both the system mask and the program mask, the following table summarizes the result if the instruction generates an interrupt request.

Program Mask	Hardware Mask PX	Hardware Mask MC	Result
0	X	X	Instruction processing continues with no pending interrupt request, no nonrecoverable error, and no program exception interrupt.
1	X	Set	Nonrecoverable error
1	Cleared	Cleared	Program exception IIS and set PX hardware mask
1	Set	Cleared	Machine check IIS and set MC hardware mask

NOTE:

X denotes hardware mask is ignored.

The interrupt codes for this level are as follows:

Interrupt Code	Interrupt Cause
0000 0001	Operation exception: An illegal operation has been attempted or an operation using a noninstalled processor feature has been attempted.
0000 0010	Privileged operation exception: A privileged operation has been attempted by a program operating in the problem mode (PS, bit 14 of current PSW, set to 1).
0000 0011	Execution exception: The subject instruction of an execute instruction is an execute instruction.
0000 0100	Protection exception: A storage protection violation occurs on a program-generated address when the storage protect feature is installed.
0000 0101	Address exception: A main storage location outside the range of the installed main storage is referenced by a program-specified address. For the load-control-storage (LCS) instruction only, the referenced control storage location is nonexistent.

Interrupt Code	Interrupt Cause
0000 0110	Specification exception: ■ The unit of information referenced is not on an appropriate boundary. ■ An invalid modifier field is specified in the service timer register (STR) instruction. ■ The r_1 field of an instruction that uses an even/odd pair of registers (64-bit operand) does not specify an even register. ■ A floating-point register other than 0, 2, 4, or 6 is specified. ■ A multiplier or divisor in decimal arithmetic exceeds 15 digits and sign. ■ The first operand field is shorter than, or equal in length to, the second operand in decimal, multiply, and divide instructions. ■ The four low-order address bits specified by the contents of r_2 comprise a set storage key (SSK) or insert storage key (ISK) instruction and are not equal to 0. ■ The function specified by the I_2 field of a diagnose instruction was not loaded in the transient area of control storage. ■ A SOFTSCOPE instruction (SSFS or SSRS) was issued without the supporting microcode loaded in the control storage transient area.
0000 0111	Data exception: ■ An invalid sign or digit code is detected in decimal operands. ■ Fields in decimal arithmetic overlap incorrectly. ■ The first operand of the multiply decimal instruction does not have sufficient number of high-order 0 digits.
0000 1000	Fixed-point overflow exception: A fixed-point add or subtract operation exceeds the capacity of the first operand field. This interrupt is masked by b, bit 36 of the current PSW.
0000 1001	Fixed-point divide exception: The quotient of a fixed-point divide operation exceeds the capacity of the first operand (including division by 0), or the result of a convert-to-binary instruction exceeds 31 bits.
0000 1010	Decimal overflow exception: The result of an add decimal, subtract decimal, or zero-and-add instruction exceeds the capacity of the first operand location. This interrupt is masked by d, bit 37 of the current PSW.
0000 1011	Decimal divide exception: The quotient of a divide decimal (DP) instruction exceeds the capacity of the quotient part of the first operand field.

Interrupt Code	Interrupt Cause
0000 1100	Exponent overflow exception: The final characteristic resulting from a floating-point arithmetic operation exceeds 127.
0000 1101	Exponent underflow exception: The final characteristic resulting from a floating-point arithmetic operation is less than 0. This interrupt is masked by e, bit 38 of the current PSW.
0000 1110	Significance exception: The final fraction resulting from a floating-point addition or subtraction is equal to 0. This interrupt is masked by s, bit 39 of the current PSW.
0000 1111	Floating-point divide exception: The divisor fraction in a floating-point divide operation is equal to 0.

2.7.7. Monitor Level

This level of interrupt is used by the monitor mode. The system mask does not apply at this level, and 0's are inserted in the associated interrupt code. The MON bit (bit 19 of the current PSW) specifies the monitor mode. The monitor interrupt indicates that a processor instruction has been completed and the next instruction may be premonitored. All instructions are completed in this level of interrupt, and the ILC stored in the old PSW is set to 0.

2.7.8. Supervisor Call Level

A supervisor call interrupt occurs when a supervisor call (SVC) instruction is executed. When the supervisor call IIS occurs, the contents of the i field of the instruction are stored in the interrupt code. The hardware does not examine or modify this code. The supervisor call level interrupt cannot be masked. The ILC stored in the old PSW is 01_2 unless the SVC instruction is the subject instruction of an execute instruction, in which case the ILC is set to 10_2.

2.7.9. Input/Output Status Tabler (IOST) Level

An IOST interrupt request occurs when the IOST interrupt request flip-flop is set to 1 and the system mask bit (bit 1 of the PSW) is set to 1. If the mask bit is 0, the interrupt request remains pending until the mask bit is set to 1. When permitted, the interrupt request enters the IIS for the IOST level. Interrupts of this level may occur only after execution of the current processor instruction is completed. The ILC stored in the old PSW is 0, and the interrupt code in the IOST old PSW is cleared to all 0's.

It should be noted that in order for the IOST to store more status entries in the status table, the IOST tabling request also must interrupt the processor function in a manner similar to that described for the IOST timer and monitor IIS, that is, between instruction boundaries after completion of the current instruction. The status tabling request sequence, which does not include an IIS, is performed then and, upon successful completion, the processor restores the system environment to what it was prior to the tabling request. Both IOST requests (status tabling and IIS) share the same level of interrupt in the interrupt priority scheme (Table 2-3). These two requests compete for entering their respective sequence. When such a case is encountered, priority is granted to the request for accessing the table rather than the one for entering the IOST IIS, and the latter is held pending. This enables a more efficient use of the tabling concept by allowing the handling of more than one table entry per interrupt.

Should the status tabler encounter a hardware malfunction during the IOST tabling sequence, an IOST machine check interrupt request results in entering a machine check level IIS and stores the appropriate interrupt code. The ILC stored in the machine check bit of the old PSW becomes 0, and the instruction address field points to the next instruction to be executed as if no interrupts had occurred. If a processor machine check occurs during the IOST tabling sequence and the MC hardware mask is set, the IOST machine check interrupt request is held pending.

2.7.10. Interval Timer

An interval timer is provided an an integral part of the 90/30 processor. The timer count, located in a 24-bit interval timer register (ITR), is decremented once every millisecond if it is turned on by the service timer register (STR) instruction. The binary count contained in the ITR allows the interval between interrupts to range from 1 millisecond to 4.66 hours. An interval timer request occurs whenever the ITR count steps from 1 to 0.

2.7.10.1. Interval Timer Register

The interval timer register (ITR) contains a 24-bit unsigned count value and is under direct control of the STR instruction. When the ITR steps from 1 to 0, an interval timer interrupt request occurs. The ITR continues to be stepped from 0 to all 1's and again toward 0. After decrementing to 0 and setting an interrupt request take place, the content of the ITR becomes the twos complement of the count past 0 (overrun count).

2.7.10.2. Interval Timer Level Interrupt

A timer interrupt request (TIR) occurs when the interval timer register is decremented from 1 to 0. The TIR is under control of the system mask t bit (bit 0 of the current PSW). If this bit is set to 0, the TIR is held pending until the t mask bit is set to 1, the STR instruction specifying the OFF state is executed, or an STR instruction with the I bit set to 1 is issued. When allowed, the TIR causes the processor to store the current PSW at the interval timer old PSW location (location 60_{16}) and reads out the interval timer new PSW location (location 68_{16}).

Should the interval timer interrupt be masked off, the content of the ITR continues to be decremented each millisecond. When the timer interrupt request is set, the content of the ITR is the overrun count. If the content of the ITR steps from 1 to 0 while a previous timer interrupt request is held pending, the two interrupts are merged into a single interrupt, when permitted. No unique indication is given for this merger. Interrupts of this level can occur only after execution of the current processor instruction is completed and the ILC stored in the old PSW is set to 0.

The interrupt code in the interval timer old PSW is also set to all 0's.

2.7.10.3. Interval Timer Operation

System reset sets the interval timer to the off state. The off state is defined as the state where the 24-bit ITR cannot be decremented.

Once turned on by way of an STR instruction, the ITR decrementation continues unless turned off by the STR instruction, the inhibit timer switch, or system reset.

Activating the inhibit timer switch on the maintenance panel suspends the interval timer operation if the timer had been previously turned on by the STR instruction. When the inhibit timer switch is deactivated, the interval timer operation resumes. When the inhibit timer switch is active, any STR instruction issued is then executed. If a turn-off operation is specified, the timer remains off after the inhibit timer switch is deactivated.

The interval timer includes a hardware queue to store 1-millisecond intervals. When the interval timer is turned off by way of an STR instruction, the queue can accumulate the transition of a 1-millisecond interval occurring without an update of the ITR. Should two or more consecutive 1-millisecond intervals occur and an ITR decrement not be made due to the interval timer being in the off state, the queue accumulates only one 1-millisecond interval and all other are lost. When the interval timer is turned on by way of the STR instruction (after being turned off), the ITR contents may be decremented by 1 if the hardware queue was set. The queue is cleared following the decrement. The queue assures software a minimum of 1 millisecond to make ITR updates without the loss of decrement to the ITR. System reset or the inhibit timer switch maintains the hardware queue in the cleared state.

2.8. PROCESSOR HARDWARE-DETECTED ERRORS

Various parts of this manual refer to error and abnormal conditions detected by the hardware. The following subsections further define these conditions.

2.8.1. Control Storage Errors

Control storage parity checks and control storage addressing exceptions are detectable whenever control storage is used.

2.8.1.1. Control Storage Parity Check

A control storage parity check is indicated if one of the following conditions is detected:

- Control Storage Read Bus Check

  The microinstruction transmitted from control storage includes two parity bits. The parity bits are not generated by the control storage but are stored as part of the microinstruction loaded from the media used for the initial load control storage (ILCS) or loaded by a load control storage (LCS) instruction. On a read from control storage, the processor performs a parity check on the microinstruction. An error thus detected is called a control storage read bus check and results in a nonrecoverable error condition at the processor.

- Control Storage Write Bus Check

  Parity bits, one per byte, are attached to the half words sent to control storage to be assembled into a microinstruction. The control storage performs a parity check on each byte to be written. The two parity bits associated with each half word transferred are not stored in control storage. If a parity error is detected, control storage so indicates by signalling the write check line, and the half word is written into control storage. If the address specifies the read-only memory (ROM) section of control storage during a write, the write-check line indicates the error.

  A write check indication from control storage during the execution of an LCS instruction results in a processor machine check. A nonrecoverable error condition results at the processor if the write check indication occurs during ILCS.

2.8.1.2. Control Storage Addressing Exception

If, in the LCS instruction, the specified control storage location is not installed, an addressing exception is indicated when the control storage fails to signal on the address-accepted line.

During the execution of the ILCS, the initial program load (IPL), or the normal microinstruction execution if the specified address pertains to a control storage location that is not installed, the control storage does not signal on the address-accepted line. The error thus detected results in a nonrecoverable error condition at the processor. It should be noted that at the access of each microinstruction from control storage, the processor checks the contents for a specific code that is loaded into all unused locations. Should the addressed location contain that code, the error thus detected causes the processor to stop instruction execution and the HPR (halt and proceed) indicator lights.

2.8.2. Main Storage Errors

Error conditions detected and indicated to the processor and I/O channels when main storage is used are described in 2.8.2.1 through 2.8.2.5.

2.8.2.1. Address Check

An address check is indicated when either of the following conditions is detected:

- Address Bus Check

  An 18-bit half word is supplied to storage with 2-byte write-enable lines. A signal on a specified write-enable line indicates that the corresponding byte in the specified half-word location in main storage is to be written.

  Absence of a signal on both write-enable lines specifies that a read operation will occur from the specified half-word location in main storage. The 18 address bits and 2-byte write-enable lines to storage are accompanied by 3 parity bits. One parity bit is associated with the low order 6 address bits. The second parity bit is associated with the next 7 address bits. The third parity bit is associated with the high order 5 address bits and 2-byte write-enable lines.

  If a parity error is detected, main storage indicates an error on the address parity check line by the ADDRESS CHECK signal. When main storage detects an error, it suppresses the erroneous information from being written unless the INHIBIT PROCESSOR CHECK switch on the system maintenance panel is set to ON. This check condition causes an address check indication.

- Storage Select Exception

  If internal and external storage provide signals simultaneously on the address-accept line, a storage-select-exception error condition occurs. If specified, data to be written into main storage is suppressed.

2.8.2.2. Storage Addressing Exception

Each 65K byte bank of storage examines all addresses supplied to those sections of storage. If external or internal storage recognizes an address as pertaining to one of the 65K byte banks, storage indicates this by signalling the address accepted line. If storage does not indicate acceptance of an address after the processor or I/O channel presents the address, an address-exception error occurs.

2.8.2.3. Storage Parity Check

If any of the following check conditions are detected, a storage parity check results:

- Storage Read Check

  A parity bit is associated with each byte read from main storage. If main storage detects a parity bit error during a read operation, a STORAGE READ CHECK signal is placed on the storage data parity check line.

- Read Bus Check

  Two parity bits accompany a half word transmitted from main storage; one parity bit is associated with each of the two bytes that are read by the processor or I/O channel, which performs a parity check on the bytes. An error is indicated with a READ BUS CHECK signal.

- Write Bus Check

  A parity bit is attached to each byte of information to be written in main storage, where the parity bit is checked. Storage indicates a detected parity error by the WRITE BUS CHECK signal. The half word containing the parity error is not written in storage unless the INHIBIT PROCESSOR CHECK switch in the system maintenance panel is set to ON.

2.8.2.4. Storage Hold Check

An address bus check, storage read check, or write bus check detected by storage while it is in hold mode is indicated by a STORAGE HOLD CHECK signal. The storage hold mode is activated with the HALT ON ERROR switch on the system maintenance panel. The switch is generally enabled only for maintenance purposes.

A storage hold condition sent to the processor causes operation to halt and prevents its use by the system. The condition is cleared with system reset, requiring manual intervention, and a nonrecoverable error is generated in the processor. The STORAGE HOLD signal is not sent to the I/O channels when the HALT ON ERROR switch is activated.

2.8.2.5. Storage Protection Exception Feature

A storage protection capability is included to prevent unauthorized writing or reading into various storage sections. The feature causes a protection exception error to occur.

2.8.3. Processor Stall Check (Recovery Timer)

A minimum 16-millisecond timer is triggered at the beginning of the execution of each processor instruction and at the beginning of an IIS. If, due to a hardware fault, a new instruction is not staticized within the 16 milliseconds, a processor machine check interrupt is generated. The effect of the expiration of this timer is inhibited under the following conditions:

- The processor is stopped due, for example, to a cleared state, halt and proceed (HPR) instruction, stop, etc.
- The processor is operating in the 1-instruction mode or the cycle mode.
- The processor is executing a softscope instruction.
- Initial load is being performed.

2.8.4. Power Control Faults

Various switches and indicators are provided on the system status panel located in the processor cabinet, for monitoring and controlling power to the system, as well as indicating certain abnormal conditions. However, only two affect the operation of the processor. One indication generates a nonrecoverable error requiring manual intervention to reestablish the normal state for the processor. The second indication generates an equipment check interrupt request.

2.8.4.1. Equipment Check - Early Warning

For the purpose of discussing this fault, the system can be considered to consist of the following physical elements: the processor, external storage cabinet, integrated peripheral channel and associated devices, integrated disk adapter and disk drives, and expansion cabinet. An early warning temperature fault detected anywhere in the system generates an equipment check interrupt request; power to the system remains on and an audiable alarm sounds at the system console.

2.8.4.2. Power Faults

The detection of a power fault (ac, dc, blower, or overtemperature) in the external storage cabinet or expansion cabinet generates a nonrecoverable error condition at the processor, and power to the cabinet containing the fault is automatically shut down immediately. The detection of a power fault in the processor automatically shuts power down within the complete system immediately. The detection of certain power faults at the devices associated with the integrated peripheral channel also automatically shuts down power within the system immediately.

2.8.5. Program Exceptions

Program exceptions listed in the following subparagraphs apply to the processor when it is operating in the native mode defined by the current PSW bits 16 through 18. If the control hardware of the processor attempts to execute an instruction which is not included in the 151-instruction repertoire, an operation exception occurs.

2.8.5.1. Operation Exception

An operation exception occurs when:

- the control hardware of the processor attempts to execute an instruction that is not included in the set of instructions for the 90/30 system;
- the micrologic expansion feature is not installed and the processor attempts to execute one of the floating-point or other instructions furnished with the optional expansion feature; or
- the storage protect feature is not installed and the processor attempts to execute a set storage key or insert storage key instruction.

When an operation exception is detected, the specified instruction is suppressed.

2.8.5.2. Privileged Operation Exception

A privileged operation exception occurs when the processor attempts to execute one of the privileged instructions and the processor is in the problem mode (bit 14, the PS bit, of the current PSW is set to 1). If a privileged operation exception is detected, the instruction is suppressed.

2.8.5.3. Execute Exception

An execute exception occurs where an execute instruction has as its subject instruction another execute instruction and causes the instruction to be suppressed.

2.8.5.4. Protection Exception

If the storage protect feature is installed, a protection exception can occur during the fetch of an instruction or an operand. When detected during the fetch of an instruction, the execution of the instruction is suppressed.

If a protection exception occurs during the fetch of the subject instruction of an execute (EX) instruction, the subject instruction is suppressed and the instruction address contained in the program exception portion of the old PSW points to the instruction following the EX instruction.

If a protection exception occurs during the fetch of an operand, the subject instruction is suppressed or terminated.

2.8.5.5. Addressing Exception

This exception occurs if a program-specified address lies either outside the range of main storage in the given system or outside the maximum addressing capacity of main storage. The conditions described for protection exception (2.8.5.4) also apply. An addressing exception also occurs if the control storage address specified in the load control storage (LCS) instruction is nonexistent or if the second operand of a set storage key (SSK) or insert storage key (ISK) instruction specifies a nonexistent block of storage. In addition, an addressing exception occurs if the address of the destination of any branch instruction or the instruction address associated with the loading of a new PSW (LPSW,IIS) lies outside the range of storage for the given installation. When this is detected, upon fetch of the destination instruction or upon fetch of the instruction specified by the new PSW instruction address, the program exception of the old PSW points to the unmodified (nonexistent) address.

2.8.5.6. Specification Exception

A specification exception occurs if any of the following conditions occurs during the operation of the specified type of instruction.

- Fixed-Point Instructions

  These instructions are suppressed if a double-word operand is not on an even 8-byte boundary, a word is not on an even 4-byte boundary, a half word is not on a 2-byte boundary, or an odd register address is specified where an even/odd pair of registers is required.

- Decimal Instructions

  These instructions are suppressed if a multiplicand or a divisor exceeds 15 digits and sign, or is equal to or exceeds the multiplicand or dividend size, respectively.

- Floating-Point Instructions

  These instructions are suppressedif a short operand is not on a word boundary, a long operand is not on a double-word boundary, or a floating-point register address is other than 0, 2, 4, or 6.

- Logical Instructions

  These instructions are suppressed if a word operand is not on 4-byte boundary or an odd register address is specified where an even/odd pair of registers is required.

- Branching Instructions

  These instructions are suppressed if the subject instruction address in an execute instruction is odd.

- Status Switching Instructions

  These instructions are suppressed if the PSW address specified in the LPSW instruction is not on a double-word boundary or the storage address held in the register specified by r_2 in an SSK or ISK instruction does not contain four low-order 0 binary bits.

- Diagnostic Instructions

  These instructions are suppressed if an odd register address is specified when an even/odd pair of registers is required, or the address is not on a half-word boundary. In addition, suppression takes place when the processor attempts to execute an SSFS, SSRS, or DIAG ($I_2 \neq 00$) instruction when the corresponding supporting microcode has not been loaded into the transient area.

- Interval Timer Instructions

  These instructions are suppressed if an invalid modifier field is specified.

- Instruction Fetching

  This instruction is suppressed if the instruction address in the current PSW is odd.

2.8.5.7. Instruction Termination and Suppression

An instruction that has been terminated or suppressed can be reconstructed by the software after the program exception interrupt has occurred by subtracting the ILC (bit positions 32 and 33) from bit positions 61 and 62 of the instruction address in the program exception old PSW. This address then points to the first byte of the instruction that was terminated or suppressed. However, if the terminated or suppressed instruction was the subject instruction of an execute instruction, the aforementioned calculated address points to the first byte of the execute instruction. Since this check is made only when an address is sent to storage, a branch address (including the one loaded into the current PSW by an LPSW instruction) containing a protection exception or addressing exception is not examined until the fetch of the instruction pointed to by the branch address occurs. Therefore, the contents of the instruction address, software-modified by the ILC contained in the program exception portion of the old PSW, points to the instruction branched to and not to the branch instruction itself.

2.9. 9200/9300 COMPATIBILITY MODE

The 9200/9300 compatibility mode is specified in the mode field (bits 16–18) of the current PSW. Programs running in this mode may be monitored when the MON bit (bit 19) of current PSW is set to 1. Protection for 9200/9300 references to restricted low-order storage locations and detection of storage wraparound are implemented by using relocation and storage protection. The 9200/9300 compatibility mode provides a rapid and simplified method of converting 9200/9300 applications programs to the 90/30 system.

2.9.1. 9200/9300 INSTRUCTIONS

Table 2–4 lists the instructions that execute in the 9200/9300 compatibility mode. Those instructions specifying OPEX generate an operation exception interrupt if an execution is attempted for that instruction.

Table 2—4. 9200/9300 Compatibility Instructions

Opcode Hexadecimal	Instruction Type	Mnemonic	Instruction	Operation Exception Interrupt
40	RX	STH	Store half word	
45	RX	BAL	Branch and link	
47	RX	BC	Branch on condition	
48	RX	LH	Load half word	
49	RX	CH	Compare half word	
91	SI	TM	Test under mask	
92	SI	MVI	Move immediate	
94	SI	NI	AND	
95	SI	CLI	Compare logical	
96	SI	OI	OR	
A0	SI	SPSC	Store state	OPEX
A1	SI	SRC	Supervisor call	OPEX
A4	SI	XIOF	Execute I/O	OPEX
A5	SI	TIO	Test I/O	OPEX
A6	SI	AI	Add immediate	
A8	SI	LPSC	Load state	OPEX
A9	SI	HPR	Halt and proceed	OPEX
AA	RX	AH	Add half word	
AB	RX	SH	Subtract half word	
D1	SS	MVN	Move numerics	
D2	SS	MVC	Move	
D4	SS	NC	AND	
D5	SS	CLC	Compare logical	
D6	SS	OC	OR	
DC	SS	TR	Translate	
DE	SS	ED	Edit	
F1	SS	MVO	Move and offset	
F2	SS	PACK	Pack	
F3	SS	UNPK	Unpack	
F8	SS	ZAP	Zero and add	
F9	SS	CP	Compare decimal	
FA	SS	AP	Add decimal	
FB	SS	SP	Subtract decimal	
FC	SS	MP	Multiple decimal	
FD	SS	DP	Divide decimal	

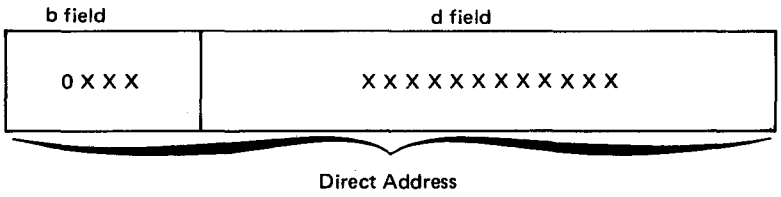
2.9.2. 9200/9300 Compatibility Registers

The 9200/9300 registers are 16 bits in length while the 90/30 registers consist of 32 bits. The processor does not write into the upper 16 bits of any register when operating in the 9200/9300 compatibility mode. Software should clear the eight registers (registers 8 through F of the 90/30 processor) used for the 9200/9300 programs before entering the 9200/9300 compatibility mode.

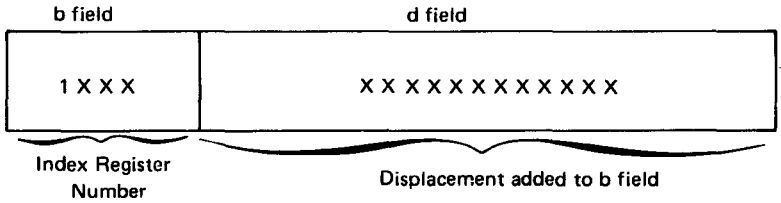
2.9.3. 9200/9300 Compatibility Instruction Execution

Instructions staticize when operating in the 9200/9300 compatibility mode is subject to the following restrictions.

Direct addressing capability of the 9200/9300 is implemented. The most significant bit of the b field is tested. If zero, the remaining three bits of the b field are attached to the most significant bit of the d field to form a 15-bit address.



If the most significant bit of the b field is set to 1, the next three bits specify one of eight registers (8 through F), the contents of which are added to the 12-bit d field.



In either case, the normal 90/30 relocation addition is performed after the emulated effective address has been formed. Bits 12–15, the x_2 field of the RX format (BAL, BC, STH, LH, CH, AH, and SH), are ignored by the 90/30 processor when 9200/9300 compatibility mode is set. No double indexing occurs regardless of the value of x_2.

2.9.3.1. Immediate Instructions (SI Format)

The TM, NI, OI, CLI, MVI, and AI instructions execute the same in the 9200/9300 compatibility mode as when not in the compatibility mode.

2.9.3.2. Indexed Instructions (RX Format)

The BAL, BC, STH, LH, CH, AH, and SH instructions execute in the 9200/9300 compatibility mode with the following restrictions.

- The x_2 field is ignored.
- Condition code calculations are based on half-word operands and not on half words that have been sign-extended to full-word operands.
- As a result of the restrictions on general register usage, the BAL instruction stores only a 16-bit (relative) address in the link register.

2.9.3.3. SS Instructions

The MVN, MVC, NC, CLC, OC, and TR instructions execute the same in the 9200/9300 compatibility mode as when not in the compatibility mode. The MVO, PACK, UNPK, ZAP, CP, AP, and SP instructions execute based on the following restrictions.

- The ZAP, CP, AP, and SP instructions terminate upon exhausting the OP1 field. If OP1 is longer than OP2, OP2 will be zero filled. If OP2 is longer than OP1, all digits of OP2 beyond the length of OP1 will be ignored. Condition codes are generated based on the conditions existing at the time OP1 is exhausted. Under this rule, ZAP cannot generate an overflow.
- A check for correct overlap is made, except for the ZAP instruction
- No data exception is generated for invalid signs.
- The following sign conventions apply.
  - Detected sign:

    9, B, and D are minus: all other digits are plus.
  - Generated signs:

    ASCII mode - A is plus; B is minus.
    EBCDIC mode - C is plus; D is minus.
- The following generated zones for UNPK apply:

  ASCII mode: 5
  EBCDIC mode: F
- Invalid data is checked except for ZAP.

The ED instruction executes based on the following restrictions:

- A data exception is not generated if the left-most digit of a byte in OP2 contains a nonnumeric hexadecimal digit.

- The following EDIT control characters are used:

Digit select	20_{16}
Significance start	21_{16}
Field separator	22_{16}

- The following sign conventions apply:

  B,D = minus
  A,C,E,F = plus

- The following generated zones apply:

  ASCII mode = 5
  EBCDIC mode = F

The MP and DP instructions execute based on the following restrictions:

- The following conventions apply:

  - Detected signs:

    OP2 9,B,D = minus. All other hexadecimal digits = plus.
    OP1 B,D = minus. A,C,E,F = plus.

  - Generated signs:

ASCII mode	A = plus; B = minus.
EBCDIC mode	C = plus; D = minus.

- All 90/30 specification errors are in effect.

- Leading zeros are checked for OP1 of MP.

- Invalid overlap is checked.

- Invalid digit check is made.

2.10. IBM 360/20 COMPATIBILITY MODE

The 360/20 compatibility mode is specified by the mode field (bits 16-18) of the current PSW. Programs running in this mode may be monitored when the MON (bit 19) of current PSW is set to 1.

Protection for 360/20 references to restricted low-order storage locations and detection of storage wraparound are implemented by using relocation and storage protection of the 90/30 system.

The 360/20 compatibility mode provides a rapid and simplified method of converting 360/20 application programs to the 90/30 system. The 360/20 compatibility mode operates identically to native mode except for differences indicated in subsequent paragraphs.

2.10.1. 360/20 INSTRUCTIONS

Table 2–5 lists the instructions that execute in the 360/20 compatibility mode. Those instructions specifying OPEX generate an operation exception interrupt if an execution is attempted for that instruction.

Table 2—5. 360/20 Compatibility Instructions

Opcode Hexadecimal	Instruction Type	Mnemonic	Instruction	Operation Exception Interrupt
07	RR	BCR	Branch on condition	
0D	RR	BASR	Branch and store	
1A	RR	AR	Add	
1B	RR	SR	Subtract	
40	RX	STH	Store half word	
47	RX	BC	Branch on condition	
48	RX	LH	Load half word	
49	RX	CH	Compare half word	
4A	RX	AH	Add half word	
4B	RX	SH	Subtract half word	
4D	RX	BAS	Branch and store	
81	SI	SPSW	Set PSW	OPEX
83	SI	DIAG	Diagnose	OPEX
91	SI	TM	Test under mask	
92	SI	MVI	Move immediate	
94	SI	NI	AND	
95	SI	CLI	Compare logical	
96	SI	OI	OR	
99	SI	HPR	Halt and proceed	OPEX
9A	SI	TIOB	Test I/O and branch	OPEX
9B	SI	CIO	Control I/O	OPEX
D0	SS	XIO	Transfer I/O	OPEX
D1	SS	MVN	Move numerics	
D2	SS	MVC	Move	
D3	SS	MVZ	Move zones	
D5	SS	CLC	Compare logical	
DC	SS	TR	Translate	
DE	SS	ED	Edit	
F1	SS	MVO	Move and offset	
F2	SS	PACK	Pack	
F3	SS	UNPK	Unpack	
F8	SS	ZAP	Zero and add	
F9	SS	CP	Compare decimal	
FA	SS	AP	Add decimal	
FB	SS	SP	Subtract decimal	
FC	SS	MP	Multiply decimal	
FD	SS	OP	Divide decimal	

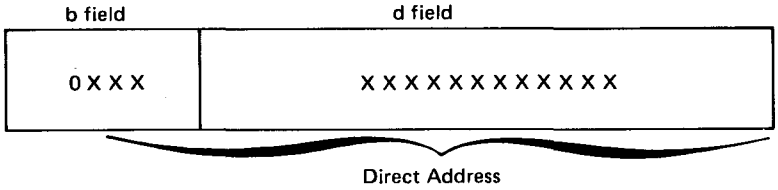
2.10.2. 360/20 Compatibility Registers

The 360/20 compatibility registers are 16 bits in length while the 90/30 registers consist of 32 bits. The processor does not write into the upper 16 bits of any register when operating in the 360/20 compatibility mode. Software should clear the eight registers (registers 8 through F of the 90/30) used for the 360/20 programs before entering the 360/20 compatibility mode.

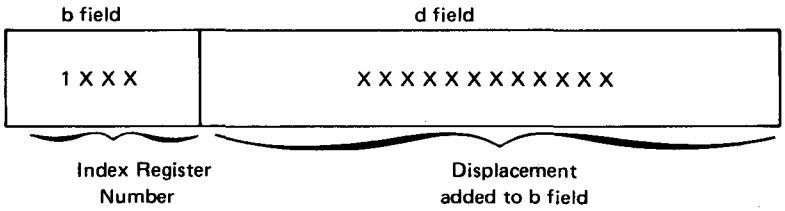
2.10.3. 360/20 Compatibility Instruction Execution

Instructions staticize when operating in the 360/20 compatibility mode is subject to the following restrictions:

- Direct addressing capability of the 360/20 processor is implemented. The most significant bit of the b field is tested. If zero, the low-order three bits of the b field are attached to the most significant bit of the d field to form a 15-bit address.



If the most significant bit of the b field is 1, the next three bits specify one of eight registers (8 through F), the contents of which are added to the 12-bit d field.



In either case, the normal 90/30 relocation addition is performed after the emulated effective address has been formed.

- A specification exception interrupt is generated for the following conditions (in addition to the native mode 90/30 conditions).
  - The r_1 or r_2 fields of an RR or RX type binary instruction (STH, LH, CH, AH, SH, AR, SR) have values less than 8.
  - The r_1 field of BAS or BASR contains a value less than 8.
  - The r_2 field of BCR or BASR contains a value in the range of 01 through 07.
  - The x field of an RX instruction is not zero.

2.10.3.1. 360/20 Compatibility Mode Special Considerations

The following additional differences exist from the 90/30 native mode for the LH, AH, SH, CH, STH, AR, and SR instructions.

- Condition code calculations and overflow determinations are based on half-word operands and not on half words that have been sign-extended into full-word operands.
- Overflow generates a fixed-point overflow exception independent of the PSW mask bit. The overflow condition sets condition code 00, 01, or 11, depending on the resultant half-word operand upon detection of overflow.

The following additional differences exist from the 90/30 native mode for the PACK, UNPK, MVO, ZAP, AP, SP, CP, MP, and DP instructions.

- The following generated zones for UNPK apply:

  ASCII = 5
  EBCDIC = F

- The following sign conventions apply to ZAP, AP, SP, MP, and DP:

  - Detected signs:

    ASCII/EBCDIC – A, C, E, F = plus
    B and D = minus

  - Generated signs:

    ASCII A = plus; B = minus
    EBCDIC C = plus; D = minus

- ZAP, AP, SP, and CP instructions:

  If OP1 is exhausted before OP2, a specification exception is generated and the construction is suppressed. Therefore, ZAP cannot generate an 11_2 condition code.

The only instruction of the group consisting of MVI, MVC, MVN, MVZ, CLI, CLC, ED, NI, OI, TM, and TR that differs in 90/30 operation compared with the 360/20 mode is the ED instruction.

- The following edit control characters are used:

  - Digit select 20_{16}
  - Significant start 21_{16}
  - Field separator 22_{16}

- The following sign conventions apply:

  - B and D = minus
  - A, C, E, and F = plus

- The following generated zones apply:

  - ASCII = 5
  - EBCDIC = F

2.10.3.2. BCR, BC, BASR, and BAS Instructions

The following additional differences exist from the 90/30 native mode only for the BASR and BAS instructions:

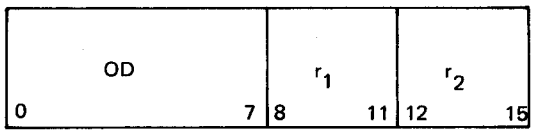
- BAS and BASR opcodes do not exist in the 90/30 native mode set. Those opcodes (BAS=ID and BASR= OD) are effective only when operating in the 360/20 compatibility mode.

- The BAS and BASR instructions are implemented exactly as the BAL and BALR instructions, with the exception that the link information consists of only 16 bits, by way of the relative address of the instruction following the branch instruction.

- The BASR and BAS instruction formats are as follows:

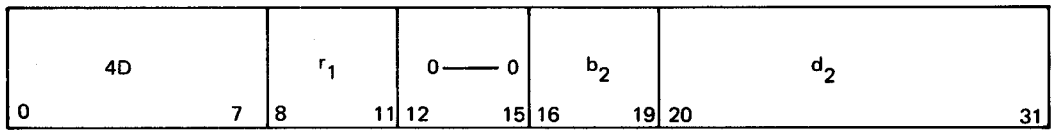
BASR

(RR)



BAS

(RX)



The offset value contained in the current relocation register is subtracted from the updated instruction address to form the link address. The 16 least significant bits of the link address are stored in bits 16 through 31 of the first operand location.

The instruction address field of the current PSW is replaced by the second operand address (branch address). The branch address is determined before the link information is stored. This permits correct operation if the r_1 and r_2 registers or the r_1 and b_2 registers are the same.

In the RR format, if the r_2 field is 0, the link information is stored in the first operand location, but no branch is accomplished; instruction sequencing continues with the updated instruction address.

For the branch and store instruction, possible condition codes remain unchanged, and no possible program exceptions will exist.

2.11. MONITOR MODE

The monitor mode of operation is provided to facilitate dynamic software monitoring of running programs. During this mode, a program analysis level interrupt occurs before the execution of every instruction in the program being monitored. The monitor mode is specified by bit 19 of the current PSW. It is not necessary to mask other interrupts that could be expected to occur during the running of a monitored program. The execution of a load PSW instruction in a program being monitored could establish a new mode in the current PSW and monitoring would cease.

2.11.1. Operation

Upon either the setting of the monitor mode in the current PSW by means of a load PSW instruction or the execution of an interrupt initialization sequence (IIS), a program analysis interrupt request is immediately honored, provided no other interrupt requests are present and unmasked. Since this interrupt request has the lowest priority of all interrupt requests, its associated IIS is the last executed in any string of contiguous IISs and, therefore, the program analysis routine is entered only if the monitor mode is set in the current PSW resulting from the the last executed IIS prior to the monitor IIS. The monitor routine should not specify the monitor mode. The load PSW instruction with a secondary operation code of 01_{16} must be used to exit from this routine. This instruction inhibits all pending interrupt requests until the next instruction to be executed has been fetched. Thereafter, interrupt requests are honored as usual.

If the monitor mode is set, a program analysis interrupt request is again made following the execution of the instruction allowing the premonitoring of the next instruction. If, during the execution of an instruction running under monitor mode, another level of interrupt request is generated, its associated IIS occurs if the level is unmasked. If the monitor mode is not set in the other level, its interrupt routine is then entered. Upon exiting from this routine, the monitor IIS occurs, provided the monitor mode has been reestablished, and the monitor program is entered.

2.12. INITIAL LOAD

Hardware is provided in the processor to enable initial loading of control storage and initial program load into main storage. Both functions are initiated and controlled from the system operator/maintenance panel.

Control storage is implemented in a separate module from main storage by means of a unique interface to the processor. Basic control storage contains 128 microwords of read-only storage and 1024 microwords of writable storage. The read-only portion of control storage provides a nondestructive microprogram to perform diagnostic functions and control the initial load control storage (ILCS) and initial program load (IPL). The 1024 microwords of writable control storage are sufficient to implement the basic instructions. An additional 1024 microwords of control storage are provided when the micrologic expansion feature is installed.

Writable control storage has the following advantages:

- Microcode for the diagnostic instructions can overlay transient areas of control storage when the use of these instructions (writable) is required.
- System diagnostic routines can overlay the microcode for servicing and fault isolation.
- Changes implemented due to engineering requirements or extended functional capabilities can be easily implemented with the loading of an updated microcode into control storage.

2.12.1. Initial Program Load Operation

The RUN switch, in conjunction with the INITIAL LOAD control switch, initiates initial load microprogram in read-only control storage. The microprogram first performs a write/read check of the first 320 half words of main storage by writing a fixed data pattern to each half word, reading this data, and then performing a comparison check. The microprogram sets the channel address word (CAW), the channel command word (CCW), and buffer control words (BCW) in main storage at locations 0090_{16} through $027F_{16}$ and initiates the specified operation on the selected channel.

For initial program load (IPL), the I/O operation on any selected channel generally causes one record to be read from the input device into main storage starting at location 0; for example, one card image, one disk IPL record, or one tape block. An exception is the integrated disk adapter that transfers 96 records (256 bytes per record) to main storage. For other channels, the size of the record determines the amount of data transferred. An upper bound on the amount of data transferred is set by the initial contents of the applicable BCW or CCW. Maximum byte counts are set at 96 for the IPC, 7264 for the multiplexer channel, and 32,768 for the selector channels. When loading from input devices on the IPC multiplexer channel takes place, care must be exercised on data that can be loaded into the BCW that is controlling the current I/O operation. If the input record size is large enough to extend into the main storage area containing the selected BCW, the data may cause alteration of the initial parameters set in the BCW.

Once operation is initiated, the microprogram checks for the proper condition code to be returned from the selected channel, and data transfers proceed under control of the channel in a normal manner. Upon completion of the operation, the channel stores the interrupt word (IOSTIW) in the BCSW in low-order main storage.

Upon recognition of the storage of an IOSTIW by the channel, the microprogram clears the current PSW and the current relocation register to 0's, transfers control to the writable portion of control storage, and stores the machine ID (MM), load ID (LL), revision level (RR), and 00 (unassigned) in main storage locations $00A4_{16}$ through $00A7_{16}$. The microprogram sets bit 1 of the current PSW (IOST level interrupt mask) to 1. This permits an IIS for the IOST level. The completion of IIS transfers control to the program specified in the IOST of the new PSW. Software analysis of the IOSTIW and BCW (if applicable) is necessary to determine if the I/O operation was successfully completed.

2.12.2. Initial Load Control Storage Operation

The initial load control storage (ILCS) is initiated by the RUN switch in a manner similar to the IPL function. The microprogram, which is read-only control storage, controls ILCS. Low-order main storage is checked, and the initiated I/O operation proceeds under normal control of the selected channel. The selector channel CCW is configured only for ILCS operation with disk subsystems.

When the loading of control storage from the integrated disk adapter (IDA) or selector channel takes place, data is transferred directly into control storage. Data transfer proceeds until 1024 or 2048 control storage locations have been loaded.

When control storage is loaded from the IPC or multiplexer channel, 80 bytes are read into main storage and transferred to control storage under microprogram control after completion of the I/O operation. The block/record size on the input must be 80 bytes. For all channels, the condition code and status bytes are checked by the microprogram for error-free operation. For the IPC or multiplexer channel, I/O operations continue to be initiated until the control storage load is complete.

Once control storage is loaded, the microprogram checks the hash total (card format only on IPC/multiplexer channel) and initializes a test read of control storage. The test read consists of sequential accesses of all control storage locations and checking for correct parity. After completion of the test read, the processor is stopped.

2.12.3. Initial Load Errors

Errors detected during initial loading have two basic sources. First, hardware-detected errors resulting from checking circuits are built into the processor and related components and result in a nonrecoverable error condition. This type of error can be detected in the processor, main storage, or control storage. The PROCESSOR CHECK and CONTROL STORAGE CHECK indicators on the system operator/maintenance panel give the proper indication, such as a storage parity check, control storage read bus check, etc. The second source of errors that can occur relates to these unique to the initial load microprogram. Examples of such errors are condition code or status byte error conditions reported by the I/O channel or device, hash total errors (ILCS on IPC/multiplexer channel), and miscomparisons on low-order main storage write/read check; any of which cause the PROCESSOR CHECK indicator to light.

Error-free completion of initial load of control storage is indicated by the HPR indicator being lit with no check indicator lit.

3. Input/Output Channels

3.1. CHANNEL NUMBERING

The input/output (I/O) channels are numbered 0 through 6, with channel number 7 assigned to the I/O status tabler (IOST). The corresponding 3-bit binary code of 000 through 111 is used to address the channel in I/O instructions and to identify the channel presenting status. The channel number order does not reflect the main storage priority or the status tabling priority of the channels. The IOST interrupt priority encompasses all the channels.

3.2. CHANNEL ASSIGNMENTS

The I/O channel assignment numbers, their identification, and their physical locations are as follows:

Channel Number	Identification	Cabinet Location
0	Integrated peripheral channel	Processor
1	Multiplexer channel or integrated multiplexer channel*	Expansion
2	Not assigned	–
3	Integrated disk adapter	Processor
4	Selector channel no. 1	Expansion
5	Not assigned	–
6	Selector channel no. 2	Expansion
7	I/O status tabler	Processor

3.3. MAIN STORAGE PRIORITY

This paragraph lists the channel priority assignments in decreasing priority. The IOST, which is an integral part of the processor, shares priority with the processor:

1.	Selector channel no. 1	Channel 4
2.	Integrated disk adapter	Channel 3
3.	Selector channel no. 2**	Channel 6
4.	Unassigned	Channel 5
5.	Multiplexer channel or integrated multiplexer channel	Channel 1
6.	Integrated peripheral channel	Channel 0
7.	Processor or I/O status tabler	Channel 7

**The multiplexer channel and integrated multiplexer channel are mutually exclusive. The integrated multiplexer channel is not included if an expansion cabinet is used.*

***When the IDA and two selector channels are included, selector channel 2 operates at a reduced rate.*

3.4. CHANNEL/SYSTEM FUNCTIONAL INTERFACES

The I/O channels in the processor complex communicate with other system components by way of the following relatively independent types of interfaces:

- I/O
- Instruction
- Main storage
- I/O status tabler (IOST)

3.4.1. I/O Interface

The I/O interface provides the means by which all communications between the channel and various subsystems are accomplished. The integrated peripheral channel (IPC) contains unique interfaces to meet the specific requirements of the attached control units. The equivalent interface within the integrated disk adapter (IDA) is unique to the IA and is not available to any other subsystems. The multiplexer and selector channels provide an 8-bit-plus-parity compatible interface and permits as many as eight subsystems to be connected to each channel by way of this interface.

3.4.2. Instruction Interface

The instruction interface allows the processor to issue I/O instructions to a selected channel and subsystem, and the channel returns a condition code (cc) upon accepting the SIO instruction. The processor starts an I/O operation to execute the instruction by activating the SIO line to the appropriate channel and by transferring the contents of the instruction device address field and a request to the selected channel. Depending on the state of the channel, subchannel, and subsystem, the operation is either initiated or not. The channel then returns the appropriate cc and the signal acknowledging receipt of the instruction request.

3.4.3. Main Storage Interface

This is the interface by which the channel requests are allowed to use main storage and by which transfers between main storage and the channel are accomplished. Each channel can independently request the use of storage by sending a storage request signal to the processor. Based on the storage priority scheme, the processor determines which channel is allowed use of main storage and issues a signal to the selected channel. This interface also is used for the transfer of main storage indications, such as addressing exceptions, protection exceptions, address checks, storage read checks, and write bus checks from main storage and the processor to the channel.

3.4.4. I/O Status Tabler Interface

The presentation of status from the channel is made by way of the IOST interface. The status information is made available to the program by means of the IOST interrupt words. The IOST is the entity that manages the presentation of the status into the table. The channel to IOST interface consists of a low-order main storage buffer for the intermediate storage of a word of status. The physical channel/processor interface includes the controls necessary for the channel to carry out its functions with the IOST, including error indications.

3.5. I/O STATUS TABLER

The I/O status tabler (IOST) section of the processor directs and monitors transfer of status between main storage and I/O channels. As an integral part of the processor, the IOST shares priority for main storage access with the processor. The IOST has some of the characteristics of an I/O channel and is controlled by way of an SIO instruction. The IOST uses a storage protection key of 0 for all accesses to the low-order main storage locations.

3.5.1. Status Handling

The IOST provides the capability of accepting status from all channels, subchannels, and subsystems and storing or tabling the status in a contiguous area of main storage (status table). The tabling of status is handled in a manner similar to the handling of data on the channels. That is, there are software-generated control words in low-order storage (IOSTCW), a dedicated working area in low-order storage (BCSW), and a limited ability to link noncontiguous table areas in storage or cause table wraparound.

The following words are used by the IOST throughout its operation:

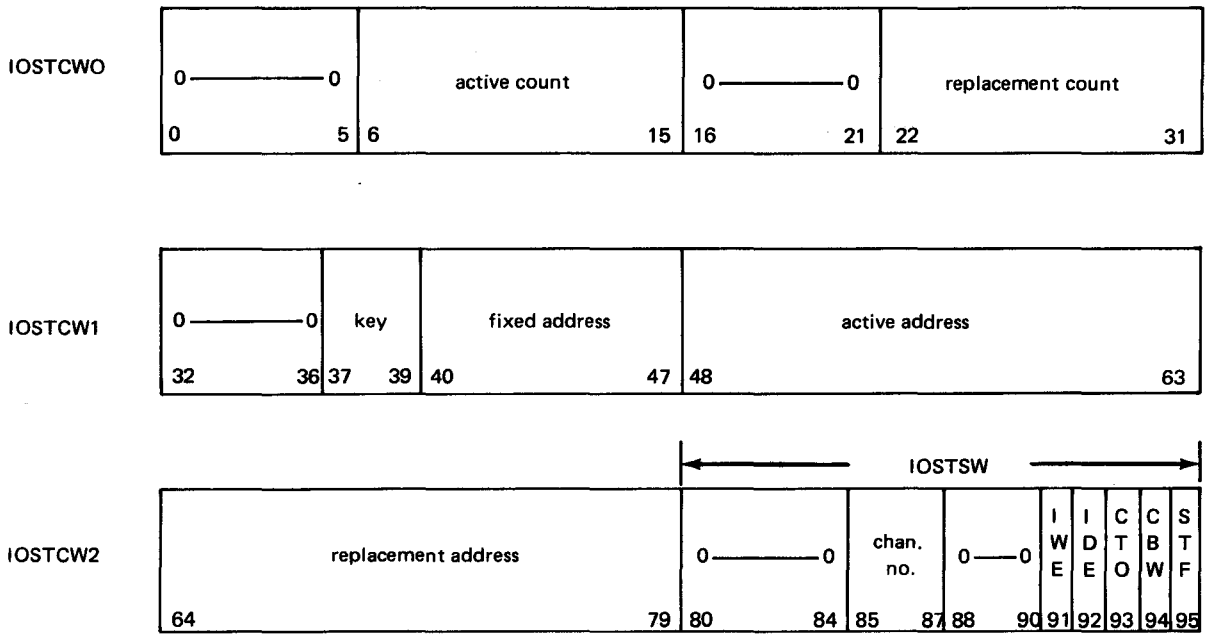
IOST control words (IOSTCW)

IOST interrupt words (IOSTIW)

3.5.1.1. IOST Control Words

The IOSTCWs consist of a 3-word area in low-order main storage locations 010_{16} through $01B_{16}$. The IOSTCW controls the operation of the IOST by pointing to the location in the status table where the next IOSTIW is to be stored, and by allowing the software to link two noncontiguous areas in storage for the status table or cause table wraparound. The software must set up the initial IOSTCW at system initialization time. Thereafter, the IOST provides for the replacement of the IOSTCW fields.

The format and description of the IOSTCW are:



Bits 0–5, 16–21, 32–36, 80–84, and 88–90
These bits are unassigned and must be set to 0 by the software.

Active count (bits 6–15)
A 10-bit count field that specifies the number of words remaining in the status table. The field is decremented by 1 each time a status word is stored in the status table by the IOST. Maximum count is 1024 words specified by all 0's.

Replacement count (bits 22–31)
A 10-bit count field that replaces the active count field when the active count field is decremented to 0.

Key (bits 37–39)
Contains the 3-bit storage protection key that is used by the IOST for all status table accesses. Unless the software replaces it at some later time, the key supplied in the initial IOSTCW is used throughout the operation of the IOST. The key is used for storage protection only, and addresses in the IOST are always absolute.

Address (bits 40–63)
Contains the absolute address of the full-word location in the status table area in storage where the next IOSTIW is to be stored. The contents of the address field must be set up to point to the first (most significant) byte of the status table. The address must be on a full-word boundary, or unpredictable status table entries result.

Fixed (bit 40–47)
Contains the fixed 8-bit field of the status table address. No carry-in from the active field occurs if the active count is greater than the maximum active address. The active address wraps around to an all 0 address.

Active (bits 48–63)
Contains the IOSTIW address and is incremented by 4 after each IOSTIW is stored. This field is replaced by the replacement address when the active count is decremented to 0. This field also points to the most significant byte of the next IOSTIW location.

Replacement address (bits 64–79)
Contains the address written into the active portion of the address field (bits 48–63) when the active count steps to 0.

IOSTSW (bits 80–95)
Contains the IOST status word with any error conditions encountered by the IOST in attempting to store status in the status table. The error conditions are not mutually exclusive.

Chan. no. (bits 85–87)
Channel number: Contains the number of the selected channel being serviced by the IOST when an error condition occurred.

IWE (bit 91)
Interrupt word error: When set, indicates that a protection exception or address exeption error was detected by the IOST when it attempted to read or write an IOSTIW from or into the status table or to read a buffer control status word. The IOSTIW to be stored is saved in the buffer control status word in low-order main storage at location $0E0_{16}$ through $0EF_{16}$.

IDE (bit 92)

Interrupt data error: When set, indicates that an address check or data parity check error was detected by the IOST when it attempted to read or write an IOSTIW from or into the status table, or to read a buffer control status word. The IOSTIW to be stored is saved in the buffer control status word in low-order main storage in locations $0E0_{16}$ through $0EF_{16}$.

CTO (bit 93)

Channel time out: Is set when the selected channel fails to respond to the STATUS SERVICE REQUEST ACKNOWLEDGE (SSRA) signal within the specified time. The contents of the BCSW in low-order main storage are unpredictable.

CBW (bit 94)

Channel buffer word error: Is set when an error comprising an address exception, protection exception, address check, or a write bus check, was detected by the selected channel when it attempted to write its IOSTIW into the buffer-storage-status-word storage location. The contents of the buffer control status word are unpredictable.

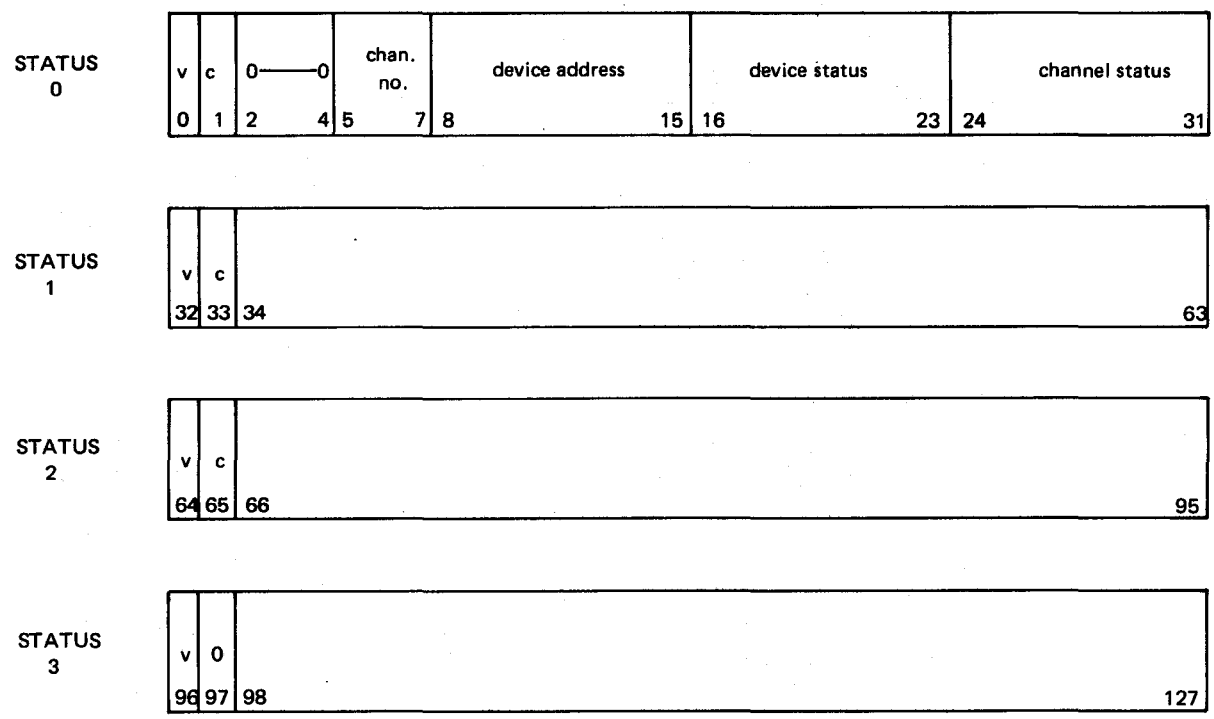
STF (bit 95)

Status table full: When set to 1, indicates the IOST detected that a v bit of the IOSTIW was set to 0 at the status table location where the IOSTIW was to be stored, and thus indicates a full status table. The IOSTIW to be stored is saved in the buffer control status word location.

3.5.1.2. IOST Interrupt Words

Each channel presenting status for entry into the status table can store a maximum of four full words when selected by the IOST. The I/O channel can write from one to four full-word IOSTIWs into the buffered channel status word (BCSW) in low-order main storage locations $0E0_{16}$ through $0EF_{16}$ and clear the status in the channel or subchannel. Control of this status tabling is maintained by the IOST.

The format and description of the IOSTIW are:



v (bits 0, 32, 64, 96)

Validation: Set to 0 by the channel before storing the IOSTIW in the BCSW to indicate new status information for the status table. They must be set to 1 by software after processing status. Should the IOST encounter a v bit set to 0 before storing an IOSTIW, the IOST enters a suspend state and stores an IOSTSW.

c (bits 1, 33, 65)

Continuation: Indicates the length of the IOSTIW. The channel sets the c bit to 1 to indicate that more than one full word of status is to be presented. The last full word of status presented must have the c bit set to 0.

Bits 2–4, 97

Unassigned and are set to 0 by the hardware.

Chan. no. (bits 5–7)

Channel number: Specifies the channel number of the channel presenting status.

Device address (bits 8–15)

Contains the address of the device or subsystem that was active on the channel at the time status was generated.

Device status (bits 16–23)

Contains one byte of status generated by the device or subsystem. The bits of the device status are:

Bit	Status
16	Attention
17	Status modifier
18	Control unit end
19	Busy
20	Channel-end
21	Device-end
22	Unit check
23	Unit exception

Channel status (bits 24–31)

Contains a byte of status information generated by and pertinent to the channel or subchannel. It should be noted that the status bits are not mutually exclusive and the presence of one type of error condition can cause the generation of more than one error condition:

Bit	Identification
24	Unassigned; are set to 0 by channel
25	Incorrect length
26	Program check
27	Invalid address
28	Channel data check
29	Interface control check
30	Channel control check
31	Buffer terminate

Status 1 (bits 34–63)

Contains 30 bits of additional status that can be presented by a channel in the same format as status 0.

Status 2 (bits 66–95)
Contains 30 bits of additional status that can be presented by a channel.

Status 3 (bits 98–127)
Contains 30 bits of additional status that can be presented by a channel.

3.5.2. IOST Sequences

The IOST and I/O channel operations can result in one of the following IOST sequences:

1. The IOST setting a condition code at the end of an operation specified by an SIO instruction, with the IOST channel number specified in the SIO instruction
2. An IOST machine check interrupt
3. The tabling of I/O channel IOSTIW
4. The IOST entering the suspend state upon attempting to store an IOSTSW (bits 80–95 of IOSTCW2)

3.5.2.1. Start I/O Instruction

A start I/O (SIO) instruction to channel 7 causes the IOST to go out of the suspend state. (Bits 24 through 31 of the SIO are unassigned and must be set to 0 by the software.)

The IOST responds to the SIO instruction by setting the appropriate condition code as follows:

00_2
The IOST was in the suspend state, was cleared, and resumed normal operation.

11_2
The IOST was not in the suspend state, rejected the command, and continued normal operations.

3.5.2.2. IOST Machine Check Interrupt

An IOST machine check level interrupt request is generated by the IOST on certain hardware malfunctions. Because the processor attempts to enter a machine check level IIS, the 8-bit interrupt code is written into the interrupt code field in the machine check portion of the old PSW. The processor then enters an IIS. The IOST machine check interrupts are used to alert the software that the IOST is in a nonrecoverable state that enter cannot be otherwise indicated or that is of such a nature as to require immediate action. The specific IOST machine check codes and their descriptions are as follows:

0000 0101
An addressing exception or protection exception occurred on any IOSTCW access by the IOST.

0000 1111
The processor stall timer expires during an IOST tabling sequence.

0000 1000
An address check or storage parity check occurred on any IOSTCW access by the IOST.

The IOST enters the suspend state whenever an IOST machine check interrupt is generated during an IOST sequence. In the suspend state, the IOST maintains the status service request acknowledge (SSRA) line active to the selected channel, inhibits further status requests, and inhibits scanning for status requests from the other channels. An IOST interrupt request also is set when the IOST is in the suspend state. The IOST machine check interrupt is cleared at the conclusion of the machine check IIS. Normal status tabling sequences are resumed by the IOST if the suspend state is cleared by way of an SIO instruction. Note that an IOST interrupt request can be held pending and is not cleared until an IOST IIS is honored.

It should be noted that the software should reinitialize the IOSTCW, IOSTIW, and BCSW after an IOST machine check interrupt since the contents of these words are unpredictable.

3.5.2.3. IOST Tabling Sequences

Each channel can independently request to present and store status in the BCSW low-order main storage in location $0E0_{16}$ through $0EF_{16}$. Device status or channel status that develops during the initiation of an I/O operation or during the course of an I/O operation causes the channel to generate a status request or status service request (SSR) and initiate a table sequence. Based on a unique priority scheme, the IOST determines which channel is permitted to store status in the BCSW and grants a priority scheme, the IOST determines which channel is permitted to store status in the BCSW and grants a priroity with an SSRA. The selected channel then attempts to store its status information in the BCSW low-order storage location.

The IOST accepts status requests from all I/O channels and assigns priority in the following sequence:

Highest priority	Integrated peripheral channel	Channel 0
↓	Multiplexer channel or integrated multiplexer channel	Channel 1
	Integrated disk adapter	Channel 3
	Selector channel 1	Channel 4
Lowest priority	Selector channel 2	Channel 6

When a channel completes the transfer of status to the BCSW low-order main storage location, it notifies the IOST by way of the IOST servicing (IOSTS) request. At the same time, the channel drops its SSR and resumes data transfers.

The IOSTS signal enables the IOST to interrupt the processor at an instruction boundary and move the IOSTIW from the BCSW low-order main storage location to the status table location in main storage specified by the IOSTCW.

An error-free completion of the above sequence results in the IOST deactivating the selected channel SSRA line. The IOST resumes scanning for status requests from other channels. The channel that had completed presenting status can again raise its status service request line after deactivating its IOSTS. Based on its priority, the channel can again be selected to present status by the IOST. Should a selected channel fail to present status and signal, by way of the IOSTS line, within 3 milliseconds, the IOST timer times out and an interrupt selective reset (ISR) is then sent to the channel. Should an error occur during the channel transfer of status into the BCSW, the channel notifies the IOST by way of the BCSW servicing error (BCSWSE).

Following the tabling of the IOSTIW, the IOST sets its interrupt request. The IOST interrupt request is under control the system mask IOST bit (bit 1 of the current PSW). If this bit is set to 0, the interrupt remains pending. However, IOST tabling of status from various channels continues with status tabling requests interrupting the processor at instruction boundaries. The IOST interrupt, when allowed, causes the processor to enter an IIS for the IOST level. Note that the multiple IOSTIW may have been stored by way of the status tabling requests before the IOST interrupt was allowed, and subsequently cleared during the IIS.

3.5.2.4. IOST Timer

The IOST is provided with a 3-millisecond IOST timer. The function of the IOST timer is to detect a stall condition in the IOST-to-channel interface due to a hardware malfunction in either the IOST or channel. If, after receiving the status service request acknowledge (SSRA) from the IOST, the selected channel fails to present status within 3 milliseconds, the IOST disconnects from the channel by way of an interrupt selective reset (ISR). The ISR causes the channel to drop its SSR line. The channel makes no attempt to present the same status again, the IOST writes an IOSTSW and goes into the suspend state.

The operation of the IOST timer is inhibited when the processor is operating in cycle mode.

3.5.2.5. IOST Suspend State

The IOST enters a suspend state whenever an error condition is encountered by the IOST or a channel presenting status. A machine check level interrupt by the IOST also results in the suspend state.

The IOST suspend state is characterized by the following conditions:

1. The IOST maintains the SSRA line active to the selected channel, inhibits further status requests from this channel, and inhibits scanning for status requests from the other channels.

2. The IOST stores an IOSTSW except on an IOST machine check error.

3. The IOST interrupt request is set to 1.

A channel-detected error is indicated to the IOST by way of the buffered channel status word servicing error (BCSWSE). The IOST sets the CBW bit and stores an IOSTSW with the appropriate channel number. The channel resumes normal data transfers after deactivating the SSR. The IOST enters a suspend state until an SIO instruction to clear the suspend state is issued. The SIO deactivates the SSRA and allows the channel to deactivate the BCSWSE. The IOST and channel then resume normal status operations.

A time-out error causes the IOST to set the CTO bit to 1 and to store an IOSTSW with the appropriate channel number, then the suspend state condition is started and exists until cleared by the SIO instruction. The IOST and channel resume normal status operations after the suspend state is cleared.

Errors detected by the IOST in tabling status are indicated by either the IWE, IDE, or STF bit in the IOSTSW (bits 91, 92, and 95, respectively). The channel number stored in the IOSTSW is of the currently selected channel. The IOST enters the suspend state until cleared by an SIO instruction. The IOST and channel resume normal status tabling operations after the suspend state has been cleared.

The IOST detecting a CTO or CBW error during initial load control storage (ILCS) or initial program load (IPL) results in a nonrecoverable error.

3.5.3. Programming Considerations

The size of the status table can vary from 1 word to a maximum of 1024 words. If a 1-word table is specified (active count in IOSTCW), the IOST enters the suspend state after storing the first word of an IOSTIW if a channel presents a multiple IOSTIW. The entire IOSTIW can be recovered from the BCSW location.

Software must maintain its own pointer to the next location where it expects to find an IOSTIW. On receipt of an interrupt, the software must check the IOSTSW location for error indications prior to looking for an IOSTIW. While processing IOSTIWs, the software can take advantage of handling more than one IOSTIW per interrupt and thus reduce the overhead of returning to the supervisor mode to process subsequent IOSTIWs.

Software must use care in handling multiple word IOSTIWs. The status could be partially stored in the status table; however, the entire IOSTIW still is contained in the BCSW and the IOST is in the suspend state. This would occur if an STF, IWE, or IDE error is encountered by the IOST after having transferred partial status out of the BCSW. The v bit of all the multiple words of an IOSTIW must be checked for 0; if all v bits are not 0, an error condition has occurred. The software must check the IOSTSW, recover status from the BCSW location, and issue an SIO instruction go out of the suspend state.

As each IOSTIW is processed, its v bit must be set to 1 to return the IOSTIW location for use by the IOST in the revolving table.

Status table entries should be processed in the order of their presentation.

When the IOST is in the suspend state, the contents of the IOSTCW can be altered only after alleviating any error conditions that may be indicated in the IOSTSW and before issuing an SIO instruction to clear the IOST suspend state. Software must clear the IOSTSW location before issuing an SIO instruction to clear the IOST suspend state.

3.6. INTEGRATED PERIPHERAL CHANNEL

The integrated peripheral channel (IPC) consists of the hardware required to perform all normal channel functions, such as data collection, data distribution, data buffering, storage addressing, and BCW maintenance for the integrated peripherals. This hardware is shared by all of the control units for the peripheral devices. The system software controls the I/O activity of the IPC and the associated controls through the preparation of the appropriate BCW and the initiation of an SIO instruction.

This instruction transfers a command to a peripheral control to specify certain types of information transfer. Commands are executed by the controls.

Information concerning the IPC that is not discussed in this section is available in the 90/30 System integerated peripheral channel programmer reference, UP-8041 (current version).

3.6.1. IPC Controls

The IPC accommodates six peripheral controls with circuitry contained in the processor. The controls are associated with the following peripheral devices:

- System console
- 0773 printer subsystem or 0778 printer subsystem
- 0717 card reader subsystem
- 0719 card reader subsystem
- 0605 card punch subsystem
- 8413 diskette subsystem
- Communications adapter (2 maximum) with line adapters

The system console, both card readers, either printer, and the card punch are each assigned 12 device addresses. The diskette subsystem is assigned four device addresses. A particular device address correlates with an associated device, line adapter, or diskette drive. The correlation cannot be waived because properties unique to a particular control are identified in the device address.

3.6.2. IPC Operation

IPC band width permits simultaneous operation of all integrated peripherals; thus, the IPC can operate as a time division multiplexer with respect to main storage interface. The IPC can support a maximum aggregate data rate of 50K bytes per second, which can be maintained unless interferance from higher priority channels is encountered.

Data requests from each control are honored on a priority basis. Main storage access is granted to each control for sufficient time to fill or unload data registers in the peripheral control. The amount of information transferred during a data transfer sequence may consist of a byte, half word, or full word, depending on the particular control and its mode of operation.

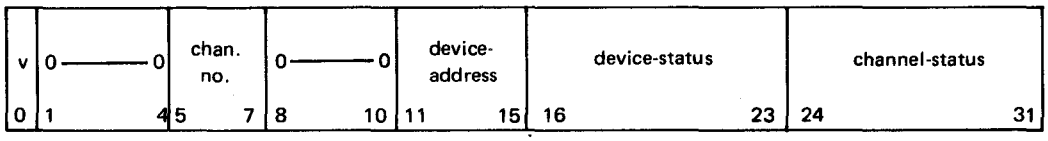
3.6.3. Status Sequences

The IPC presents all channel and device status by way of the storage of an interrupt word (IOSTIW). The IPC stacks channel status and each control unit stacks device status. Channel status and device status are presented to the IOST by separate status sequences. The IOSTIW is stored in fixed locations of low-order main storage (locations $0E0_{16}$ through $0E3_{16}$) in the first four bytes of the buffered channel status word (BCSW). Control for the storage of the IOSTIW by the IPC is furnished by the IOST within the processor.

If an error occurs during the storage of the IOSTIW, the IPC generates a BCSW service error (BCSWSE) indication to the IOST. The IPC attempts to store both half words of its IOSTIW. Either or both half words can be in an intermediate state. The state of the IOSTIW is a function of the error and its time of occurrence. The IPC does not generate another IOSTIW for the interrupt that encountered the error condition.

If the error occurs during the storage of a buffer terminate interrupt, eventual depletion of the byte count results in termination of the control unit since the f bit of the associate BCW is set to 0. The IPC generates a BCSWSE indication as first described.

The format and description of the IOST interrupt word (IOSTIW) of the IPC are as follows. This word is prepared by the IPC and is stored in the BCSW for each IPC and/or control status requests.



IOSTIW for IPC

v (bit 0)
A single bit used by the IOST for control purposes. The IPC sets this bit to 0 when storing the IOSTIW.

Bits 1–4 and 8–10
Unassigned and are set to 0 by the hardware.

Chan. no. (bits 5–7)
A 3-bit field that identifies the channel that presented the status. The IPC channel number is 0.

Device address (bits 11–15)
A 5-bit field that identifies the subchannel and/or device to which the channel and/or device status pertains.

Device status (bits 16–23)

An 8-bit field that identifies successful completion or error conditions that are directed at the control unit and device level. Individual bit descriptions are as follows:

Attention (bit 16)

Indicates an operator-initiated transition from the stop to the run state. This condition is asynchronous to and independent of program-initiated operations.

Bits 17–20

Set to 0 by the hardware.

Device-end (bit 21)

Indicates the completion of a previously initiated command by the subsystem and readiness to accept a new command. Successful completion is indicated if the device-end status is presented without unit check status.

Unit check (bit 22)

Indicates some unusual condition has been detected at the subsystem level:

- during the execution of the current command and presented with device-end status; or
- presented without the device-end status; indicates command rejection during an SIO sequence prior to the initiation of the current command.

Unit exception (bit 23)

Indicates an unusual condition has occurred as the result of initiating an operation. The condition may not necessarily represent an error. It may be presented as a result of an SIO or at the completion of the operation.

Channel status (bits 24–31)

An 8-bit field that identifies certain error conditions and the buffer terminate condition for data chaining sequences. The error condition is detected by the processor or by the IPC. Individual bit descriptions are as follows:

Bits 24–26

Set to 0 by the hardware.

Invalid address (bit 27)

Indicates that a protection exception or an addressing exception condition occurred during any storage access excluding IOSTIW references.

Channel data check (bit 28)

Indicates that a storage parity check has been detected on the fetch or storage of data from or to main storage during a data transfer sequence.

Bit 29

Set to 0 by the hardware.

Channel control check (bit 30)

Indicates that an address check has occurred during an IPC operation, with main storage, excluding IOSTIW references. This bit is set in conjunction with the invalid address if a protection exception or an addressing exception occurs during the BCW fetch or restoration. This bit also is set if a storage parity check is detected during a fetch or storage of the BCW.

Buffer terminate (bit 31)

Indicates that the IPC has performed the replacement operation required in the data chaining operations. The software must store new replacement parameters in the BCW(r) fields.

3.7. INTEGRATED DISK ADAPTER

The integrated disk adapter (IDA) is a combination channel and control unit designed to operate with 8416 and 8418 disk drive units. It receives and transmits data between the processor and disk subsystems.

In addition to storing and retrieving information on the disk surfaces, the IDA can search for specific information before transferring data to main storage. Controls are program-controlled to move accessor arms to reach any of the concentric cylinders on the disk surfaces.

The IDA interfaces the processor and 8416 or 8418 disk drives. The 8416 and 8418 both use removable disk packs, each having a storage capacity of 28.96 megabytes. However, the 8418 disk subsystem may use a high density recording method that doubles capacity to 57.9 megabytes.

Configurations using 8416, or 8416 and 8418 disk subsystems provide a total of 231.6 megabytes of online storage. Configurations using all 8418 disk drives with high density recording provide a total of 463.3 megabytes of online storage. The nominal data transfer rate for both subsystem types is 625K bytes.

The disk subsystems and IDA use a modified frequency modulation (MFM) recording technique. The IDA receives or transmits data through the processor interface in half-word parallel form, and receives or transmits data through the disk drive interface in bit-serial form. Data flows in only one direction at any given time. No data translation is performed within the IDA except the encoding the decoding required for the MFM scheme.

Additional programming information for the 8418 disk subsystem is provided in the 8418 disk subsystem reference, UP-8362 (current version),

Checking is performed within the IDA, with odd parity being carried throughout. Error correction code (ECC) bytes are used after each field on the disk surface of the disk units. A listing of the checking performed in the IDA is as follows:

- Read bus checking

  Data (including the BCW) received from main storage is parity checked as it enters the IDA.

- Internal checking

  As portions of the BCW are used to compare data on the disk surface, the parity carried with them is checked and data is checked for correct parity before being written on the disk surface.

- Command validity and certain address violations are checked.

- Correction code check

  The integrity of a field is checked as it is processed through the use of the ECC bytes.

3.7.1. Disk Pack Configurations

Data on the 8416 and 8418 disk packs are organized in a similar manner except that the 8416 has a track density of 192 tracks per inch (TPI) and the 8418 has a density of 370 TPI. Reference should be made to 8418 disk subsystem reference, UP-8362 (current version), for additional information on the data organization and track format and their parameters. The information applies to both the 8416 and 8418 disk subsystems.

3.7.1.1. Addressing

The IDA is identified by channel number 3. (See 4.5.) The disk drives attached to the IDA are identified by an 8-bit address field corresponding to bits 24 through 31 of the start I/O device address field. Since a maximum of eight devices may be attached to the IDA, only bits 29, 30, and 31 are used; bits 24 through 28 are unassigned and must be set to 0 (Table 3-1).

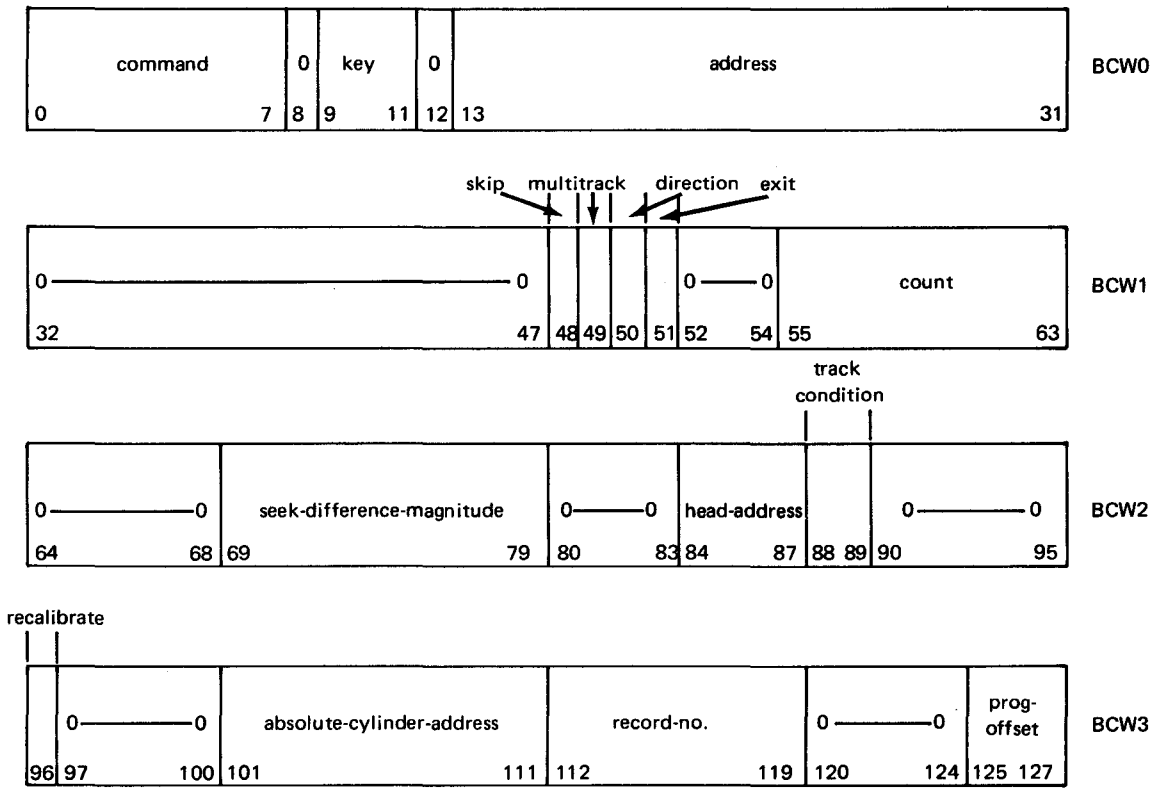
Table 3—1. IDA Device Addresses

Device	Device Address		
	29	30	31
Disk drive 0	0	0	0
Disk drive 1	0	0	1
Disk drive 2	0	1	0
Disk drive 3	0	1	1
Disk drive 4	1	0	0
Disk drive 5	1	0	1
Disk drive 6	1	1	0
Disk drive 7	1	1	1

3.7.2. Buffer Control Word

Buffer control words (BCW) are fixed areas of main storage that contain the parameters needed to permit the IDA to execute any of its commands. The BCW for the IDA is composed of four words located at $0F0_{16}$ in low-order main storage. The command code is transferred to the IDA during SIO. If the 00_2 condition code is returned, the remaining portion of the BCW is transferred to the IDA. The BCW in low-order main storage should not be altered until the I/O operation has been completed. Bytes within the IDA are manipulated by the IDA as the command progresses. The BCW is loaded by either the software or automatically by the processor (initial load).

The format and description of the IDA BCW are:



BCW for IDA

Command (bits 0–7)

Specifies command code to be executed by the IDA. The first four bits (0–3) must be set to 0.

Bits 8, 12, 32–47, 52–54, 64–68, 80–83, 90–95, 97–100, and 120–124

Unassigned and must be set to 0 by the software.

Key (bits 9–11)

A 3-bit field containing the I/O storage protection key. This key is presented to the main storage interface during each data transfer sequence. The IDA does not modify the key field during data transfer sequences; however, addresses that are generated internally by the IDA (for example, references to low-order main storage) are presented to main storage interface with a key of 0.

Address (bits 13–31)

Indicates the storage address on which the command is to operate. Within the IDA, this field is incremented as the command progresses and must begin on a half-word boundary.

Skip sentinel (bit 48)

Only effective with the read-data or search/read commands. When set to 1 with:

- Read-Data Command

  Inhibits data transfers to main storage.

- Search/Read Commands

  Specifies that searching begin at index.

Mulitrack sentinel (bit 49)

Used in conjunction with the search/read commands. When set to 1, indicates that the search operation is to be limited to cylinder boundaries rather than a single track. A multitrack search terminates with the search using head address 6 if the search conditions have not been satisfied.

Direction sentinel (bit 50)

When set, specifies the accessor moved in the negative direction (direction of decreasing cylinder numbers). This bit, even through transmitted to the device, has no meaning if a restore condition is indicated by bit 96 or if zero difference is specified, but is effective for all seek operations.

Exit sentinel (bit 51)

When set, a data error in the data field has caused the operation to terminate. An error condition is presented immediately instead of being deferred to the normal end of the operation.

Count (bits 55–63)

Used as follows:

- Search/Read Commands

  To specify the number of bytes to be searched (search key length) up to a maximum of 256 bytes. A count of 0 causes the first record processed to satisfy the search requirements and results in transferring that data field to main storage.

- Read-Data or Write-Data Commands

  To specify the number of records to be processed (record count). No records are processed if count is 0.

Seek difference magnitude (bits 69–79)

Used during a seek operation and specifies the magnitude of the difference between the accessor arm present position and the location desired.

Head address (bits 84–87)

Specifies the head address for the current operation. Head addresses 0 through 6 specify heads of the removable pack. Any addresses other than those above are invalid.

Track condition (bits 88–89)

Specifies the condition of the track on which the operation is to act.

Recalibrate (bit 96)

Specifies that the accessor is to be reoriented and moved to cylinder 0. This bit overrides the contents of bits 71 through 79 and the direction bit (bit 50).

Absolute cylinder address (bits 101–111)

Specifies the intended final position of the accessor after a seek or recalibrate command has been completed.

Valid addresses are:

- 8416 or 8418-04/05

  0_{10}–410_{10}

- 8418-02/03

  0_{10}–814_{10}

Record number (bits 112–119)

Specifies the number of the first records on which the operation is to be performed or initiated. This number should not exceed 40_{10}, the highest numbered record on a track.

Programmed offset (bits 125–127)

Used to specify to the selected disk drive a small offset in the positioning mechanism to allow data recovery from an otherwise unrecoverable segment of the recording surface. The bits are used to specify one of four offset positions, two on each side of the nominal track position.

The three bits are defined as follows:

Bit 125

Specifies whether offset is to be employed for the command.

1 Programmed offset to be employed for command
0 Programmed offset not to be employed and bits 126 and 127 ignored

Bit 126

Specifies magnitude of offset.

1 Offset major
0 Offset minor

Bit 127

Specifies direction of offset.

1 Offset away from hub
0 Offset toward hub

Programming Note:

Some BCW fields are meaningful only for specific commands. Table 3-2 summarizes the BCW fields applicable for each command.

Table 3—2. Summary of BCW Fields Applicable for Each Command

BCW Field \ Command	Format Write	Write Data	Search/ Read Equal	Search/ Read HI or Equal	Read ID	Read Data	Seek	Sense	ECC Sense	Diagnostic	ECC Diagnostic
Command	X	X	X	X	X	X	X	X	X	X	X
Key	X	X	X	X	X	X		X	X	X	X
Address	X	X	X	X	X	X		X	X	X	X
Skip			X	X		X					
Multitrack			X	X							
Direction	X	X	X	X	X	X	X				
Count	X	X	X	X		X				X	X
Seek difference magnitude	X	X	X	X	X	X	X				
Read address	X	X	X	X	X	X	X				
Track condition	X	X	X	X		X					
Recalibrate	X	X	X	X	X	X	X				
Absolute cylinder address	X	X	X	X		X					
Record number	X	X				X					
Programmed offset			X	X	X	X	X				

3.7.3. Command Repertoire

The command repertoire (codes) for the IDA provides the capability to read, write, search, or interrogate the subsystem. Table 3-3 lists these commands with their hexadecimal command codes.

3.7.3.1. Write Commands

During a write command, information is transferred from main storage through the IDA and written by the device. The IDA automatically generates and affixes the ECC bytes at the end of the appropriate fields. The IDA also generates gap constants and data patterns for these gaps. All write commands are preceded by an implied seek. The two write commands are format write and write data.

Table 3—3. IDA Command Codes

Command	Command Code in Hexadecimal
Format write	01
Write data	05
Search/read equal	09
Search/read-high-or-equal	OD
Read ID	0E
Read data	02
Seek	08
Sense	04
ECC sense	03
*Diagnostic	06
*ECC diagnostic	07

*For diagnostic use only

3.7.3.1.1. Format Write

This command (01) is used to format an entire track. The BCW must contain the track condition flags, the cylinder address, the head address, the number of records to be written on the track (40), and the record number, which must be set to 1. The main storage area specified by the address field of the BCW must contain two identical bytes that are used to fill the data fields of each record. The format-write command is inherently limited to one track. There can be no multiple track format-write commands. This command starts to write valid data at index time and continues to write the ID field and the data field for each of the 40 records on that track.

If the track to be formatted is designated as a defective track, the ID field should contain the cylinder and head address of its alternate track. The head numbers of the alternate and defective tracks must be identical, since any attempt to write a different head number results in a seek to the new head by a virtue of the implied seek.

Channel-end and device-end status are presented at the termination of the writing on a track. No implied search exists with the format-write command.

3.7.3.1.2. Write Data

The write-data (05) command writes 256 bytes into the data field of each record to be written. The information written is transferred from main storage with the storage address of the BCW specifying the address of the first data byte to be written. The BCW contains a count indicating the number of records to be written. This command can be of a multiple track nature if a sufficient number of records are specified. A maximum of 40_{10} records can be written on each track of a cylinder. Channel-end and device-end status are presented at the conclusion of the last record written.

No data fields are written unless the ID for that record compares exactly, including error correction code information. If the ID does not compare exactly, the operation is terminated and verification of unit check status is presented to the processor while the appropriate sense bits are set within the IDA.

The write-data command normally progresses by searching the ID of each record on a track until it finds one specified by the BCW. It commences writing the following data field after which it decrements by 1 the count of records to be written. Bytes stored in the IDA that reflect the ID of a record are altered to permit the search of the next ID in sequence; this occurs only if the residual count (within the adapter) of the number of records to be written is greater than 0. Head switching occurs at index time. A write-data command cannot cause writing to exceed a cylinder.

3.7.3.1.3. Error Exits

If an error occurs when processing a data field, immediate action is taken within the IDA. During a read operation, the command and sequences are immediately terminated and an interrupt word is stored. If a write operation is in progress, the remaining portion of that data field is filled with 0's to avoid creating discontinuities within the field. The interrupt word is stored when the data field is ended. Note that even though the entire field is written, the field may not be recoverable as a result of the error, nor can the ECC bytes be expected to be valid.

3.7.3.2. Search/Read Commands

The search/read commands of the IDA are unique in that they search the data field of a record for the number of bytes specified by the count field of the BCW. If the conditions of the search are satisfied, the remaining bytes of the data field are read and transferred to main storage. Error correction code information is computed for the entire field and is checked at the end of the data field. The search/read command is a single record operation. During the search portion of these commands, the IDA is in an output (write) mode while the device is in a read mode. The device remains in a read mode throughout, but upon satisfying the search conditions, the IDA switches to an input (read) mode. If bit 48 of the BCW is set to 1, searching does not start until the index is detected.

All search/read commands commence by ascertaining that cylinder and head positions, as specified by the BCW, are correct. (Note that the record number is not checked with these commands.) This is accomplished by comparing the information of the ID field as it is read from the disk surface (the same procedure used on other commands). Satisfaction of correct positioning permits the data field of that record to be compared against the information from main storage to the extent as specified here.

Bit 49 of the BCW is used to specify multiple track operations. This extends the limit of the search/read command to a cylinder rather than a head (track). On the search/read-equal-or-high, the heads do not increment at index time unless a low condition has been found. Two passes of index on a single track cause a no-record-found indicator to be set in the sense bytes.

The utilization of the search/read-equal-or-high command requires that data be written in ascending order. Data of the least magnitude starts on the first record of head 0. Comparison is always made from device to main storage; hence, a 2 on the disk compared to a 1 from main storage satisfies the 'greater than' (high) requirement.

The search/read commands require a main storage buffer area of 256 contiguous bytes. The search key is contained in the first section (number of bytes specified by the count field) while the data is read into the remainder of that buffer after the search conditions are satisfied.

An all 1's byte (FF_{16}) transferred from main storage automatically compares with the corresponding byte read from the record being searched.

When the IDA determines that the search conditions are not satisfied, it continues reading through the data field and checks the error correction code information. An error correction code check error on any data field is delayed until either the search is satisfied or some other condition terminates the operation. Unit check status is presented if an error correction code check error occurred on any of the encountered data fields.

The search/read commands include search/read-equal and search/read-high-or-equal.

3.7.3.2.1. Search/Read-Equal

This command (09) requires that the search key information compare exactly before the remainder of the data field can be recovered. This command allows the index to pass twice on the first track before incrementing the head if the multiple track indicator is set. The no-record-found indicator is set if the index is detected twice without the compare being satisfied. No-record-found and end-of-cylinder sense indicators are set in the sense bytes in the event that the search conditions are not met by the time the index is detected on the highest order track. If the multitrack sentinel is set and the index is encountered, the head address is incremented by 1. This continues until the search conditions are satisfied or the operation is otherwise terminated.

The search/read-equal command may be used as an absolute byte-to-byte check of the data written in a data field. If it is desired to use this capability, the search/read-equal command should be issued immediately following the write-data command. The check should be made on the same information that was written; that is, the same data buffer can be used. The byte count should be set to the full record length of 256 bytes.

3.7.3.2.2. Search/Read-High-or-Equal

The search/read-high-or-equal (0D) command operates in much the same manner as the search/read-equal command. When this command is first initiated, positioning information is checked, and upon detection of the first data field, a compare of the search data is initiated. If the data from the disk drive is greater than that from main storage, all recovery operations cease until the index is detected. (The conditional requirement of a low before a high must be detected before a high can be accepted.) The multiple track operation can proceed (increment head) only if a low condition is detected on the first track searched. There is no differentiation made between the condition being satisfied by the equal or high requirements.

3.7.3.2.3. Implied Searches

The ID of each record, from which or to which data is to be transferred (except format-write command), is searched against the applicable contents of the BCW. This is an implied search.

3.7.3.3. Read Commands

Read commands transfer recovered data to main storage. Error correction code (ECC) information is checked at the end of the appropriate field, causing the proper sense and status bits to be set in the event of an error. ECC information is not transferred to main storage.

3.7.3.3.1. Read ID

The read (OE) command attempts to recover the first ID (microcode identification) field encountered, after the command is initiated, and transfers the content of the field to main storage. No comparison is made on the content of the ID, with the exception of ECC information. If bad cyclic check information is encountered, the next ID is read. This operation continues until either a good ID is recovered or some error (for example, no record found) other than an ECC check error occurs. This command attempts to recover the first ID encountered after initialization, which is not necessarily the first record after index. Five bytes of information (contents of the ID field) are transferred to main storage.

3.7.3.3.2. Read Data

The read-data (02) command is used to retrieve the data fields of a disk pack. The read-data command begins by searching for the ID of that record specified in the BCW. When the ID compares exactly, the data field of that record is transferred to main storage. If a multiple record read is specified, the record number and record count are updated at the end of a recovered data field. This operation continues until the total number of records requested have been transferred. This command is limited by cylinder boundaries, not head limits.

The ID specified in the BCW is sought until two indices have been detected, in which case the no-record-found sense bit is set. If more records than the cylinder can contain are specified, the end-of-cylinder sense bit is set at index time of the highest order track. Note that the total data capacity of a cylinder can exceed the storage capacity of the processor; therefore, care must be exercised when specifying multiple record reads.

The read-data command also can be used for write checking. If bit 48 in the BCW is present with the read-data command, the data field of the specified record or records is clocked through and the ECC information verified. This operation works exactly as the normal read-data command except that no data is transferred to main storage.

The first ID of this command must compare exactly, including ECC information, before the recovery of data can take place. Once the operation has commenced, subsequent IDs need not compare if they contain an ECC check error. The detection of an ECC check error on a subsequent ID causes the unit exception status to be presented when the entire operation is completed. Note that a miscompare of an ID that has good ECC information terminates the operation; this again is other than the first ID encounter. If errors are encountered, the sense bytes indicate that either an ECC check error occurred in an ID field or an ID miscompare occurred.

Any ECC check errors detected in a data field cause unit check status to be generated at the end of the command. An ECC check error on any data field in a multiple read operation causes the appropriate sense bits to be set. Because an error is deferred until the operation is completed and there is no provision for multiple errors of the same type, it is necessary that a series of single record reads be executed to isolate and identify the record or records in error. If a faulty ID is encountered during these single record reads, it is necessary to issue a multiple record read (that is, two records), starting with the good ID immediately preceding the faulty ID, to recover the data of the record having the faulty ID. The multiple record read must begin on a record with a good ID.

3.7.3.4. Seek Command

The seek command and the implied seek perform identical cylinder and head selection functions. In cases where more than one device is present in the subsystem, use of seek commands to initiate simultaneous seeks at the device level results in a saving of accessor positioning time. In cases where only one device is to be addressed, however, use of the implied seek is the most efficient method of positioning. If only head address modification is required, the implied seek is sufficient and no time saving results if the seek command is utilized.

3.7.3.4.1. Seek

The seek (08) command is used to reposition the accessor, as well as to select new heads. The BCW contains the magnitude and direction of accessor movement. Recalibrate, which is an automatic return to cylinder 0, overrides any difference or direction bits that may be in the BCW.

As with all commands, the device status is presented by way of an interrupt word. The seek command presents the device status with channel-end and device-end status if no accessor movement is required. After the IDA has completed the sequences necessary to start accessor movement, the channel-end status is presented and the IDA is free to accept another command. When accessor movement is completed, device-end status is then presented. The IDA has the ability to store device-end status for each device independently. Once device-end status is accepted, that status is cleared in both the IDA and device. The device is busy to software, from the onset of a seek until the interrupt word containing device-end status has been accepted.

It is possible to issue overlapping seeks with the seek command. This is accomplished by issuing a seek command to the next device to be positioned after receiving channel-end status indication from the previous device. The device-end interrupt words are sent to the processor by way of an arbitrary priority scheme within the IDA, with disk drive 0 having highest priority; disk drive 1, second highest priority,; and so forth.

3.7.3.4.2. Implied Seek

Seeks are automatically issued before all commands except the sense, ECC sense, diagnostic, and ECC diagnostic commands, which in effect are the implied seek. Information necessary for the seek is contained in the BCW. This function is completed prior to the main portion of the command. The implied seek acts as a chained operation and the adapter is busy while this operation takes place. If a difference is specified, accessor movement takes place as specified in the BCW. Recalibrate overrides any difference magnitude. The direction bit has no meaning if a 0 (no) difference is specified or recalibrate is specified. Head selection is effected by this function, but head switching (incrementing) is accomplished only after the main portion of the command has been entered.

3.7.3.4.3. Programmed Offset

Programmed offset is used to specify to the selected disk drive a small offset in the positioning mechanism to allow data recovery from an otherwise unreadable portion of the recording surface. Three bits in the BCW are used to specify the offset. No offset is permitted during the write command. If offset is attempted with any write command, the command is rejected and an error is indicated.

Programmed offset is treated exactly as a normal seek operation would be treated; that is, the device is busy until the offset movement is complete. Typically, there is a 0 seek difference specified when offset is used; however, movement between cylinders is allowed with offset when specified. It should be noted that programmed offset does not occur when recalibrate is specified. No integral cylinder movement is permitted from an offset position to a different cylinder when offset is specified.

3.7.3.5. Sense Commands

The sense commands interrogate the hardware and inform software of any existing errors.

3.7.3.5.1. Sense

The sense (04) command is used to interrogate both the IDA and the device. The issuance of a sense command causes five bytes of sense information to be transferred to main storage. The sense command does not clear the sense information within the IDA. The interrupt word presented as a result of the sense command reflects the status for that sense command. The sense command is issued following the magnitude search/read command to determine on which head and record the search conditions were satisfied. The definitions of the bits, the combinations in which they can be set, and the conditions that set them follow. Tables 3–4 and 3–5 follow the description of the sense bytes and are a summary of the IDA bytes and the setting of sense bits for each command, respectively.

Sense Byte 0

Command reject (bit 0) SB0, 0

When set alone, indicates an invalid command code, an invalid head address, or an attempted write command when programmed offset was selected in the BCW. (Bit 125 of BCW is set to 1.) It can also be set with file protect (SB1,5) to indicate that a write command was issued to a file-protected device.

Intervention required (bit 1) SB0, 1

Is set with the stop state (SB0, 6) to indicate that an operation was attempted on a device that was either nonexistent or in the stop state.

Output parity check (bit 2) SB0, 2

Indicates that a parity error was detected at the end of the output queue, such as the input to the shifter.

Equipment check (bit 3) SB0, 3

Is set to indicate that a serious hardware malfunction has occurred within the subsystem. If set alone, it indicates that the IDA control logic contains an error. If set with other sense bits, it indicates an error within a device. Specifically, it may be set with device check (SB0, 7) to indicate that the device selected has a serious hardware failure requiring maintenance; with seek incomplete (SB2, 0) to indicate that a selected device was not able to complete a seek within a specified period of time; or with unselected status (SB2, 5) to indicate that status has been received from a device not selected. It may also be set with track overrun (SB1, 1) to indicate that the index has occurred while a record was being processed. This indication implies that the disk is rotating too fast relative to the write oscillator.

Data check (bit 4) SB0, 4

Is set with no clocks (SB2, 7) to indicate that no clocks have occurred for 10 milliseconds while reading and writing, or with comparison parity check (SB2, 1) to indicate a parity bit miscompare was detected at the comparator. It also is set with head/cylinder miscompare (SB2, 2), with record number miscompare (SB2, 3), or flag byte miscompare (SB2, 4) to signify difficulty with a seek operation, difficulty in maintaining orientation in the read circuitry, or an incorrect flag byte, respectively; and no ECC check error is detected.

Data check also is set with ID field check (SB1, 0) or data field check (SB1, 7), together with either sync region error (SB1, 6) or ECC check (SB2, 6). These bit combinations indicate either that an error (such as an improperly detected index mark or an incorrect sync byte) has been detected in the sync region or that an ECC check error was detected. They also indicate whether the error was detected in an ID field or a data field.

Overrun (bit 5) SB0, 5

Indicates during a write or search operation that main storage has not supplied data at a sufficient rate to satisfy the data rate of the device. It sets when the output queue is empty and a request is made of the queue for more data.

It also sets during a read operation if main storage accepts information too slowly. Overrun sets when the input data register is full and the shifter is ready to transfer a full byte to it.

Stop state (bit 6) SB0, 6

Sets with intervention required (SB0, 1) if an operation was attempted on a device nonexistent or in the stop state. It is set alone if the device addressed by the sense command is nonexistent or in the stop state but no operation was attempted on the device.

Device check (bit 7) SB0, 7

Sets with equipment check (SB0, 3) to indicate that the device selected has a serious hardware error requiring maintenance or operator intervention. If conditions in the device cause device check to set when that device is not selected, device check may be presented alone; this occurs only if no attempt has been made to use that device. Examples of errors that can cause this are multiple head selection and write current on without having been selected. Note that device check can be set if a seek or an implied seek is issued to a nonexistent cylinder.

Sense Byte 1

ID field check (bit 0) SB1, 0

Sets with data check (SB0, 4) and either sync region error (SB1, 6) or an ECC check (SB2, 6) to indicate either than an error was encountered in the ID field preamble or in the ID field ECC bytes, respectively.

Track overrun (bit 1) SB1, 1

Sets with equipment check (SB0, 3) to indicate that the index was encountered while processing a record.

Cylinder end (bit 2) SB1, 2

Indicates an operation was complete at the end of a cylinder.

High density mode (bit 3) SB1, 3

Set to 1 when LOW DENSITY switch on 8418-02/03 disk drive is reset.

No record found (bit 4) SB1, 4

Is set alone during a search/read command to indicate that the search conditions were not satisfied after two passes of the index if the multitrack bit is not set in the BCW. If the multitrack bit is set, it is set in conjunction with cylinder end (SB1, 2) if the search conditions were not satisfied by the end of the cylinder. No record found will also set in conjunction with sync region error (SB1, 6) if no data was found on the selected track; for example, two passes of the index occurred without an intervening ID preamble. It will also be set alone on read-data or write-data commands if the record number specified in the BCW cannot be found.

File protect (bit 5) SB1, 5

Sets if the device selected is in file protect mode. Command reject (SB0, 0) also sets if a write operation was attempted on the device.

Sync region error (bit 6) SB1, 6

Sets with data check (SB0, 4) and either ID field check (SB1, 0) or data field check (SB1, 7) to indicate that an error was determined in the preamble of either the ID field or the data field of a record, respectively. This may set either because of an erroneous sync region or because an 8418 disk pack formatted in high density mode was used on a low density device, or vice versa.

Data field check (bit 7) SB1, 7

Sets with data check (SB0, 4) and either sync region error (SB1, 6) an ECC parity check (SB2, 6) to indicate that an error was encountered in the data field preamble or in the data field ECC bytes, respectively.

Sense Byte 2

Seek incomplete (bit 0) SB2, 0

Sets with equipment check (SB0, 3) to indicate that a seek operation could not be completed within a specified time, or that the positioner on a disk drive drifted off track while the drive was selected but no head had yet been selected. Seek incomplete can be cleared only by issuing a recalibrate command to the drive that caused the error.

Comparison parity check (bit 1) SB2, 1

Indicates parity bits associated with bytes found to be identical by the comparator are not identical.

Head/cylinder miscompare (bit 2) SB2, 2

Sets with data check (SB0, 4) to indicate that the head or cylinder number specified in the BCW does not match the corresponding bytes on a record read from the disk and no ECC check error occurred.

Record number miscompare (bit 3) SB2, 3

Sets with data check (SB0, 4) to indicate that the record number read from the disk is not the one expected after a string of records has begun to be processed. This bit implies that one or more records have been skipped accidentally.

Flag byte miscompare (bit 4) SB2, 4

Indicates that the flag bits read from the disk did not match those in the BCW. Data check (SB0, 4) is also set when this bit is set.

Unselected status (bit 5) SB2, 5

Sets with equipment check (SB0, 3) to indicate that a device has raised a status line when no device was selected.

ECC check (bit 6) SB2, 6

Sets with data check (SB0, 4) and either ID field check (SB1, 0) or data field check (SB1, 7) to indicate that an ECC error was detected in either the ID field or the data field, respectively.

No clocks (bit 7) SB2, 7

Sets with data check (SB0, 4) to indicate that no clocks have occurred between 7 and 13 milliseconds while reading and writing.

Sense Byte 3

Bits 0–3 SB3, 0–3
Unassigned and are set to 0 by the IDA.

Head address (bits 4–7) SB3, 4–7
Indicates which head was last selected.

Sense Byte 4

Bits 0–1 SB4, 0–1
Unassigned and are set to 0 by the IDA.

Record number (bits 2–7) SB4, 2–7
Indicates the last record number partially or completely processed at the time of termination of the operation.

Table 3—4. IDA Sense Bytes

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
0	Command reject	ID field check	Seek incomplete		
1	Intervention required	Track overrun	Comparison parity check		
2	Output parity check	Cylinder end	Head/cylinder miscompare		Record Number
3	Equipment check	High density mode	Record number miscompare		
4	Data check	No record found	Flag byte miscompare	Head Number	
5	Overrun	File protect	Unselected status		
6	Stop state	Sync region error	ECC check		
7	Device check	Data field check	No clocks		

Table 3—5. Summary of Sense Bits Possible for Each Command

Commands	Sense Bits																							
	0 Command reject	1 Intervention required	2 Output parity check	3 Equipment check	4 Data check	5 Overrun	6 Stop state	7 Device check	0 ID field check	1 Track overrun	2 Cylinder end	3 High density mode	4 No record found	5 File protect	6 Sync region error	7 Data field check	0 Seek incomplete	1 Comparison parity check	2 Head/cylinder miscompare	3 Record number miscompare	4 Flag byte miscompare	5 Unselected status	6 ECC check	7 No blocks
	Byte 0								Byte 1								Byte 2							
Format write	X	X	X	X			X	X		X				X			X					X		X
Write data	X	X	X	X	X	X	X	X	X	X	X		X	X	X		X	X	X	X	X	X	X	X
Read data	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X
Search data	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X
Read ID	X	X	X	X	X	X	X	X	X	X			X	X	X		X	X				X		X
Seek	X	X		X			X	X						X			X					X		

3.7.3.5.2. ECC Sense

The ECC sense (03) command is used to provide information to software relative to the correction of errors encountered when reading or searching a data field. Determination of whether an error correction should be attempted is made by analysis of the sense bytes transferred by way of the sense (04) command. The combination of data check (SB0, 4), date field check (SB1, 7), and ECC check (SB2, 6), when set alone and reflecting results of a single record read-data (02) command or the search-read commands (09, 0D), indicates an ECC check error on a data field. Correction then can be attempted by software recovery procedure. Special consideration must be made for the occurrence of this condition on a search-read operation. In particular, it must be determined that the error did not occur within the key area of the data field being operated on by the search-read command. If the error did not occur in the key area, the record must be read (read-data command), the error correction applied, if required, and a software search accomplished on the corrected data in main storage.

Issuance of the ECC sense command causes the hardware to calculate the error pattern and displacement and determine whether the error encountered is correctable. Once the calculation is complete, three bytes of ECC sense information are transferred to main storage. Channel-end and device-end statuses are presented upon completion of data transfer. The format of the three bytes transferred is shown in Table 3-6. The first byte is the byte displacement, which (when the error is correctable) indicates the location of the error within the data field in question. The second and third bytes are the first and second pattern bytes, respectively, which define the error pattern on which an exclusive OR must be performed, with the data at the relative location specified by the byte displacement. If the first pattern byte contains all 1's, the error is then uncorrectable. In cases where the correction is unnecessary, the pattern bytes are 0's and the exclusive OR function does not alter the data patterns.

The error correction function is available for software use and should be properly applied within the framework of recommended recovery procedures.

Table 3—6. ECC Sense Bytes

Interpretation		Byte Displacement	First Pattern Byte	Second Pattern Byte
Correction necessary	Record read without error.	0000 1010	0000 0000	0000 0000
	Error burst starts within first byte of data field.	0000 0000 or 0000 0001 or 0000 010	0000 0000	0000 0000
Correctable error	Error burst starts within first byte of data field.	0000 0000	PPPP PPPP*	PP00 0000
	Error burst starts within second through 255th byte of data field.	0_{10}<displacement<255_{10}	PPPP PPPP*	PP00 0000
	Error burst starts within last byte of data field.	1111 1111	PPPP PPPP	0000 0000
Uncorrectable error		0000 0011	1111 1111	0000 0000

*P.....P is an error pattern. Any P bit set to 1 causes a reversal of the data bit at that position after an exclusive OR function takes place.

3.7.3.5.3. Sense Clearing

Upon decoding a valid command, other than a sense or ECC sense command, normal and ECC sense information are cleared, providing no errors generating channel status are present.

3.7.3.6. Diagnostic Commands

Diagnostics permit checkout of the input and output paths of the IDA.

3.7.3.6.1. Diagnostic

The diagnostic (06) command is used to exercise the output and input data paths of the IDA. The command causes data to be transferred from main storage, circulated through the output data path and shifter, and returned to main storage by way of the input data path. Data is transmitted from and stored back into the data area specified by the address field of the BCW. Upon completion of the command, output data and input data are interleaved in alternating half-word segments in the data area. Input data half words can be compared against corresponding output data half words to verify the integrity of the IDA data path. Termination of data transfers is controlled by the count field in the BCW, which is interpreted as a byte count (number of output data bytes to be transmitted). Due to the alternating nature of the output and input data, the data area provided must be approximately twice that specified by the byte count. This command is for diagnostic use only.

3.7.3.6.2. ECC Diagnostic

The ECC diagnostic (07) command is used to exercise the output data path and ECC circuitry within the IDA. This command is similar to the write-data (05) command, with the exception that no device is required or selected, implied seeks or searches are not performed, and only a single record is processed. By the use of the ECC sense command, the displacement and error pattern information can be compared against expected results, determined by the data transferred on the ECC diagnostic operation. It should be noted that the count field of the BCW must be set to a nonzero value.

3.7.4. Start I/O

The start I/O (SIO) instruction is used to initiate all input/output operations. In its operation with the IDA, the SIO has associated with it an address to determine which of the attached devices is to be accessed. The presence of the SIO and an address are sufficient to probe the addressed device to determine the availability of that device. This initial probe is reflected by a condition code, being a unique 2-bit code returned to the processor. Table 3–7 summarizes the IDA condition codes. In the case where the IDA is busy executing a previously initiated command, the IDA returns a 10_2 condition code. No interrupt word is stored as a result of this operation. The normal interrupt word for the command in progress is presented when that operation is complete. The command is rejected (not called for) with this condition. It should be noted that finding the IDA busy implies a programming error. If a new SIO is issued, there is reason to expect that the BCW was changed while a command was in progress; this is not permitted since main storage or the recording surface may be erroneously altered.

Table 3—7. IDA Condition Codes

Condition Code	IDA State	Device State	Command Action	Interrupt Word Stored	Applicable Status Field in IOSTIW
00	Available	Run	Accepted	Yes ①	Device ②
01	Available	Busy	Rejected ③	Yes	Device
01	Available	Stop	Rejected ③	Yes	Device
10 ④	Busy	– –	Rejected	No	– –

NOTES:

① IOSTIW is stored in the BCSW at completion of command.

② Device status is stored if no other errors occur (such as channel status).

③ Sense, ECC sense, diagnostic, or ECC diagnostic commands are not rejected. Condition code = 00.

④ IDA is considered busy if pending status condition prevails.

Pending status is defined as the condition where IDA is attempting to present an interrupt word (e.g., containing the device-end status) that is stored in the IDA. Pending status is treated exactly as the IDA busy sequence. With the exception of this case (IDA busy), the IDA makes requests for the command byte. Once the command code has been received and examined by the IDA, the condition code is presented.

In all cases (except for the sense and diagnostic commands) where the IDA is available but the addressed device is busy, the IDA returns a 01_2 condition code. The command just received is not executed. The device busy condition occurs only if the device is completing some positioning activity that normally causes an interrupt word to be stored when accessor movement is complete. The appropriate status is entered into main storage after the command is rejected.

The IDA returns a 01_2 condition code if the IDA is available but the addressed device is in the stop state. The command (with the exception of the sense command) is not executed, but an interrupt word is stored.

If both device and adapter are available, a 00_2 condition code is presented and the remainder of the BCW subsequently is loaded with the IDA (if required). The IDA than executes the specified command and presents status by way of an interrupt word when the operation is complete.

Aside from the conditions previously noted, the 01_2 condition code is presented if an invalid command is issued to the IDA, if a channel error is detected, or unselected status (SB2, 5) is detected and the command is not a sense or diagnostic command. In all cases where 01_2 condition code is presented, the command is rejected. The interrupt word (IOSTIW) is stored after the SIO sequence. The state of the device is not reflected in the device status field of the interrupt word stored if a channel error is detected. This field is set to 0. Table 3-8 lists these error conditions.

Table 3—8. IOSTIW Error Conditions (IDA)

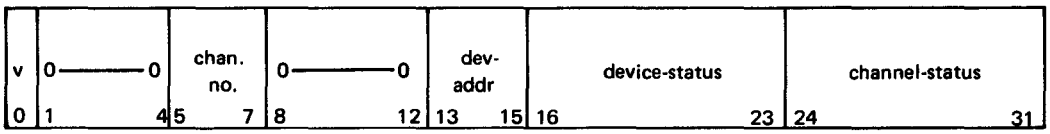
Error Condition	Applicable Status Field in IOSTIW
Invalid command or unselected status	Device
Protection exception on command reference	Channel
Address check on command reference	Channel
Storage parity check on command reference	Channel
Addressing exception on command reference	Channel

3.7.5. Status Sequence

The IDA uses an interrupt word (IOSTIW) to store all channel and device status. (See 3.5.1.2.) The interrupt word is stored in fixed locations of low-order main storage at location $0E0_{16}$ through $0E3_{16}$; this interrupt word corresponds to the first four bytes of the buffered channel status word (BCSW). Control for the storage of status is provided by the processor.

3.7.5.1. IOSTIW Format (IDA)

The format and description of the IOST interrupt word (IOSTIW) for the IDA are:



IOSTIW FOR IDA

v (bit 0)
A control bit used by the IOST. The IDA always sets this bit to 0 when storing an interrupt word.

Bits 1–4 and 8–12
Unassigned and are set to 0 by the hardware.

Chan. no. (bits 5–7)
Indicates channel number of the IDA and is set to 011_2 by the IDA.

Dev. addr. (Bits 13–15)
Indicates in binary code the device associated with this status entry.

Device status (bits 16–23)
Used to define conditions within the subsystem at the time the status entry is initiated. They reflect the condition of the device and the control section of the IDA only. Normal status (successful completion) in this byte is represented by channel-end or device-end status only. Excluding attention and busy, all other status bits require sense byte interrogation to further define the condition.

Attention (bit 16)
Indicates that one of the attached drives has undergone an operator-initiated transition from the stop state to the run state. This is an asynchronous status presentation not related to any software-initated sequences.

Bits 17 and 18
Unassigned and are set to 0 by the IDA.

Busy (bit 19)
Indicates that the addressed device is completing a previously initiated seek order, programmed offset, or has a pending gated attention. This bit is presented without end status (i.e., channel-end, device-end). This bit is not set with the sense or diagnostic commands.

Channel-end (bit 20)
Indicates that the IDA is able to accept another command.

Device-end (bit 21)

Is set with channel-end status at the normal end of all commands except the seek (08) command. A device-end status by itself indicates that accessor movement is complete and the disk drive is available to be accessed.

Unit check (bit 22)

Indicates that some problem exists with either the addressed disk drive or the IDA. Any data transferred during an operation causing unit check status to be presented must be rejected. A condition was detected, with any of the following sense bits set:

1. Sense byte 0, bits 0–5

2. Sense byte 1, bits 2–4

If this bit is presented without a channel-end or device-end status, it indicates that the abnormal condition occurred before any changes were made to the addressed device.

Unit exception (bit 23)

Indicates some abnormal condition occurred during an operation that does not affect the validity of the data transferred. This is not necessarily an error condition, but the sense bytes should be interrogated to define the condition. The followingconditions cause unit exception status to be presented:

1. An ECC check error occurs on an ID (other than the first record to be read), with the read-data command (sense byte 1, bit 0 and sense byte 2, bit 6).

2. An ID of a record not associated with the fields to be written (write-data command) has an ECC check error; unit exception status is presented (sense byte 1, bit 0 and sense byte 2, bit 6).

3. An ECC check or improper number of missing clocks are detected in the ID field with the search/read commands, causing unit exception bit rather than unit check bit to be set.

It should be noted that unit exception and unit check bits may be set together.

Channel status (bits 24–31)

Associated with the channel section of the IDA. The interface between the IDA and the processor or main storage is the source for these bits.

Bits 24–26, 29, and 31

Unassigned and are set to 0 by the IDA.

Invalid address (bit 27)

Indicates an addressing exception or a protection exception when accessing main storage for data or the BCW.

Channel data check (bit 28)

Indicates detection of a storage parity check on a data access (to or from main storage).

Channel control check (bit 30)

Is set when a storage parity check, addressing exception, or protection exception is encountered when accessing any portion of the BCW. It is also set upon detection of an address check on BCW or data accesses.

3.8. SELECTOR CHANNEL

The selector channel uses the four independent interfaces described in 3.4. Operation of the selector channel is in burst mode, allowing only one of eight subsystems connected to the channel to retain control of the interface throughout the I/O operation. Simultaneously, other subsystems may be executing previously initiated operations (e.g., rewinding tape) that do not involve operations with the channel via the I/O interface.

The processor initiates all I/O operations by issuing I/O instructions to a selected channel and a selected subsystem connected to the channel. After successful initiation, the channel maintains control of data transfers between main storage and the subsystem, independently of the processor. Upon completion of an I/O operation, the state of the channel and subsystem is presented to software via the appropriate status words by means of the I/O status tabler (3.5.).

3.8.1. Status Service Request

For the selector channel a STATUS SERVICE REQUEST (SSR) signal is set by the channel. When the IOST grants priority with a STATUS SERVICE REQUEST ACKNOWLEDGE (SSRA) signal, the selector channel then can write three words of status into the BSCW. The channel notifies the IOST of a successful or unsuccessful entry into the buffer. Interlocks exist to ensure that subsequent requests do not destroy status previously stored in the BCSW until serviced by the IOST. Once status has been successfully stored in the BCSW, all further handling of status is under control of the IOST and its governing program. Thus, the channel can carry out other activities.

3.8.2. SELECTIVE RESET Signal

The SELECTIVE RESET signal is the means by which the channel, upon detection of an equipment failure or certain software errors, signals the subsystem to disconnect from the I/O interface. The system immediately terminates any data or status transfers and disconnects completely within 6 microseconds from the receipt of the SELECTIVE RESET signal, with no additional action taken by the channel.

The channel issues the SELECTIVE RESET signal under the following conditions:

- The address or status byte received from a device has incorrect (even) parity.
- The address returned by a device during any initial selection sequence (ISS) had good parity but did not compare to the address transmitted by the channel.
- A signal from a device occurred at the wrong time or was active or inactive for an excessive duration.
- Two inbound control lines on the I/O interface were found active simultaneously when the OPERATIONAL IN signal was active.
- The channel detected any of the program, invalid address, or channel control check conditions during the first ISS associated with an I/O instruction.
- The selector channel detected any of the program, invalid address, or channel control check conditions during the ISS associated with the initiation of a new operation under command chaining.

In all these cases, the appropriate subchannel status (interface control check, channel control check, invalid address, program check) is included in the status word (IOSTIW) presented to the I/O status tabler.

On a control unit-initiated request, if the channel does not receive an OPERATIONAL IN and ADDRESS IN signal, or a device address parity check is detected, the following takes place:

1. The SELECTIVE RESET signal is issued.

2. NO INTERFACE CONTROL CHECK signal is set and a STATUS SERVICE REQUEST signal is not generated.

3.8.3. Selector Channel Device Addresses

The format of the device address assigned to each subsystem (control unit and device) for the selector channel is as follows:

- 1xxx xxxx

  For diagnostic use only. The channel disconnects the I/O interface and connects the control unit simulator. (x bits may be 1 or 0.)

- 0ccc dddd

  Normal format, where:

  ccc
  : Is the binary representation of the address of one of the eight possible control units connected to the I/O interface.

  dddd
  : Is the binary representation of the device number of 1 of 16 devices that may be connected to a shared control unit. A nonshared control unit may use any 1 of the 16 combinations of d's.

The 8-bit device address is used by the processor when it issues an I/O instruction to the selector channel and by the channel and subsystem during the prescribed operation.

3.8.4. Selector Channel Data Transfer Rates

The data transfer rate of the selector channel varies as a function of the response time of the attached control unit, the length of the I/O cable between the channel and control unit, and main storage interference by other channels. Control unit response time is the internal delay between the leading edge (rise) of the SERVICE OUT signal and the following edge (drop) of the SERVICE IN signal.

Main storage interference is interference by other channels and must be within the internal data buffering capability of the channel. Internal data buffering of the selector channel permits it to absorb, given a sufficiently low data rate, a certain number of main storage interfering cycles by other higher priority channels (or by the main storage refresh cycle) per selector channel main storage access.

Selector channel maximum average data rates with main storage interference by higher priority channels are:

Main Storage Level①	Maximum Data Rate (K bytes/sec)③	Required Operating Area (K bytes/sec)②③
1 (highest)	825	833
2	660	833
3	550	556

NOTES:

① Main storage level represents the actual selector channel main storage priority if all the higher priority channels are active. If some channels of higher priority are inactive (or not installed), one level may be subtracted for every inactive channel to obtain the main storage level.

② The required operating level specifies the required I/O interface data rate capability.

The listed maximum data rates for the selector channel are valid for any higher priority channel operating at a data rate not exceeding 825K bytes per second.

Maximum data rates take into account interference (1 percent degradation) by the main storage refresh cycle determined by main storage. Conversely, the absence of a refresh cycle would allow the listed maximum average data rates to increase by approximately 1 percent.

③ The aforementioned rates represent the capabilities of the selector channels. Active channels cause main storage interference to lower priority channels operating concurrently. An insufficient resultant data rate at a lower priority channel may require a reduction in selector channel maximum allowable data rate or a reduction in the degree of channel operations overlap.

3.8.5. I/O Instructions

The software initiates all I/O operations by means of the start I/O (SIO) operation. The execution of the I/O instruction is initiated by the processor to the appropriate channel. After the activation of the instruction to the channel, the processor enters the stall state. The processor remains in this state until it receives the ACKNOWLEDGE signal from the channel, or until the 16-millisecond processor stall timer expires. If the channel or subsystem is either not present in the particular installation or not operational, the condition code in the current PSW is set to 11_2 and an immediate acknowledge is returned to the processor. If the channel or subsystem is not in a state that would allow the initiation of the operation, a condition code of 10_2 and an ACKNOWLEDGE signal are returned to the processor. If the channel and subsystem are in the proper state, the operation is initiated.

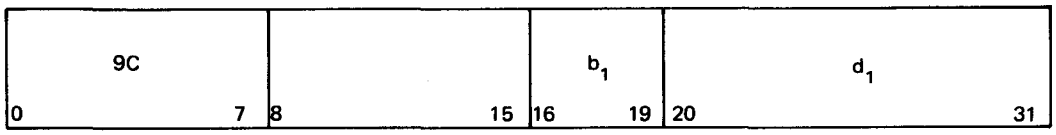
The states of the channel, subchannel, and subsystem are defined in Table 3-9.

Table 3—9. I/O States of Channel, Subchannel, and Subsystem

State	Unit	Description
Available	Channel	The channel is available when it is ready to accept an I/O instruction and to execute the operation prescribed by the instruction.
	Subchannel	In the selector channel, an available subchannel is the same as an available channel.
	Subsystem	A subsystem is considered available if neither the control unit nor the addressed device is executing a previously initiated operation or holding pending status. The channel recognizes this state when the subsystem presents status equal to 0000 0000 during the ISS (first CCW), or if bit 33 of the CCW is 0, and status equals 0000 xx 00. Also, if cc equals 1, status equals 0x00 x100 (x=1 or 0), or status equals 0000 1000 which establishes this state, causing the channel to recognize the available state of a subsystem.
Working (W)	Channel	The selector channel is working when it is operating in burst mode, is in the midst of a chaining sequence, or is in the midst of a control unit-initiated sequence to present status.
	Subchannel	In the selector channel, a working subchannel is the same as a working channel.
	Subsystem	A working subsystem is one where the control unit is executing a previously initiated operation or the control unit is holding status for a nonaddressed device. This state results in a control unit busy sequence during the initial selection sequence.
Status pending (P)	Channel	In the selector channel, the status pending state implies that a device status or channel status signal has generated a status service request signal and all other operations have been halted.
	Subchannel	In the selector channel, the subchannel in the status pending state is the same as the channel in the status pending state.
	Subsystem	A subsystem in the status pending state is one in which the control unit is holding pending status for the addressed device or the addressed device is executing a previously initiated operation. The channel recognizes this state when, during the ISS, the subsystem presents status equal to xxx1 xxxx, where the x bits may be 0 or 1. The channel stacks status at the control unit.
Non-operational (N)	Channel	A channel that is nonoperational is either offline or not present in the particular installation.
	Subchannel	In a selector channel, a nonoperational subchannel is the same as a nonoperational channel.
	Subsystem,	A nonoperational subsystem is one that is offline, powered down, or not installed, or it does not recognize its address during the initial selection sequence.

The SIO instruction is used to initiate all read forward, read backward, write, control, and sense operations. If in the proper state, the specified channel reads the CAW and the first CCW and initiates operation with the device.

The format of the SIO instruction is:



SIO Format for Selector Channel

The contents of the 32-bit register specified by b_1 are added to the contents of the d_1 field to form a 32-bit field with the following format:

$(b_1) + d_1$ (bits 0–20 are ignored.)

The channel specified by bits 21 through 23 and the device specified by bits 24 through 31 are addressed, and the operation proceeds as follows: (Also refer to Table 3-10.)

1. State: AAA

   If the addressed channel, subchannel, and subsystem are in the available state, the condition code is set to 00_2 and the prescribed I/O operation proceeds under the control of the channel and subchannel.

   During the initial selection sequence (ISS), the following takes place on the basis of status received by the channel:

Status	Command Chaining	Status Accepted by Channel	SSR Generated
0000 0000	Immaterial	Yes	No
0000 xx00	No	Yes	Yes
0x00 x100	Yes	Yes	No
000 1000	Yes	Yes	No

   where:

   The x bits can be either 0 or 1.

Table 3—10. Selector Channel SIO Condition Codes

Condition Code	State ④	Command Action	IOSTIW Stored	Applicable Status Field in IOSTIW	Initial Selection Sequence
00	A A A	Accepted	Yes ①	Device	Yes
01	A A P	Rejected	Yes ②	Device	Yes
	P P X	Rejected	Yes ③	Device and/or channel	No
10	W W X	Rejected	No	–	No
	A A W	Rejected	No	–	Attempted (immediate busy)
11	N X X	Rejected	No	–	No
	A A N	Rejected	No	–	Attempted

NOTES:

① Refer to AAA state.

② The COMMAND OUT signal response to a STATUS IN signal is given when the control unit status has the busy bit set. A subsequent control unit request to present status is accepted by the channel, if possible, and a STATUS SERVICE REQUEST signal is generated.

③ A STATUS SERVICE REQUEST signal has been generated as a result of a previous operation. An IOSTIW containing status pertaining to the previous operation is stored. The IOSTIW contains both CHANNEL STATUS and DEVICE STATUS signals if both were stored in the channel and the device address pertains to both. No new status results fron the SIO instruction.

④ A = available
W = working
X = any state
P = status pending
N = nonoperational

2. State: PPX, WWX

   If the addressed selector channel (and subchannel) is in either the working or status pending state, the device is not addressed; the operation is not initiated and the condition code is set to 10_2.

3. State: AAW

   If the control unit is working or contains pending status for a device other than the one addressed, the operation is not initiated and the condition code is set to 10_2 (immediate busy).

4. State: AAP

   If the addressed channel and subchannel are available, but the subsystem is in the status pending state, the condition code is set to 01_2 and status is not accepted at this time. However, the control unit attempts to present status at a later time and if the channel is able to accept it, it will do so and generate a STATUS SERVICE REQUEST signal.

5. State: NXX, AAN

   If the addressed channel or subsystem is nonoperational, the condition code is set to 11_2.

If any errors are detected during the SIO sequence (Table 3-10), the I/O operation with the device is aborted by way of SELECTIVE RESET signal. The condition code is set to 01_2, a STATUS SERVICE REQUEST signal is generated, and an IOSTIW is written with the appropriate channel status bits set.

During an SIO sequence, an invalid address, channel control check, or program check error may occur during the access of the CAW or the first CCW. Errors with the I/O interface are indicated by an interface control check status bit setting.

Table 3-11 gives the error conditions that lead to the rejection of the SIO instruction and the setting of a 01_2 condition code, or that lead to the termination of a command chaining operation.

If the processor attempts to initiate an SIO instruction at the same time the channel receives a REQUEST IN signal on the I/O interface (for the presentation of status), the REQUEST IN signal is handled first and the SIO immediately afterwards. The condition code is set to 10_2.

Table 3—11. Error Conditions Leading to a Rejection of an SIO Instruction for Selector Channel (Part 1 of 2)

Condition	IOSTIW Stored	IOSTIW Status Field	ISS
A. Error on CAW, CCW, Read	Yes	Channel: Invalid address or channel control check or program check	Attempted (1)
B. Channel detected error on I/O interface	Yes (2)	Channel: Interface control check	Attempted (1)
C. Control unit develops error status during ISS	Yes (2)	Device: Status not equal to available subsystem status or status is not set to: xxx1 xxxx where: x bits may be 0 or 1.	Yes

Table 3—11. Error Conditions Leading to a Rejection of an SIO Instruction for Selector Channel (Part 2 of 2)

NOTES:

① Selector channel issues a SELECTIVE RESET signal.

② Error condition C may occur during the same ISS prior to error condition B. Both a CHANNEL STATUS and a DEVICE STATUS signal would be presented to the IOST in the same IOSTIW if the IOST request is granted after the subchannel error condition; otherwise, separate IOSTIW entries are made, with a DEVICE STATUS signal always being entered first.

③ Errors incurred during an SIO sequence cause the channel to return to the processor a condition code set to 01_2, with the exception of the following case:

During the ISS initiated by an SIO instruction, the control unit fails to drop either a STATUS IN signal or an OPERATIONAL IN signal (no control unit forced burst mode). The condition code of 00_2 is returned before the error occurred. The control unit is disconnected by way of the SELECTIVE RESET signal. The interface control check channel status bit is set and a STATUS REQUEST signal is generated.

When the error condition described in the exception case occurs, the 01_2condition code can be returned due to the AAP state (Table 3–16) or presentation of error status (of this table) by the control unit.

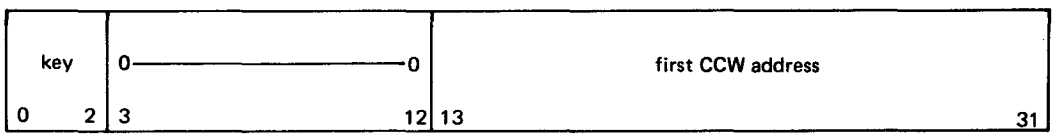
3.8.6. Control of I/O Operations

Control of all selector channel I/O operations is accomplished by means of a set of control words located in main storage (including low-order storage) and in internal channel hardware. Software conveys information to the channels by means of start I/O instructions, channel address words (CAW), and channel command words (CCW). The working copies of the information contained in the control words are maintained in the internal hardware and manipulated by the channel during operations with the I/O subsystems.

3.8.6.1. Channel Address Word (CAW)

The CAW is a 32-bit word located in low-order storage at location $0A0_{16}$. The software is required to load the CAW before each start I/O instruction is issued. The CAW specifies the location of the first CCW and the I/O storage protection key to be used in the execution of the operation initiated by the SIO instruction.

The format and description fo the CAW are:



CAW for Selector Channel

Key (bits 0–2)
A 3-bit I/O storage protection key that the channel uses for all storage accesses for data and CCWs related to the I/O operation initiated by this SIO instruction.

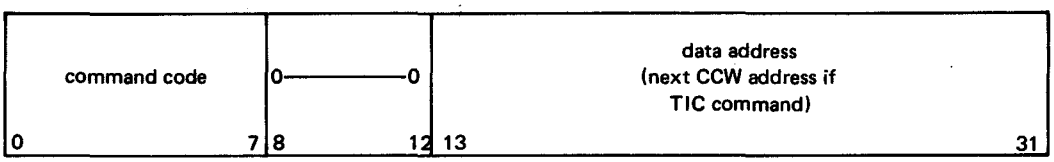
Bits 3–12
Set to 0 by the program.

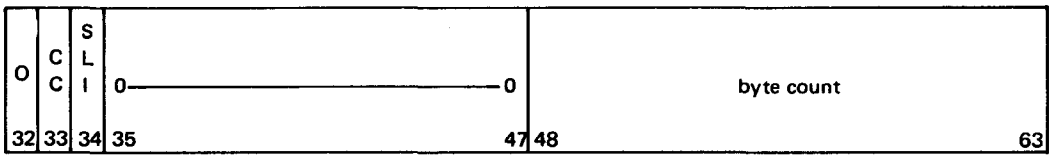
First CCW address (bits 13–31)
Specifies the address of the double word in storage that contains the first CCW used to control the I/O operation being initiated by the SIO instruction. The CAW is used to fetch the first CCW.

3.8.6.2. Channel Command Word (CCW)

The CCW is a 64-bit double word that may be located on any double-word boundary in main storage. The CCW specifies the operation to be performed by the channel or device. The first CCW is accessed during the initiation of an I/O operation by an SIO instruction. The contents of a CCW are transferred to the internal hardware and are used to control the I/O operation until either another CCW is fetched during command chaining or the operation is completed. Fetching of the CCW by the channel does not affect the contents of the location in main storage.

The format and description of the CCW are:





CCW for Selector Channel

Command code (bits 0–7)
Specifies the operation to be performed by the device and channel. The command code is transferred to the device during the initial selection sequence of an SIO instruction or a command chain operation. The command codes and their general interpretation are listed; the M bits are not interpreted by the channel.

Command Code Bits 0 1 2 3 4 5 6 7	Command	Description
MMMM 1 0 0 0	Transfer in channel (TIC)	This command code causes the channel to consider bits 13 through 31 as the address of the next CCW to be fetched. The second word of the CCW is ignored if a TIC is detected in the first word. A program check bit is set in the subchannel status if a TIC is detected in the first CCW or the CCW addressed by a TIC also contains a TIC.
MMMM 0 1 0 0	Sense	The channel interprets the sense code as an input data transfer (a write into main storage with data address incremented). The program must set up the appropriate byte count and data address fields.
MMMM MM0 1	Write	This command code causes the channel to initiate an output data transfer (a read from main storage with the data address incremented).
MMMM MM1 0	Read	This command code causes the channel to initiate an input data transfer (a write into main storage with the data address incremented).
MMMM 1 1 0 0	Read backward	This command code causes the channel to initiate an input data transfer (a write into main storage with the data address decremented).
MMMM MM1 1	Control	This command code is interpreted by the channel as an output data transfer (as in a write command). Control commands that cause the device to present either device-end status or channel-end and device-end status during the ISS are referred to as immediate commands.
MMMM 0 0 0 0	Test	The selector channel does not interpret this command. The channel transmits the command code to the subsystem.

NOTE:

The selector channel transmits the test command to the subsystem, as applies to other commands in the repertoire. (The test command is not in this system's repertoire.) Since a subsystem usually does not generate an END STATUS signal in response to the test command, the selector channel stalls until the interface control check condition is detected and a SELECTIVE RESET signal is issued to the subsystem. As a result, the test command is not recommended for use except by such programs as diagnostic programs which are designed to handle these conditions. This is not true if the device or the control unit has outstanding status.

Bits 8–12, 32 and 35–47
: Unassigned and are set to 0 by the program.

Data address (bits 13–31)
: Contains the address of the location in storage into or from which the first byte of data is transferred. If the command code specifies a TIC, these bits are considered the address of the next CCW. In this case, bits 13–31 must specify a double-word boundary.

CCW flag CC (bit 33)
: This flag (chain command) specifies that, upon receipt of a valid ending DEVICE STATUS signal, a new CCW is fetched and the operation specified by the new command code is initiated. The address of the new CCW is found in the internal channel hardware by incrementing by 8 the previous CCW address.

CCW flag SLI (bit 34)
: If this flag (suppress length indication) is set to 1, an incorrect length condition is not indicated to the program; and if CC is also set to 1, command chaining is not suppressed. An incorrect length condition arises when the subsystem attempts to transfer a fewer or greater number of bytes than specified by the byte count.

NOTE:

If a subsystem contains no means of counting the number of bytes transferred, the channel must terminate the data transfer by a COMMAND OUT signal response to the SERVICE IN signal when the channel has already transferred the number of bytes specified in the byte count. Although this is a normal termination, an incorrect length condition does result and is indicated if the SLI flag has not been set.

Byte count (bits 48–63)
: A 16-bit field that contains the byte count required for all data transfer operations. A byte count of 0 results in no data transfer.

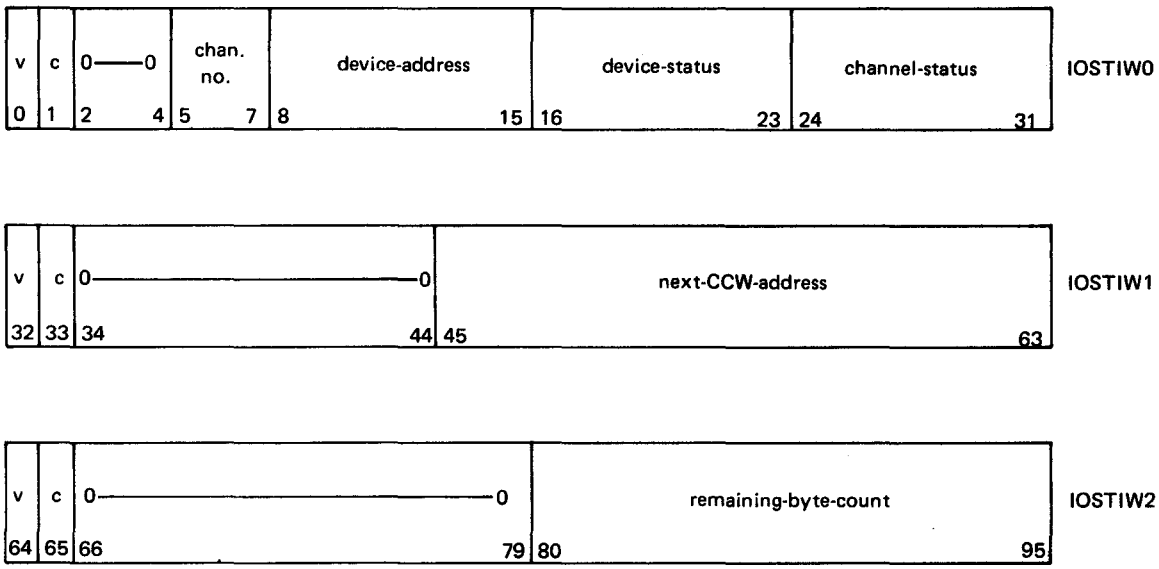
NOTE:

The byte count is transferred to the channel when the CCW is read and becomes the internal byte count. The internal byte count is decremented by 1 after a byte of data is transferred to or from a subsystem. The byte count in the CCW in main storage is not altered.

3.8.7. Selector Channel IOST Interrupt Word (IOSTIW)

The presentation of status information resulting from I/O operations is made by way of the I/O status tabler. Whenever status information is to be communicated to the program, the channel generates a STATUS SERVICE REQUEST signal and writes the appropriate IOST interrupt words (IOSTIW).

The format and description of the IOSTIW are:



IOSTIW for Selector Channel

v (bits 0, 32, 64)
Validation bits. These bits are set to 0 by the selector channel. Software must preset these bits to 1 as each IOSTIW is processed and thus cause the return of the IOSTIW location for use by the IOST with revolving table.

c (bits 1, 33, 65)
Continuation bits:

Bit 1
Always set to 1 by the channel.

Bit 33
Always set to 1 by the channel.

Bit 65
Always set to 0 by the channel.

Bits 2–4, 34–44, and 66–79
Unassigned and are set to 0 by the channel.

Chan. no. (bits 5–7)

Designates the channel number and are set by the selector channel as follows:

For selector channel number 1, set to 100_2 (processor channel number 4).

For selector channel number 2, set to 110_2 (processor channel number 6).

Device address (bits 8–15)

Contains the device address associated with the device or channel status information. If both channel status and device status are presented, the device address applies to both.

Device status (bits 16–23)

Contains one byte of status generated by the subsystem. This field is always 0 if only channel status (bits 24 through 31) is presented. Individual bit definitions are as follows:

Bit	Status
16	Attention
17	Status modifier
18	Control unit end
19	Busy
20	Channel-end
21	Device-end
22	Unit check
23	Unit exception

Channel status (bits 24–31)

Contains a byte of status information generated by the channel. Note that these bits are not mutually exclusive. The presence of one type of error condition may cause the generation of more than one error indication. That is, a valid channel control check may be accompanied by an invalid address.

The software must provide for the possibility of such indications. The channel status field is always 0 if only device status (bits 16–23) is presented. Individual bit definitions are:

Bit	Name	Definition
24	—	This bit is set to 0 by the channel.
25	Incorrect length	If the SLI flag of the current CCW is set to 0, the channel sets the incorrect length indication under the following conditions: ■ On an input or output data transfer, the subsystem attempts to transfer another byte of data after the internal byte count has been decremented to 0. ■ On an input or output data transfer, the subsystem presents ending status before the internal byte count has been decremented to 0.

Bit	Name	Definition
25 (cont)		■ On a control command (assuming at least channel end is returned with ISS status) with all of the following conditions met, the incorrect length indication is set if the internal byte count is not equal to 0 at the time the device-end status presentation has been made. – Command chaining is specified. – Separate channel-end and device-end status presentations are made by the subsystem.
26	Program check	This bit indicates that the channel has detected a software-generated error. The following conditions cause the program check bit to be set: ■ The CCW address field in a CAW or in a CCW specifying a TIC command does not specify a double-word boundary. ■ The CCW addressed by a TIC also contains a TIC. ■ The first CCW in a chain specifies a TIC.
NOTE: Under some conditions, the program check bit may be set if an address check, protection exception, or storage parity check is encountered on a main storage reference.		
27	Invalid address	This bit is set if an addressing exception or a protection exception is encountered on a main storage reference (read or write except on IOSTIW entries).
28	Channel data check	This bit is set if a storage parity check is encountered on main storage data reference (read or write). A bus in parity error or input data sets this bit.
29	Interface control check	This bit indicates that the channel has detected any one of the following conditions at the I/O interface: 1. The address or status byte received from a device has incorrect (even) parity. 2. The address returned by a device during the ISS did not compare to the address transmitted by the channel. 3. During any ISS associated with command chaining, the device appeared nonoperational or returned an immediate busy (busy and status modifier bits) indication.

Bit	Name	Definition
29 (cont)		4. A signal from a device occurred at the wrong time or was active or inactive for an excessive duration. 5. Two inbound control lines on the I/O interface were found active simultaneously when OPERATIONAL IN was active. Interface control check conditions cause the channel to disconnect the device by way of the selective reset, except the conditions listed under 3.
NOTE: On control unit-initiated requests, an address in parity error does not set interface control check, and a STATUS SERVICE REQUEST is not generated. A SELECTIVE RESET is issued.		
30	Channel control check	This bit is set under the following error conditions: ■ Errors during a control reference (CAW/CCW) – Addressing exception – Protection exception – Storage parity error ■ Error during data or control references: address check
31		This bit is unassigned and is set to 0 by the channel.

Next CCW address (bits 45–63)

Contains the value of the next CCW address that was present in the internal channel hardware at the time the status word was written, provided the channel has the first CCW address or next CCW address field then becomes invalid.

Remaining byte count (bits 80–95)

Contains the value of the byte count that was present in the internal channel hardware at the time the status word was written. The internal byte count is decremented by the actual number of bytes transferred on the I/O interface. It should be noted that the byte count does not always reflect the byte count of the subsystem.

NOTE:

All error conditions that led to the setting of channel status generate a STATUS SERVICE REQUEST.

Invalid address, channel control check, or program check errors incurred during the access of a control word (CAW or CCW) result in a SELECTIVE RESET being issued to the device. Interface control check errors also result in a SELECTIVE RESET being generated.

Invalid address, channel control check, or channel data check errors incurred during a data transfer cause the channel to disconnect the device by way of the COMMAND OUT response to SERVICE IN. Entry of the channel status is deferred until device status is received or a stall condition is detected.

An IOSTIW always contains channel status and device status if both are available.

3.8.8. Channel Programs

The selector channel has the capability of controlling a series of I/O operations by means of a channel program. A channel program consists of a set (or chain) of channel command words that are constructed by the software. The main storage address of the first CCW of the chain is designated in the first CCW address field of the CAW. When an SIO instruction is issued, the channel processes the first CCW and, independently of the processor, the remaining CCWs in the chain.

Unless exceptional conditions arise, status service requests are generated only when channel-end or channel-end and device-end status is presented during the operation controlled by the last CCW of the chain. Within the chain of CCWs, a new operation may be initiated, a given CCW may be skipped, and noncontiguous CCWs may be linked by branching. The facilities which allow channel programming include command chaining, TIC commands, and CCW skipping.

3.8.8.1. Command Chaining

Command chaining is specified to the channel by the cc flag (bit 33) in a CCW. The presence of the CCW flag in a CCW is recorded in the channel internal hardware when the operation controlled by the given CCW is initiated. If, at the end of this operation, the device presents channel-end status and no exceptional conditions have occurred, status is accepted from the device, and a status service request will not be generated. The channel-end indication is not preserved in the channel. When the device returns normal device ending status (device-end, or device-end and channel-end), the channel accesses the next CCW in the chain. A status service request is not generated and the device ending status is not preserved. The address of the CCW is stored in the channel internal hardware. The channel then proceeds to reconnect the device by way of the ISS and to initiate the operation indicated in the new CCW.

Refer to 3.8.5.1 and Table 3-11 for a description of error conditions leading to a termination of a command chaining operation.

The following device status presentations lead to command chaining:

- device-end
- channel-end and device-end
- device-end and status modifier
- channel-end, device-end, and status modifier

Status other than the preceding is considered to be abnormal status. The abnormal status is accepted and then a status service request is generated. Unless the current SLI flag of the current CCW is set to 1, an incorrect length condition causes command chaining to be suppressed.

3.8.8.2. CCW Skipping

If the current CCW specifies command chaining and the device returns device status with the device-end status indication or device-end and channel-end status indication accompanied by the status modifier bit, the channel will not access the next CCW in the chain. Rather, the channel increments by 8 the value of the next CCW address field in the internal hardware and accesses the CCW with the resulting address. In this way, the channel program can contain a branch which is dependent upon device conditions.

3.8.8.3. Transfer in Channel (TIC)

If the command code field of a CCW accessed during command chaining specifies a TIC, the data address field of the CCW is used to immediately access the next CCW. In this manner, the software can link noncontiguous chains of CCWs.

If any storage or format errors are detected in the access of the first word of the CCW containing the TIC, the operation is terminated as described. Execution of a TIC to another TIC causes the program check to be set and a status service request generated. If a TIC command is specified in the first CCW, the operation is aborted and the program check bit is set in the IOST interrupt word (IOSTIW).

3.8.9. I/O Interface Stall Timer

Each selector channel is provided with an I/O interface stall timer. The function of the timer is to detect any stall condition in the I/O interface operation due to hardware malfunctions in either the channel or the subsystem. Each stall timer consists of timers to detect two distinctive types of stall conditions. The first is a 50-microsecond timer that detects stalls in the interlocking signal sequences between the channel and the subsystem. That is, if a signal (or absence of a signal) required from the channel (or subsystem) in response to a previous signal from the subsystem (or channel) is delayed more than 50 microseconds, a stall condition is indicated.

The second is a 24-second (nominal) timer, which detects inactive periods of excessive duration. That is if, after making connection with the channel, by way of either an ISS or control unit initiated sequence, a subsystem remains connected and fails to transfer either status or data within 24 seconds, a stall condition is indicated. Upon detection of either type of stall, the channel issues a SELECTIVE RESET signal and sets the interface control check bit in the channel status.

The I/O interface stall timer is inhibited from indicating a stall condition when the processor complex is operating in cycle mode.

3.9. MULTIPLEXER CHANNEL

The multiplexer channel, like the selector channel, communicates with other components of the processor via several relatively independent types of functional interfaces, described in 3.4. The multiplexer channel, external to the processor, controls the exchange of information between the eight peripheral subsystems attached to the I/O interface and the processor and main storage. The multiplexer channel operates in multiplex mode only; that is, the channel services several concurrently operating subsystems by assigning the I/O interface to a subsystem only long enough to transfer one (or a few) bytes of information. It then services other subsystems in a similar manner if necessary, before returning to service the first subsystem again.

The multiplexer channel treats a control unit forced burst mode as if it were a normal multiplex mode transfer of long duration on a single subchannel. The multiplexer channel never forces a control unit into burst mode. It should be noted that when the multiplexer channel is operating in control unit forced burst mode, all other multiplex channel operations are halted. Therefore, requests for service from other subsystems and processor instruction requests to the channel are all ignored while the channel is operating in burst mode.

A channel that does not react to instruction requests for greater than 16 milliseconds may cause a processor-machine-check interrupt because of processor stall. Thus, the consequences of operating the multiplexer channel in control unit forced burst mode should be understood by the user.

The multiplexer channel controls eight subchannels in order to maintain concurrent operations with as many as eight subsystems. The processor initiates all I/O operations by issuing I/O instructions to a selected channel and a selected subsystem connected to the channel. Control is maintained by software-generated control words.

Once an operation is successfully initiated, the channel maintains control of data transfers between main storage and the subsystem independently of the processor. Upon completion of the I/O operation, the state of the channel and subsystem is presented to the software via the appropriate status words by means of the I/O status tabler (3.4.4).

3.9.1. Status Service Request

For the multiplexer channel only, the generation of a status request is again set by the channel. But, when the IOST grants priority with an SSRA, the multiplexer may initiate an IOST sequence within the constraints of the channel sequence priority (3.9.12).

3.9.2. SELECTIVE RESET Signal

The SELECTIVE RESET signal is the means by which the channel, upon detection of an equipment failure, signals the subsystem to disconnect from the I/O interface. Upon such detection, the subsystem immediately terminates any data or status transfers and disconnects completely within 6 microseconds from the receipt of the SELECTIVE RESET with no additional action taken by the channel. The channel issues the SELECTIVE RESET under the following conditions.

- The address or status byte received from a device has incorrect (even) parity.
- The address returned by the device during the initial selection sequence did not compare to the address transmitted by the channel.
- The signal from a device occurred at the wrong time or is inactive for an excessive duration.
- Two inbound control lines on the I/O interface were found active simultaneously when the OPERATIONAL IN signal was active.
- On a start I/O sequence, an error occurred in the writing of the BCW into main storage.

In all these cases, the appropriate subchannel status check, invalid address, channel control check, or interface control check is included in the IOST interrupt word (IOSTIW) presented to the I/O status tabler (IOST).

NOTE:

On a control unit-initiated request, if the channel does not receive OPERATIONAL IN and ADDRESS IN, or a device address parity error is detected, the following takes place:

- *The SELECTIVE RESET signal is issued.*
- *No interface control check is set and a status service request is not generated.*

3.9.3. Multiplexer Channel Device Addresses

The format of the device address assigned to each subsystem configured with the multiplexer channel is:

1. 1xxx xxxx – For diagnostic use only (x bits may be either 1 or 0.)
2. Occc dddd – Normal format

   where:

   Occc
   : Is the representation of the address of one of eight possible control units connected to the I/O interface operating with one of eight subchannels.

   dddd
   : Is the representation of the device number of 1 of 16 devices that may be connected to a shared control unit. A nonshared control unit may use any 1 of the 16 combinations of d.

The 8-bit device address is used by the processor when it issues an I/O instruction to the multiplexer channel and by the channel and subsystem during the prescribed operation. In particular, the multiplexer channel uses the cc field to point to the appropriate BCW in low-order main storage.

3.9.4. Multiplexer Channel Data Transfer Rates

The multiplexer channel data transfer rates are based on the assumption that no main storage interface error occurs on idealized control unit responses. The rates are not attainable with concurrent operation of higher priority channels.

- Multiplex Mode

  The maximum aggregate rate for the multiplexer channel is 83.3K bytes per second. This assumes that the three most significant bits of the data address field in the BCW do not need to be updated. Six main storage accesses are required per byte of data transferred. If the three most significant bits must be updated, two additional channel cycles (600 nanoseconds each) are required. These two cycles include the time of one additional main storage cycle. The instantaneous data rate then becomes 75.7K bytes per second.

- Burst Mode

  The maximum aggregate rate for the channel is 128.2K bytes per second with no significant address field update, and 111.1K bytes per second with the most significant address field update.

3.9.5. Start I/O Instruction

The software initiates all I/O operations by means of the start I/O (SIO) instruction. The execution of the I/O instruction is directed by the processor to the appropriate channel. After the activation of the instruction to the channel, the processor enters the stall state. The processor remains in this state until it receives the ACKNOWLEDGE from the channel or until the 16-millisecond processor stall timer expires. If the channel or subsystem is either not present in the particular installation or not operational, the condition code in the current PSW is set to 11_2 and an immediate ACKNOWLEDGE is returned to the processor. If the channel or subsystem is not in a state that allows the initiation of the operation, a condition code of 10_2 and an ACKNOWLEDGE signal are returned to the processor. If the channel and subsystem are in the proper state, the operation is initiated.

The states of the channel, subchannel, and subsystem are defined in Table 3–12.

Table 3—12. SIO States of Channel, Subchannel, and Subsystem (Part 1 of 2)

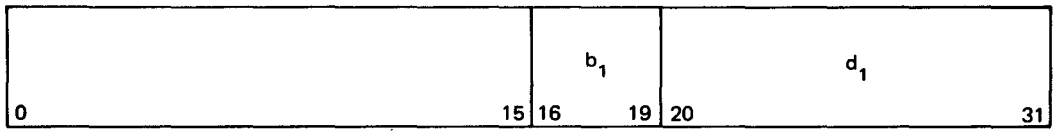
State	Unit	Description
Available (A)	Channel	The channel is available when it is ready to accept an I/O instruction and to execute the operation prescribed by that instruction. This state implies that there is no channel status to present to the IOST. NOTE: The channel is available even if it holds pending device status. Sequences that determine the acceptance of device status from a control unit in the channel depend on the device status pending condition in the channel. If device status is not pending, status is accepted and a status service request is generated. If device status is pending in the channel, status is stacked at the control unit.
	Subchannel	The subchannel is available when the channel is available.
	Subsystem	A subsystem is considered available if neither the control unit nor the addressed device is executing a previously initiated operation or holding pending status. The channel recognizes this state when the subsystem presents status equal to 0000 xx00 (x bits being either 0 or 1) during the ISS.
Working (W)	Channel	Working state does not apply to the channel.
	Subchannel	Working state does not apply to the subchannel.
	Subsystem	The working subsystem is one in which the control unit is executing a previously initiated operation or the control unit is holding status for a non-addressed device. This state results in a control unit busy sequence during the initial selection sequence (ISS).
Status pending (P)	Channel	The channel is in the status pending state when the channel has channel status to present and has generated a status service request.
	Subchannel	The status pending state does not apply to the subchannel.
	Subsystem	A subsystem in the status pending state is one in which the control unit is holding pending status for the addressed device, or the addressed device is executing a previously initiated operation. The channel recognizes this state when, during the ISS, the subsystem presents status equal to xxx1 xxxx (x bits being either 0 or 1). The channel stacks status at the control unit.

Table 3—12. SIO States of Channel, Subchannel, and Subsystem (Part 2 of 2)

State	Unit	Description
Nonoperational (N)	Channel	A nonoperational channel is either offline or not present in the particular installation.
	Subchannel	In the multiplexer channel, a nonoperational subchannel is the same as a nonoperaitonal channel.
	Subsystem	A nonoperational subsystem is one that is offline, powered down, or not installed, or one that does not recognize its address during the ISS.

The SIO instruction is used to initiate all read, read backward, write, control, and sense operations. If in the proper state, the multiplexer channel reads the BCW and initiates the operation with the device.

The format of the SIO instruction is:



SIO Format for Multiplexer Channel

The contents of the 32-bit register specified by b_1 are added to the contents of the d_1 field to form a 32-bit field with the following format:

$(b_1) + d_1$

Bits 0 through 20 are ignored. The channel specified by bits 21 through 23 and the device specified by bits 24 through 31 are addressed, and the operation proceeds as shown in Table 3–13 and the explanation that follows:

1. State: AAA

   If the addressed channel, subchannel, and subsystem are in the available state, the condition code is set to 00_2 and the prescribed I/O operation proceeds under the control of the channel and subchannel.

   During ISS, if status is equal to 0000 yy00, where y is not set to 00, the channel attempts to store the status under the available channel by the generation of a status service request. If status is equal to 0000 0000, the status is presented to the IOST. A status service request is normally generated upon completion of the operation when the control unit presents ending status to the channel.

2. State: PXX

   If the multiplexer channel is in the active status pending state (there is a pending channel status), the device is not addressed; the operation is not initiated and the condition code is set to 10_2.

3. State: AAW

   If the control unit is working or contains pending status for a device other than the one addressed, the operation is not initiated and the condition code is set to 10_2 (immediate busy).

4. State: AAP

   If the addressed channel or subchannel is available but the subsystem is in the status pending state, the condition code is set to 01_2 due to the busy bit contained in the status byte. Status is not accepted at this time. However, the control unit attempts to present status at a later time, and if the channel is able to accept this status, the channel does so and then generates a STATUS SERVICE REQUEST.

5. State: NXX, XXN

   If the addressed channel or subsystem is nonoperational, the condition code is set to 11_2.

If any errors are detected during the SIO sequence (Table 3–14), the I/O operation with the device is not initiated. A status service request is generated and an IOSTIW is written with the appropriate channel status bits set.

If an invalid address of channel control check (or both) condition is detected on a BCW fetch, the device is not addressed. If any of the previously specified errors occur on a BCW write into main storage (the subsystem-returned status is equal to 0000 0000), the operation with the device is aborted by way of a SELECTIVE RESET. An interface control check condition always results in a selective reset. An interface control check is not generated if errors exist only on the write of the BCW into low-order storage.

NOTE:

Errors incurred during an SIO sequence cause the channel to return to the processor a condition code set to 01_2, with one exception. This exception occurs during an ISS initiated by the SIO instruction when the control unit fails to drop either STATUS IN or OPERATIONAL IN (no control unit forced burst mode). The condition code of 00_2 is returned before the error occurred. The control unit is disconnected by way of the SELECTIVE RESET and no BCW update is accomplished. The interface control check channel status bit is set and a STATUS REQUEST is generated.

When the error condition of the exception just described occurs, the 01_2 condition code can be returned due to the AAP state (Table 3—13) or presentation of error status (Table 3—14) by the control unit.

Table 3—14 gives the error conditions that lead to the rejection of the SIO instruction and to the setting of an 01_2 condition code.

Table 3—13. Multiplexer Channel SIO Condition Codes

Condition Code	State ④			Command Action	IOSTIW Stored	Applicable Status Field in IOSTIW	Initial Selection Sequence
	Channel	Subchannel	Subsystem				
00	A	A	A	Accepted	Yes ①	Device	Yes
01	A	A	P	Rejected	Yes ②	Device	Yes
10	P	X	X	Rejected	Yes ③	–	No
	A	A	W	Rejected	No	–	Attempted (immediate busy)
11	N	X	X	Rejected	No	–	No
	X	X	N	Rejected	No	–	Attempted

NOTES:

① Refer to the AAA state.

② The COMMAND OUT response to a STATUS IN is given when the busy bit is set in the control unit status. A subsequent control unit request ot present status is accepted by the channel, if possible, and a status service request is generated.

③ A status service request has been previously generated.

④ A = Available
W = Working
P = Status pending
N = Nonoperational
X = Any state

Table 3—14. Error Conditions Leading to Rejection of SIO Instruction for Multiplexer Channel

IOSTIW Conditions	IOSTIW Stored	IOSTIW Status Field	ISS
A. Error on BCW read	Yes	Channel: Invalid address or channel control check	No
B. Error on BCW write	Yes	Channel: Invalid address or channel control check	Yes ①
C. Channel-detected error on I/O interface	Yes	Channel: Interface control check	Attempted ①
D. Control unit develops error status during ISS	Yes ③	Device: Status ≠ 0000 XX00 or xxx1 xxxx (x bits, 0 or 1)	Yes ②

NOTES:

① Multiplexer channel issues a SELECTIVE RESET.

② If device status is not pending in the channel, status is accepted.

If device status is pending in the channel, status is stacked at the control unit. Status is accepted later by way of a control unit-initiated request.

③ Error condition D may occur during the same ISS prior to error condition C. Both channel status and device status would be presented to the IOST in separate IOSTIWs, with channel status first.

3.9.6. Buffer Control Word (BCW)

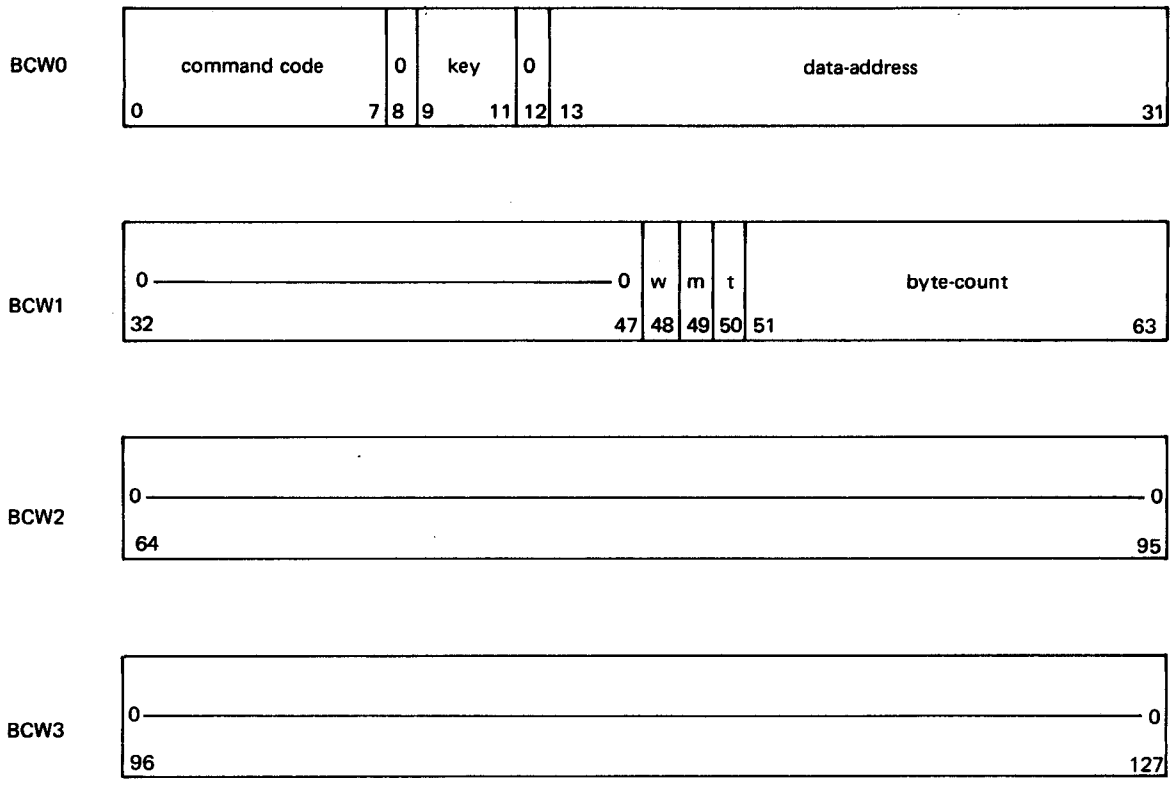
Control of all multiplexer channel I/O operations is accomplished by means of a set of control words located in low-order storage. Software conveys information to the channel by means of start I/O instructions and buffer control words.

The multiplexer channel controls eight subchannels. Each subchannel is controlled by its own BCW. The BCW is a group of four 32-bit words located in low-order storage on a 4-word boundary with the following assignments:

Subchannel No.	Byte Address of BCW (Hexadecimal)
0	200 to 20F
1	210 to 21F
2	220 to 22F
3	230 to 23F
4	240 to 24F
5	250 to 25F
6	260 to 26F
7	270 to 27F

The software loads the BCW for the corresponding subchannel before each start I/O instruction issued. A BCW must not be altered by the software until the present I/O operation is complete (ending status tabled and examined by the software). The multiplexer channel fetches the BCW in order to obtain command, data address, I/O storage protection key, byte count, and control information. Each data request from a subsystem causes the multiplexer channel to access the BCW. The data transfer is controlled by the current contents of the BCW. The multiplexer channel maintains the BCW during the I/O operation by updating and restoring the appropriate parameters for each data transfer sequence.

The format and description of the BCW are:



BCW for Multiplexer Channel

Command code (bits 0–7)

Specifies the operation to be performed by the device and channel. The command code is transferred to the device during the ISS of an SIO instruction. The command codes and their general interpretation are listed. The M bits are not interpreted by the channel.

Command Code Bit 0 1 2 3 4 5 6 7	Command	Description
MMMM 0100	Sense	The channel interprets the sense code as an input data transfer (a write into main storage, with data address incremented). The program must set up the appropriate byte count and data address fields.
MMMM MM01	Write	The write command causes the channel to initiate an ouput data transfer (a read from main storage, with the data address incremented).
MMMM MM10	Read	The read command causes the channel to initiate an input data transfer (a write into main storage, with data address incremented).
MMMM 1100	Read backward	The read-backward command causes the channel to initiate an input data transfer (a write into main storage, with the data address decremented).
MMMM MM11	Control	The command control code is interpreted by the channel as an output data transfer (as in a write command).
MMMM 0000	Test	This command is not interpreted by the multiplexer channel.
MMMM 1000	Transfer in channel (TIC)	The BCW is not modified. The channel transmits the command code to the system.

Bits 8, 12, 32–47, and 64–127

Unassigned and must be set to 0 by the software.

Key (bits 9–11)

A 3-bit field containing the I/O storage protection key. This key is presented to the main storage interface during each data transfer sequence. The multiplexer channel does not modify the key field during data transfer sequences. Addresses generated internally by the multiplexer channel are presented to the main storage interface with a key equal to 0.

Data address (bits 13–31)

A 19-bit field permitting the multiplexer channel to reference any byte in main storage during data transfer sequences. The address is updated by the channel during each data transfer sequence. The initial address can be on a byte boundary.

w (bit 48)
Indicates the following:

w = 0 Input (read) operation
w = 1 Output (write) operation

The proper state of this bit is set by the channel itself on the basis of the command code byte. The program must initialize this bit to 0.

m (bit 49)
Indicates the following:

m = 0 Ascending (forward sequence) address
m = 1 Descending (reverse sequence) address

The proper state of this bit is set by the channel itself on the basis of the command code byte. The program must initialize this bit to 0.

t (bit 50)
Indicates the following:

t = 0 Transfer data
t = 1 Termination of data transfer

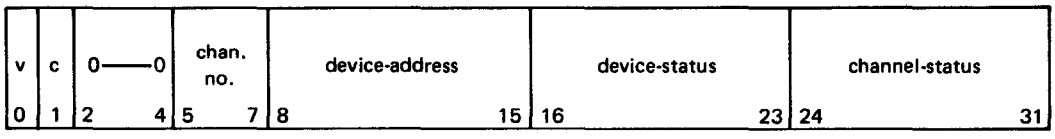
Normal data transfer operations require the t bit to be set to 0 by the software. The channel sets t = 1 only when the transfer of a byte of data causes the byte count to go to 0. The channel does not set t = 0. If no data transfer is to take place, the t bit can be initialized to 1 by the software.

Byte count (bits 51–63)
This 13-bit field contains the byte count required for all data transfer operations. The channel decrements the count during each data transfer sequence. An initial count of 0 provides for a maximum transfer of 8192 bytes if the t bit is set to 0. When the channel decrements the byte count to 0, the t bit is set to 1.

3.9.7. Multiplexer Channel IOST Interrupt Word (IOSTIW)

The presentation of status information resulting from I/O operations is made by way ot the IOST. Whenever status information is to be communicated to the program, the channel generates a status service request and writes the appropriate IOST interrupt word (IOSTIW).

The format and description of the IOSTIW are:



IOSTIW for Multiplexer Channel

v (bit 0)

Validation bit. This bit is set to 0 by the multiplexer channel. Software must preset these bits to 1 as each IOSTIW is processed so that the IOSTIW location is returned for use by the IOSTIW in the revolving table.

c (bit 1)

Continuation bit. This bit is set to 0 by the multiplexer channel.

Bits 2–4

Unassigned and are set to 0 by the channel.

Chan. no. (bits 5–7)

Designates the channel number and are set to 001 by the multiplexer channel.

Device address (bits 8–15)

Contains the device address associated with the device or channel status information.

Device status (bits 16–23)

Contains one byte of status generated by the subsystem. The bits are defined as follows:

Bit No.	Status	Bit No.	Status
16	Attention	21	Device-end
17	Status modifier	22	Unit check
18	Control unit end	23	Unit exception
19	Busy		
20	Channel-end		

Channel status (bits 24–31)

Contains a byte of status information generated by the channel. Note that bits in the channel status are not mutually exclusive. The presence of one type of error condition can cause the generation of more than one error indication, that is, a valid channel control check may be accompanied by an invalid address. Systems software provides for the possibility of such indications. Individual bit definitions are:

Bit	Name	Definition
24–26, 31	–	These bits are set to 0 by the channel.
27	Invalid address	This bit is set if an addressing exception or a protection exception is encountered on a main storage reference (read or write except IOSTIW entries).
28	Channel data check	This bit is set if a storage parity check is encountered on a main storage data reference (read or write). A bus in parity error on input data sets this bit.

Bit	Name	Definition
29	Interface control check	This bit indicates that the channel has detected any one of the following conditions at the I/O interface. ■ The address (ISS only) or a status byte received from a device has incorrect (even) parity. ■ The address returned by a device during the ISS did not compare to the address transmitted by the channel. ■ A signal from a device occurred at the wrong time or was active or inactive for an excessive duration. ■ Two inbound control lines on the I/O interface were found active simultaneously when OPERATIONAL IN was active. Interface control check conditions cause the channel to disconnect the device by way of the SELECTIVE RESET.
30	Channel control check	This bit is set under the following conditions: ■ Errors during a control reference (BCW): – Addressing exception – Protection exception – Storage parity check ■ Errors during any sequence address check (except IOST sequence).

All error conditions leading to the setting of channel status generate a status service request. If an invalid address or a channel control check is detected on a BCW fetch during an SIO sequence, the device is not addressed. If any of the aforementioned errors occur on a BCW write into main storage during an ISS, the operation with the device is aborted by way of SELECTIVE RESET. An interface control check is not generated if errors exist only on the write of the BCW into low-order storage.

Invalid address, channel control check, or channel data check errors incurred during a data transfer sequence cause the channel to disconnect the device by way of the COMMAND OUT response to SERVICE IN.

In the multiplexer channel, a new storage request can be made before an error condition from an earlier request is received and processed by the channel. The first main storage error encountered causes the channel to inhibit subsequent errors that may occur during the sequence.

Channel status and device status are not merged; separate IOST service sequences are initiated for each status. The channel status has higher priority and is presented first. When an IOSTIW contains channel status, the following facts must be considered:

- Device address field

  The dddd bits are set to 0 during channel status presentation. Only the ccc subchannel address field is valid (3.9.3).

- Device status field

  All bits are set to 0.

When an IOSTIW contains device status, all bits in the channel status field are set to 0. The entire device address field is written.

3.9.8. Channel Programming

The multiplexer channel has the capability of controlling the I/O operations of up to eight separate subsystems. Each subsystem is controlled by its own BCW assigned by a subchannel number (ccc field specified in 3.9.3).

The following steps are required to initiate an I/O operation with a given subchannel:

1. Establish that all I/O activity with the given subchannel has been completed.
2. Load the appropriate BCW with the required information.
3. Issue the start I/O instruction.

3.9.9. I/O Interface Stall Timer

The multiplexer channel is provided with an I/O interface stall timer. Its function is to detect any stall conditions in the I/O interface operation due to hardware malfunctions in either the channel or the subchannel. Each stall timer consists of timers to detect two distinctive types of stall conditions. The first is a 50-microsecond timer that detects a stall in the interlocking signal sequences between the channel and the subsystem. That is, if a signal (or absence of a signal) required from the channel (or subsystem) in response to a previous signal from the subsystem (or channel) is delayed more than 50 microseconds, a stall condition is indicated.

The second is a 24-second timer that detects inactive periods of excessive duration. That is, if after making connection with the channel (by way of either an ISS or control unit-initiated sequence) a subsystem remains connected and fails to transfer either status or data within 24 seconds, a stall condition is indicated. Upon detection of either type of stall, the channel issues a SELECTIVE RESET and sets the interface control check bit in the channel status field.

3.9.10. Polling

The multiplexer channel has the capability to poll the I/O interface. When the channel has been inactive (not executing any sequence) for at least 6 microseconds, SELECT OUT is activated by the channel. If no subsystem requires service, SELECT IN is returned to the channel. If the channel is still inactive 6 microseconds from the receipt of SELECT IN, SELECT OUT is again activated. This operation permits the channel to service subsystems that are patched to inhibit the generation of REQUEST IN. In effect, this establishes another control on subsystem priority. Subsystems that have a high data transfer rate but can wait for service may be patched to inhibit REQUEST IN; therefore, they must wait for channel poll. Thus, such subsystems would be prevented from causing channel overruns (late data) at other subsystems.

3.9.11. Channel Status Registers

The multiplexer channel contains three status registers, two dedicated to subchannel status and the third dedicated to device status.

Each subchannel status register has an associated subchannel address register (three bits). Normally, a channel error condition is stored in one of the two subchannel status registers and a status service request is generated. The channel then gives the busy response (condition code 10_2 to any start-I/O instruction. If a new error condition occurs before the first register has been relieved of its status, the second register is used to hold the new status. Start I/O instructions continue to be rejected with condition code 10_2. In addition, control unit requests for service are not accepted as long as both subchannel status registers contain pending status. If this condition lasts too long, secondary errors, such as data overrun, may occur at the device. The relief of status is transparent to the software and is handled by the IOST.

The dedicated device status register has an associated device address register with 8 bits. The channel accepts control unit status if the device status register of the channel does not contain pending status. Upon acceptance of the device status, the channel generates a status service request. As long as this status is stored in the channel, subsequent control unit-initiated requests to present status are stacked at the control unit.

3.9.12. Channel Sequences and Priorities

Three distinct sequences specify action between the multiplexer channel and attached subsystems. Each sequence is entered at the request of either the processor, channel, or subsystem. The requests are honored on a first come, first served basis whenever a single request is present.

The action taken by the multiplexer channel whenever a multiple number of requests are present is resolved by the following priority scheme:

1. IOST
2. Control unit-initiated
3. Start I/O
4. Polling

An exception to the order of priorities exists. The start I/O sequence is not deferred by more than one control unit-initiated sequence whenever the two sequences are returned concurrently.

3.9.12.1. I/O Status Tabler

This sequence may be initiated when the IOST has granted priority to the multiplexer by way of an SSR acknowledge instruction. The channel writes an IOSTIW in a low-order main storage buffer and notifies the IOST of a successful (or unsuccessful) entry. Interlocks exist to ensure that subsequent requests do not destroy status previously stored in the BCSW until serviced by the IOST. Beyond this time, all further handling of status in the BCSW is under control of the IOST and its governing program, and the channel becomes free to carry out other activities.

3.9.12.2. Control Unit Initiated

This sequence is entered by way of a control unit REQUEST IN to transfer data to or from the channel or to attempt to accept status from the subsystem.

3.9.12.3. Start I/O Instruction

All I/O operations are initiated by a start I/O instruction. The SIO sequence results in the connection of the pertinent subsystem to the multiplexer by way of an ISS, and the generation of an appropriate condition code that is returned to the processor.

3.10. INTEGRATED MULTIPLEXER CHANNEL

The integrated multiplexer channel is housed within the processor cabinet, and is not included with the system if the system uses an I/O expansion cabinet. The integrated multiplexer channel operates identically to the external multiplexer channel (3.9) and has the same capabilities. Refer to 1.1.2.5 for additional information on the integrated multiplexer channel, and to 3.9 for a functional description.

4. Main Storage

4.1. MAIN STORAGE/PROCESSOR INTERFACE

Main storage is interfaced to the processor with up to 524K bytes of main storage. No provision for interleaving is provided in main storage.

Figure 4–1 shows the processor-main storage interface signals with the number of lines required to carry the signals between processor and main storage. A brief description of the interface also is provided in the following paragraphs.

Main storage error conditions are described in 2.8.2 through 2.8.2.5.

4.1.1. STORAGE INITIATE Signal

The STORAGE INITIATE signal requires one line and initiates a main storage cycle. This signal must be active from between 20 and 180 nanoseconds with no maximum time interval between cycles. However, STORAGE INITIATE is applied only in time increments of 600 nanoseconds.

4.1.2. Address Word

The address word consists of a total of 23 lines: 2 lines for byte write enable, 18 lines for data address, and 3 lines for address word parity.

4.1.2.1. Byte Write Enable

When any combination of the write enable lines is active, the data that is available on the input data lines is written into the storage half-word location specified by the address lines. When both write enable lines are inactive, a read cycle takes place with output data from storage specified by the address available on the address lines.

4.1.2.2. Data Address

The binary information presented as the address on the data address lines determines the location of the half word to be written into or read from main storage.

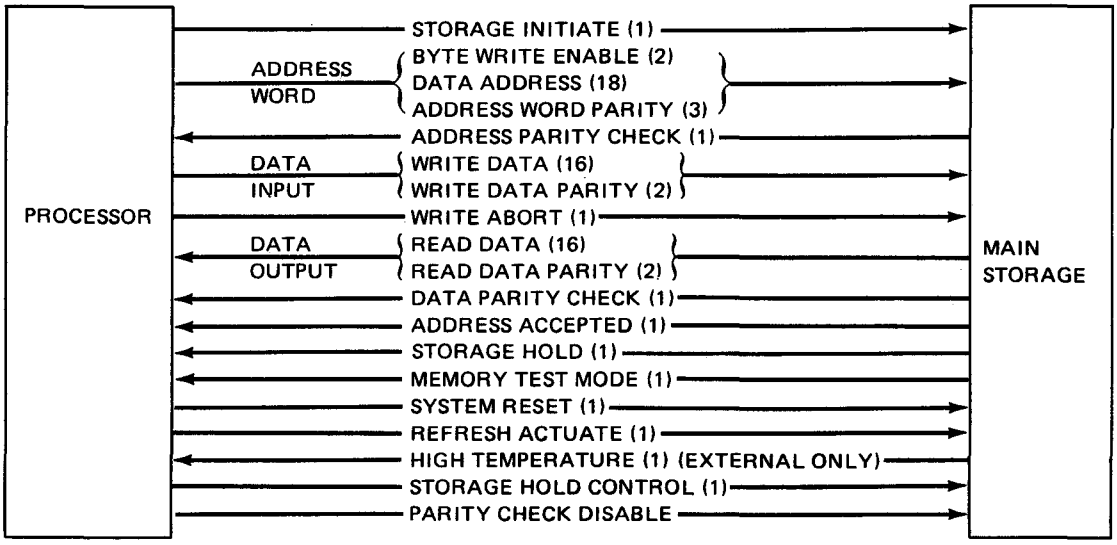
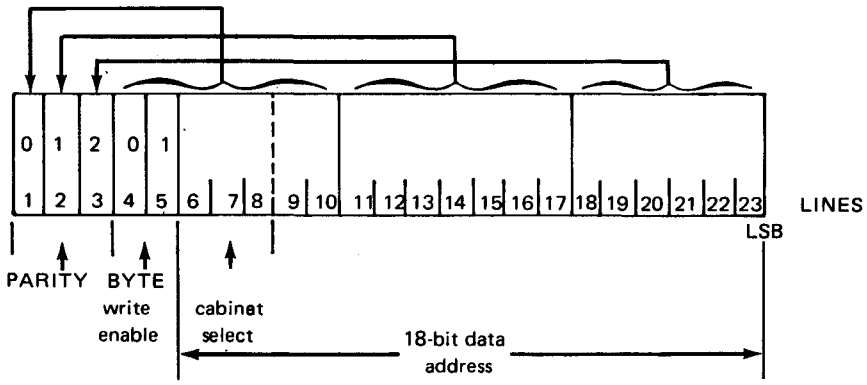


Figure 4—1. Main Storage Interface

4.1.2.3. Address Word Parity

Odd address parity is made available to the storage unit along with the data address. Address parity bit 0 furnishes odd parity for the 2-byte write enable lines and the five most significant address lines, 6 through 10. Parity bit 1 covers address lines 11 through 17; parity bit 2 covers address lines 18 through 23.



4.1.2.4. Address Range Control

Each 64K-byte storage module contains three switches. These switch settings are compared against the most significant address bits (lines 6, 7, and 8). These switches determine the low-order address of the subject storage unit.

4.1.3. Address Parity Check

An address parity check requires one line and, when active, indicates a parity error that causes the following:

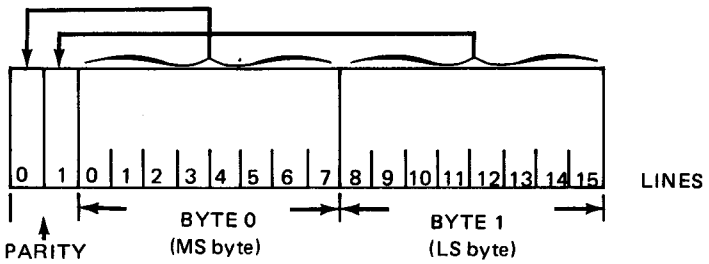
- If in the write mode, main storage aborts the write operation and does not change the address location. If in the parity check disable mode, the write proceeds as if no error ocurred.
- If in the read mode, the read continues as in a normal cycle.
- In all cases of address parity error, except with the parity check disable mode, the ADDRESS PARITY CHECK signal is sent to the system.

4.1.4. Data Input

Data input consists of a total of 18 lines: 16 lines for write data and 2 lines for write data parity.

4.1.4.1. Write Data

These lines are used to input data for the write cycle. When the appropriate write enable lines are activated, the data on these lines is written into storage. The write data format is:



4.1.4.2. Write Data Parity

Odd parity is made available to main storage along with write data. Data format parity bit 0 (line 0) checks odd parity on byte 0, and bit 1 (line 1) checks odd parity on byte 1. In the event of a write data parity error, main storage responds, depending on whether or not the storage unit is in the storage hold mode. When the storage hold control line is activated, detection of a parity error causes the storage unit to immediately halt (preserving the contents of all registers) and initiate the appropriate parity check. Reinitiation of storage operation requires resetting main storage by means of the SYSTEM RESET pushbutton.

Storage response to the storage hold control and parity check control lines is:

Storage Hold	Parity Disable	Storage Response to Parity Error
Not active	Not active	Indicates parity error; abort write operation; do not halt.
Not active	Active	Indicates parity error; do no abort write operation; do not halt.
Active	Not active	Indicates parity error; abort write operation; halt (storage hold).
Active	Active	Indicates parity error; do not abort write operation, halt (storage hold).

4.1.5. WRITE ABORT Signal

The WRITE ABORT signal requires one line and informs the storage unit to abort the write operation with no stored data being affected.

4.1.6. Data Output

Data output consists of a total of 18 lines: 16 lines for read data and 2 lines for read data parity.

4.1.6.1. Read Data

These lines are used to output data for the read cycle. Output data is placed in the output data registers regardless of parity condition. The read data lines for the selected interface are gated into the processor data path by the appropriate address accept line, gating being performed in the processor. The format is the same as shown for write data (4.1.4.1).

4.1.6.2. Read Data Parity

Odd parity is made available from main storage along with read data. Data format parity bit 0 checks odd parity on byte 0, and bit 1 checks odd parity on byte 1. The read data parity lines for the selected interface are gated (along with read data) into the processor data path by the appropriate address accept line.

4.1.7. DATA PARITY CHECK Signal

The DATA PARITY CHECK signal requires one line. This signal, when active, indicates that a parity error exists:

- in at least one of the bytes read from main storage (read cycle); or
- in at least one of the bytes to be placed in main storage (write cycle).

In both cases, parity checking is performed at the main storage module.

4.1.8. ADDRESS ACCEPT Signal

The ADDRESS ACCEPT signal requires one line. This signal, when active, confirms that the address requested is valid, is available in main storage, and is not greater than the storage capacity available. If the address is accepted by both internal and external storage, the processor recognizes this as an erroneous condition. If the address is accepted by more than one 64K-byte module in the external storage, the storage unit will not accept the initiate.

4.1.9. STORAGE HOLD Signal

The STORAGE HOLD signal requires one line and is an asynchronous signal. When active, it indicates that main storage has halted because:

- A parity error was detected in the address and/or data registers. The storage unit halts for this condition when in the storage hold mode, even if in parity check disable mode.
- The storage unit has taken control of the refresh operation. Main storage halts and properly indicates this condition independent of the selection of the storage mode selector. Reinitiation of normal main storage operation requires resetting main storage by means of the SYSTEM RESET pushbutton.

4.1.10. MEMORY TEST MODE Signal

The MEMORY TEST MODE signal requires one line and is an asynchronous signal. When active, it indicates that the internal storage module has been placed into either a test state or one of the following conditions:

- Address interchange
- Off line
- Parity check mode active

4.1.11. SYSTEM RESET Signal

The SYSTEM RESET signal requires one line. When active, it permits master clearing of the storage units of any error condition and leaves them in an online operable state.

4.1.12. REFRESH ACTUATE Signal

The REFRESH ACTUATE signal requires one line. This signal is sent from the processor every 60 microseconds (approximately once every 100 processor cycles) to initiate a main storage refresh cycle.

4.1.13. HIGH TEMPERATURE Signal

The HIGH TEMPERATURE signal is for external main storage, requires one line, and is an asynchronous signal. It is sent from main storage to inform the processor that a high-temperature condition has been sensed in a main storage module (early warning).

4.1.14. STORAGE HOLD CONTROL Signal

The STORAGE HOLD CONTROL signal requires one line and is an asynchronous signal. It is sent from the processor to control the status of the storage hold mode. Storage hold mode occurs when a parity error is detected. It causes the storage unit to immediately halt preserving the contents of all registers.

4.1.15. PARITY CHECK CONTROL Signal

The PARITY CHECK CONTROL signal requires one line and is an asynchronous signal. It is sent from the processor to control the status of main storage module parity checking.

4.2. MAIN STORAGE ADDRESSING

Although the maximum main storage capacity is 524,288 bytes, the addressing hardware of the processor accommodates a 24-bit binary number, thus permitting an addressing capability of 16,777,216 bytes. The low-order 19 bits are used to allow access to 524,288 bytes.

Byte locations are numbered consecutively beginning with 0; each number is considered to be the address of the byte stored in that location. Address fields in the processor and I/O reference an individual byte. However, the least significant address bit is not sent from the processor of I/O to main storage. Therefore, a 2-byte group (half word) in main storage is referenced by the address of the most significant byte location. On processors having serial numbers up to 745/J325, an 18-bit address capability (19 bits less the least significant bit) is provided in the processor and I/O to main storage interface to allow for future expansion to a maximum storage capacity of 524,288 bytes.

On a read cycle, main storage presents two bytes to the processor or I/O channel. If the particular reference requires byte address ability, the processor or I/O channel chooses the appropriate byte, based upon the least significant bit of its own address field. Similarly on a write cycle, two bytes are written unless the partial write capability of main storage is provided.

The processor provides for main storage addressing to wrap around from the maximum 24-bit address 16,777,215 to address 0. However, remember that the maximum byte address that can access main storage is 524,287. An addressing exception error is detected and indicated by main storage whenever the address exceeds the capacity of main storage or by the processor when the address exceeds the maximum addressing capacity of 524,288 bytes.

Processors having serial numbers below 746/J326 have each 65K-byte bank of storage equipped with three switches that are used to assign the storage bank starting address. A 65K-byte bank can be assigned to one of eight different starting addresses, each on a 65,536-byte boundary. Starting address assignments for a configuration's storage bank should be sequential beginning with address 0. No hardware verification of this assignment is made and the system software assumes responsibility of main storage management if addressing gaps are created. If it is necessary to place a 65K-byte storage bank offline, switches on the 65K bank that are set to higher addresses can be reset to new starting addresses in order to bypass the 65K bytes removed from service. This permits the remaining storage banks to have sequential addresses.

Processors having serial numbers 746/J326 and higher have each 262K-byte bank of storage equipped with one switch that is used to assign the storage bank starting address. A 262-byte bank can be assigned either of two different starting addresses, each on a 262K-byte boundary. Starting address assignments for a configuration's storage bank should be sequential beginning with address 0. If it becomes necessary to place a 262K-byte storage bank offline, the switch on the 262K-byte bank that is set to a higher address can be reset to the new starting address, thus bypassing the module removed from service.

4.3. PARTIAL WRITE CAPABILITY

Partial write capability of main storage enables the processor to specify writing of either one or two bytes on a write cycle. There are two control lines in the main storage interface for this purpose. The processor or I/O channel activates the proper control line or lines at the beginning of a storage write cycle. The unselected byte is unchanged in cases where less than the half word is written.

4.4. MAIN STORAGE DATA BOUNDARIES

Fixed-length fields (half words, words, and double words) must be placed into main storage on appropriate multiple-byte boundaries, depending on the field length. For example, words must be stored on a boundary in which the address of the most significant byte of the word is some multiple of 4. On this basis, the boundary address of these fields must contain 0's in low-order bit positions, as follows:

Field	Binary Address
Byte (8 information bits)	x....xxxx
Half word (2 consecutive bytes)	x....xxx0
Word (4 consecutive bytes)	x....xx00
Double word (8 consecutive bytes)	x....x000

Variable-length fields, such as the operands specified by the SS type instructions, are not restricted to integral boundaries. All processor instructions are restricted to half-word boundaries.

4.5. MAIN STORAGE PRIORITY

Main storage and the main storage interface are shared by the processor (including the maintenance panel functions for altering and displaying the contents of main storage) and input/output. The order of priority for access to main storage is:

Highest priority	Selector channel 1*	Channel no. 4
	Integrated disk adapter*	Channel no. 3
	Selector channel 2*	Channel no. 6
	Unassigned	Channel no. 5
	Multiplexer channel	Channel no. 1
	Integrated peripheral channel	Channel no. 0
Lowest priority	Processor or IOST	Channel no. 7

4.6. FIXED MAIN STORAGE ASSIGNMENTS

Figure 4–2 shows the organization of the first 1024 (400_{16}) bytes of low-order main storage. The locations in main storage are reserved for special functions, such as the initiation and control of input/output operations, program interrupt execution, and program analysis.

When the IDA and two selector channels are present, selector channel 2 may operate at a reduced data rate.

Byte Address (Hexadecimal)	0 1 2 3	4 5 6 7	8 9 A B	C D E F
00X				
01X	IOSTCW0	IOSTCW1	IOSTCW2	(Reserved)
02X	IOST	Old PSW	IOST	New PSW
03X	Machine Check	Old PSW	Machine Check	New PSW
04X	Program Exception	Old PSW	Program Exception	New PSW
05X	Supervisor Call	Old PSW	Supervisor Call	New PSW
06X	Interval Timer	Old PSW	Interval Timer	New PSW
07X	(Reserved)		(Reserved)	
08X	Monitor	Old PSW	Monitor	New PSW
09X	(Reserved)		(Reserved)	
0AX	CAW	MM LL RR 00	(Reserved)	
0BX	Rel. Reg. 0	Rel. Reg. 1	Rel. Reg. 2	Rel. Reg. 3
0CX	Rel. Reg. 4	Rel. Reg. 5	Rel. Reg. 6	Rel. Reg. 7
0DX	(Reserved)		(Reserved)	
0EX	IOST BCSW		IOST BSCW	
0FX	Int. Disk BCW0	Int. Disk BCW1	Int. Disk BCW2	Int. Disk BCW3
10X	Console BCW0	Console BCW1	Console BCW2	Console BCW3
11X	Reader BCW0	Reader BCW1	Reader BCW2	Reader BCW3
12X	Printer BCW0	Printer BCW1	Printer BCW2	Printer BCW3
13X	Punch BCW0	Punch BCW1	Punch BCW2	Punch BCW3
14X	CA1 LA0 BCW0	CA1 LA0 BCW1	CA1 LA0 BCW2	CA1 LA0 BCW3
15X	CA1 LA6 BCW0	CA1 LA6 BCW1	CA1 LA6 BCW2	CA1 LA6 BCW3
16X	CA1 LA1 BCW0	CA1 LA1 BCW1	CA1 LA1 BCW2	CA1 LA1 BCW3
17X	CA1 LA7 BCW0	CA1 LA7 BCW1	CA1 LA7 BCW2	CA1 LA7 BCW3
18X	CA1 LA2 BCW0	CA1 LA2 BCW1	CA1 LA2 BCW2	CA1 LA2 BCW3
19X	CA1 LA0 BCW0	CA1 LA8 BCW1	CA1 LA8 BCW2	CA1 LA8 BCW3
1AX	CA1 LA3 BCW0	CA1 LA3 BCW1	CA1 LA3 BCW2	CA1 LA3 BCW3
1BX	CA1 LA9 BCW0	CA1 LA9 BCW1	CA1 LA9 BCW2	CA1 LA9 BCW3
1CX	CA1 LA4 BCW0	CA1 LA4 BCW1	CA1 LA4 BCW2	CA1 LA4 BCW3

Figure 4–2. Fixed Main Storage Assignments (Part 1 of 3)

Byte Address (Hexadecimal)	0 1 2 3	4 5 6 7	8 9 A B	C D E F
1DX	CA1 LA10 BCW0	CA1 LA10 BCW1	CA1 LA10 BCW2	CA1 LA10 BCW3
1EX	CA1 LA5 BCW0	CA1 LA5 BCW1	CA1 LA5 BCW2	CA1 LA5 BCW3
1FX	CA1 LA11 BCW0	CA1 LA11 BCW1	CA1 LA11 BCW2	CA1 LA11 BCW3
20X	Mux. Subch. 0 BCW0	Mux. Subch. 0 BCW1	Mux. Subch. 0 BCW2	Mux. Subch. 0 BCW3
21X	Mux. Subch. 1 BCW0	Mux. Subch. 1 BCW1	Mux. Subch. 1 BCW2	Mux. Subch. 1 BCW3
22X	Mux. Subch. 2 BCW0	Mux. Subch. 2 BCW1	Mux. Subch. 2 BCW2	Mux. Subch. 2 BCW3
23X	Mux. Subch. 3 BCW0	Mux. Subch. 3 BCW1	Mux. Subch. 3 BCW2	Mux. Subch. 3 BCW3
24X	Mux. Subch. 4 BCW0	Mux. Subch. 4 BCW1	Mux. Subch. 4 BCW2	Mux. Subch. 4 BCW3
25X	Mux. Subch. 5 BCW0	Mux. Subch. 5 BCW1	Mux. Subch. 5 BCW2	Mux. Subch. 5 BCW3
26X	Mux. Subch. 6 BCW0	Mux. Subch. 6 BCW1	Mux. Subch. 6 BCW2	Mux. Subch. 6 BCW3
27X	Mux. Subch. 7 BCW0	Mux. Subch. 7 BCW1	Mux. Subch. 7 BCW2	Mux. Subch. 7 BCW3
28X	Not Used		Not Used	
29X	Not Used		Not Used	
2AX	Not Used		Not Used	
2BX	Not Used		Not Used	
2CX	Not Used		Not Used	
2DX	Not Used		Not Used	
2EX	Not Used		Not Used	
2FX	Not Used		Not Used	
34X	CA2 LA0 BCW0	CA2 LA0 BCW1	CA2 LA0 BCW2	CA2 LA0 BCW3
35X	CA2 LA6 BCW0	CA2 LA6 BCW1	CA2 LA6 BCW2	CA2 LA6 BCW3
36X	CA2 LA1 BCW0	CA2 LA1 BCW1	CA2 LA1 BCW2	CA2 LA1 BCW3
37X	CA2 LA7 BCW0	CA2 LA7 BCW1	CA2 LA7 BCW2	CA2 LA7 BCW3
38X	CA2 LA2 BCW0	CA2 LA2 BCW1	CA2 LA2 BCW2	CA2 LA2 BCW3
39X	CA2 LA8 BCW0	CA2 LA8 BCW1	CA2 LA8 BCW2	CA2 LA8 BCW3

Figure 4–2. Fixed Main Storage Assignments (Part 2 of 3)

Byte Address (Hexadecimal)	0 1 2 3	4 5 6 7	8 9 A B	C D E F
3AX	CA2 LA3 BCW0	CA2 LA3 BCW1	CA2 LA3 BCW2	CA2 LA3 BCW3
3BX	CA2 LA9 BCW0	CA2 LA9 BCW1	CA2 LA9 BCW2	CA2 LA9 BCW3
3CX	CA2 LA4 BCW0	CA2 LA4 BCW1	CA2 LA4 BCW2	CA2 LA4 BCW3
3DX	CA2 LA10 BCW0	CA2 LA10 BCW1	CA2 LA10 BCW2	CA2 LA10 BCW3
3EX	CA2 LA5 BCW0	CA2 LA5 BCW1	CA2 LA5 BCW2	CA2 LA5 BCW3
3FX	CA2 LA11 BCW0	CA2 LA11 BCW1	CA2 LA11 BCW2	CA2 LA11 BCW3

Figure 4–2. Fixed Main Storage Assignments (Part 3 of 3)

4.7. STORAGE PROTECTION OPERATION

Storage protection is provided as a feature to ensure the integrity of the supervisor and the individual user programs located in main storage. Up to eight programs, including the supervisor, can be protected against adverse interaction. The storage protection capability protects the contents of main storage from unwanted destruction or misuse in both single-program and multiprogramming environments.

Two types of protection, write only or read/write, are provided in blocks of 512, 1024, or 2048 bytes in processors with serial numbers up to 745/J325, or in 2048 bytes with higher serial numbers. Storage protection is implemented for both processor and input/output references.

4.7.1. Storage Key

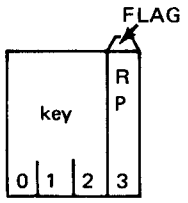
Storage protection is achieved by dividing main storage into blocks of 512, 1024, or 2048 bytes and by associating a 4-bit storage key with each block. Each storage key is stored in a location in key storage, which is a set of hardware registers in the processor. The capacity of main storage determines the number of locations required in key storage in a given configuration.

For main storage configurations up to 131,072 bytes, main storage is divided into blocks of 512 bytes. For example, 64 four-bit storage keys are provided when the main storage is minimum size (32,768 bytes); 256 four-bit storage keys are provided when the main storage capacity is 131,072 bytes.

For main storage configurations greater than 131,072 bytes, up to the maximum size of 262,144 bytes, main storage is divided into blocks of 1024 bytes; for example, 256 four-bit storage keys are provided when main storage capacity is 252,144 bytes.

For main storage configurations greater than 262,144 bytes, main storage is divided into blocks of 2048 bytes; for example, 256 four-bit storage keys are provided when main storage capacity is 524,288 bytes.

Storage keys are held on locations in key storage having the following format and description:



Bits 0–2
A 3-bit code assigning the associated 512-, 1024- or 2048-byte block of main storage to one of eight programs.

RP–Bit 3
Read protect flag. This bit specifies the type of protection as follows:

RP=0
Specifies write protection.

RP=1
Specifies read and write protection.

The supervisor component of the operating system establishes the storage key for 512-, 1024-, or 2048-byte block. This is accomplished by executing a set storage key (SSK) instruction that loads a storage key into a key storage location. The insert storage key (ISK) instruction is used for reading out the contents (key) of a location in key storage. These two instructions are privileged and, therefore, must be executed when the processor is operating in the supervisor mode. Note that there may be many locations in key storage that contain the same storage key because many blocks of main storage have been assigned to the same program. These two instructions (SSK and ISK) are provided with the storage protection feature.

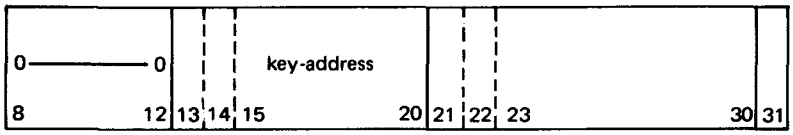
4.7.2. Storage Protection and Relocation Key

The storage protection facility also uses a 3-bit protection key that identifies the program or operation that is requesting access to main storage. Storage references initiated by the processor use the key field (bits 8–11) in the current PSW; accesses by I/O channels are controlled by the protection key in the I/O control word. In addition to its function in storage protection, this key also is used by the processor for accessing the appropriate relocation register in low-order main storage as part of the processor address relocation capability provided in the system. Address relocation, which is independent of main storage protection, applies to processor references only. The protection key, therefore, is called the current PSW storage protection and relocation key or I/O storage protection key.

4.7.3. Key Comparison

When access to main storage is initiated by the processor or I/O, a location in key storage is selected that corresponds to the 512-, 1024-, or 2048-byte block of storage that is accessed. The processor and I/O storage provide the key storage selection as follows:

1. Key address generation from the 24-bit processor storage address:



Bits 8–12
Are an illegal storage address. This field is checked for all 0's by the processor hardware. A nonzero field generates an address exception error and aborts the storage cycle.

Bits 13–30
Is an 18-bit (excluding bit 13) main storage address and is the half-word address on the main-storage-to-processor interface.

Bits 13–20, 14–21 or 15–22

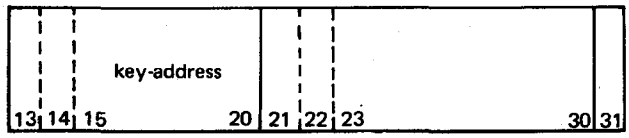
Bits 13–20
Is the key address and selects one of the 256 storage protection keys to provide storage protection in blocks of 2048 bytes when main storage configuration exceeds 262K bytes.

Bits 14–21
Is the key address field and selects one of the 256 storage protection keys to provide storage protection in blocks of 1024 bytes when main storage configurations are between 131K and 262K bytes.

Bits 15–22
Is the same as for bits 14–21 except it provides storage protection in blocks of 512 bytes for main storage configurations up to 131K bytes.

Bit 31
Is the byte select and is carried internally in the processor.

2. Key address generation from the 19-bit I/O storage address:



Bits 13–30
Is an 18-bit main storage address. This is the half-word address on the main-storage-to-I/O interface.

Bits 13–20, 14–21 or 15–22

Bits 13–20
Is the key address and selects one of the 256 storage protection keys to provide storage protection in blocks of 2048 bytes when main storage configuration exceeds 262K bytes.

Bits 14–21
: Is the key address field and selects one of the 256 storage protection keys to provide storage protection in blocks of 1024 bytes when main storage configurations are between 131K and 262K bytes.

Bits 15–22
: Is the same as for bits 14–21 except it provides storage protection in blocks of 512 bytes for main storage configurations up to 131K bytes.

Bit 31
: Is the byte select and is carried internally in the I/O channel.

The key portion (three high-order bits) of the storage key accessed from the key storage location is compared for equality with the 3-bit I/O storage protection key or the current PSW storage protection and relocation key. If the result is either that the two 3-bit keys are equal or that the PSW or I/O key is binary 0, a match condition exists and access to main storage is granted. If a match condition does not exist and the RP flag bit (bit 3) is set to 0, a read access is granted.

Whenever main storage access is not granted because a match condition does not exist, a protection exception occurs. If the reference is for a write operation, the processor signals main storage to abort the write cycle and the original data is preserved. If the reference is for a read operation, main storage executes the read cycle; however, the processor blocks the read data from being used either in the processor or I/O channels.

When a protection exception occurs on a program-specified reference, a program exception interrupt request is generated. An I/O channel takes the same action when protection exception occurs on an access initiated by an I/O channel. When the storage protection feature is not present, protection exception does not occur on the processor or the I/O-specified storage references; all references to main storage are granted, and address relocation is performed if applicable.

4.7.4. Summary of Storage Protection Rules

Writing is allowed if any of the following is true:

- The storage protection and relocation key equals the key portion of the storage key.
- The storage protection and relocation key equals 0.
- The storage protection feature is not installed.

Reading is allowed if any of the following is true:

- The storage protection and relocation key equals the key portion of the storage key.
- The storage protection and relocation key equals 0.
- The RP flag in the storage key equals 0.
- Storage protect feature is not installed.

4.8. ADDRESS RELOCATION

Two methods for modification of program-specified addresses are provided and are independent of each other. However, in the case of operand and branching addresses in instructions, either one or both methods can be applied.

One method, which is compatible to that used on IBM System 360, is an indexing scheme for the modification of program-specified operand and branch addresses in RX, RS, SI, and SS type instructions. In this method, the address is formed by the addition of the operand address displacement value, the contents of the d field in the instruction, and the base (b), which is the contents of the register specified by the b field in the instruction. Addresses specified in RX type instructions can be modified further by the addition of the contents of the index register (x_2). (See 2.4.1.)

The second method, called address relocation, also can be effective on program-specified operand and branch addresses. In addition, it can modify the addresses of instructions and data. It is this relocation method for address modification that is described herein.

4.8.1. Absolute and Relative Addresses

Main storage addresses are specified by the program and the hardware. The value of the address when it references main storage is the number of the absolute location that is being accessed in main storage. Hardware-generated addresses are always in this form. The value of program-generated addresses, as specified in the program, may be different from the value of the corresponding address used to reference main storage. For example, program-specified addresses of instructions and data in object code may appear relative to a starting address of the program. This starting address and, hence, all addresses relative to it may or may not specify the same values as the values of the absolute locations where the program or data is placed in main storage when it is loaded; that is, the addresses may be relocated. There are also special cases where a program address always must specify the absolute location in main storage; for example, addresses specified in all I/O BCWs.

- Relative Address

  An address as it appears prior to being modified by address relocation.

- Absolute address

  - An address not capable of being relocated, that is, address relocation does not apply. For example, hardware-generated addresses, addresses used by the IOST, and all I/O references.

  - A relative address that has been relocated. This includes cases when a relocation value of 0 is added to a relative address resulting in the absolute address having the same value as the relative address.

4.8.2. Address Relocation Characteristics

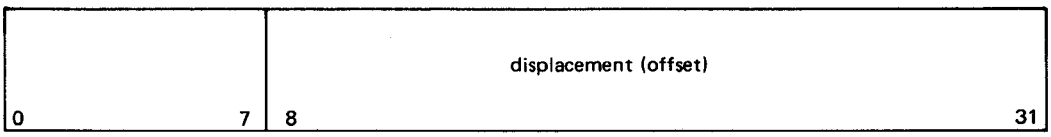
As user programs are loaded into main storage, the OS/3 supervisor loads the program relocation register with the absolute location of the programs.

Address relocation is accomplished by adding a 24-bit relocation displacement value to bits 8 through 31 (most significant bits) of the main storage address that would normally have been used to access main storage if relocation had not been used. The 24-bit displacement value provides a relocation base that is on a byte boundary.

Relocation is controlled by a set of eight relocation registers that are in low-order main storage locations OBO_{16} to OCF_{16}. Each register numbered 0_{16} through 7_{16} is associated with a corresponding storage protection and relocation key. These registers are used by the processor. The storage protection and relocation key (processor only) is used to select the appropriate register. There are no special instructions to store the relocation register in low-order main storage. The processor also retains (in the current relocation register) information from the relocation register that is actively in use by the processor. The current relocation register is loaded in the execution of a load PSW instruction and in an interrupt initialization sequence.

4.8.3. Relocation Register Format

Each relocation register in main storage and the current relocation register in the processor have the following format:



Bits 0–7
: Ignored and are not altered by the hardware.

Offset (bits 8–31)
: The 24-bit relocation value. This value is added to the 24 bits of the resultant address that is to be relocated (2.4.1) to form the absolute main storage address. If the software sets the value of the offset to 0, the relative address equals the absolute address.

4.8.4. Processor Address Relocation

Address relocation is not implemented for I/O channels; addresses are always specified as absolute. Subsequent paragraphs further describe address relocation.

4.8.4.1. Loading the Current Relocation Register

The current relocation register is loaded from main storage with the contents of a relocation register whenever there is a new processor storage protection and relocation key placed in the current PSW. This occurs when the current PSW is changed by the execution of a load PSW instruction or by an interrupt initialization sequence (IIS). The new key is used as a pointer to access the appropriate relocation register in low-order main storage. The contents of the relocation register are loaded into the current relocation register in the processor. The panel system RESET ACTIVE switch causes the processor current relocation register to be cleared.

4.8.4.2. Instruction Address Relocation

The instruction address in the current PSW is converted from relative to absolute. The relocation displacement is added if required, and processing of instructions is initiated and continues with the instruction address in the current PSW maintained in absolute. The instruction address in the current PSW can be changed by the following conditions:

- Execution of a branch instruction where branching occurs. The branch address becomes the new instruction address in the current PSW. When the branch address is loaded into the instruction address field in the current PSW, it is converted from relative to absolute. The relocation offset is added, if required.

- Execution of an LPSW instruction or execution of an interrupt initialization sequence (IIS). In the case of an IIS, the instruction address in the current PSW that is being replaced is stored in the old PSW as a relative address; that is, the displacement in the current relocation register is subtracted from the absolute instruction address. The values in the current relocation register are determined prior to the loading of a new relocation register during the execution of an IIS.

- Cleared state of the current relocation register. When the processor is in the cleared state and the RUN switch is pressed, the contents of the current relocation register are 0.

  The contents of the current PSW, and therefore the processor storage protection and relocation key, are also 0 when the RUN switch is pressed following system reset.

4.8.4.3. Operand Address Relocation

Operand addresses in instructions are converted from relative to absolute, if required, depending upon the particular instruction being executed. Address relocation does not apply to all instruction operand address fields; for example, in the load address, shift, and the I/O instructions, the operand address field is never converted from relative to absolute.

The relocation displacement is added to an instruction operand address at the same time the d + (b) modification takes place. In RX type instructions, the contents of the indexing register (x_2) is added, if required, following the d + (b) offset addition.

Appendix A. IPC BCW Command Codes

This appendix lists command codes used for the integrated peripheral subsystems connected to the processor via the IPC. Additional programming information on these subsystems is available in the 90/30 IPC programmer reference, UP-8041 (current version).

Table A—1. 0717 or 0719 Card Reader Command Codes

Command	Bit Number							
	0	1	2	3	4	5	6	7
Read	a	b	x	d	e	f	1	0
Sense	x	x	x	x	0	1	0	0

LEGEND:

x = bit is ignored by the reader control
d,e,f = modifier bits.
a = diagnostic modifier bit. Normal operation of the reader should not be predicated on the use of this modifier bit.
b = normal read operation; when set to 1, selects read station 2 only.

Modifier Bit	Description
a = 0	Normal read operation
a = 1	Diagnostic use only
b = 0	Normal read operation
b = 1	Select read station 2 only; inhibit compare error (SB1,2)
d = x and e = 0	80-column read. Normal read mode for standard size 80-column punch cards
d = 0 and e = 1	Short card read (51-column). This mode permits reading of 51-column cards (feature installed).
d = 1 and e = 1	Short card read (66-column). This mode permits reading of 66-column cards (feature installed).
f = 0	Read in translate mode. The extended code card
f = 1	to EBCDIC translator is selected. Read in image mode.

Table A—2. Card Punch Command Codes

Command	Bit Number							
	0	1	2	3	4	5	6	7
Control	a	b	x	x	e	f	r	P
Sense	x	x	x	x	0	1	0	0

LEGEND:

x = bit is ignored by punch control
a,b,f,r,p = modifier bits.
e = diagnostic modifier bit. Normal operation of card punch should not be predicated on the use of this modifier bit. This bit is ignored if p is set to 0.

Modifier Bit	Description
a = 0	Normal punch/read operation
a = 1	Diagnostic use only
b = 0	Stop on error.
b = 1	Short errors and remain in run state.
e = 0	Normal punch operation
e = 1	Select reject stacker, terminate data transfers; card reject based on punch data.
f = 0	Punch or read in translate mode. The extended card code/EBCDIC translator is selected.
f = 1	Punch or read in image mode.
r = 0 and p = 0	Invalid code; results in a command reject
r = 1 and p = 0	A read operation is specified with no punch operation. This code is invalid if the read station feature is not installed and results in a command reject.
r = 0 and r = 1	A punch operation is specified with no read operation.
r = 1 and p = 1	A punch and read operation is specified. This code is invalid if the read station feature is not installed and results in a command reject.

Table A—3. 0773 or 0778 Printer Command Codes

Command	Bit Number							
	0	1	2	3	4	5	6	7
Load vertical format buffer	0	1	1	0	0	0	1	1
Load code buffer	1	1	1	1	1	0	1	1
Print advance	a	c	d	e	f	0	0	1
Advance	a	c	d	e	f	1	1	1
Load print line buffer*	1	1	1	0	0	0	1	1
Read print line buffer*	x	x	x	0	0	0	1	0
Read load code buffer*	x	x	x	0	1	0	1	0
Read vertical format buffer*	at	x	x	1	0	0	1	0
Diagnostic*	x	x	x	x	x	1	0	1
Sense	x	x	x	x	0	1	0	0

LEGEND:

x = bit is ignored by printer control.
a,c,d,e,f = modifier/detail bits.

*These commands are for diagnostic use only. Normal operation of the printer should not be predicated on the use of these commands.

The a,c,d,e, and f detail bits of the print-advance command specify two modes of paper advance:

Mode 1:

- If a=0, the c,d,e, and f bits specify a line advance from 0 to 15 lines.

  The eight codes assigned to home paper/end of form, form overflow, five unique skip codes that may be used at various times perform, and the filler code are ignored.

- If a=1, the d,e, and f bits are interpreted as skip codes that are compared with the stop codes in the vertical format buffer. The c bit is ignored in this case.

Mode 2:

- If a=0, the read vertical format buffer command initiates the transfer of the contents of the vertical format buffer to main storage. The buffer is unloaded sequentially beginning at the home paper location. Unloading continues until 144 bytes (192 bytes if the character set control feature is installed) have been transferred or the IPC indicates the end of data transfers (early termination). Bits 0–4 of each byte transferred are set to 0.

- If a=1, the vertical format buffer command initiates the transfer of the contents of the print line buffer followed by the contents of the vertical format buffer.

Table A—4. 0773 or 0778 Printer Command Detail Bit Interpretation

a=0 Advance Line(s)	C	D	E	F	a=1 Interpretation
0	0	0	0	0	Filler code*
1	0	0	0	1	Form overflow
2	0	0	1	0	
3	0	0	1	1	Program selectable
4	0	1	0	0	skip codes
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	Home paper/end of forms
8	1	0	0	0	Filler code*
9	1	0	0	1	form overflow
10	1	0	1	0	
11	1	0	1	1	Program selectable
12	1	1	0	0	skip codes
13	1	1	0	1	
14	1	1	1	0	
15	1	1	1	1	Home paper/end of forms

*This code normally should not be specified in the command. If a=1 and d,e,f = 0,0,0, paper then advances to a position corresponding to the next position in the vertical format buffer (VFB) containing the filler code.

Table A—5. System Console Command Codes

Command	Bit Number							
	0	1	2	3	4	5	6	7
Read	a	x	x	x	x	f	1	0
Write	a	b	c	x	x	f	0	1
Sense	x	x	x	x	0	1	0	0

LEGEND:

x = bit is ignored by console control.
a,b,c,f = modifier bits.

Modifier Bit	Description
a = 0	Normal read operation
a = 1	Diagnostic use only
f = 0	Read in translate mode. The ASCII to EBCDIC translator is enabled; all data transferred to main storage is in EBCDIC.
f = 1	Read in ASCII mode. The ASCII to EBCDIC is bypassed; all data transferred to main storage is in ASCII.

Table A—6. System Console Write Command Detail Bit Interpretation

Detail Bit	Description
a = 0	Normal write operation
a = 1	Diagnostic use only
b = 0	Lock keyboard at completion of the write sequence. (Only MESSAGE WAITING key is available to the operator.)
b = 1	Unlock keyboard at completion of write sequence (operator data entry permitted).
c = 0	Data transfer is to system console only. No print sequence is to be initiated.
c = 1	Data transfer is to the system console and console printer.
f = 0	Write in translate mode. The EBCDIC to ASCII translator is enabled; all EBCDIC data from main storage is translated to ASCII required by the system console.
f = 1	Write in ASCII mode. The EBCDIC to ASCII translator is bypassed; all data from processor main storage must be ASCII, as is required for the system console.

Table A—7. Communications Adapter Command Codes (Part 1 of 3)

Command	Hex.	Bits							
		0	1	2	3	4	5	6	7
No-op	00	0	0	0	0	0	0	0	0
Enable data output	01	0	0	0	0	0	0	0	1
	or								
	41	0	1	0	0	0	0	0	1
Enable data output with automatic turnaround	81	1	0	0	0	0	0	0	1
	or								
	C1	1	1	0	0	0	0	0	1
Dial	05	0	0	0	0	0	1	0	1
Send space	11	0	0	0	1	0	0	0	1
Send mark	0D	0	0	0	0	1	1	0	1
Send idle	09	0	0	0	0	1	0	0	1
Enable data input	02	0	0	0	0	0	0	1	0
New sync	0A	0	0	0	0	1	0	1	0
Look for sync	06	0	0	0	0	0	1	1	0
Turn off	03	0	0	0	0	0	0	1	1
Clear active	0E	0	0	0	0	1	1	1	0
Disconnect	13	0	0	0	1	0	0	1	1
Sense	04	0	0	0	0	0	1	0	0
Set busy	1F	0	0	0	1	1	1	1	1

Table A—7. Communications Adapter Command Codes (Part 2 of 3)

Command	Hex.	Bits							
		0	1	2	3	4	5	6	7
LA clear	0F	0	0	0	0	1	1	1	1
Enable data set ready monitor	17	0	0	0	1	0	1	1	1
Set full duplex	1B	0	0	0	1	1	0	1	1
Load control bytes:									
Byte 1, 2, 3, 4	15	0	0	0	1	0	1	0	1
Byte 2, 3, 4	55	0	1	0	1	0	1	0	1
Byte 3,4	95	1	0	0	1	0	1	0	1
Byte 4	D5	1	1	0	1	0	1	0	1
Load control character detect table:									
Table 1	19	0	0	0	1	1	0	0	1
Table 2	59	0	1	0	1	1	0	0	1
Table 3	99	1	0	0	1	1	0	0	1
Table 4	D9	1	1	0	1	1	0	0	1
Load control interpretation table:									
Table 1	1D	0	0	0	1	1	1	0	1
Table 2	5D	0	0	0	1	1	1	0	1
Table 3	9D	1	0	0	1	1	1	0	1
Table 4	DD	1	1	0	1	1	1	0	1
LA test (diagnostic)	0B	0	0	0	0	1	0	1	1
Modem text (diagnostic)	07	0	0	0	0	0	1	1	1
Read port control word (diagnostic)	16	0	0	0	1	0	1	0	0
Read control character detect table:									
Table 1	1A	0	0	0	1	1	0	1	0
Table 2	5A	0	1	0	1	1	0	1	0
Table 3	9A	1	0	0	1	1	0	1	0
Table 4	DA	1	1	0	1	1	0	1	0

Table A—7. Communications Adapter Command Codes (Part 3 of 3)

Command	Hex.	Bits							
		0	1	2	3	4	5	6	7
Read control interpretation table:									
Table 1	1E	0	0	0	1	1	1	1	0
Table 2	5E	0	1	0	1	1	1	1	0
Table 3	9E	1	0	0	1	1	1	1	0
Table 4	DE	1	1	1	0	1	1	1	0

Table A—8. Line Adapter Device Addresses

LA Number	Device or Port Address (hexadecimal)
LA–0	4
LA–1	6
LA–2	8
LA–3	A
LA–4	C
LA–5	E
LA–6	5
LA–7	7
LA–8	9
LA–9	B
LA–10	D
LA–11	F

Table A—9. 8413 Diskette Command Codes

Command Name	Mnemonic	Code (hexadecimal)	Transfer Mode	Bytes Transferred (maximum)
Load track and sector	LTS	21/31	Byte	2
Read index	RI	12	Word	128
Read index deleted	RID	52	Word	128
Write index	WI	11	Word	128
Write index deleted	WID	51	Word	128
Read	R	02	Byte and word	–
Read deleted	RD	41	Byte and word	–
Write	W	42	Byte and word	–
Wirte deleted	WD	41	Byte and word	–
Diagnostic read	DR	82-F2*	Word	–
Diagnostic write	DW	81-F1*	Word	–
Sense	Sense	X4**	Byte	6

*Any hexadecimal value within the stated ranges will execute the command.

**Any hexadecimal value may be used for the first digit; the second value ($_{16}$) is the significant value for the sense command.

Appendix B. Instructions

The SPERRY UNIVAC 90/30 System is provided with a total of 148 instructions, consisting of:

- 84 basic instructions, including
  - 73 nonprivileged instructions
  - 11 privileged instructions
- 64 additional instructions that include 44 floating-point instructions
- 2 storage protection instructions (optional, part of basic)

Tables B–1 and B–2 list the basic and extended instruction timing, alphabetically according to instruction name, with the execution time required for each instruction.

Tables B–3 and B–4 provide instruction execution timing and legend assumptions to aid in determining exact timing requirements.

Table B–5 lists the instructions according to the hexadecimal opcode along with instruction type and mnemonic code.

Table B–6 lists the SPERRY UNIVAC 9200/9300 and IBM 360/20 compatibility instructions according to hexadecimal opcode.

Table B–7 lists the operation exception instructions for the compatibility modes.

Table B—1. Instruction Timing (Part 1 of 3)

Instruction	Mnemonic Code	Operation Code	Type	Time in Microseconds*
Add	AR	1A	RR	3.0
Add	AR (360/20)	1A	RR	3.6
Add	A	5A	RX	5.4
Add decimal	AP	FA	SS	36.6 + 0.75n1 + 0.375n2 + 6.0t1 + 3.0s5
Add half word	AH	4H	RX	5.4
Add half word	AH (9300 only)	AA	RX	5.4
Add immediate	AI	9A	SI	6.0
Add immediate	AI (9300 only)	A6	SI	6.0
AND	NR	14	RR	3.0
AND	N	54	RX	5.4
AND	NI	94	SI	6.0
AND	NC	D4	SS	10.2 + 1.5n**
Branch and link	BALR	05	RR	3.6 + 0.6s
Branch and link	BAL	45	RX	6.0
Branch and store	BASR	0D	RR	3.0 + 0.6s
Branch and store	BAS	4D	RX	5.4
Branch on condition	BCR	07	RR	3.0
Branch on condition	BC	47	RX	3.6
Branch on count	BCTR	06	RR	3.6
Branch on count	BCT	46	RX	4.2
Compare	CR	19	RR	3.0
Compare	C	59	RX	5.4
Compare decimal	CP	F9	SS	31.8 + 0.375n1 + 0.375n2 + 2.4s6
Compare half word	CH	49	RX	5.4
Compare logical	CLR	15	RR	3.0
Compare logical	CL	55	RX	5.4
Compare logical	CLI	95	SI	4.8
Compare logical	CLC	D5	SS	9.6 + 1.2b
Convert to binary	CVB	4F	RX	36.0
Convert to decimal	CVD	4E	RX	66.6 + 6.0s4
Divide	D	5D	RX	68.4 + 1.2s1 + 0.6rn
Divide decimal	DP	FD	SS	37.8 + 0.75n1 + 6.375n2 + 24.6(n1 - n2)
Edit	ED	DE	SS	9.0 + 3.0n + 0.6n3 + 3.0n4 + 0.6n6
Exclusive OR	XR	17	RR	3.0
Exclusive OR	X	57	RX	5.4
Exclusive OR	XI	97	SI	6.0
Exclusive OR	XC	D7	SS	10.2 + 1.5n
Execute	EX	44	RX	3.6 + 0.6r + 0.6nrr + e
Halt and proceed (privileged)	HPR	99	SI	3.6
Insert character	IC	43	RX	4.2
Insert storage key (privileged)	ISK***	09	RR	4.2
Load	LR	18	RR	3.0
Load	L	58	RX	4.8
Load address	LA	41	RX	4.2
Load and test	LTR	12	RR	3.0
Load control storage (privileged)	LCS	B1	RS	5.4 + 24.0w + 4.8s8
Load half word	LH	48	RX	5.4
Load multiple	LM	98	RS	3.0 + 1.8gr
Load PSW (privileged)	LPSW	82	SI	11.4
Move	MVC	D2	SS	7.8 + 0.6n + 0.6t4 (n - 1)
Move	MVI	92	SI	4.8
Move numerics	MVN	D1	SS	10.2 + 2.1n
Move with offset	MVO	F1	SS	10.2 + 1.2n1 + 1.2n2
Move zones	MVZ	D3	SS	10.2 + 2.1n

Table B—1. Instruction Timing (Part 2 of 3)

Instruction	Mnemonic Code	Operation Code	Type	Time in Microseconds*
Multiply	M	5C	RX	49.2 + 0.6s1 + 0.6s2 + 0.6rn
Multiply decimal	MP	FC	SS	36.0 + 0.75n1 + 14.4(n1 - n2) + 0.375n2
OR	OR	16	RR	3.0
OR	O	56	RX	5.4
OR	OI	96	SI	6.0
OR	OC	D6	SS	10.2 + 1.5n
Pack	PACK	F2	SS	12.0 + 1.2(n1 - 1) + 1.2(n2 - 1)
Service timer register (privileged)	STR	03	RR	6.0 + 0.6t3
Set program mask	SPM	04	RR	3.0
Set storage key (privileged)	SSK***	08	RR	4.2
Set system mask (privileged)	SSM	80	SI	4.8
Shift left single logical	SLL	89	RS	5.4 + 0.6p + 0.6q
Shift right single logical	SRL	88	RS	5.4 + 0.6p + 0.6q
SOFTSCOPE forward scan (privileged)	SSFS	A2	RS	7.2
SOFTSCOPE reverse scan (privileged)	SSRS	A3	RS	7.2
Start I/O (privileged)	SIO	9C	SI	
IDA				10.2 to 12.6
IPC communication				15.6 to 68.0
IPC paper peripheral				15 to 59
Multiplexer				16.8 to 18.6†
Selector				25.2 to 26.4†
Store	ST	50	RX	5.4
Store character	STC	42	RX	4.8
Store half word	STH	40	RX	4.8
Store multiple	STM	90	RS	4.2 + 1.2gr
Subtract	SR	1B	RR	3.0
Subtract	SR (360/20)	1B	RR	3.6
Subtract	S	5B	RX	5.4
Subtract decimal	SP	FB	SS	3.6.6 + 0.75n1 + 0.375n2 + 6t1 + 3.0s6
Subtract half word	SH	4B	RX	5.4
Subtract half word	SH (9300 only)	AB	RX	5.4
Supervisor call	SVC	0A	RR	15.0
Supervisor load multiple (privileged)	SLM	B8	RS	4.2 + 1.8gr
Supervisor store multiple (privileged)	SSTM	B0	RS	4.2 + 1.2gr
Test under mask	TM	91	SI	6.0
Translate	TR	DD	SS	7.2 + 2.4n
Translate and test	TRT	DD	SS	8.4 + 1.8b
Unpack	UNPK	F3	SS	12.0 + 1.2(n1 - 1) + 1.2(n2 - 1)
Zero and add	ZAP	F8	SS	16.2 + 1.8n7 + 1.2n8 + 1.8t2(n2 - n1)
Diagnose (privileged)	DIAG	83	SI	
(I2 = 00)				22.8
(I2 = 01)				42.0
(I2 = 02)				48.0
(I2 = 0E)				12 + y (0.6 + 1.8[z])

Table B—1. Instruction Timing (Part 3 of 3)

Instruction	Mnemonic Code	Operation Code	Type	Time in Microseconds*
Diagnose	DIAG	83	SI	
(nonprivileged)				
I2 = (OF)				13.8 + 3.6d1 + 6.6d2 + 3.6d3
secondary function				+ 11.4d4 - 4.2d5 + 0.6d6
				+ 1.2d7
I2 = (OF)				13.8 + 3.6d1 + 6.6d2 + 3.6d3
primary function				+ 10.2d4 - 4.2d5 + 0.6d8
				+ 3.0d9 + 3.0d10 + 3.6d11
Interrupt PSW swap				
Machine check				14.4
Program exception				15.6
Timer interrupt				14.4
IOST tabling channel status				
PIOST (IPC)				5.4 + 14.4wl + 3.0z1
IOST (IPC)				6.0 + 13.8wl
Other channels				5.4 + 13.8wl

*See Table B–4.
**Five cycles per half word equals 3.0 microseconds per half word, which yields the 1.5n component.
***Storage protect feature.
† Realistic times, not maximum.

Table B—2. Extended Instruction Timing (Part 1 of 2)

Instruction	Mnemonic Code	Operation Code	Type	Time in Microseconds*
Add logical	ALR	1E	RR	3.0
Add logical	AL	5E	RX	5.4
Add normalized (long)	ADR	2A	RR	16.2 + 1.2ce + 1.2pr + 1.2t1 + 1.2rp
Add normalized (long)	AD	6A	RX	19.2 + 1.2ce + 1.2pr + 1.2t1 + 1.2rp
Add normalized (short)	AER	3A	RR	14.4 + 1.2ce + 1.2pr + 1.2t1 + 1.2rp
Add normalized (short)	AE	7A	RX	16.8 + 1.2ce + 1.2pr + 1.2t1 + 1.2rp
Add unnormalized (long)	AWR	2E	RR	16.2 + 1.2ce + 0.6rp
Add unnormalized (long)	AW	6E	RX	19.8 + 1.2ce + 0.6rp
Add unnormalized (short)	AUR	3E	RR	14.4 + 1.2ce + 0.6rp
Add unnormalized (short)	AU	7E	RX	17.4 + 1.2ce + 0.6rp
Branch on index high	BXH	86	RS	7.2 - 1.2s3
Branch on index low or equal	BXLE	87	RS	7.2 - 1.2s3
Compare (long)	CDR	29	RR	18.0 + 1.2ce
Compare (long)	CD	69	RX	22.2 + 1.2ce
Compare (short)	CER	39	RR	15.6 + 1.2ce
Compare (short)	CE	79	RX	18.6 + 1.2ce

Table B—2. Extended Instruction Timing (Part 2 of 2)

Instruction	Mnemonic Code	Operation Code	Type	Time in Microseconds*
Divide	DR	1D	RR	67.8 + 1.2s1
Divide (long)	DDR	2D	RR	205.2 + 0.6p1 + 0.6p2 + 15.0pn + 0.6rn
Divide (long)	DD	6D	RX	208.2 + 0.6p1 + 0.6p2 + 15.0pn + 0.6rn
Divide (short)	DER	3D	RR	45.0 + 0.6p1 + 0.6p2 + 6.6pn + 0.6rn
Divide (short)	DE	7D	RX	47.4 + 0.6p1 + 0.6p2 + 6.6pn + 0.6rn
Edit and mark	EDMK	DF	SS	9.0 + 3.0n + 1.2n3 + 3.0n4 + 1.2n6
Halve (long)	HDR	24	RR	7.8 + 1.2pr + 0.6pn + 0.6(s2)
Halve (short)	HER	34	RR	7.2 + 1.2pr + 0.6pn
Load and test (long)	LTDR	22	RR	4.8
Load and test (short)	LTER	32	RR	4.2
Load complement	LCR	13	RR	3.0
Load complement (long)	LCDR	23	RR	4.8
Load complement (short)	LCER	33	RR	4.2
Load (long)	LDR	28	RR	4.2
Load (long)	LD	68	RX	7.2
Load negative	LNR	11	RR	4.2
Load negative (long)	LNDR	21	RR	4.2
Load negative (short)	LNER	31	RR	3.6
Load positive	LPR	10	RR	4.2
Load positive (long)	LPDR	20	RR	4.2
Load positive (short)	LPER	30	RR	3.6
Load (short)	LER	38	RR	3.6
Load (short)	LE	78	RX	6.0
Multiply	MR	1C	RR	48.6 + 0.6s1 + 0.6s2 + 0.6rn
Multiply half word	MH	4C	RX	28.8 + 0.6s1 + 0.8s2 + 0.6rn
Multiply (long)	MDR	2C	RR	115.2 + 0.6p1 + 0.6p2 + 1.2pn + 0.6rn
Multiply (long)	MD	6C	RX	118.2 + 0.6p1 + 0.6p2 + 1.2pn + 0.6rn
Multiply (short)	MER	3C	RR	48.6 + 0.6p1 + 0.6p2 + 0.6pn + 0.6rn
Multiply (short)	ME	7C	RX	51.0 + 0.6p1 + 0.6p2 + 0.6pn + 0.6rn
Shift left double logical	SLDL	8D	RS	4.8 + 1.2p + 1.2q
Shift left double	SLDA	8F	RS	7.8 + 1.2p + 1.2q
Shift left single	SLA	8B	RS	7.2 + 0.6p + 0.6q
Shift right double logical	SRDL	8C	RS	4.8 + 1.2p + 1.2q
Shift right double	SRDA	8E	RS	6.0 + 1.2p + 1.2q
Shift right single	SRA	8A	RS	5.4 + 0.6p + 0.6q
Store (long)	STD	60	RX	7.2
Store (short)	STE	70	RX	6.0
Subtract logical	SLR	1F	RR	3.0
Subtract logical	SL	5F	RX	5.4
Subtract normalized (long)	SDR	2B	RR	16.2 + 1.2ce + 1.2pr + 1.2t1 + 1.2rp
Subtract normalized (long)	SD	6B	RX	19.2 + 1.2ce + 1.2pr + 1.2t1 + 1.2rp
Subtract normalized (short)	SER	3B	RR	14.4 + 1.2ce + 1.2pr + 1.2t1 + 1.2rp
Subtract normalized (short)	SE	7B	RX	17.4 + 1.2ce + 1.2pr + 1.2t1 + 1.2rp
Subtract unnormalized (long)	SWR	2F	RR	16.2 + 1.2ce + 0.6rp
Subtract unnormalized (long)	SW	6F	RX	19.2 + 1.2ce + 0.6rp
Subtract unnormalized (short)	SUR	3F	RR	14.4 + 1.2ce + 0.6a
Subtract unnormalized (short)	SU	7E	RX	17.4 + 1.2ce + 0.6a
Test and set	TS	93	SI	6.0

*See Table B–4.

Table B—3. Instruction Execution Timing Assumptions

1.	Times are given from start of staticize of one instruction to the start of staticize of the next instruction, unless otherwise specified.
2.	No allowance is included for performance degradation due to the refresh cycles required by the storage units. This typically reduces performance by approximately 1%.
3.	The time to perform single indexing of each operand address is included in the times given. Double indexing on RX instructions is not included. For each double indexing operation, add 0.6μs to the execution time.
4.	Storage relocation is included in the specified times.
5.	Execution times apply to native mode operation. The times also apply to the 9200/9300 and 360/20 compatibility modes unless otherwise specified.
6.	Instructions may start on even or odd half-word boundaries with equal probability.
7.	Operands for MVN, MVZ, NC, OC, ZC are processed a half word at a time, unless the first operand address is greater than the second operand address by 1, in which case the operands are processed a byte at a time. When processed a byte at a time, the execution time, dependent on the length of the operands (i.e., n), doubles.
8.	On decimal multiply instructions, a digit of the multiplier (op1) may be any decimal digit with equal probability. The average multiplication time is used in the equation.
9.	On decimal divide instructions, a given digit of the quotient may be any decimal digit with equal probability. The average time to produce a digit of the quotient is used in the equation.
10.	Operands in main storage for AP, SP, CP, MP, and DP may be located on byte boundaries or half-word boundaries with equal probability.
11.	Decimal add and subtract assume a first operand more than eight bytes long and a positive result.
12.	The execution time given for SSFS is from the start of staticize to the point where the sync condition and the first SOFTSCOPE bus sample (after sync) is stored. Static presence of the sync condition, an initial sweep delay of zero, and a scan rate of zero are assumed. Additional time for completion of the instruction depends on the scan rate of nonzero, and the number of bytes to be stored. When the initial sweep delay is nonzero, this time also contributes to the overall execution time.
13.	The execution time given for SSRS is from the start of staticize to the point where the first SOFTSCOPE bus sample is taken. An initial sweep delay and scan rate of zero are assumed. When the initial sweep delay is nonzero, an additional 0.6μs should be added to the time listed, as well as the specified initial sweep delay value, to determine the time to the first sample. When the initial sweep delay is zero and the scan rate is nonzero, an additional 0.6μs should be added to the time listed, as well as the specified scan rate value, to determine the time to the first sample. The completion of the remainder of the instruction depends on the specified scan rate, the byte count, and the detection of a sync condition on the SOFTSCOPE bus.
14.	SOFTSCOPE time is for the fastest SOFTSCOPE instruction that detects sync and has initial sweep = zero and scan rate = zero; otherwise SOFTSCOPE instruction execution time is dependent on instruction parameters and external device operation.
15.	The execution time listed for the LCS instruction corresponds to the case for cc = 0 (i.e., no sentinel is detected). For the cases when a sentinel is detected (cc = 1,3), an additional 4.8μs should be added to the execution time.
16.	Decimal mutiply timing assumes an average of 12 cycles per digit, which equals 14.4 μs per multiplier byte.
17.	SS operand fetch or store timing components assume 4-byte operands averaged over byte and half-word addressing routines.
18.	Decimal divide timing assumes an average of 20.5 cycles per quotient digit, which equals 24.6 μs per byte of quotient.

Table B—4. Legend for Instruction Execution Timings (Part 1 of 2)

a	A 1, if overflow adjustment is necessary; otherwise, 0.
b	Number of bytes in the first operand which are processed.
ce	Number of digit shifts required to equalize the characteristics.
d1	Number of zero addresses in switch list.
d2	A 1 if initial r odd general register has a nonzero value; otherwise, 0.
d3	A 1 if sentinel is found; otherwise, 0.
d4	Number of task control blocks scrutinized.
d5	Number of linked task control blocks scrutinized.
d6	A 1 when exclusive search is specified; otherwise, 0.
d7	A 1 when a match is found; otherwise, 0.
d8	Number of control blocks with absolute wait bits set.
d9	Number of control blocks with wait bits set and ICOR bit clear.
d10	A 1 if ICOR = 1; otherwise, 0.
d11	A 1 if ICOR = 0 and no wait bits set; otherwise, 0.
e	Execution time of the subject instruction of an execute instruction.
gr	Number of general registers loaded or stored.
n	Number of bytes in the first operand (for instructions with a single field length).
n1	Number of bytes in the first operand.
n2	Number of bytes in the second operand.
n3	Number of field separator characters in the pattern.
n4	Number of digit select or significance starter characters in the pattern.
n6	Number of significant digits encountered when the significance indicator is not set before the digit is examined.
n7	Lowest number of bytes specified by I1 or I2.
n8	Number of bytes I1 exceeds I2; n8 = 0 if I1 $\leqslant$ I2.
nrr	A 1 if subject instruction of an execute instruction is not an RR type; otherwise, 0.
p	Number of 4-place shifts.
p1	Number of digit shifts required to prenormalize op1.
p2	Number of digit shifts required to prenormalize op2.

Table B—4. Legend for Instruction Execution Timings (Part 2 of 2)

pn	A 1 if the result requires postnormalization; otherwise, 0.
pr	Number of digit shifts required to postnormalize result.
q	Number of 1-place shifts.
r	A 1 if r1 $\neq$ 0; otherwise, 0.
rn	A 1 if the result (product or quotient) is negative; otherwise, 0.
rp	A 1 if recomplementing without postnormalization is required; otherwise, 0.
s	A 1 if branch is successful; otherwise, 0.
s1	A 1 if the sign of op1 is negative; otherwise, 0.
s2	A 1 if the sign of op2 is negative; otherwise, 0.
s3	A 1 if the sum of the first and third operand is equal to the comparand; otherwise, 0.
s4	A 1 if the result is greater than one word (eight decimal digits); otherwise, 0.
s5	A 1 if the signs of op1 and op2 are the same; otherwise, 0.
s6	A 1 if the signs of op1 and op2 are different; otherwise, 0.
s8	A 1 if sentinel detected; otherwise, 0.
t1	A 1 if the result is recomplemented; otherwise, 0.
t2	A 1 if n2 $>$ n1; otherwise, 0.
t3	A 1 if timer stored; otherwise, 0;
t4	A 1 if one operand address is even and the other is odd; otherwise, 0.
w	Number of control storage words loaded.
w1	Number of channel status words.
y	A zero for byte count = 0; a 1 for byte count $\neq$ 0.
z	Number of half words in sum.
z1	Priority level (1 through 7) of the device address in the device address list during IOST tabling.

Table B—5. List of Instructions by Opcode (Part 1 of 3)

Opcode Hexadecimal	Instruction Type	Mnemonic	Instruction
03	RR	STR (P)	Service timer register
04	RR	SPM	Set program mask
05	RR	BALR	Branch and link
06	RR	BCTR	Branch on count
07	RR	BCR (C)	Branch on condition
08	RR	SSK (SPF) (P)	Set storage key
09	RR	ISK (SPF) (P)	Insert storage key
0A	RR	SVC	Supervisor call
0D	RR	BASR (C)	Branch and store
10	RR	LPR (F)	Load positive
11	RR	LNR (F)	Load negative
12	RR	LTE	Load and test
13	RR	LCR (F)	Load complement
14	RR	NR	AND
15	RR	CLR	Compare logical
16	RR	OR	OR
17	RR	XR	Exclusive OR
18	RR	LR	Load
19	RR	CR	Compare
1A	RR	AR (C)	Add
1B	RR	SR (C)	Subtract
1C	RR	MR (F)	Multiply
1D	RR	DR (F)	Divide
1E	RR	ALR (F)	Add logical
1F	RR	SLR (F)	Subtract logical
20	RR	LPDR (FP)	Load positive (long)
21	RR	LNDR (FP)	Load negative (long)
22	RR	LTDR (FP)	Load and test (long)
23	RR	LDCR (FP)	Load complement (long)
24	RR	HDR (FP)	Halve (long)
28	RR	LDR (FP)	Load (long)
29	RR	CDR (FP)	Compare (long)
2A	RR	ADR (FP)	Add normalized (long)
2B	RR	SDR (FP)	Subtract normalized (long)
2C	RR	MDR (FP)	Multiply (long)
2D	RR	DDR (FP)	Divide (long)
2E	RR	AWR (FP)	Add unnormalized
2F	RR	SWR (FP)	Subtract unnormalized (long)
30	RR	LPER (FP)	Load positive (short)
31	RR	LNER (FP)	Load negative (short)
32	RR	LTER (FP)	Load and test (short)
33	RR	LCER (FP)	Load complement (short)
34	RR	HER (FP)	Halve (short)
38	RR	LER (FP)	Load (short)
39	RR	CER (FP)	Compare (short)
3A	RR	AER (FP)	Add normalized (short)
3B	RR	SER (FP)	Subtract normalized (short)
3C	RR	MER (FP)	Multiply (short)
3D	RR	DER (FP)	Divide (short)
3E	RR	AUR (FP)	Add unnormalized (short)
3F	RR	SUR (FP)	Subtract unnormalized (short)
40	RX	STH (C)	Store half word
41	RX	LA	Load address
42	RX	STC	Store character
43	RX	IC	Insert character
44	RX	EX	Execute
45	RX	BAL (C)	Branch and link

Table B—5. List of Instructions by Opcode (Part 2 of 3)

Opcode Hexadecimal	Instruction Type	Mnemonic	Instruction
46	RX	BCT	Branch on count
47	RX	BC (C)	Branch on condition
48	RX	LH (C)	Load half word
49	RX	CH (C)	Compare half word
4A (AA)	RX	AH (C)	Add half word
4B (AB)	RX	SH (C)	Subtract half word
4C	RX	MH (F)	Multiply half word
4D	RX	BAS (C)	Branch and store
4E	RX	CVD	Convert to decimal
4F	RX	CVB	Convert to binary
50	RX	ST	Store
54	RX	N	AND
55	RX	CL	Compare logical
56	RX	O	OR
57	RX	X	Exclusive OR
58	RX	L	Load
59	RX	C	Compare
5A	RX	A	Add
5B	RX	S	Subtract
5C	RX	M	Multiply
5D	RX	D	Divide
5E	RX	AL (F)	Add logical
60	RX	STD (FP)	Store (long)
68	RX	LD (FP)	Load (long)
69	RX	CD (FP)	Compare (long)
6A	RX	AD (FP)	Add normalized (long)
6B	RX	AD (FP)	Subtract normalized (long)
6C	RX	MD (FP)	Multiply (long)
6D	RX	DD (FP)	Divide (long)
6E	RX	AW (FP)	Add unnormalized (long)
6F	RX	SW (FP)	Subtract unnormalized (long)
70	RX	STE (FP)	Store (short)
78	RX	LE (FP)	Load (short)
79	RX	CE (FP)	Compart (short)
7A	RX	AE (FP)	Add normalized (short)
7B	RX	SE (FP)	Subtract normalized (short)
7C	RX	ME (FP)	Multiply (short)
7D	RX	DE (FP)	Divide (short)
7E	RX	AU (FP)	Add unnormalized (short)
7F	RX	SU (FP)	Subtract unnormalized (short)
80	SI	SSM (P)	Set system mask
82	SI	LPSW (P)	Load PSW
83	SI	DIAG (P)	Diagnose
86	RS	BSH (F)	Branch on index high
87	RS	BXLE (F)	Branch on index low or equal
88	RS	SRL	Shift right single logical
89	RS	SLL	Shift left single logical
8A	RS	SRA (F)	Shift right single
8B	RS	SLA (F)	Shift left single
8C	RS	SRDL (F)	Shift right double logical
8D	RS	SLDL (F)	Shift left double logical
8E	RS	SRDA (F)	Shift right double
8F	RS	SLDA (F)	Shift left double
90	RS	STM	Store multiple
91	SI	TM (C)	Test under mask
92	SI	MVI (C)	Move immediate

Table B—5. List of Instructions by Opcode (Part 3 of 3)

Opcode Hexadecimal	Instruction Type	Mnemonic	Instruction
93	SI	TS (F)	Test and set
94	SI	NI (C)	AND
95	SI	CLI (C)	Compare logical
96	SI	OI (C)	OR
97	SI	XI	Exclusive OR
98	RS	LM	Load multiple
99	SI	HPR (P)	Halt and proceed
9A	SI	AI	Add immediate
9C	SI	SIO (P)	Start I/O
A2	RS	SSFS (P)	SOFTSCOPE forward scan
A3	RS	SSRS (P)	SOFTSCOPE reverse scan
A6	RS	AI (C)	Add immediate
A9	RS	HPR (P)	Halt and proceed
AA	RS	AH (C)	Add half word
AB	RS	SH (C)	Subtract half word
B0	RS	SSTM (P)	Supervisor store multiple
B1	RS	LCS (P)	Load control storage
B8	RS	SLM (P)	Supervisor load multiple
D1	SS	MVN (C)	Move numerics
D2	SS	MVC (C)	Move character
D3	SS	MVZ (C)	Move zones
D4	SS	NC (C)	AND
D5	SS	CLC (C)	Compare logical
D6	SS	OC (C)	OR
D7	SS	XC	Exclusive OR
DC	SS	TR (C)	Translate
DD	SS	TRT	Translate and test
DE	SS	ED (C)	Edit
DF	SS	EDMK (F)	Edit and mark
F1	SS	MVO (C)	Move and offset
F2	SS	PACK (C)	Pack
F3	SS	UNPK (C)	Unpack
F8	SS	ZAP (C)	Zero and add
F9	SS	CP (C)	Compare decimal
FA	SS	AP (C)	Add decimal
FB	SS	SP (C)	Subtract decimal
FC	SS	MP (C)	Multiply decimal
FD	SS	DP (C)	Divide decimal

Table B—6. Compatibility Mode Instructions

Opcode Hexadecimal	Instruction Type	Mnemonic	Instruction	Valid Modes	
				360/20	9200/9300
07	RR	BCR	Branch on condition	X	
OD	RR	BASR	Branch and store	X	
1A	RR	AR	Add	X	
1B	RR	SR	Subtract	X	
40	RX	STH	Store half word	X	X
45	RX	BAL	Branch and line		X
47	RX	BC	Branch on condition	X	X
48	RX	LH	Load half word	X	X
49	RX	CH	Compare half word	X	X
4A	RX	AH	Add half word	X	
4B	RX	SH	Subtract half word	X	
4D	RX	BAS	Branch and store	X	
91	SI	TM	Test under mask	X	X
92	SI	MVI	Move immediate	X	X
94	SI	NI	AND	X	X
95	SI	CLI	Compare logical	X	X
96	SI	OI	OR	X	X
A6	SI	AI	Add immediate		X
AA	RX	AH	Add half word		X
AB	RX	SH	Subtract half word		X
D1	SS	MVN	Move numerics	X	X
D2	SS	MVC	Move	X	X
D3	SS	MVZ	Move zones	X	
D4	SS	NC	AND		X
D5	SS	CLC	Compare logical	X	X
D6	SS	OC	OR		X
DC	SS	TR	Translate	X	X
DE	SS	ED	Edit	X	X
F1	SS	MVO	Move and offset	X	X
F2	SS	PACK	Pack	X	X
F3	SS	UNPK	Unpack	X	X
F8	SS	ZAP	Zero and add	X	X
F9	SS	CP	Compare decimal	X	X
FA	SS	AP	Add decimal	X	X
FB	SS	SP	Subtract decimal	X	X
FC	SS	MP	Multiply decimal	X	X
FD	SS	DP	Divide decimal	X	X

Table B—7. Operation Exception Instructions for Compatibility Modes

Opcode Hexadecimal	Instruction Type	Mnemonic	Instruction	Operation Exception	
				360/20	9200/9300
81*	SI	SPSW	Set PSW	X	
83	SI	DIAG	Diagnose	X	
99	SI	HPR	Halt and proceed	X	
9A	SI	AI	Add immediate	X	
9B*	SI	CIO	Control I/O	X	
A0*	SI	SPSC			X
A1*	RS	SRC	Supervisor call		X
A4*	RS	XIOF	Execute I/O		X
A5*	RS	TIO	Test I/O		X
A8*	RS	LPSC	Load state		X
A9*	RS	HPR	Halt and proceed		X
DO*	SS	XIO	Translate I/O	X	

* Not valid for 90/30.

Appendix C. Abbreviations and Acronyms

A

A
Active

ASCII
American Standard Code for Information Interchange

B

BCA
Bit count appendage

BCSW
Buffered channel status word

BCSWE
Buffered channel status word error

BCSWSE
Buffered channel status word servicing error

BCW
Buffer control word

BDW
Branch discipline word

C

CA
Communications adapter

CAF
Communications attachment feature

CAW
Channel address word

CBWE
Channel buffer word error

CC
Command chaining or condition code

CCW
Channel command word

CTO
Channel timeout

E

EBCDIC
Extended Binary Coded Decimal Interchange Code

ECC
Error correction code

I

ID
Microcode identification

IDA
Integrated disk adapter

IDE
Interrupt data error

IGR
Integrated general register

IIS
Interrupt initialization sequence

ILC
Instruction length code

ITR
Interval timer register

ILCS
Initial load control storage

IOST
Input/output status tabler

IOSTCW
Input/output status table control word

IOST IIS
Input/output status table interrupt initialization sequence

IOSTIW
Input/output status table interrupt word

IOSTS
Input/output status table service request

IOSTSW
Input/output status table status word

IPC
Integrated peripheral channel

IPL
Initial program load

ISK
Insert storage key

ISR
Interrupt selective reset

ISS
Initial selection sequence

IWE
Interrupt word error

L

LA
Line adapter

LECS
Low end communications subsystem

LPSW
Load program status word

LSB
Least significant bit
Least significant byte

M

MC
Microcontrol; machine check

MFM
Modified frequency modulation recording

MON
Monitor

MS
Main storage

MSB
Most significant bit
Most significant byte

P

PR
Problem register (supervisor registers)

PS
Problem/supervisor mode

PSW
Program status word

PX
Program exception

R

r
Replacement, when applied to BCW fields
Register, when applied to instructions

RDG
Read data generator

RDR
Read data register

REV
Revision level

RP
Read protect

RR
Register to register

RS
Register to storage

RX
Register to indexed storage

S

SBX
Sense byte x, where x can be 0, 1, 2, etc

SI
Storage and immediate operand

SIO
Start input/ouput

SLI
Suppress length indication

SPM
Set program mask

SS
Storage to storage

SSK
Set storage key

SSM
Set system mask

SSR
Status service request

SSRA
Status service request acknowledge

STF
Status table full

STR
Service timer register

T

TIC
Transfer in channel

TIR
Timer interrupt request

TPI
Tracks per inch

V

VFB
Vertical format buffer

W

WE
Write enable

Index

N

Comments concerning this manual may be made in the space provided below. Please fill in the requested information.

System: _____

Manual Title: _____

UP No: _____ Revision No: _____ Update: _____

Name of User: _____

Address of User: _____

Comments:

CUT

FOLD

FIRST CLASS
PERMIT NO. 21
BLUE BELL, PA.

BUSINESS REPLY MAIL NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

SPERRY UNIVAC

P.O. BOX 500
BLUE BELL, PA.
19424

ATTN: SYSTEMS PUBLICATIONS DEPT.

FOLD

CUT