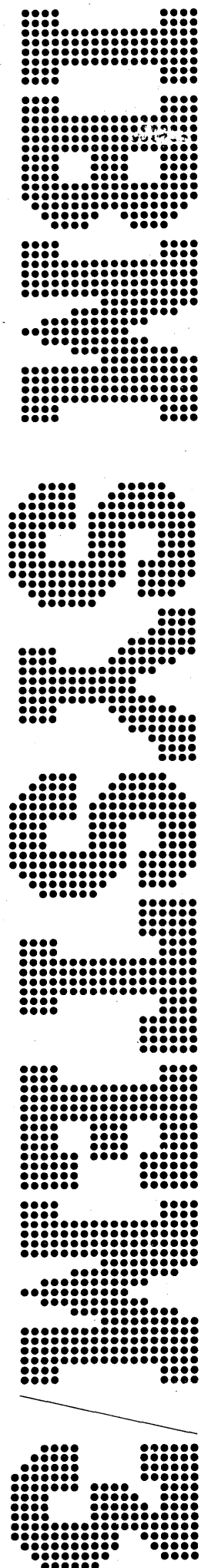


IBM System/3 Model 6
Components Reference Manual

GA34-0001-3
File No. S3-01

Fourth Edition (August 1974)

This is a reprint of GA34-0001-2 incorporating changes released in the following Technical Newsletter: GN21-0173 (dated 8 April 1974).

Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM systems, refer to the latest System/3 Bibliography, GC20-8080, for the editions that are applicable and current.

Text for this manual has been prepared with the IBM Selectric® Composer.

Some illustrations in this manual have a code number in the lower corner. This is a publishing control number and is not related to the subject matter.

Requests for copies of IBM publications should be made to your IBM representative or the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, send your comments to IBM Corporation, Systems Publications, Department 27T, P. O. Box 1328, Boca Raton, Florida 33432. Comments become the property of IBM.

© Copyright International Business Machines Corporation 1970, 1971, 1972, 1974

Contents

This manual contains the organizational concepts and the instruction set for the IBM System/3 Model 6.

The publication is intended as a reference manual for those interested in writing or maintaining programs written in assembler language for the System/3 Model 6.

Explanations of the data formats, interruption structure, addressing and storage organization, and machine language instruction set are included. Additionally, the data formats, machine language instructions, programming requirements, and special considerations for all I/O units currently available on the system are explained.

It is assumed that the reader is familiar with good programming practice and with the terminology of data processing. More information on the IBM 1255 Magnetic Character Reader is contained in *IBM 1255 Magnetic Character Reader Components Description,* Order No. GA24-3542. For off-line data-recorder operating procedures, refer to *IBM 5496 Data Recorder Operator's Guide,* Order No. GA21-9086. Off-line operating procedures for the 80-column data recorder are included in *IBM 129 Card Data Recorder Machine Description,* Order No. GA22-6980.

No assembler language program support exists for the System/3 Model 6. The program listings for IBM program products that support the system, however, contain assembler mnemonics. Therefore, machine mnemonics are included in this manual as an aid to reading any program listings which use the mnemonics.

This manual makes reference to calling the customer engineer (CE) for assistance. The service or assistance resulting from some CE calls may be subject to billing. If your system is serviced by an organization other than IBM, please call them.

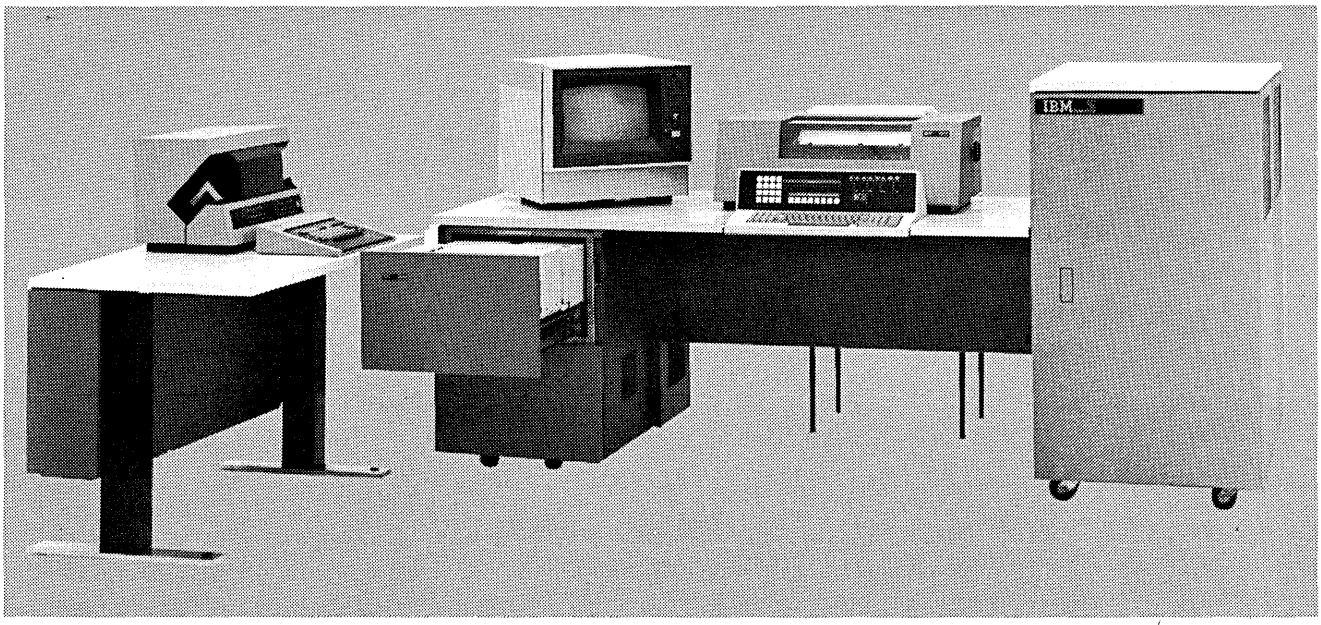


Figure 1-1. IBM System/3 Model 6

Chapter 1. IBM System/3 Model 6

The IBM System/3 Model 6 (Figure 1-1) is an operator-oriented data processing system. Data entry is centered in an operator's keyboard-console; data output is centered in a console-type serial printer. Storage is provided by the internal (main) storage of the processing unit and by magnetic-disk storage units.

SYSTEM COMPONENTS AND CONFIGURATION

Figure 1-2 shows the units of the system. System/3 Model 6 must include the following:

- IBM 5406 Processing Unit. The basic storage size is 8192 bytes. Additional storage is available as shown in Figure 1-2. Included as special features are a serial input/output channel adapter and a binary synchronous communication adapter. The operator's keyboard-console is part of the processing unit.
- IBM 5444 Disk Storage Drive. The disk storage drive is available in four combinations of three different models to provide 2,457,600 bytes; 4,915,200 bytes; 7,372,800 bytes; or 9,830,400 bytes of storage.
- IBM 5213 Printer, Model 1, 2, or 3, or IBM 2222 Printer Model 1 or 2. The 5213 can print 132 characters per line. The 2222 can print 220 characters per line. Both printers operate at 85 characters per second. Other capabilities of the printers include (1) a bi-directional printing feature for increased line-per-minute printing speeds, (2) vertical forms control, (3) a ledger card device, and (4) an enhanced-print-rate feature that increases the per-character print rate to a nominal 115 characters per second. The bi-directional printing capability is standard on the 5213 Model 3 and on the 2222 Model 2, but not available on the 5213 Model 1 or 2 or on the 2222 Model 1. Vertical forms control is standard on the 5213 Models 2 and 3 and the 2222 Models 1 and 2, but is not available on the 5213 Model 1. The ledger card device is standard on the 2222 Models 1 and 2, but is not available on the 5213 Model 1, 2, or 3. Enhanced print rate is available only on the 5213 Model 3.

The following special features can be included in System/3 Model 6:

- IBM 2265 Display Station Model 2. The 2265 can display the contents of 960 adjoining storage locations.
- IBM 5496 Data Recorder Model 1 with the On-Line Feature for System/3 Model 6. The 5496 is cable-connected to the processing unit and operates either on-line or off-line. This unit permits (1) entering data into the processing unit storage from punched cards, and (2) punching data into cards from storage. The data punched into the cards can also be printed on the cards. Punching without printing can be performed, but printing without punching cannot. Card data is in the 6-bit, 96-column format.
- IBM 1255 Magnetic Character Reader. The 1255 reads magnetic ink characters and sorts paper documents. This unit is attached to the system through the serial I/O channel.
- Serial I/O Channel. This is an adapter unit built into the processing unit to allow special I/O devices to be attached to the system.
- Binary Synchronous Communication Adapter. This is an adapter unit built into the processing unit to allow communication between the system and remote systems or terminals.
- Additional Command Keys. This feature increases the number of command keys on the keyboard from 8 to 16.

- IBM 129 Card Data Recorder Model 1, 2, or 3 with the Card Input/Output Feature. The 129 is cable-connected to the processing unit and operates either on-line or off-line. This unit permits (1) entering data into the processing unit storage from punched cards, and (2) punching data into cards from storage. The data punched into the cards can also be printed on the cards. Punching without printing can be performed, but printing without punching cannot. Card data is in the 12-bit, 80-column format.

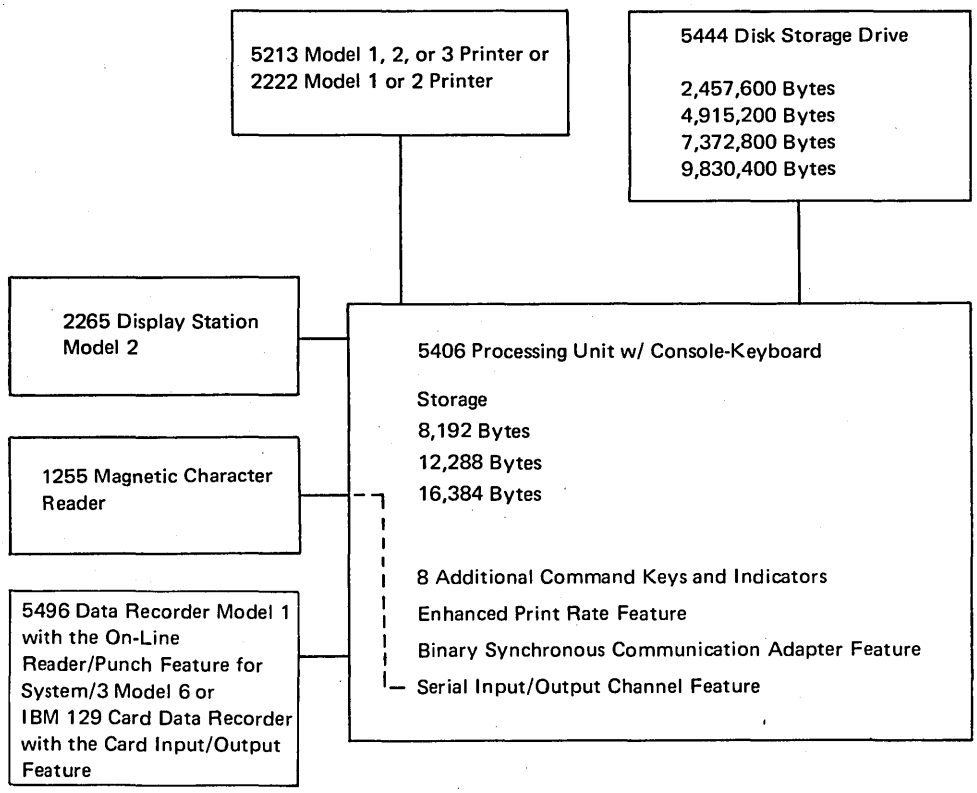


Figure 1-2. System/3 Model 6 Configuration

SYSTEM/3 DATA INPUT/OUTPUT AND INTERPRETATION

Data is entered into System/3 through the keyboard or from punched cards via the 5496 or 129. The card used in the 5496 (Figure 1-3) has 96 positions for recording data and 96 positions for printing by the data recorder. A fourth print line position is included for use by other systems. Data is recorded in three tiers of 32 columns each. Each of the three print lines used by System/3 Model 6 contains 32 positions for printing. The card used in the 129 (Figure 1-4) has 80 positions for recording data and 80 positions for printing by the card data recorder. The 80-column card is not divided into tiers of data and provides only a single print line. Data can also be entered into the system from paper documents (via the magnetic character reader), from the disk storage unit, or from the communication adapter.

Data output from the system is by way of the printer, the display station, the disk storage drive, or the communication adapter. Each of these devices (as well as the processing unit itself) uses a somewhat different method of representing data.

Data Formats

Data resides in main storage as binary bits in eight-bit bytes. Each byte also contains a parity bit. All data bytes in storage look the same to the processing unit. The instruction the processing unit is executing determines how the data is interpreted. The data contained in a byte can be interpreted as:

1. A numeric, alphabetic, or special character
2. An unsigned binary number (logical data)

The bytes that are interpreted as characters constitute a code called Extended Binary Coded Decimal Interchange Code (EBCDIC).

Bytes interpreted as binary numbers can also be used as bit-significant conditional or status information.

The disk storage drive and the magnetic character reader transfer data in the same formats used by the processing unit—that is, as EBCDIC characters or binary numbers. Data read from cards by the data recorder is transferred as EBCDIC characters only.

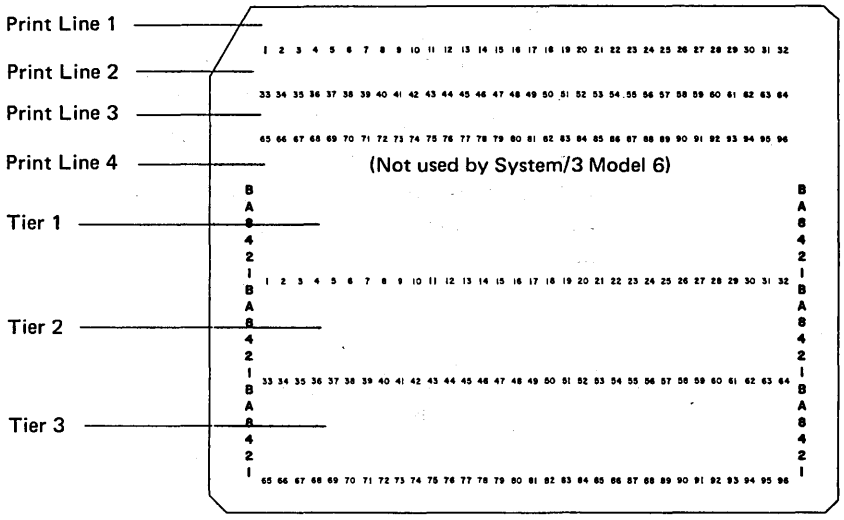


Figure 1-3. 96-Column Card

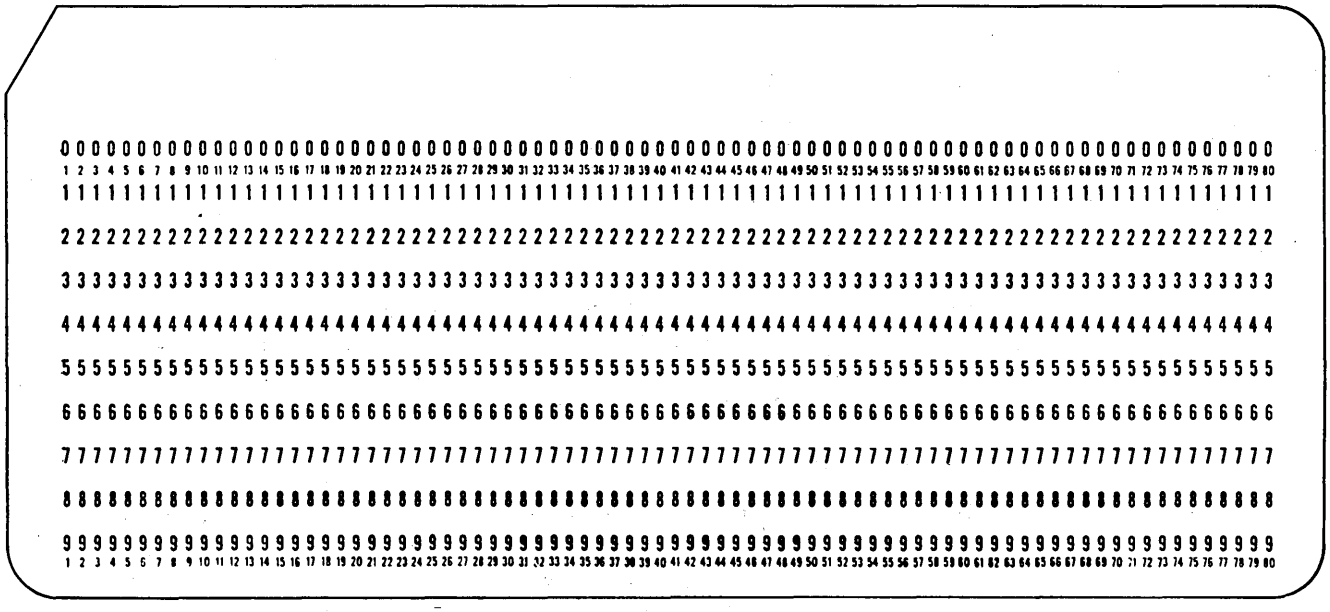


Figure 1-4. 80-Column Card

Data in 96-column cards is in a six-bit code. When the cards are read, this six-bit code is converted to EBCDIC. Data from main storage that is to be punched into cards can be either EBCDIC characters or binary numbers. If the data is binary numbers, some six-bit code is punched, but the code may not be the code expected from EBCDIC. See "Appendix B."

Data in 80-column cards is in a 12-bit code. When the cards are read, this 12-bit code is converted to EBCDIC. Data from main storage that is to be punched into cards can be either EBCDIC or binary numbers. If the data is binary numbers, some 12-bit code is punched, but the code may not be the code expected from EBCDIC.

The keyboard transfers data to the processing unit in an eight-bit code that is entirely different from EBCDIC.

The printer and the display station do not use the two high-order bits of the byte for data. The display station uses one of the high-order bits for a character position indicator. The data code for the printer and the display station is composed of six bits.

Character Format

In character format, each byte of data is interpreted as though it were divided into two groups of four bits each. Bits 0-3 constitute the zone portion, and bits 4-7 constitute the digit portion. The character format can represent a decimal digit, a communication adapter control character, or one of the graphics (printable characters) defined for the system. Figure 1-5 shows the byte as interpreted for character format.

When data is handled in this format, the zone bits do not participate in any arithmetic operations. The zone bits of the low-order byte indicate the sign of the field for arithmetic operations. Zone bits equal to 1101 indicate a negative number; all other zone bit patterns indicate a positive number.

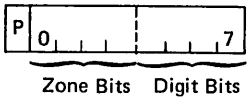


Figure 1-5. Character Format

Binary Format (Logical Data)

When data is handled in binary format, it is treated as an eight-bit binary integer as shown in Figure 1-6.

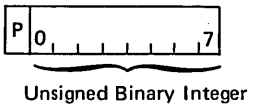


Figure 1-6. Binary Format

Parity

In addition to the eight data bits, each byte contains one other bit called a parity bit. This bit is used to maintain an odd number of bits in the byte. The processing unit checks each byte to ensure that it contains an odd number of bits. If an even number of bits is detected, the system stops and turns on an indicator.

Six-Bit Card Code

Data is stored in the 96-column card in six-bit form. Each of the ninety-six columns in the card can contain one six-bit character, which is converted during input operations to the eight-bit character format (EBCDIC). On output to the data recorder, character format is converted to card code. Figure 1-7 shows the card code representation of each character.

Twelve-Bit Card Code

Data is stored in the 80-column card in 12-bit form. Each of the 80 columns in the card can contain one character, which is converted during input operations to the eight-bit character format (EBCDIC). On output to the card data recorder, character format is converted to card code. Figure 1-8 shows the card code representation of each character.

Keyboard Code

Data entered through the keyboard enters the processing unit as eight binary bits plus a parity bit. These bits are not in character format. The specific eight-bit code for each key and key shift on the keyboard is shown in "Appendix B." Translation between the keyboard code and the character format or binary format must be performed by the program.

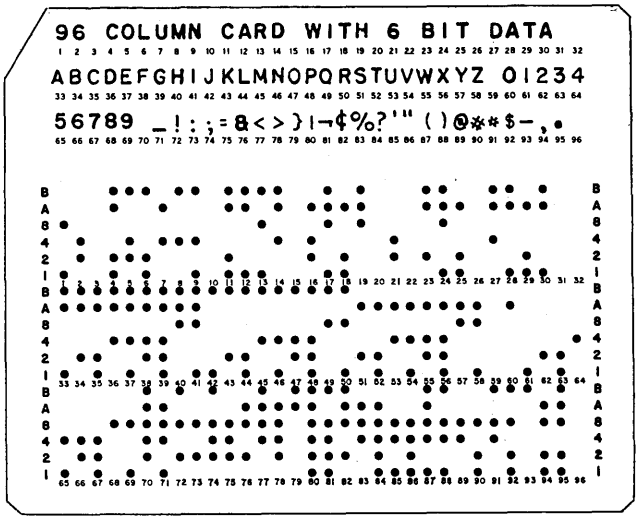


Figure 1-7. 96-Column Card Code

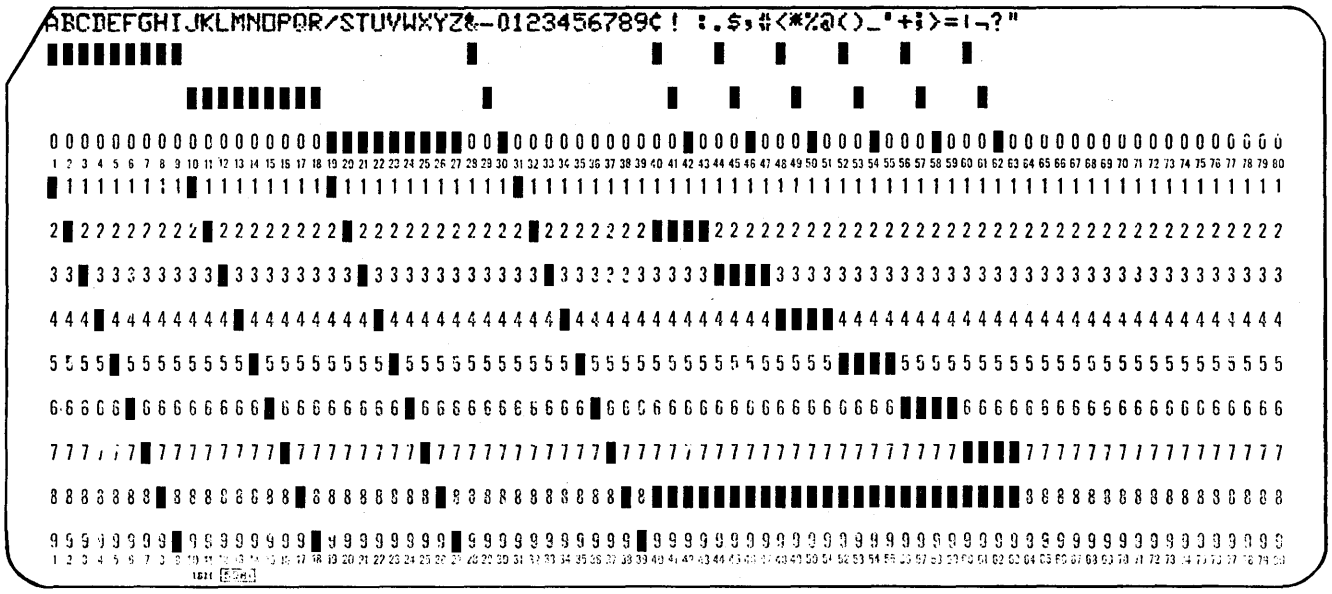


Figure 1-8. 80-Column Card Code

Serial Printer and Display Unit Code

This six-bit code is a subset of the eight-bit character format used by the processing unit. Only the low-order six bits of the byte are considered in developing the character, but any defined graphic in character format is displayed as that graphic on the display unit or is printed by the printer.

ADDRESSING

Byte location addresses in storage are expressed in binary form and consecutively numbered from hexadecimal 0000 to the upper limit of storage. "Appendix D" explains the binary number system, and "Appendix E" contains the hexadecimal representation for addresses 0000 to 4095. The location of any field or group of bytes is specified by the address of either the leftmost (high-order, lowest address) byte or the rightmost (low-order, highest address) byte of the field, depending on the instruction.

An address used to refer to main storage can be specified by either of two methods; *direct addressing* or *base-displacement addressing*. The type of addressing to be used is specified by bits 0 through 3 of the first byte of the instruction. These four bits are treated as two groups of two bits each, group 0 and 1 and group 2 and 3. Bits 0 and 1 control addressing for operand 1 in a two-address instruction; bits 2 and 3 control addressing for operand 2 in a two-address instruction.

Direct Addressing

When both bits of either bit group are 0, the corresponding operand is selected from storage by direct addressing. Both bit groups can equal 00, causing both operands to be selected by direct addressing.

When direct addressing is employed, the storage address is taken directly from the instruction. The address in the instruction is two bytes long.

Base-Displacement Addressing

When either or both bit groups have one bit equal to 1 and the other bit equal to 0, the corresponding operand is selected from storage by base-displacement addressing.

In base-displacement addressing, the positive binary number in the one-byte address of the instruction is added to the positive binary number in a two-byte index register. The result of this addition is used to select the operand from storage. The index register to be used is determined by the bit of the bit group that is 1. If the low-order bit (bit 1 or bit 3) is 1, index register 1 is used. If the high-order bit of the bit group (bit 0 or bit 2) is 1, index register 2 is used. Both bit groups can specify the same index register during the execution of an instruction.

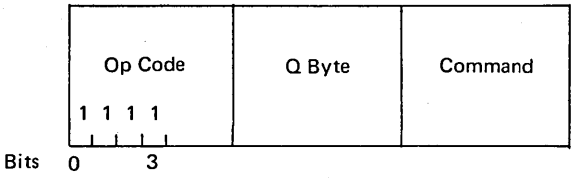
Because the maximum value that the one-byte address in the instruction can contain is decimal 255, any one value of an index register allows access to 256 storage positions.

INSTRUCTION FORMATS

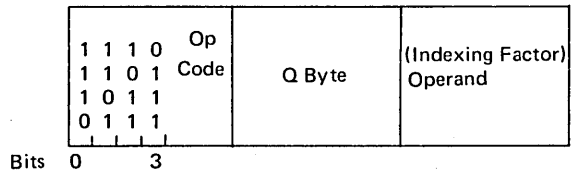
System/3 provides three instruction formats of varying length. These instruction formats are distinguished by their ability to address storage. The length of each instruction is determined by the type of addressing being performed.

As Figure 1-9 shows, all instruction formats have two elements in common, the operation (op) code and the Q byte. Each of these elements is one byte. The op code determines the type of addressing to be performed (and thereby the length of the instruction) and the operation to be performed. Figure 1-10 summarizes addressing in instruction format sequence. The function of the Q byte is determined by the instruction being performed.

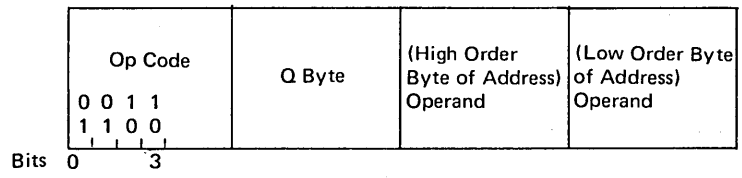
Command Instruction



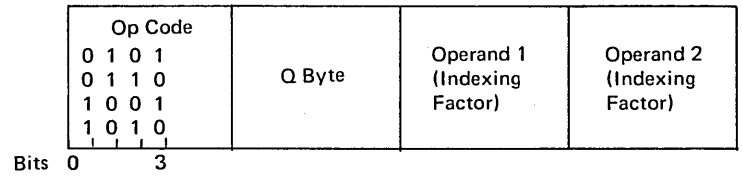
One Address Instruction—Base-Displacement Addressing



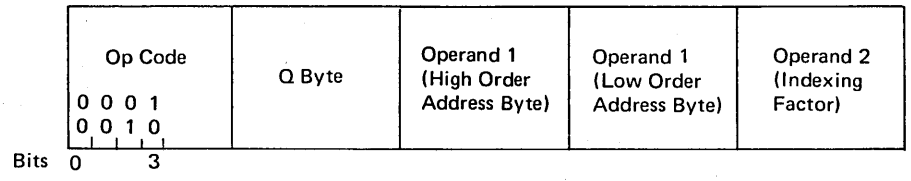
One Address Instruction—Direct Addressing



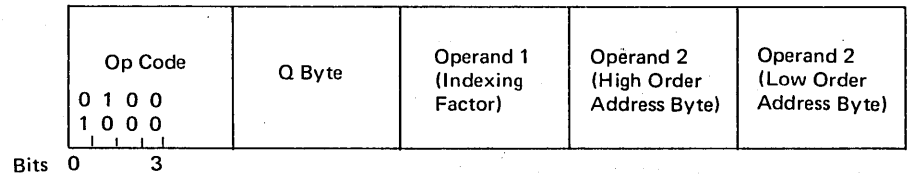
Two Address Instruction—Both Addresses Base-Displacement



Two Address Instruction—Operand 1 Address Direct



Two Address Instruction—Operand 2 Address Direct



Two Address Instruction—Both Addresses Direct

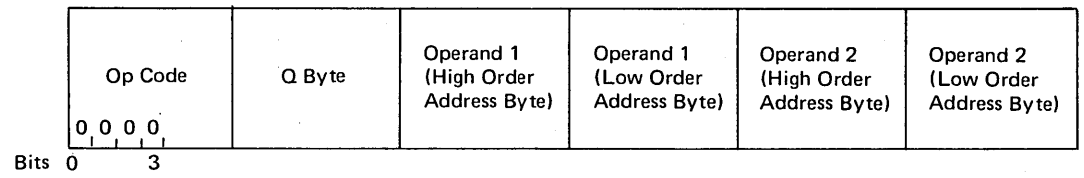


Figure 1-9. Instruction Formats

Instruction Type	Op Code Bits 0 1 2 3	Instruction Length	Addressing Type and Index Register	
			Operand 1	Operand 2
Command	1 1 1 1	3 Bytes		
One Address	0 0 1 1	4 Bytes	Direct	
One Address	0 1 1 1	3 Bytes	Base-Displacement Reg 1	
One Address	1 0 1 1	3 Bytes	Base-Displacement Reg 2	
One Address	1 1 0 0	4 Bytes		Direct
One Address	1 1 0 1	3 Bytes		Base-Displacement Reg 1
One Address	1 1 1 0	3 Bytes		Base-Displacement Reg 2
Two Address	0 0 0 0	6 Bytes	Direct	Direct
Two Address	0 0 0 1	5 Bytes	Direct	Base-Displacement Reg 1
Two Address	0 0 1 0	5 Bytes	Direct	Base-Displacement Reg 2
Two Address	0 1 0 0	5 Bytes	Base-Displacement Reg 1	Direct
Two Address	1 0 0 0	5 Bytes	Base-Displacement Reg 2	Direct
Two Address	0 1 0 1	4 Bytes	Base-Displacement Reg 1	Base-Displacement Reg 1
Two Address	0 1 1 0	4 Bytes	Base-Displacement Reg 1	Base-Displacement Reg 2
Two Address	1 0 0 1	4 Bytes	Base-Displacement Reg 2	Base-Displacement Reg 1
Two Address	1 0 1 0	4 Bytes	Base-Displacement Reg 2	Base-Displacement Reg 2

Figure 1-10. Addressing by Instruction Types

Command Instructions

Command instructions are always three bytes long and do not address storage. In a command instruction, the Q byte contains the following information, depending on the instruction:

1. Device address and function specification
2. Jump condition
3. Halt identifier (alpha code)

Bits 0 through 3 of the command-instruction op code are always hexadecimal F.

One-Address Instructions

One-address instructions can be either three or four bytes long. These instructions always have either bits 0 and 1 or bits 2 and 3 of the op code byte both ones. The two bits that are not both ones (0 and 1, or 2 and 3) can be 01, 10, or 00. If these bits are 00, addressing is direct and the instruction is four bytes long. If the bits are 01 or 10, base-displacement addressing is used; the instruction is three bytes long; and index register 1 (01) or index register 2 (10) is used.

The Q byte of a one-address instruction can contain:

1. An immediate operand
2. A mask
3. A branch condition
4. A data selection

Two-Address Instructions

Two-address instructions can be four, five, or six bytes long. This instruction type is distinctive in that neither bit group 0 and 1 nor bit group 2 and 3 of the op code byte is both ones. If all four of bits 0 through 3 are zero, addressing is direct; and the instruction is six bytes long. If any one of bits 0 through 3 is one, one of the addresses is direct and the other is base-displacement; and the instruction is five bytes long. If one bit from each of the bit groups is one, all addressing is done by the base-displacement method; and the instruction is four bytes long.

The index register to be used in base-displacement addressing for either operand is determined by the bit in the bit group that is 1. If the bit group equals 01, index register 1 is used; if the bit group equals 10, index register 2 is used. Both addresses can use the same index register during one instruction.

The processing unit (Figure 2-1) is the heart of the system. It controls the input of data to the system (by calling for data when required), the output of data from the system, and the operations performed on the data while it is in the system.

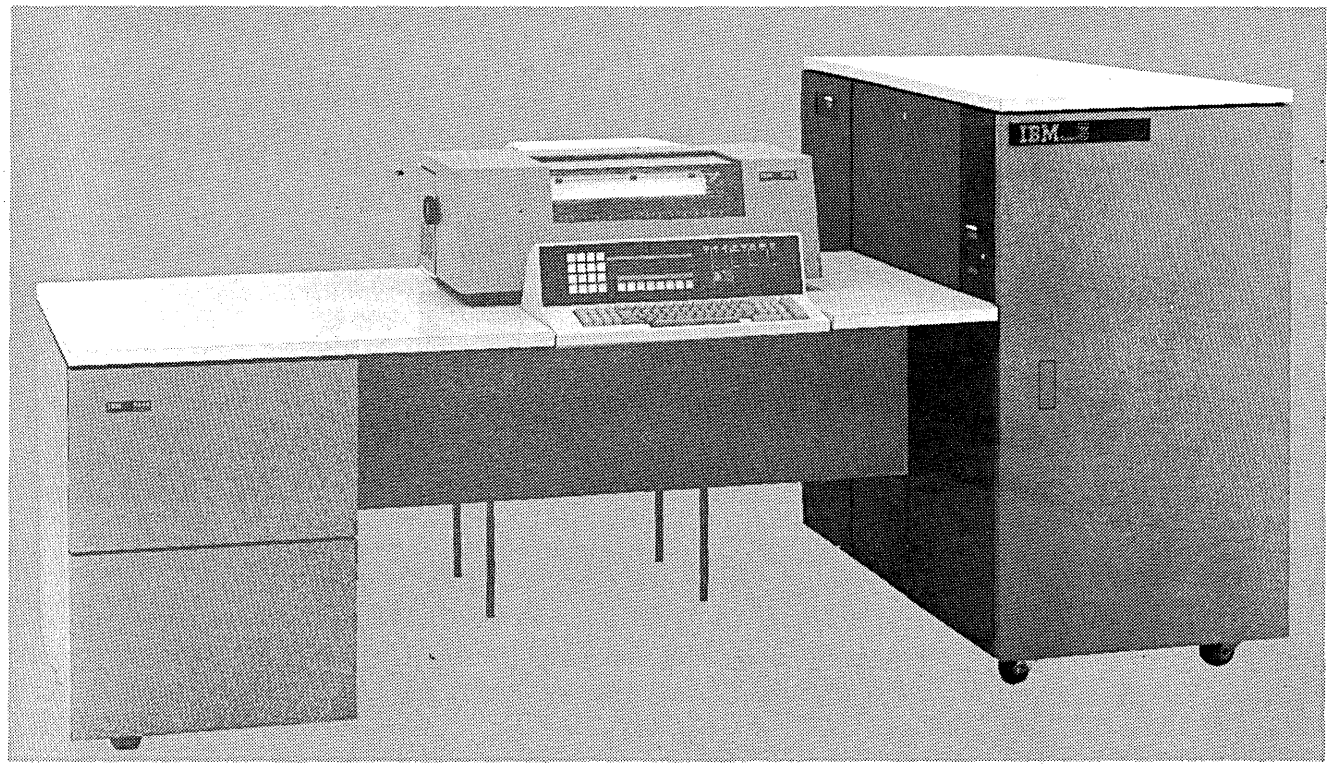
PROCESSING UNIT DATA FLOW

Figure 2-2 shows the data flow for the basic processing unit. Data is taken from storage through the storage data register (SDR) to the B register. From the B register data enters the arithmetic and logic unit (ALU) to be operated on and directed to one of the following units:

1. Op register
2. Q register
3. Condition register
4. Local storage registers (LSR's)
5. I/O unit

The data can also be sent to the SDR to be returned to storage. In certain operations, part of the data is returned to storage from the SDR without passing through the B register and ALU.

Data coming into the system enters the A register, passes through the ALU, and enters storage through the SDR.



BR0299

Figure 2-1. IBM 5406 Processing Unit

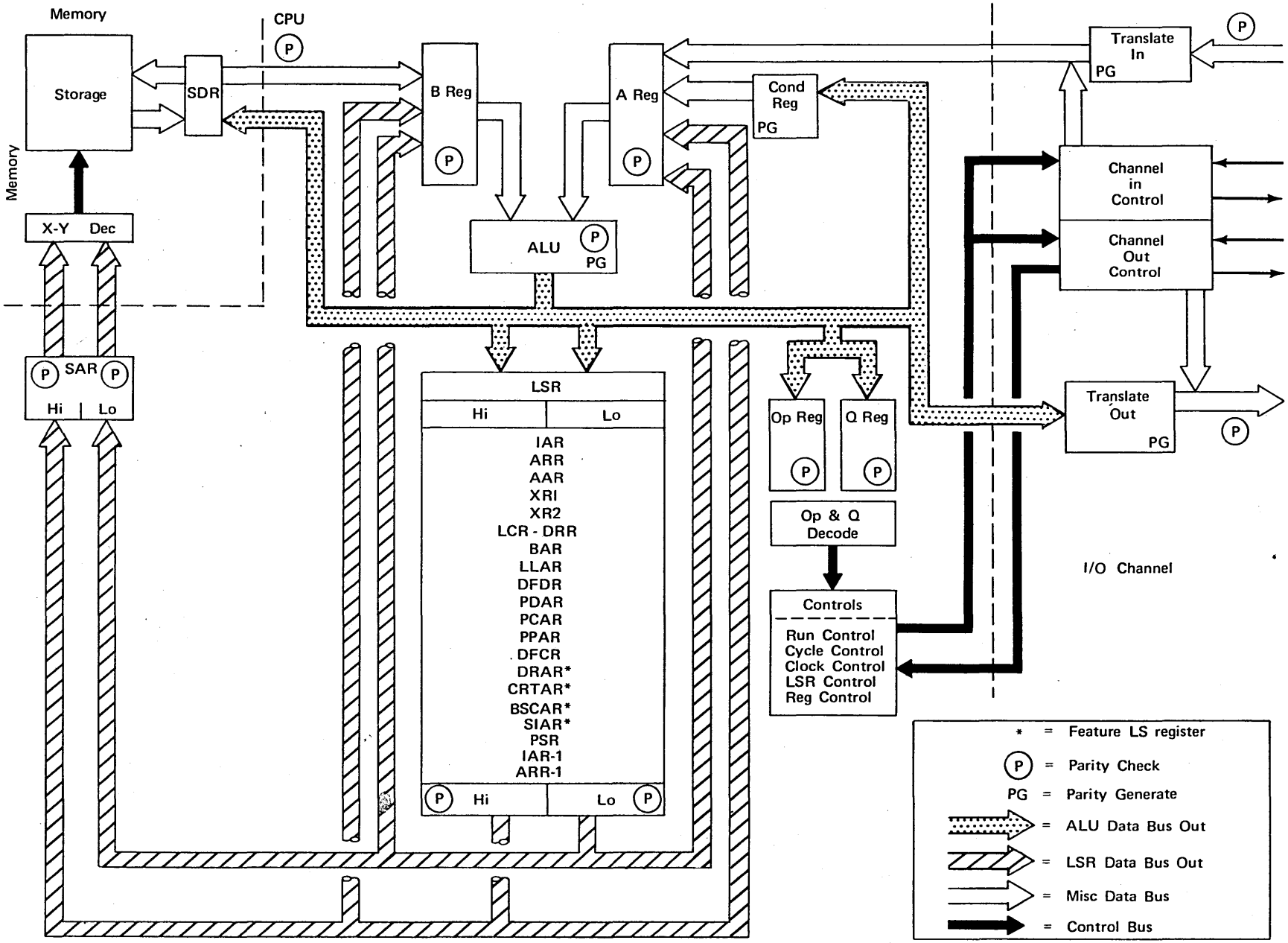


Figure 2-2. Processing Unit Data Flow

Main Storage

Main storage consists of 8,192; 12,288; or 16,384 positions of magnetic core storage. Each position has an address and contains an eight-bit unit of information called a byte. Coded combinations of bits within a byte can represent alphabetic, numeric, or logical data.

Main storage holds all the data that is processed at a given time. It also holds the instructions, or program, that control the operation of the system. For each individual job performed by the system, certain portions of main storage are assigned to store instructions and certain portions to store data to be processed.

Storage Data Register (SDR)

This register serves as a place to store data temporarily as it passes between the processing portions of the processing unit and main storage. Data can enter the storage data register from the ALU or from main storage. Data can be sent from the storage data register to either the B register or to main storage. The storage data register holds one byte of data.

A and B Registers

The A and B registers serve as temporary storage for data to be processed by the ALU. Each of these registers holds one byte of data and each can be entered from several other units in the data flow.

Arithmetic and Logic Unit

The ALU performs all the arithmetic and logic functions for the processing unit. It is capable of decimal add, decimal subtract, binary add, binary subtract, compare, logical AND-OR, pass A register, and pass B register functions. All data that is to be moved from any unit in the data flow to any other unit in the data flow (except the storage address register and the A and B registers) must pass through the ALU. (Data enters the A and B registers only if it is to pass through the ALU.) The ALU accepts two bytes of input and produces one byte of output.

Storage Address Register

The storage address register (SAR) holds the address that is to be accessed in main storage. The SAR holds two bytes.

Condition Register

The condition register holds results of various operations, such as compare operations, and overflow and positive or negative indications from arithmetic operations. It can be tested for these conditions.

The various bits of the condition register are assigned names as follows:

Bit	*Name*
0	Unassigned
1	Unassigned
2	Binary overflow
3	Test false
4	Decimal overflow
5	High
6	Low
7	Equal

The bits are referred to by these names in the remaining sections of this manual.

Q Register

This register accepts the Q byte from the instruction. The Q byte is read from this register to circuitry that controls the operations performed by instructions.

Op Code Register

The op code register holds the op code byte of each instruction taken from storage to enable the control circuitry to perform the desired operation.

Local Storage Registers (LSR)

The local storage registers are a group of sixteen registers that contain data required for the execution of instructions. Sixteen registers are required for the system with standard I/O devices (disk storage and printer) and for certain optional feature I/O devices. Additional local storage registers (called feature LSR's) are installed to accommodate optional feature I/O devices when they are required. Each of the local storage registers contains two bytes.

During system reset (caused by operating the system reset switch) and during program loading, certain of the local storage registers are reset to zero. System reset causes the instruction address register and the program status register to be reset to all zeros. Operating the program load switch resets the instruction address register, the program status register, and the disk file data address register to all zeros. All other addressable local storage registers must be set to their initial values by specific instructions before they can be used. During the first instruction after a system reset or program load, the program status register and the condition register are set to condition equal. This condition can be changed by the next instruction.

Instruction Address Register (IAR)

This register contains the address of the instruction bytes as they are addressed. It is increased by one each time an instruction byte is taken from storage. When the last byte of any instruction has been taken from storage, the instruction address register contains the address of the first byte of the next sequential instruction (NSI).

The basic group of registers contains two instruction address registers: the instruction address register for the main program and the instruction address register for interrupt level 1 (the interrupt level for the keyboard). The feature local storage registers contain instruction address registers for interrupt level 2 (if the communication adapter is installed) and interrupt level 4 (if the serial I/O channel is installed).

Address Recall Register (ARR)

The address recall register is used by certain instructions to store a beginning address for execution of the instruction or the address of the next sequential instruction. As with the instruction address register, there are address recall registers for the main program and for interrupt levels 1, 2, and 4. The address recall register is affected only by branch, decimal, and insert-and-test-characters instructions.

Operand 2 Address Register (AAR)

This register, which is set during instruction readout from storage, addresses the various bytes of operand 2. It is updated as each individual byte of operand 2 participates in the execution of the instruction. This register cannot be addressed by the program.

Index Register 1 (XR1)

This register is used to contain the base address for base-displacement addressing. This register is shared by the main program and by all interrupt levels.

Index Register 2 (XR2)

This register is used as a base register for base-displacement addressing. It is used by the main program and by all interrupt levels.

Program Status Register (PSR)

The high-order byte of this register is used as a length count recall register during instructions that may require recomplement operations. The low-order byte is used as a condition recall register for storing the value of the condition register when the processing unit is interrupted by an I/O unit. Loading this register automatically sets the condition register to the same value as the low-order byte.

Operand 1 Address Register (BAR)

This register is set during instruction readout and is used to address the various bytes of operand 1 as they are required by the instruction. This register is updated as each individual byte of operand 1 participates in the instruction. This register cannot be addressed by the program.

Ledger Card Device Locate Line Address Register (LLAR)

This register is used as a counter for the locate line operation of the ledger card device.

Length Count/Data Recall Registers (LCR and DRR)

These two registers occupy one LSR. Each is one byte in size. The length count register stores the length count from two-address instructions. The data recall register is used in two-operand type instructions to hold a byte of one operand while a byte of the other operand is retrieved from storage. These registers are not accessible to the program.

Disk File Control Address Register (DFCR)

This register holds the address of the high-order byte of a four-byte disk control field.

Disk File Data Address Register (DFDR)

This register stores the address of the high-order byte of the disk-file-data area of storage.

Printer Data Address Register (PDAR)

This register holds the address of the high-order byte of the print-data area of storage for the 5213 or 2222 Printer.

Printer Command Address Register (PCAR)

This register holds the storage address of the high-order byte of a group of printer-command bytes for the 5213 or 2222 Printer.

Data Recorder Address Register (DRAR)

This register contains the address of the high-order byte of the area in storage that contains data to be transferred to or that has been transferred from the data recorder.

Binary Synchronous Communication Adapter Address Register (BSCAR)

This register contains the address of the high-order byte of a group of bytes in storage that are to be used for exchanging data with the communication adapter. The register is available only when the communication adapter special feature is installed.

Serial I/O Channel Address Register (SIAR)

This register is available only when the serial I/O channel special feature is installed. It stores the address of the high-order byte of the field into which or from which serial I/O channel data is transferred.

CRT Address Register (CRTAR)

This register is available only when the display station special feature is installed. It holds the high-order address of a 960-character block of storage from which characters are taken for display on the cathode ray tube in the display station.

CYCLES AND PHASES

Each processing unit operation is performed in two phases, instruction phase and execution phase. Some instructions combine the phases so that there is no distinct execution phase.

During the instruction phase, the processing unit retrieves an instruction from storage. The op code byte of the instruction is sent to the op register, the Q byte is sent to the Q register, and the operand addresses are developed and sent to the address LSR's.

During the execution phase, the instruction just retrieved from storage is executed to perform the desired operation. The data contained in the operands is retrieved from storage and examined, moved, or modified as directed by the instruction.

The time interval in which the processing unit reads one byte from storage and writes one byte into storage is known as a cycle. The processing unit must perform at least three cycles for each instruction (3 bytes, 1 cycle per byte). Cycles (accesses to storage) can also be taken by the I/O units.

Cycles are designated by the phase in which they occur and the type of operations performed in them as follows:

Cycle	*Operation*
I-Op	The op code is moved from storage to the op code register.
I-Q	The Q byte is moved from storage to the Q register.
I-R	Third instruction cycle when the instruction uses no address.
I-X1	Establishes the first operand address in BAR when the first operand uses base-displacement addressing.
I-H1	Establishes the high-order byte of the first operand address in the high-order byte of BAR when the first operand is directly addressed.
I-L1	Establishes the low-order byte of the first operand address in the low-order byte of BAR when the first operand is directly addressed.
I-X2	Establishes the second operand address in AAR when the second operand uses base-displacement addressing.
I-H2	Establishes the high-order byte of the second operand address in the high-order byte of AAR when the second operand is directly addressed.
I-L2	Establishes the low-order byte of the second operand address in the low-order byte of AAR when the second operand is directly addressed.
E-A	Moves a byte of the second operand from storage to the data recall register.
E-B	Moves a byte of the first operand from storage, operates on it, and returns it to storage.
I/O	Moves a byte of data between storage and an input/output unit.

CYCLE STEALING

The system operates in a mode known as cycle stealing. Cycle stealing is a mode of operation in which I/O operations are overlapped with processing operations so that processing operations can continue while I/O operations are in process. This is accomplished by allowing I/O units to *steal* processing unit cycles between processing cycles. The processing unit must check the I/O units to determine when an I/O operation is completed and the input data can be used or the output data can be replaced with new data.

INTERRUPT

Certain I/O units require special subroutines to handle data entered by them within a limited period of time or for other similar reasons. To provide for these special subroutines, a system of interruptions is installed. This interrupt system permits the processing unit to change state as a result of a condition external to the system. External conditions encountered in the system originate at an I/O device that has requested special attention by the processing unit. Generally, an interrupt implies that the processing unit must interrupt a current instruction sequence, perform an intervening instruction sequence requested by the interrupting I/O device, and return to the interrupted program.

A means of retaining the stopping point of an interrupted program and the starting point of an intervening program is important. The system provides an instruction address register and an address recall register for each level of interrupt.

Interrupt Priorities

There are four levels of interrupt in the system. I/O devices and their interrupt levels, in order of descending priority are:

Serial I/O channel	Level 4
Unassigned	Level 3
Binary synchronous communication adapter	Level 2
Keyboard-console	Level 1

Any level of interrupt can interrupt the main program or any lower level of interrupt.

Interrupt Operation

An I/O device requests an interrupt as the result of a previously issued start I/O instruction. When an I/O unit requests an interrupt, the processing unit completes the instruction in progress, then begins retrieving instructions from storage, using the interrupting program's instruction address register. The interrupted program's instruction address register and address recall register remain intact. The interrupting program is responsible for storing and restoring index registers 1 and 2 and the program status register for the interrupted program. The end of the interrupt routine is signalled by a start I/O instruction telling the interrupting device to reset its request. Figure 2-3 shows the recommended generalized interrupt routine.

INPUT/OUTPUT FACILITIES

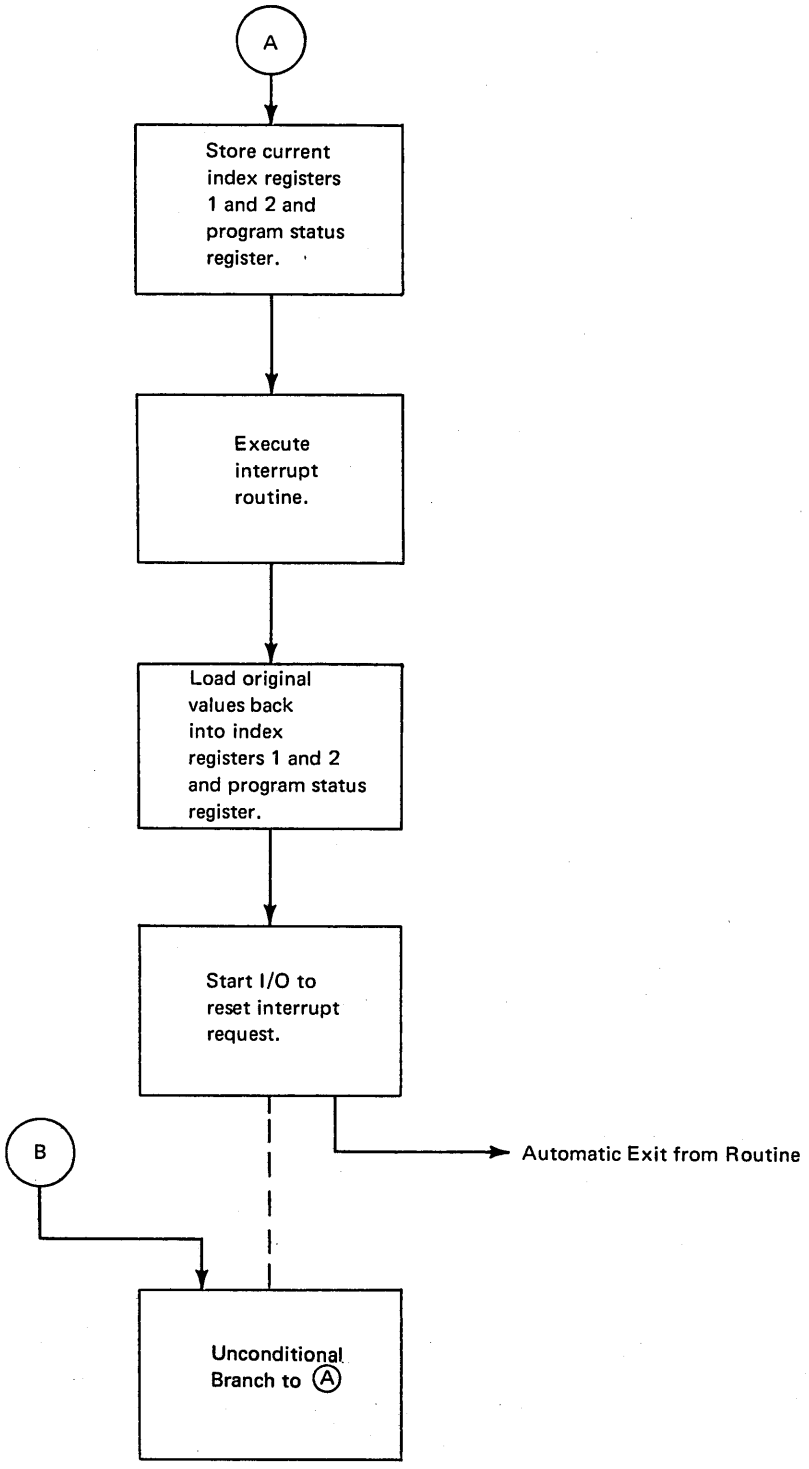
The processing unit acts as a controller for all I/O devices operating over a single I/O attachment interface. The I/O devices time-share the processing unit according to defined priorities established for each device.

The processing unit communicates with the I/O devices via an interface called the input/output channel. The I/O channel consists of:

1. A set of signal lines that carry information to and from the processing unit.
2. Logic to establish cycle steal and interruption priorities and to translate card code data into EBCDIC and EBCDIC to card code.

Channel Organization

The channel serves as a data, status, and instruction path between the processing unit and the I/O attachment circuits of the attached I/O devices. The I/O device attachments are integral with the processing unit. All I/O attachments are connected to the same set of signal lines in the channel. Thus, the recognition of its own address by a device is the only indication a device has that its services are required.



Note:

The interrupt instruction address register must be set to the address of (A) or (B) before the first interrupt occurs. Normally, the processing unit leaves the interrupt instruction address register set to the address of (B) at the end of the interrupt routine.

Figure 2-3. Interrupt Routine

Device Control

All devices respond to the following instructions:

1. Start I/O
2. Load I/O
3. Sense I/O
4. Test I/O
5. Advance program level

Each of these instructions carries within itself the address of the device that is to perform the operation and the exact operation to be performed. The individual formats of these instructions are discussed in the chapters dealing with the individual I/O devices.

The channel operates in either of two modes, depending on the device to be operated. The interruption mode is discussed earlier in this chapter. The second mode of operation is the cycle steal mode. In this mode of operation the I/O device is started by a start I/O instruction, then is left to perform its operations until access to main storage is required. When main storage is needed, the I/O device is allowed to steal one or more cycles from other processing unit operations in order to store or retrieve from storage the necessary data. The processing unit then continues to perform other operations until the I/O unit requires main storage cycles again.

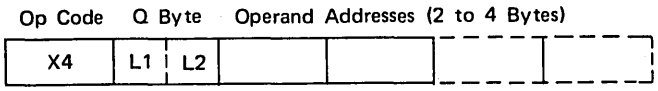
INSTRUCTIONS AND PROGRAMMING CONVENTIONS

Because the instruction format is so variable and the length of the instruction is determined by the high-order hexadecimal digit of the op code byte, the following conventions are used in discussing the instructions:

1. A high-order digit of X in the op code designates a two-address type of instruction. The actual high-order hexadecimal digit of the op code can be 0, 1, 2, 4, 5, 6, 8, 9, or A.
2. A high-order digit of Y in the op code designates a one-address instruction addressing operand 1. The actual high-order hexadecimal digit of the op code can be 3, 7, or B.
3. A high-order op code digit of Z designates a one-address instruction addressing operand 2. The actual high-order hexadecimal digit in the op code can be C, D, or E.
4. A high-order op code digit of F designates a command type instruction and is the true high-order hexadecimal digit of that op code.
5. Op codes are expressed in hexadecimal notation. Q codes may be expressed in either hexadecimal or binary notation or may have symbols to indicate the significance of the individual bits or groups of bits of the Q byte.
6. Minimum length of the instruction is shown in solid blocks; maximum and intermediate lengths are indicated by dotted blocks attached to the minimum length blocks.

Zero and Add Zoned

Mnemonic: ZAZ



Operation: The second operand is placed byte by byte in the first operand starting with the rightmost byte of each operand. Extra high-order zeros are inserted into those positions by which the first operand exceeds the second in length. The zone bits in each byte of the result except the rightmost are set to all ones. The zone bits of the rightmost byte of the first operand are set to all ones if the result is positive or zero. If the result is negative, the zone bits of the rightmost byte are set to 1101. If the second operand is negative, the result is in complement form, and an automatic recomplement operation is performed to convert the result to a positive number.

The first and second operand fields may overlap when the rightmost byte of the first operand is coincident with or to the right of the rightmost byte of the second operand.

The Q byte designates the length of the two operands. L2 is 1 less than the number of bytes in the second operand. L1 is the number of bytes by which the length of the first operand exceeds the length of the second operand. L1 and L2 are expressed in binary notation. The maximum length of operand 2 is 16 bytes. The maximum length of operand 1 is 31 bytes.

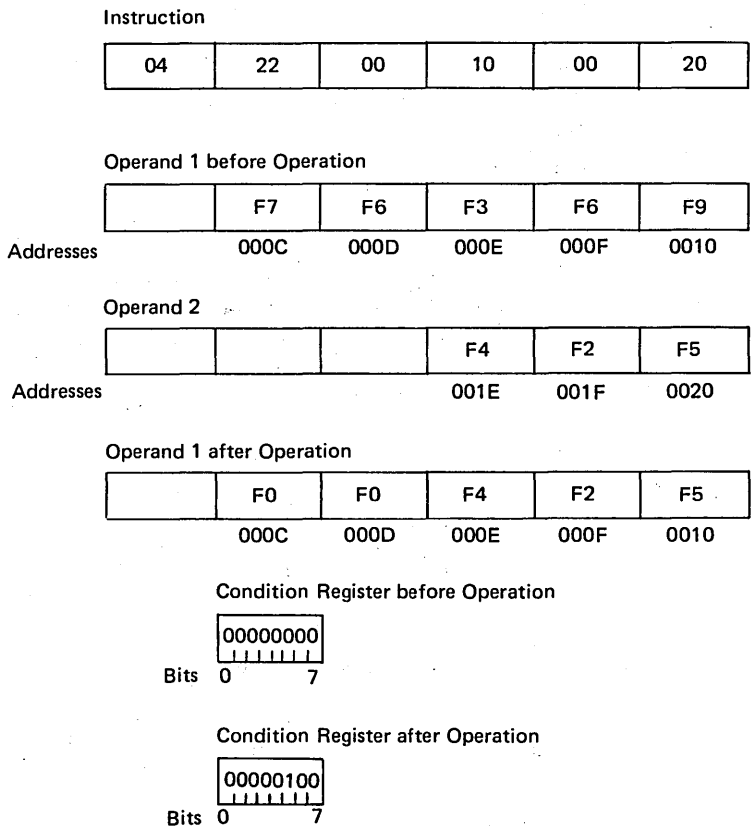
The second operand remains unchanged unless the fields overlap. No check is made for valid decimal digits in either operand.

Resulting Condition Register Settings:

Equal	Result is zero.
Low	Result is negative.
High	Result is positive.
Decimal overflow	Not affected.
Test false	Not affected.
Binary overflow	Not affected.

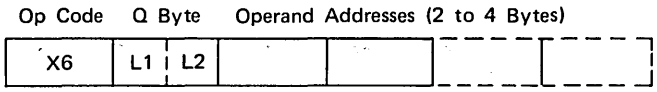
Program Note: Overlapping the operands with the rightmost byte of the first operand to the left of the rightmost byte of the second operand results in destroying part of the second operand before it takes part in the operation.

Example:



Add Zoned Decimal

Mnemonic: AZ



Operation: The second operand is added algebraically to the first operand, byte by byte, and the result is stored in the first operand. The operands are addressed by their rightmost bytes. The zone bits of all except the rightmost byte of operand 1 are set to all ones. The zone bits of the rightmost byte are set to all ones if the result is positive or zero. If the result is negative, the zone bits of the rightmost byte of operand 1 are set to 1101.

The Q byte specifies the length of the operands. L2 is one less than the length in bytes of the second operand. L1 is the number of bytes by which the length of the first operand exceeds the length of the second operand. The maximum length of operand 2 is 16 bytes. The maximum length of operand 1 is 31 bytes.

The first and second operand fields may overlap if the rightmost byte of the first operand is coincident with or to the right of the rightmost byte of the second operand.

The second operand remains unchanged unless the fields overlap.

If operand 2 is negative and larger than operand 1, the result is in complement form and an automatic recomplement operation is performed to produce a first operand in true form.

No check is made for valid decimal digits in either operand.

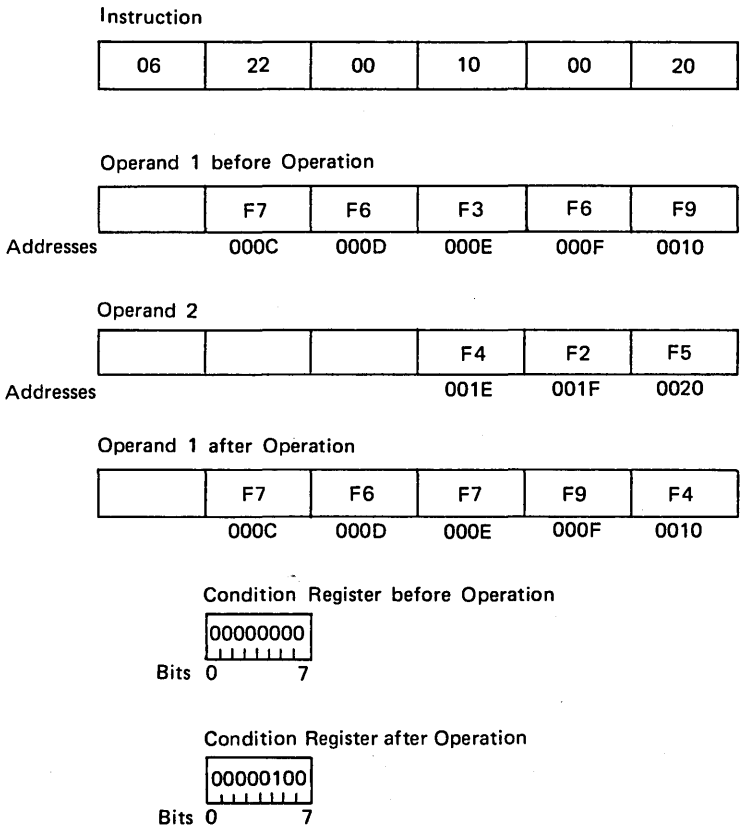
Resulting Condition Register Settings:

Equal	Result is zero.
Low	Result is negative.
High	Result is positive.
Decimal overflow	Carry occurred from the high-order position of operand 1.
Test false	Not affected.
Binary overflow	Not affected.

Program Notes:

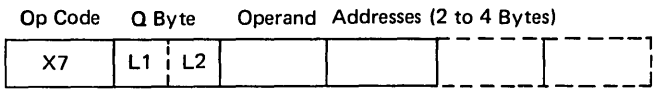
1. The decimal overflow condition code is reset only by system reset or by testing decimal overflow with a branch-on-condition or jump-on-condition instruction.
2. Overlapping the operands with the rightmost byte of the first operand to the left of the rightmost byte of the second operand results in destroying part of the second operand before it takes part in the operation.

Example:



Subtract Zoned Decimal

Mnemonic: SZ



Operation: Operand 2 is subtracted algebraically from operand 1, byte by byte, and the result is placed in operand 1. The operands are addressed by their rightmost bytes. The zone bits of all except the rightmost byte of operand 1 are set to all ones. The zone bits of the rightmost byte of operand 1 are set to all ones if the result is positive or zero. If the result is negative, the zone bits of the rightmost byte of operand 1 are set to 1101.

The Q byte specifies the length of the operands. L2 is one less than the number of bytes in the second operand. L1 is the number of bytes by which operand 1 exceeds the length of operand 2. The maximum length of operand 2 is 16 bytes. The maximum length of operand 1 is 31 bytes.

The first and second operand fields may overlap if the rightmost byte of the first operand is coincident with or to the right of the rightmost byte of the second operand.

The second operand remains unchanged unless the fields overlap.

If operand 2 is larger than operand 1 in a subtract operation, the result is in complement form. An automatic recomplement operation is performed to produce a first operand in true form.

No check is made for valid decimal digits in either field.

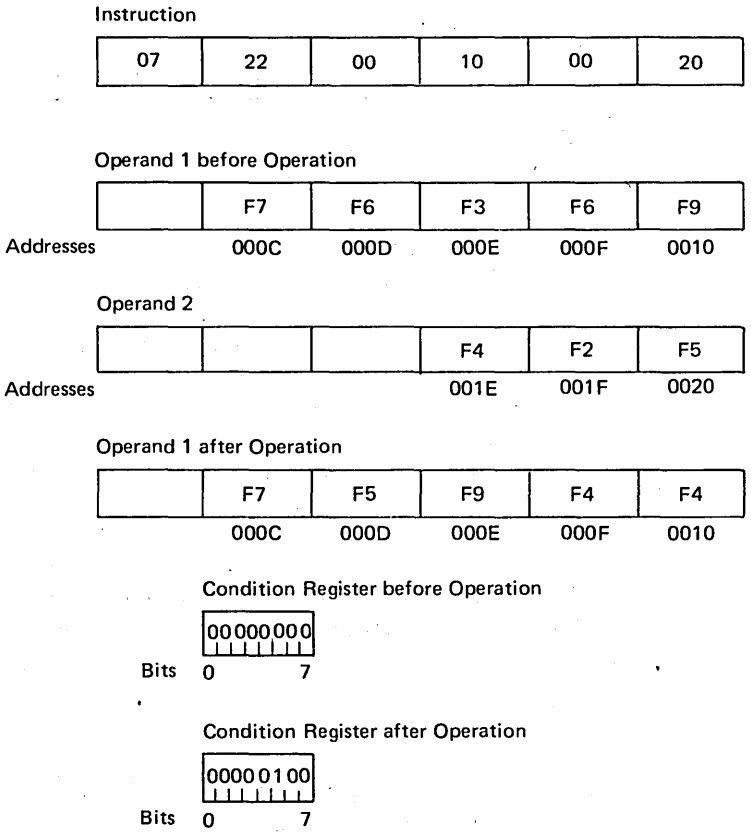
Resulting Condition Register Settings:

Equal	Result is zero.
Low	Result is negative.
High	Result is positive.
Decimal overflow	Carry occurred from the high-order position of operand 1.
Test false	Not affected.
Binary overflow	Not affected.

Program Notes:

1. The decimal overflow condition code is reset only by system reset or by testing decimal overflow with a branch-on-condition or jump-on-condition instruction.
2. Overlapping the operands with the rightmost byte of the first operand to the left of the rightmost byte of the second operand results in destroying part of the second operand before it takes part in the operation.

Example:



Add Logical Characters

Mnemonic: ALC

Op Code	Q Byte	Operand	Addresses (2 to 4 Bytes)		
XE	L				

Operation: The positive binary number contained in the bytes of operand 2 is added, byte by byte, to the positive binary number contained in the bytes of operand 1. The result is stored in operand 1. The operands are addressed by their rightmost bytes.

The Q byte specifies the length of the operands. L is one less than the length in bytes of either operand. Both operands must be the same length. The maximum length of the operands is 256 bytes.

The operands may overlap if the rightmost byte of operand 1 is coincident with or to the right of the rightmost byte of operand 2.

Operand 2 is not changed unless it overlaps operand 1.

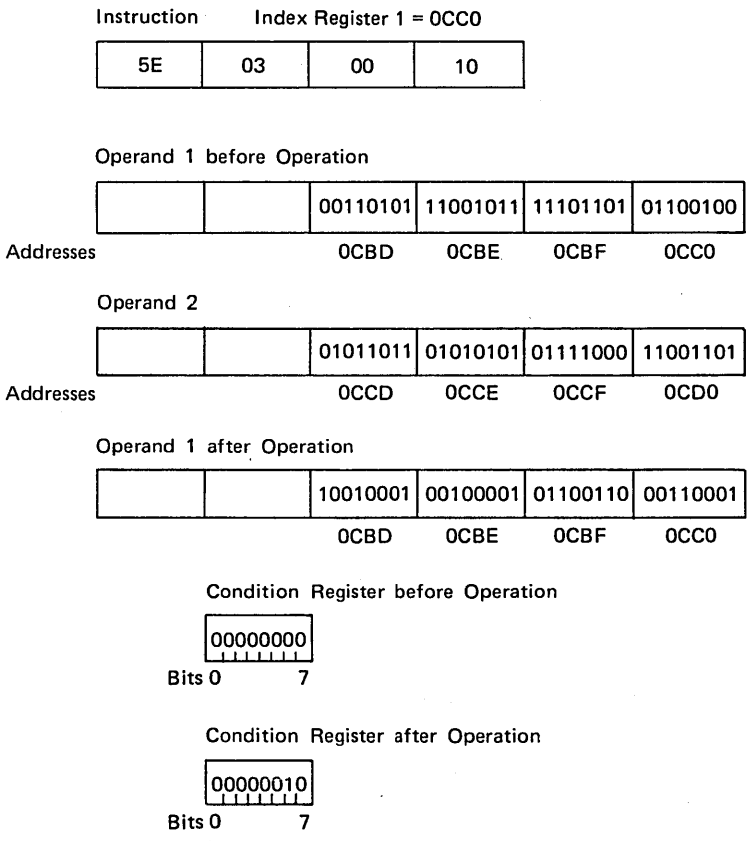
Resulting Condition Register Settings:

Equal	Result is zero.
Low	No carry occurred out of the high-order byte and the result is not zero.
High	Carry occurred out of the high-order byte and the result is not zero.
Decimal overflow	Not affected.
Test false	Not affected.
Binary overflow	Carry occurred out of the high-order byte.

Program Notes:

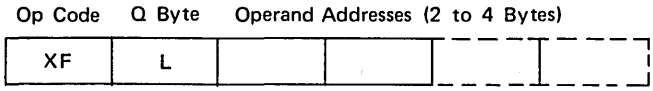
1. Binary overflow bit is reset during the instruction phase of this operation.
2. Overlapping the operands with the rightmost byte of the first operand to the left of the rightmost byte of the second operand results in destroying part of the second operand before it takes part in the operation.

Example:



Subtract Logical Characters

Mnemonic: SLC



Operation: The positive binary number contained in the bytes of the second operand is subtracted, byte by byte, from the positive binary number contained in the bytes of the first operand. The result is stored in the first operand. The operands are addressed by their rightmost bytes. If the second operand is larger than the first operand, the answer is developed as though the first operand had an additional high-order binary digit. The result can never be negative. For example:

First Operand	0110 1101
Second Operand	0111 1110
Result	1110 1111

The Q byte specifies the length in bytes of the operands. L is one less than the length of either operand. Both operands must be the same length. The maximum length of the operands is 256 bytes.

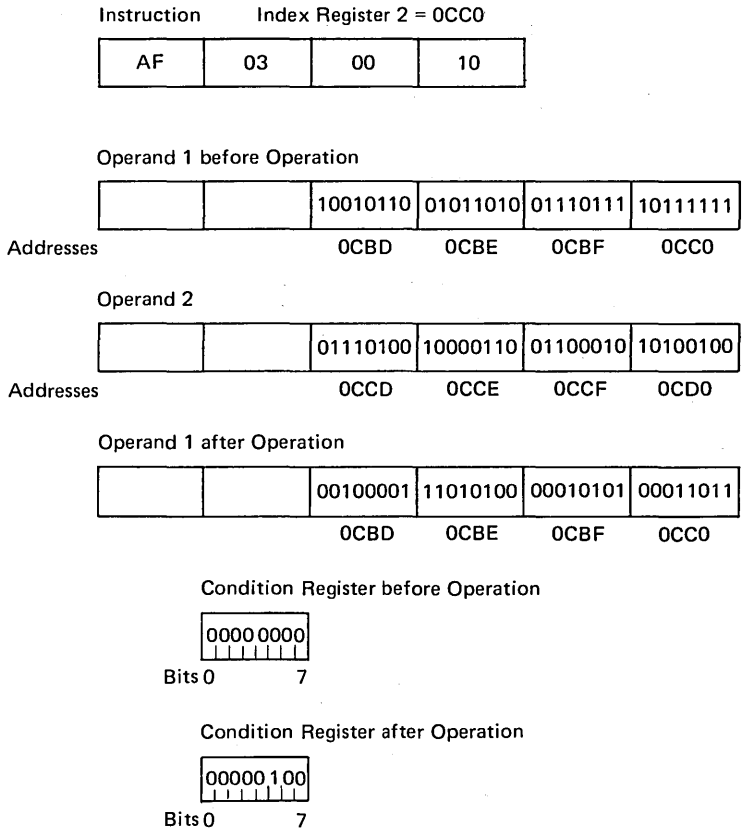
The operands can overlap if the rightmost byte of operand 1 is coincident with or to the right of the rightmost byte of operand 2. The second operand is not changed unless the operands overlap.

Resulting Condition Register Settings:

Equal	Result is zero.
Low	First operand is smaller than the second operand.
High	First operand is greater than the second operand.
Decimal overflow	Not affected.
Test false	Not affected.
Binary overflow	Not affected.

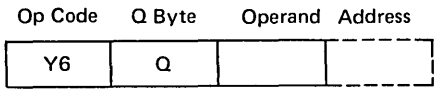
Program Note: Overlapping the operands with the rightmost byte of the first operand to the left of the rightmost byte of the second operand results in destroying part of the second operand before it takes part in the operation.

Example:


Instruction
Index Register 2 = 0CC0
AF
03
00
10
Operand 1 before Operation
10010110
01011010
01110111
10111111
Addresses
0CBD
0CBE
0CBF
0CC0
Operand 2
01110100
10000110
01100010
10100100
Addresses
0CCD
0CCE
0CCF
0CD0
Operand 1 after Operation
00100001
11010100
00010101
00011011
0CBD
0CBE
0CBF
0CC0
Condition Register before Operation
0000 0000
Bits 0
7
Condition Register after Operation
00000 100
Bits 0
7

Add to Register

Mnemonic: A



Operation: The positive binary number contained in the two-byte field addressed by the operand address is added to the contents of the two-byte register selected by the Q code. The result replaces the contents of the register. The operand is addressed by its rightmost byte and is not changed by the operation.

The Q code selects the register to be modified. The high-order bit (bit 0) of the Q code determines which of two groups of registers will be modified. The remaining bits of the Q code determine which specific register within the group will be modified. If bit 0 of the Q code is zero, the remaining bits cause modification of the registers as follows:

Bit	*Register*
1	Not used
2	Main program level instruction address register
3	Instruction address register in use when the add-to-register instruction is executed
4	Address recall register
5	Program status register
6	Index register 2
7	Index register 1

If the high-order bit of the Q code is 1, the selected group is the four instruction registers for the four interrupt levels. The instruction address registers are selected by the remaining bits as follows:

Bit	*Interrupt Level*
1	Interrupt level 1
2	Interrupt level 2
3	Interrupt level 3
4	Interrupt level 4

This instruction must not be used to add to more than one register at a time. The result of attempting to add to two registers simultaneously can be either incorrect parity or incorrect results in the registers.

Resulting Condition Register Settings:

Equal	Result is zero.
Low	No carry occurred from the high-order byte and the result is not zero.
High	Carry occurred from the high-order byte and the result is not zero.
Decimal overflow	Not affected.
Test false	Not affected.
Binary overflow	Carry occurred from the high-order byte.

Program Notes:

1. Even though this instruction can modify the program status register, the contents of the condition register will be placed in the low-order byte of the program status register during I-phase of this instruction.
2. This instruction can be used to subtract from the value contained in a register by placing the two's complement of the value to be subtracted in the operand.

Example:

Instruction

36	00000010	00	04

Operand 1

01001000	00100000
0003	0004

Index Register 2

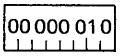
00110101	01101010

Before Operation

01111101	10001010

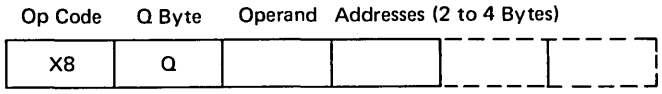
After Operation

Condition Register after Operation



Move Hexadecimal Character

Mnemonic: MVX



Operation: The numeric (low-order 4 bits) portion or the zone (high-order 4 bits) portion of the single-byte second operand is placed in the numeric portion or zone portion of the single-byte first operand. The second operand is not changed, unless both operands address the same byte.

The Q code specifies which portion of each operand is to be used in the operation.

Hexadecimal Value of Q Code	*Operand 2*	*Operand 1*
00	Zone	to zone
01	Numeric	to zone
02	Zone	to numeric
03	Numeric	to numeric

The high-order 6 bits of the Q byte should all be zeros.

Resulting Condition Register Settings: The condition register is not affected by this instruction.

Example:

Instruction

98	01	A0	65

Index Register 1 = 2B15
Index Register 2 = 1F20

Operand 1 before Operation

2F

1FC0

Operand 2

4C

2B7A

Operand 1 after Operation

CF

1FC0

Move Characters

Mnemonic: MVC

Op Code	Q Byte	Operand Addresses (2 to 4 Bytes)			
XC	L				

Operation: The second operand is placed byte by byte in the first operand location. The second operand is not changed. The operands are addressed by their rightmost bytes. One character can be propagated through an entire field by setting the operand 1 address one byte to the left of the operand 2 address. Operand 2 is not changed unless the fields are overlapped.

The Q code specifies the length of the operands. L is one less than the length in bytes of either operand. Both operands must be the same length. The maximum length of the operands is 256 bytes.

Resulting Condition Register Settings: This instruction does not affect the condition register.

Example:

Instruction

0C	06	1A	06	2B	5A

Operand 1 before Operation

	D1	C1	D4	C5	E2	
Addresses	1A01	1A02	1A03	1A04	1A05	1A06

Operand 2

	D9	D6	C2	C5	D9	E3
Addresses	2B55	2B56	2B57	2B58	2B59	2B5A

Operand 1 after Operation

D9	D6	C2	C5	D9	E3
1A01	1A02	1A03	1A04	1A05	1A06

Edit

Mnemonic: ED

Op Code	Q Byte	Operand Addresses (2 to 4 Bytes)			
XA	L1				

Operation: The decimal numeric characters in the second operand replace the bytes containing hexadecimal 20 in the edit pattern contained in the first operand. All characters other than hexadecimal 20 in the edit pattern remain unchanged. The zone bits of all the replaced characters are set to all 1's. The result of the edit operation occupies the first operand; the second operand is not changed. The operands are addressed by their rightmost bytes. The operands cannot be overlapped.

The Q byte specifies the length of operand 1. L1 is one less than the length in bytes of operand 1. Operand 2 contains the same number of bytes as the number of hexadecimal 20 characters in operand 1.

Resulting Condition Register Settings:

Equal	Second operand is zero.
Low	Second operand is negative.
High	Second operand is positive.
Decimal overflow	Not affected.
Test false	Not affected.
Binary overflow	Not affected.

Example:

Instruction

0A	0A	00	BF	00	07

Operand 1 before Operation

$	20	,	20	20	20	.	20	20	ъ	*
00B5	00B6	00B7	00B8	00B9	00BA	00BB	00BC	00BD	00BE	00BF

Operand 2

0	1	0	8	0	R
0002	0003	0004	0005	0006	0007

Note: "R" represents "-9"

Operand 1 after Operation

$	0	,	1	0	8	.	0	9	ъ	*
00B5	00B6	00B7	00B8	00B9	00BA	00BB	00BC	00BD	00BE	00BF

Note: Location 00BD contains a 9 because the zone bits of all replaced characters (zeros) in the edit pattern are set to all ones.

Condition Code

0	0	0	0	0	0	1	0

Bits 0 7

Insert and Test Characters

Mnemonic: ITC

Op Code	Q Byte	Operand Addresses (2 to 4 Bytes)			
XB	L1				

Operation: The single character at the second operand address replaces all the characters to the left of the first significant digit in the first operand. Only the digits 1 through 9 are significant. The first operand is addressed by the leftmost byte that can contain a character that should be replaced. (For example, if the high-order byte of the field for the first operand contains a dollar sign, the first operand address is the address of the byte to the right of the dollar sign.) The operation proceeds from left to right. Filling operand 1 with the character from operand 2 or encountering a significant digit in operand 1 ends the operation.

The Q byte specifies the length in bytes of operand 1. L1 is one less than the number of bytes in operand 1 from the first byte addressed to the end of the field. The second operand is a single byte.

At the end of this operation, the address of the first significant digit is placed in the address recall register. If no significant digit is found, the address of the byte to the right of the first operand is placed in the address recall register. The address recall register can be changed again only by a decimal add, decimal subtract, branch, or insert-and-test-character instruction.

Resulting Condition Register Settings: This instruction does not affect the condition register.

Example:

Instruction

0B	09	00	B6	00	10

Operand 1 before Operation

$	0	,	1	0	8	.	0	9	ъ	*
00B5	00B6	00B7	00B8	00B9	00BA	00BB	00BC	00BD	00BE	00BF

Operand 2

*
0010

Operand 1 after Operation

$	*	*	1	0	8	.	0	9	ъ	*
00B5	00B6	00B7	00B8	00B9	00BA	00BB	00BC	00BD	00BE	00BF

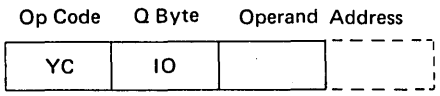
Note: Address 00B5 was not included in the first operand.

Address Recall Register after Operation

00	B8

Move Logical Immediate

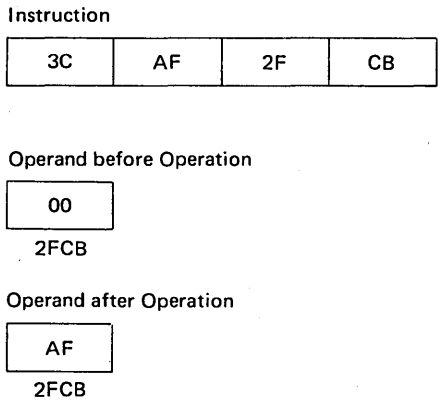
Mnemonic: MVI



Operation: The data contained in the Q byte of the instruction is moved into the byte located at the operand address.

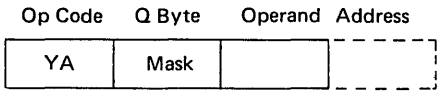
Resulting Condition Register Settings: This instruction does not affect the condition register.

Example:



Set Bits On Masked

Mnemonic: SBN



Operation: The byte of data contained in the mask byte (M) is used to set to one the corresponding bits in the byte located at the operand address. Any bits in the operand that are already set to one remain set to one. A mask bit equal to one indicates that the corresponding operand bit is to be set to one. A mask bit equal to zero indicates that the corresponding bit is to remain unchanged.

Resulting Condition Register Settings: This instruction does not affect the condition register.

Example:

Instruction

3A	01011010	00	20

Operand before Operation

00001100

0020

Operand after Operation

01011110

0020

Set Bits Off Masked

Mnemonic: SBF

Op Code	Q Byte	Operand Address	
YB	Mask		

Operation: The data contained in the mask byte (M) is used to set to zero the corresponding bits of the byte located at the operand address. Any bits in the operand that are already set to zero remain zero. A mask bit equal to one indicates that the corresponding operand bit is to be set to zero. A mask bit equal to zero indicates that the corresponding operand bit is to remain unchanged.

Resulting Condition Register Settings: This instruction does not affect the condition register.

Example:

Instruction

3B	10000001	00	30

Operand before Operation

01111001

0030

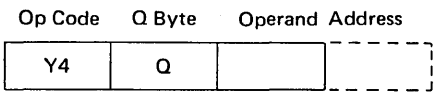
Operand after Operation

01111000

0030

Store Register

Mnemonic: ST



Operation: The contents of the two-byte register specified by the Q code are placed in the two-byte field addressed by the operand address. The operand is addressed by its rightmost byte.

The Q byte specifies the register to be stored. The high-order bit of the Q byte, bit 0, specifies which of two groups of registers is to be addressed. The low-order bits specify which register within each group is to be stored.

If the high-order bit is zero, the selected group consists of the following seven local storage registers, each represented by a single bit.

Bit	*Register*
1	Not used
2	Main program level instruction address register
3	Instruction address register in use when the store register instruction is executed
4	Address recall register
5	Program status register. The high-order byte of this register is the length count recall register and has no program significance. The low-order byte is the image of the condition register.
6	Index register 2
7	Index register 1

If the high-order bit of the Q code is one, the interrupt instruction address registers are selected as follows:

Bit	*Interrupt Level*
1	Interrupt level 1
2	Interrupt level 2
3	Interrupt level 3
4	Interrupt level 4

Program Note: This instruction must not be used to store more than one register at a time. The attempt to store more than one register at a time can result in either incorrect parity and a parity check or incorrect data in the specified registers at the end of the operation.

Resulting Condition Register Settings: This instruction does not affect the condition register.

Example:

Instruction

34	00001000	32	BB

Address Recall Register

0A	CD

Operand before Operation

2F	C2
32BA	32BB

Operand after Operation

0A	CD
32BA	32BB

Load Register

Mnemonic: L

Op Code	Q Byte	Operand	Address
Y5	Q		

Operation: The contents of the two-byte field addressed by the operand address are placed in the local storage register specified by the Q byte. The operand is addressed by its right-most byte. The operand is not changed.

The Q byte specifies the register to be loaded. The high-order bit (bit 0) of the Q byte specifies which of two groups of registers is to be loaded.

If the high-order bit of the Q byte is zero, the selected group consists of the following seven local storage registers, each represented by a single bit.

Bit	*Register*
1	Not used
2	Main program level instruction address register
3	Instruction address register in use when the load register instruction is executed
4	Address recall register
5	Program status register. The high-order byte of this register is the length count recall register and has no program significance. The low-order byte of this register holds a condition code and is loaded under special conditions described in the programming notes for this instruction.
6	Index register 2
7	Index register 1

If the high-order bit of the Q byte is one, the interrupt instruction address registers are selected as follows:

Bit	*Interrupt Level*
1	Interrupt level 1
2	Interrupt level 2
3	Interrupt level 3
4	Interrupt level 4

Program Notes:

1. This instruction must not be used to load more than one register at a time. The attempt to load more than one register can result in incorrect register contents.
2. When the program status register is selected, the contents of the low-order byte of the operand has the following significance: bit 7 equals 1, set equal condition; bit 7 equals 0 and bit 6 equals 1, set low condition; bit 7 equals 0 and bit 6 equals 0, set high condition; bit 4 equals 1, set decimal overflow condition; bit 3 equals 1, set test false condition; bit 2 equals 1, set binary overflow condition. When bit 7 of the operand equals 0, bit 6 controls setting bit 5 of the program status register low-order byte. Bit 5 is set to 1 when bit 6 equals 0 and is set to 0 when bit 6 equals 1. Bits 0, 1, and 5 of the operand are ignored. The condition register is set at the same time as the program status register under the same controls.
3. If the main program level has been halted and this instruction is used by an interrupt routine to load the main program level instruction address register, the main program level is reset from the halt state and will proceed at the new address when the interrupt routine exits to the main program level. The main program level halt indicators are turned off.
4. An unconditional branch can be performed without disturbing the address recall register by using this instruction to load the instruction address register.

Resulting Condition Register Settings: This instruction does not affect the condition register setting unless the program status register is being loaded.

Example:

Instruction

35	00000100	00	11

Operand

00000000	00000000
0010	0011

Program Status Register before Operation

00001100	00110001

Program Status Register after Operation

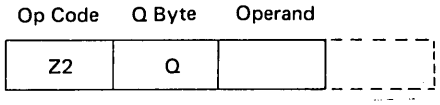
00000000	00000100

Condition Register after Operation

00000100
Bits 0 7

Load Address

Mnemonic: LA



Operation: If the instruction is in the four-byte format (op code C2), the two-byte operand is taken from the instruction stream and loaded into the register specified by the Q byte.

If the instruction is in the three-byte format (op code D2 or E2), the single-byte operand is taken from the instruction stream and added to the contents of the index register specified by the op code. The result of this addition is loaded into the register specified by the Q byte. Only index registers can be loaded with this instruction. Bits 6 and 7 of the Q code specify which index register to load as follows:

Bit	*Register*
6	Index register 2
7	Index register 1

Program Note: This instruction must not be used to load both index registers at the same time. The attempt to load both registers can result in incorrect data in the register.

Resulting Condition Register Settings: This instruction does not affect the condition register.

Example:

Instruction

D2	02	05

Index Register 1

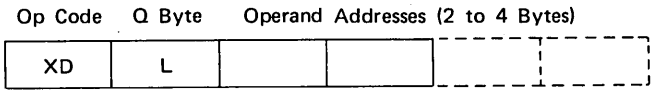
2A	15

Index Register 2 after Operation

2A	1A

Compare Logical Characters

Mnemonic: CLC



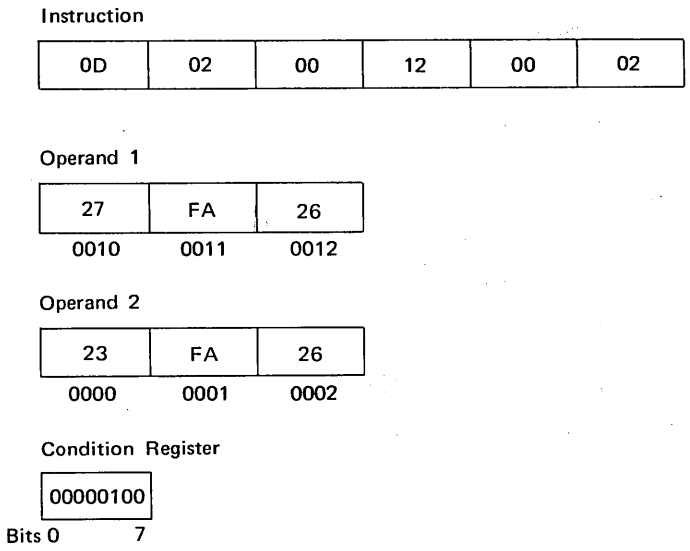
Operation: The first operand is compared with the second operand, byte by byte, and the condition register is set according to the result of the comparison. Each operand is treated as a binary quantity. Each operand is addressed by its rightmost byte. Neither operand is changed as a result of this operation.

The Q byte specifies the length of the operand. L is one less than the length in bytes of either operand. Both operands are the same length.

Resulting Condition Register Settings:

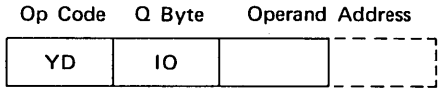
Equal	Operands are equal.
Low	First operand is smaller than the second operand.
High	First operand is greater than the second operand.
Decimal overflow	Not affected.
Test false	Not affected.
Binary overflow	Not affected.

Example:



Compare Logical Immediate

Mnemonic: CLI

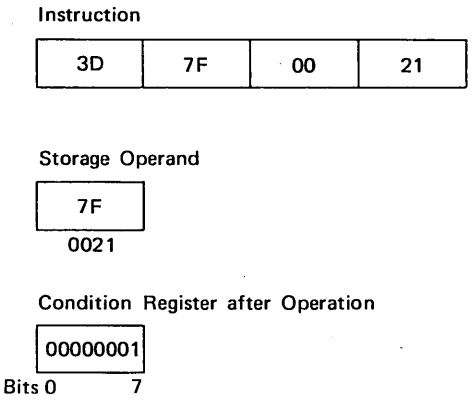


Operation: The binary immediate operand contained in the Q byte is compared with the binary operand in storage located at the operand address. The result sets the condition register. Neither operand is changed as a result of this operation.

Resulting Condition Register Settings:

Equal	Operands are equal.
Low	Storage operand is smaller than the immediate operand.
High	Storage operand is greater than the immediate operand.
Decimal overflow	Not affected.
Test false	Not affected.
Binary overflow	Not affected.

Example:



Test Bits On Masked

Mnemonic: TBN

Op Code	Q Byte	Operand Address	
Y8	Mask		

Operation: The bits of the storage operand located at the operand address are tested for bit-equal-1 as defined by the mask contained in the Q byte. A mask bit equal to 1 indicates that the corresponding storage operand bit is to be tested; a mask bit equal to 0 indicates that the corresponding storage operand bit is to be ignored. The result of the test controls the setting of the condition register. The storage operand is not changed.

Resulting Condition Register Settings:

Equal	Not affected.
Low	Not affected.
High	Not affected.
Decimal overflow	Not affected.
Test false	Turned on if any of the designated bits is not equal to 1.
Binary overflow	Not affected.

Program Notes:

1. If the mask is all zeros, the test false condition cannot be turned on.
2. Test false condition can be turned off only by a system reset or by using test false as a condition in a branch-on-condition instruction.

Example:

Instruction

38	00010110	00	21

Storage Operand

10010101

0021

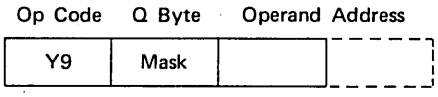
Condition Register after Operation

00010000

Bits 0 7

Test Bits Off Masked

Mnemonic: TBF



Operation: The bits of the storage operand located at the operand address are tested for bits equal to 0 as defined by the mask contained in the Q byte. A mask bit equal to 1 indicates that the corresponding storage operand bit is to be tested; a mask bit equal to 0 indicates that the corresponding storage operand bit is to be ignored. The result of the test controls setting the condition register. The storage operand is not changed.

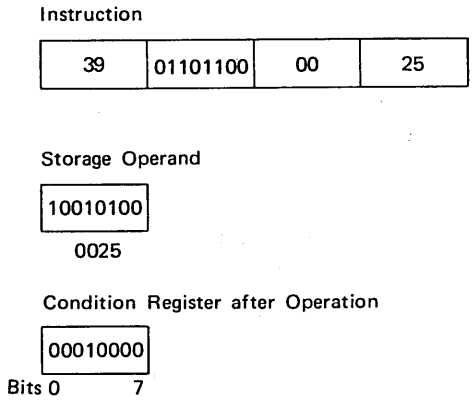
Resulting Condition Register Settings:

Equal	Not affected.
Low	Not affected.
High	Not affected.
Decimal overflow	Not affected.
Test false	Turned on if any tested bit is not zero.
Binary overflow	Not affected.

Program Notes:

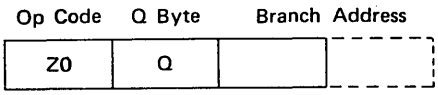
1. If the mask is all zeros, the test false condition cannot be turned on.
2. Test false condition can be turned off only by a system reset or by using test false as a condition in a branch-on-condition instruction.

Example:



Branch On Condition

Mnemonic: BC



Operation: The condition register is tested under control of the Q byte. If the condition register satisfies the condition or conditions established by the Q code, the next instruction is taken from the branch address; and the next sequential instruction address is placed in the address recall register. If the condition or conditions are not satisfied, the next instruction is taken from the next sequential instruction address contained in the instruction address register; and the branch address is placed in the address recall register. The address placed in the address recall register is retained there until a decimal add, decimal subtract, test-I/O-and-branch, insert-and-test-characters, or another branch-on-condition instruction is executed.

The Q byte defines what condition or what conditions are to be tested in the condition register and whether the branch is to occur on condition true (condition register bit equal to 1) or condition false (condition register bit equal to 0). When bit 0 of the Q byte equals 1, the branch occurs on condition true; when bit 0 equals 0, the branch occurs on condition false.

Bits 2 through 7 of the Q byte define the condition register bits to be tested. More than one condition code bit can be tested at the same time. The Q code bits and the conditions tested are:

Bit	*Condition Tested*
2	Binary overflow
3	Test false
4	Decimal overflow
5	High
6	Low
7	Equal

When bit 0 equals 1 (condition true), if any of the conditions tested is one, the branch occurs. When bit 0 equals 0 (condition false), the branch occurs if all of the conditions tested are zero.

Resulting Condition Register Settings:

Equal	Not affected.
Low	Not affected.
High	Not affected
Decimal overflow	Turned off if tested, otherwise, not affected.
Test false	Turned off if tested, otherwise, not affected.
Binary overflow	Not affected.

Program Notes:

1. The Q code 80, X7, or XF (where X equals 0 through 7), causes the branch operation to perform as a no-op.
2. An unconditional branch occurs when the Q byte contains 00, X7, or XF (where X equals 8 through F).

Example:

Instruction

C0	10001000	02	BF
0BCC	0BCD	0BCE	0BCF

Condition Code before Operation

00011001

Instruction Address Register after Operation

02	BF

Address Recall Register after Operation

0B	D0

Condition Register after Operation

00010001

Bits 0 7

Jump On Condition

Mnemonic: JC

Op Code	Q Byte	Control Code
F2	Q	

Operation: The condition register is tested under control of the Q Code. If the condition register satisfies the condition or conditions established by the Q code, the one-byte control code is added to the value in the instruction address register (the address of the next sequential instruction), and the sum becomes the address of the next instruction.

The Q byte defines the condition or conditions that are to be tested in the condition register. Bit 0 of the condition register specifies a condition true jump (tested bit equals 1), or a condition false jump (tested bit equals 0).

When bit 0 of the Q byte equals 1, the jump occurs on condition true; when bit 0 equals 0, the jump occurs on condition false.

Bits 2 through 7 of the Q byte define the condition register bits to be tested. More than one condition register bit can be tested at the same time. The Q byte bits and the conditions tested are:

Bit	*Condition Tested*
2	Binary overflow
3	Test false
4	Decimal overflow
5	High
6	Low
7	Equal

Under condition true (bit 0 equal to 1) testing, the jump occurs if any of the conditions equal one. Under condition false (bit 0 equal to 0) testing, the jump occurs if all of the conditions tested equal zero.

Resulting Condition Register Settings:

Equal	Not affected.
Low	Not affected.
High	Not affected.
Decimal overflow	Turned off if tested, otherwise, not affected.
Test false	Turned off if tested, otherwise, not affected.
Binary overflow	Not affected.

Program Notes:

1. The Q code 80, X7, or XF (where X equals 0 through 7) causes the operation to perform as a no-op.
2. An unconditional jump occurs when the Q code is 00, X7, or XF (where X equals 8 through F).

Example:

Instruction

F2	00110000	0F
0BBD	0BBE	0BBF

Condition Register before Operation

00001001

Instruction Address Register after Operation

0B	CF

Condition Register after Operation

00001001

Bits 0 7

Halt Instructions

Halt Program Level

Mnemonic: HPL

Op Code	Halt Identifier	
F0	Alpha Code	Numeric Code

Operation: This instruction prevents the execution of the next sequential instruction and displays a halt identifier in the display unit on the console keyboard. The display unit consists of nine indicators as follows: four alphabetic indicators labeled A, B, C, and D and five numeric indicators labeled 1, 2, 3, 4, and 5. The indicator lamps are individually controlled by the bits in the halt indicator bytes as follows:

Bit	*Alpha Indicator Lighted*	*Numeric Indicator Lighted*
0	Reserved	Reserved
1	A	1
2	B	2
3	C	3
4	D	4
5	Reserved	5
6	Reserved	Reserved
7	Reserved	Reserved

The halt operation is performed for any bit combinations of the identifier bytes including all zeros. The display unit is turned off (reset to blank) whenever the program is no longer in the halt operation.

Resulting Condition Register Settings: This instruction does not affect the condition register.

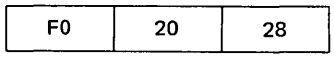
Program Notes:

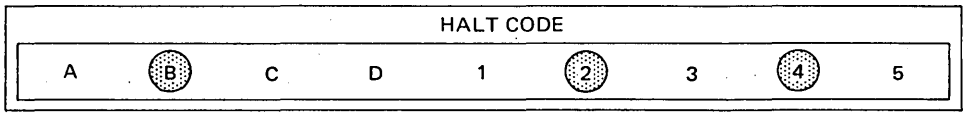
1. A halt program level instruction is functional only in the main program and operates as a no-op in interrupt level sequences.
2. The main program level can be stopped with a halt program level instruction to wait for an interrupt request. The interrupt routine can modify the main program instruction address register with a load register instruction to return to the main program at an instruction other than a halt instruction. The main program resumes operation and the display unit is turned off immediately after such a load register instruction is executed.
3. For checks requiring CE intervention, we recommend the following unique halt identifier combinations:

Hard equipment check	CD13
Error while logging I/O error on fixed disk	CD15
Disk file one	ABCD1
Disk file two	ABCD15
Keyboard	ABCD14
Printer	ABCD145
Ledger card device	ABCD13
Data recorder	ABCD135
CRT display station	ABCD134
Communication adapter	ABCD1345
Serial I/O adapter	ABCD12

The remaining combinations of ABCD1 should not be used.

Example:





Input/Output Instructions

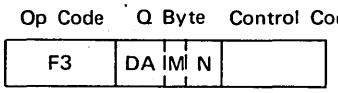
Five specific instructions are provided for I/O operations. The op code for these instructions is the same for all I/O devices. Bits 0 through 4 of the Q byte of these instructions provide a device address (DA) and modifier (M) bits. In the following table, both the device address (DA) and modifier (M) are shown because the modifier specifies the device in some cases.

Device	*Address*				
	DA				*M*
	0	*1*	*2*	*3*	*4*
Binary synchronous communication adapter	1	0	0	0	0
Data recorder	1	1	1	1	0
Disk storage drive:					
Drive 1 removable disk	1	0	1	0	0
Drive 1 fixed disk	1	0	1	0	1
Drive 2 removable disk	1	0	1	1	0
Drive 2 fixed disk	1	0	1	1	1
Display station	1	0	0	1	0
Keyboard-console	0	0	0	1	0
Magnetic character reader	0	0	1	1	0
Printer	1	1	1	0	0
Printer, ledger card device	1	1	1	0	1
Serial input/output channel adapter	0	0	1	1	0

Bits 5 through 7 of the Q byte provide an N code. These instructions and the specific bits that should occupy each of these Q byte bit positions are discussed under the specific I/O unit or operation that uses them.

Start I/O

Mnemonic: SIO



Operation: The operation of start I/O for each individual device is discussed under that device.

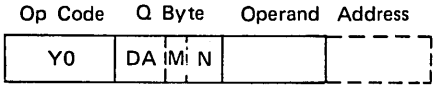
Resulting Condition Register Settings: This instruction does not affect the condition register.

Program Notes:

1. For some I/O units the start I/O instruction is no-oped if a unit check condition that prevents the execution of a start I/O instruction exists in the addressed device. No-oping the instruction results in setting a no-op status bit in the device.
2. A start I/O instruction that specifies the reset of an interrupt condition is executed regardless of any unit check condition in the addressed device.
3. Any unit check condition that does not prevent the execution of a start I/O instruction is reset by the start I/O instruction, and the instruction is executed.
4. A start I/O instruction addressed to a device that is busy or not ready results in a loop on that instruction until that device is ready or not busy .

Sense I/O

Mnemonic: SNS

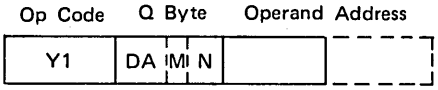


Operation: Data is taken from a source specified by the N code portion of the Q byte and placed in the two-byte field specified by the operand address. The operand is addressed by its rightmost byte.

Resulting Condition Register Settings: This instruction does not affect the condition register.

Load I/O

Mnemonic: LIO



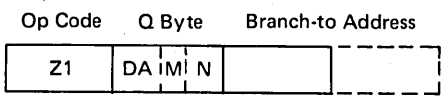
Operation: The contents of the two-byte field addressed by the operand address are transferred to a destination specified by the N code of the Q byte. The operand is addressed by its rightmost byte. A Q byte of 00 results in a no-op condition. If the no-op status bit for the addressed device is on when the load I/O instruction is executed, the instruction is no-oped.

Program Note: A load I/O instruction issued to a busy device causes the program to loop at the load I/O instruction until the device becomes not busy.

Resulting Condition Register Settings: The load I/O instruction does not affect the condition register.

Test I/O and Branch

Mnemonic: TIO

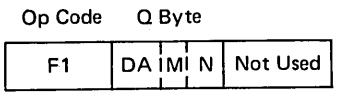


Operation: The condition specified by the Q byte is tested in the addressed device. If the condition is present, the branch-to address is placed in the instruction address register and the next sequential instruction address is placed in the address recall register. If the condition is not present, the next sequential instruction address is used and the branch-to address is placed in the address recall register. The address placed in the address recall register remains there until the next decimal add, decimal subtract, insert-and-test-characters, or branch instruction is executed.

Resulting Condition Register Settings: This instruction does not affect the condition register.

Advance Program Level

Mnemonic: APL



Operation: If the condition specified by the N code of the Q byte exists at the addressed device, the program loops on the advance program level instruction until the specified condition no longer exists at the device. The program then proceeds with the next sequential instruction. If the specified condition does not exist at the addressed device, the next sequential instruction is taken.

Program Note: The advance program level instruction should not be used to test for unit check. An advance program level instruction specifying unit check becomes a one-instruction loop when unit check exists. In this type of operation there would be no indication to the operator that the program was in a one-instruction loop and that a unit check condition existed.

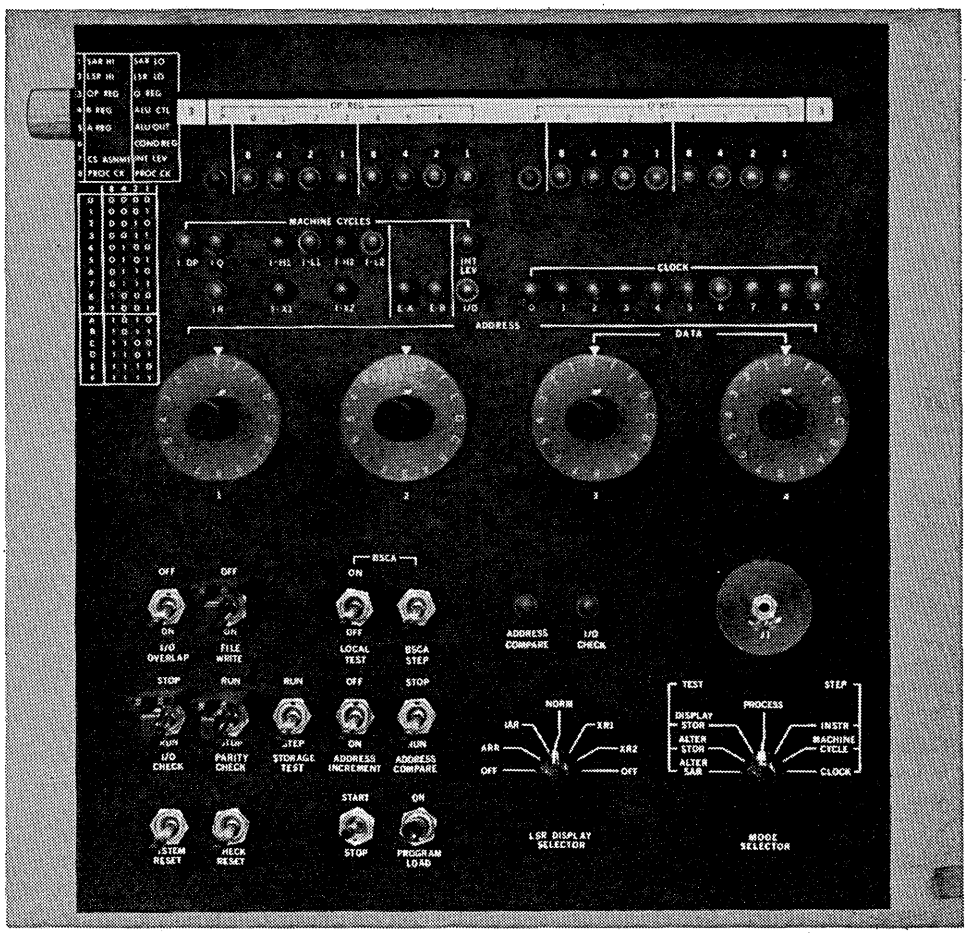
Resulting Condition Register Settings: This instruction does not affect the condition register.

CE CONTROL PANEL

The customer engineer (CE) control panel (Figure 2-4) is located behind the cover on the front of the processing unit. It contains the lights and switches necessary for customer engineering to maintain the system. Some of these lights and switches are useful for analyzing programming errors.

Address/Data Compare Switches

These switches are used in conjunction with controls on the CE panel to enter data into storage or to set up addresses for accessing storage. Each switch controls the setting of four bits in either storage or the storage address register.



BR0354A

Figure 2-4. Customer Engineer Control Panel

Register Display Unit

The register display unit consists of a row of twenty lights and eight legend strips mounted on an eight-position roller-type switch. Rolling the switch knob selects the legend strip and the register to be displayed. The legend strips display the following information:

Strip Number	*Display*
1	SAR HI and SAR LO–contents of the two-byte storage address register
2	LSR HI and LSR LO–contents of a local storage register selected by a switch on the CE panel
3	OP REG–contents of the op register Q REG–contents of the Q register
4	B REG–contents of the B register ALU CTL–state of all of the following ALU controls: DIG CAR Digital carry DEC Decimal instruction RECOMP Recomplement ADD Addition SUB Subtraction TEMP CAR Temporary carry AND Logical AND OR Logical OR
5	A REG–contents of the A register ALU OUT–output of the ALU
6	COND REG–contents of the condition register as follows: BIN OVF Binary overflow TF Test false DEC OVF Decimal overflow HI High LO Low EQ Equal
7	CS ASNMT–cycle steal assignment as it is presented to the I/O devices INT LEV–interrupt level indicating which I/O device is interrupting the program
8	PROC CHK–the following causes of processor checks are displayed. Most of these indications are useful only to the customer engineer, but some of them are useful in analyzing programming errors. I/O LSR–indicates that the selection of the LSR by an I/O device was not performed correctly. The CE LSR selector switch must be set to normal to obtain an indication of an LSR parity check. LSR F1–indicates that parity is incorrect on the output of the LSR's associated with certain optional features. LSR F2–this indicator is nonfunctional on Model 6. LSR HI–indicated that parity is incorrect on the high-order byte output LSR LO–indicates that parity is incorrect on the low-order byte output of the LSR's associated with the system with standard I/O devices. SAR HI–indicates that parity is incorrect in the high-order byte of the storage address register. SAR LO–indicates that parity is incorrect in the low-order byte of the storage address register. INV ADDR–indicates that the address contained in the storage address register is outside the address range of the system.

Strip Number	*Display*
8 (continued)	SDR–indicates that parity is incorrect in the storage data register. CAR–indicates that the carry out of the ALU is incorrect. DBI–parity is incorrect on data received from an I/O device. A/B–parity is incorrect in the A register or the B register. ALU–parity is incorrect at the output of the ALU. CPU DBO–indicates that the processing unit attempted to send data with incorrect parity to an I/O device. OP/Q–indicates that the op register or the Q register contains incorrect parity. INV OP–indicates that the byte in the op register does not specify a valid operation. CHAN DBO–indicates that the processing unit sent data with correct parity to an I/O device, but the I/O device received data with incorrect parity. INV Q–indicates that the Q byte in the Q register is not valid.

Machine Cycles Display

These lamps are used by the CE in servicing the system. None of these lamps are on during test mode operations, after a system reset or stop key operation, or during address compare stop.

Clock Cycles Display

These lamps are used by the CE in servicing the system.

CE Mode Selector

This rotary switch selects one of three processing unit modes: the normal process mode, the step mode, or the test mode. PROCESS is the normal mode for programmed system operation. In the STEP mode the rotary switch setting controls the manner in which the processing unit performs the stored program.

1. Instruction step–each time the start key is pressed and released the processing unit performs one complete instruction.
2. Machine cycle step–each time the start key is pressed and released, the processing unit executes one machine cycle.
3. Clock step–each time the start key is pressed and released the processing unit executes two clock cycles.

I/O operations operate in their normal manner in step mode operations. I/O requests for interruptions are honored in step mode, and the interrupting routine operates in step mode.

The switch settings under the test mode permit the following operation:

1. Alter SAR–with the CE mode selector switch set to this position, pressing the start key transfers the setting in the address/data compare switches to the instruction address register in use at the time and into the storage address register.
2. Alter storage–with the switch in this position, pressing and releasing the start key transfers the data specified by the two rightmost address/data compare switches into the A register and into storage at the address specified by the SAR. (For altering storage from the keyboard, see "Altering Storage" in the "I/O Operations" section.)
3. Display storage–pressing and releasing the start key transfers the data at the address specified by the storage address register to the B register, and then to the Q register, where it can be displayed by the roller switch on the display panel. The data is not destroyed in storage.

The storage test switch must be in the step position to avoid a processor check when the CE mode selector switch is moved between the alter storage position and the display storage position. Moving the mode switch through the alter-storage position to the alter-SAR position causes the keyboard to unlock. If having the keyboard unlocked is detrimental to your operation, operate the system-reset switch before you operate the start switch after the mode switch has been returned to the run position.

Note: No test is made for invalid storage addresses when the CE mode selector switch is in one of the test positions. No storage location will be selected by the invalid address.

System Reset Key

Pressing this key resets the power check condition and causes all I/O and machine registers (not local storage registers) to be reset to zero. The instruction address register for the main program and the program status register are reset to zero; all other local storage registers are unaffected. A complete program restart is normally required after a system reset.

Check Reset Key

Pressing this key resets the processor check and input/output checks. Resetting the checks allows the processing unit to resume processing when the start key is pressed.

File Write Switch

When this switch is in the off position, write operations cannot be performed on the disk file. Issuing a start I/O write instruction with this switch off causes the processing unit to loop on the start I/O write instruction.

Start/Stop Switch

When placed in the start position, this switch will take the processor out of the halt state (i.e., it will turn off the halt code lights and stop indicator and allow the processor to resume normal operation).

When placed in the stop position, this switch causes the processing unit to halt at the end of the operation during which the switch was activated. The halt is indicated by the turn-on of the stop indicator on the system keyboard console. I/O data transfers are completed without loss of information.

The processing unit may be restarted, without loss of information, only by placing the switch in the start position.

Address Compare Switch

When this switch is set to STOP, the CE mode selector switch is set to PROCESS, and the register display switch is set to SAR, the processing unit stops at the end of the cycle in which the storage address matches the address specified by the address/data compare switches. The processing unit is restarted by operating the start switch.

CE Servicing Switches

The following switches are normally used only by the customer engineer. They should be left in the positions indicated in parentheses:

1. Storage test (step)
2. Address increment (on)
3. I/O overlap (on)
4. I/O check (run)
5. Parity check (stop)

During normal processing, all two-position switches should be in the down position.

Address Compare Light

This indicator lights whenever the address/compare switches match the address in the storage address register, the register display switch is set to SAR, and the address compare switch is set to STOP.

I/O Check Light

This lamp lights when certain I/O errors are detected by an addressed I/O device. The light is turned off by system reset or by the I/O device. This light is used primarily by the CE.

LSR Display Selector Switch

This switch selects the local storage register to be displayed by the LSR position of the register display switch. The LSR's displayed are the LSR's in use (main program level or interrupt level). In the normal position, the register in use at any particular instant is the one displayed. The switch must be in the normal position for LSR parity checks to be displayed.

MANUAL OPERATION PROCEDURES

Altering Storage Addresses

This procedure is used to begin at a specific point in a program:

1. Move the system start switch to STOP.
2. Turn the CE mode selector switch to ALTER SAR.
3. Set the address/data compare switches to the desired address.
4. Move the system start switch to START.

If the CE mode selector switch is now turned to PROCESS, and the system start switch is operated, processing begins with the instruction located at the address set in the SAR.

Altering Storage

1. Move the system start switch to STOP.
2. Set the CE mode selector switch to ALTER SAR.
3. Set the address of the storage position you want to alter in the address/data compare switches.
4. Move the system start switch to START.
5. Turn the CE mode selector switch to ALTER STOR.
6. Set the two rightmost address/data compare switches to the hexadecimal value you want to enter into storage.
7. Move the system start switch to START.

In order to resume normal operation, set the storage address register to the address of the instruction at which you wish to begin. This is accomplished by the procedure described under "Altering Storage Addresses".

Displaying Local Storage Registers

1. Move the system start switch to STOP.
2. Turn the register display roller switch to LSR HI/LSR LO.
3. Turn the LSR display selector switch to the desired LSR.

Stopping at a Particular Address

1. Set the address compare switch to STOP.
2. Turn the register display roller switch to SAR HI/SAR LO.
3. Set up the desired address in the address/data compare switches.
4. Move the system start switch to START.

At the end of the cycle in which the desired address is used to access storage, the processing unit stops with the address compare light on. After the system is stopped, a manual operation such as alter storage, display storage, or manual branching can be performed. Care must be taken to insure that the operation in progress at the time that the address compare occurs is completed before altering storage, or manual branching. Storage can be altered or displayed in the middle of an operation such as during the E-A and E-B cycles of an add or subtract operation.

To perform this type of operation:

1. Set address compare to STOP.
2. Note the contents of the instruction address register.
3. Alter the storage address register to the address in storage that is to be altered or displayed.
4. Alter or display storage.
5. Restore the contents of the instruction address register by placing the instruction address register contents in the storage address register.
6. Turn the CE mode selector switch to PROCESS.
7. Operate the system start switch.

If a manual branch is to be performed, the operations following the manual branch must be considered. If the program status register and/or the results of the operation in progress are to be used and a return to the main program is required after the manual branch:

1. Address compare stop.
2. Turn the CE mode selector switch to INSTR.
3. Press and release the start key once.
4. Write down the contents of the instruction address register.
5. Change the instruction address register by altering the storage address register to the first instruction address in the subroutine.
6. At the end of the subroutine, restore the instruction address register by altering the storage address register to the address recorded in step 4.
7. Turn the CE mode selector switch to PROCESS.
8. Operate the system start switch to continue the main program.

If only a return to the main program is required after the address compare stop, and the address compare stop occurs in an instruction cycle:

1. Address compare stop.
2. Operate the system reset switch.
3. Manually branch to the subroutine by altering the instruction address register.
4. Restore the instruction address register to the op code of the stopped instruction.
5. Turn the CE mode selector switch to PROCESS.
6. Operate the system start switch to continue the main program.

If a return to the main program is not required after the manual branch, but either the program status register or the result of the operation being performed is required by the subroutine:

1. Address compare stop.
2. Turn the CE mode selector switch to INSTR.
3. Operate the system start switch once.
4. Perform the manual branch to the subroutine by altering the storage address register to the first instruction step in the subroutine.

If a return to the main program is not required, and only the results of the last operation are required by the subroutine:

1. Address compare stop at the desired address.
2. Address compare stop at the I-op address of the next sequential instruction.
3. Operate the system reset switch.
4. Perform the manual branch to the subroutine by altering the storage address register.

If none of the conditions (return to main program, use the program status register, or use the results of the last operation) are needed by the subroutine:

1. Address compare stop.
2. Operate the system reset switch.
3. Perform the manual branch to the subroutine.

CHECK CONDITIONS

Processor Checks

Detection of any one of the following processor checks causes the system to come to an immediate stop and terminates all I/O data transfers. The processor check light turns on for each of these checks. The kind of processor check that stopped the system can be determined by turning the register display roller switch to the PROC CHK position.

Invalid Address

This check indicates that the storage address register contains an address outside the address range of the processing unit. Recovery from this check usually requires correcting and reloading the program.

Invalid Op Code

This check indicates that the op register contains a code that is not recognized as a valid op code. Recovery from this check usually requires correcting and reloading the program.

Parity Check in the Processing Unit

This check indicates that an even number of bits has been detected in a byte at one or more of the data or addressing check points in the processing unit. Parity errors in data transferred from I/O units cause this check to occur. Restart procedure for this operation must be determined by the programmer. It usually requires reloading the program.

Invalid Q Byte in an I/O Instruction

This check indicates that the device address (contained in the Q code of an I/O instruction) addressed a unit that is not available to that system or that the N code in the Q byte is not valid for that I/O device. Recovery from this check usually requires correcting and reloading the program.

Unit Check

Unit check is the result of a positive response to testing testable check indicators in the I/O devices with a test I/O and branch instruction. The existence of these check conditions is signaled to the operator by having the processing unit come to a programmed halt. The halt identifier is keyed to the operator's restart/recovery procedure listing. The testable error indicators are discussed in the chapters of this manual dealing with I/O devices.

Usage Meter

This meter is located on the panel at the left side front of the processing unit. This meter records the time that the system is in operation. The meter records all the time that the processing unit is in operation from the time the start or load key is pressed until the job is completed. Time is not recorded when the processing unit is halted by either a manual or programmed halt, when a processor check stop occurs, when power is lost, or when the CE is servicing the system. When I/O operations are being performed during a programmed halt, time is recorded on the meter until all I/O operations are completed.

Disk Drive Metering

These two meters are located on the panel at the left rear side of the processing unit. Each meter is connected to a separate disk drive thereby recording the time that the drive is in use. The disk drive meters are activated by turning on the system console switches marked DISK DRIVE 1 and DISK DRIVE 2.

The first IPL (initial program load) or SIO (start I/O) instruction issued to a disk drive starts recording time on that drive's meter. Time is then recorded as long as the usage meter is recording time or until that disk drive is switched off.

Chapter 3. Keyboard-Console

The keyboard-console is the prime means of operator control of the system and data entry into the system. Control is exercised over the system through switches and lights on the console. Data is entered from the keyboard (Figure 3-1) through any of three groups of keys: command keys, alpha-numeric and function keys, or numeric keys.

The command keys are grouped to the left side of the keyboard. Eight command keys are standard. An optional feature provides an additional eight command keys. The standard keys are labeled 01 through 08. The optional keys are labeled 09 through 16.

The alpha-numeric and function keys resemble an electric typewriter keyboard. All of the graphics can be entered from this keyboard, including alphabetic, numeric, and special characters. The function keys should be programmed to perform the function stated.

The ten-key numeric keyboard is used to enter numeric data and decimal points. Three function keys (enter+, enter-, and erase) are associated with these keys for use in controlling the data entered.

OPERATIONAL CHARACTERISTICS

The keyboard-console operates in interrupt mode on interrupt level 1. Each key (including the inquiry request key located on the console section) generates a unique 8-bit code. Keys with upper and lower shift characters generate two unique codes, the code generated depending on whether the keyboard is in upper or lower shift.

Three of the I/O instructions can be used with the console keyboard: start I/O, sense I/O, and load I/O. Test-I/O-and-branch and advance program level are not effective for the console keyboard and result in a processor check with invalid-Q indication if issued with the keyboard-console device address.

Keyboard Operations

Data handling operations in the keyboard require some preparation. A load register instruction must be issued to load the address of the first instruction of the interrupt level 1 routine into the interrupt level 1 instruction address register. A start I/O instruction must be issued to enable keyboard interruptions (interrupt level 1).

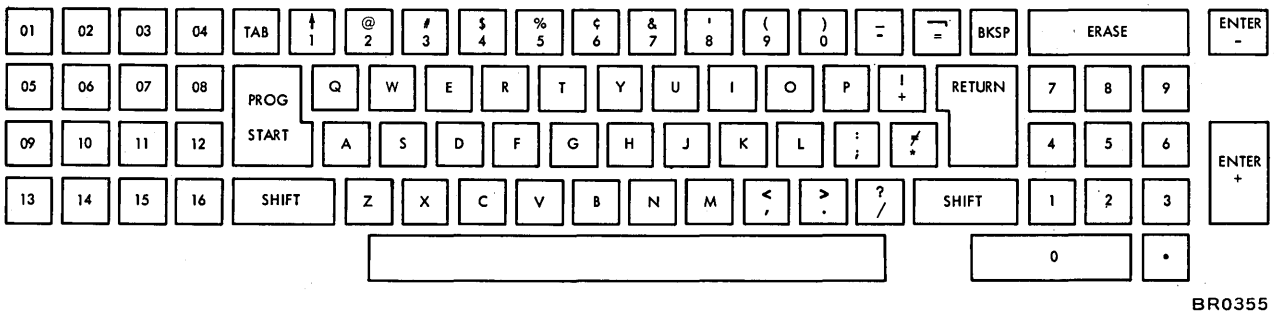


Figure 3-1. System Keyboard

After the keyboard has been prepared for operation by the start I/O instruction, the keys on the keyboard are freed so that they can be pressed. Pressing one of the keyboard keys causes two actions:

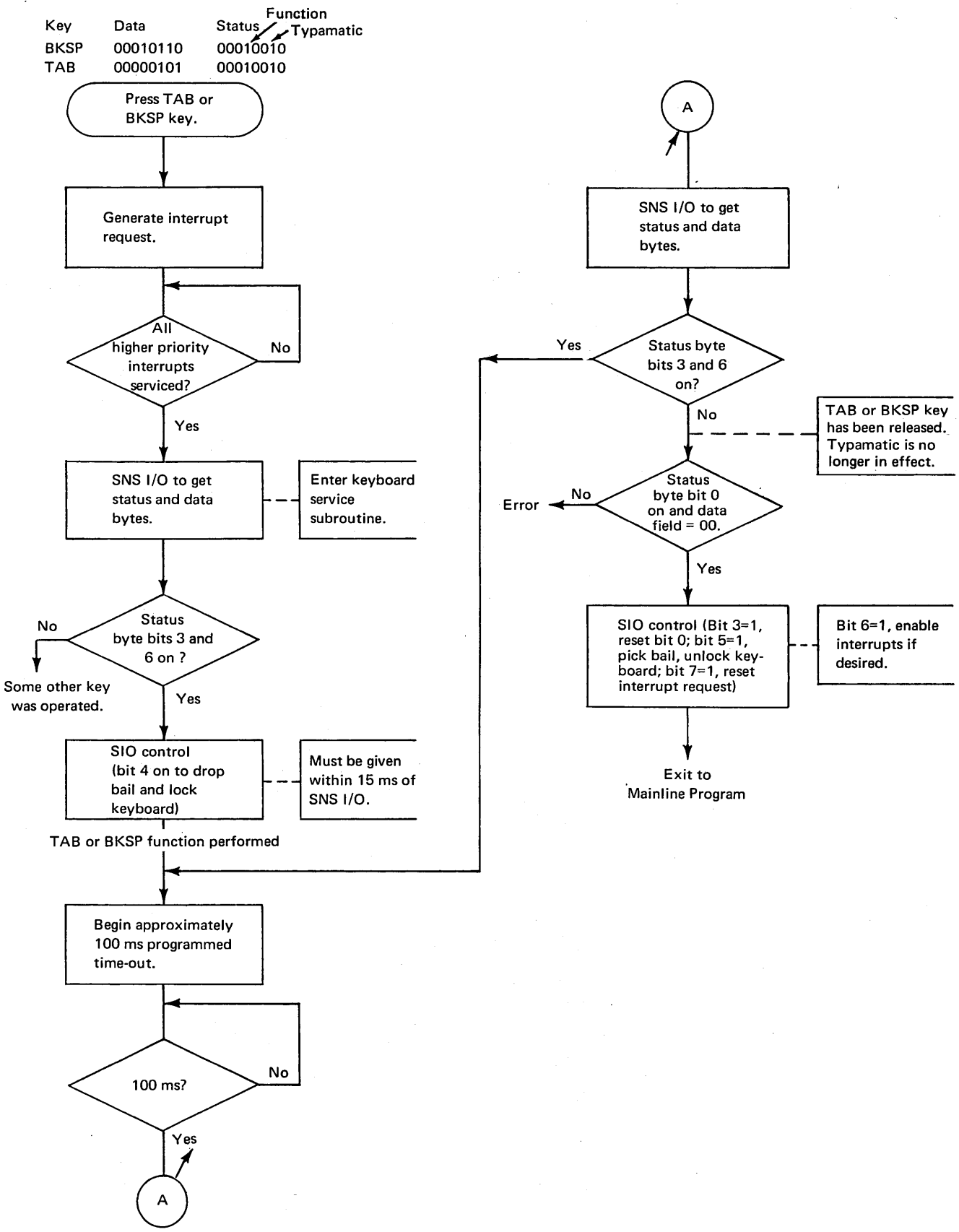
1. Two bytes are stored in the keyboard attachment circuitry. The first of these bytes is a status byte that defines whether the key pressed was a data key, a function key, or a command key. The status byte also carries information about the accuracy of the data and whether the key is typamatic or not. The second byte contains the character generated by the key contacts.
2. An interrupt request on interrupt level 1 is generated. If no higher-priority interrupt is being serviced, the processing unit honors the interrupt request. The interrupt subroutine must perform a sense I/O instruction to transfer the two bytes stored in the adapter circuitry into storage. The routine must also restore the keyboard, reset the interrupt request, and enable the keyboard if further keyboard operations are desired. Pressing any key automatically disables the keyboard.

All the keys on the keyboard except the backspace key and the tab key are prevented from storing characters in the adapter unit by the drop bail function of the start I/O instruction. This applies only when the key is held down after being first struck. The tab key and the backspace key are called typamatic keys and continue to store characters in the adapter circuits as long as they are held down. The tab and backspace keys operate on a two-level principle. Pressing these keys to the first level results in an operation that is the same as the non-typamatic keys. Pressing the keys to the second level (past the first level) results in a continuous generation of the unique code for that key. For typamatic operation to function completely, sense I/O and start I/O instructions must be issued repetitively under certain time constraints to store the characters and restore the keyboard.

The unique codes generated by each key on the keyboard are not the EBCDIC configurations for the same graphic characters in storage. The data placed in storage from the keyboard must be translated to EBCDIC data before it can be processed. The unique character codes generated by the keyboard are shown in "Appendix B".

Typamatic Key Programming

Operating the typamatic keys requires special programming procedures because repetitive keyboard operations have to be programmed even though the key is struck only once. Pressing a typamatic key causes a special sense bit to be set and transferred to the processing unit along with the data. The procedure shown in the following flowchart is recommended for processing typamatic key data.


Function
Key Data Status Typamatic
BKSP 00010110 00010010
TAB 00000101 00010010
Press TAB or BKSP key.
Generate interrupt request.
All higher priority interrupts serviced?
No
Yes
SNS I/O to get status and data bytes.
Enter keyboard service subroutine.
Status byte bits 3 and 6 on ?
No
Some other key was operated.
Yes
SIO control (bit 4 on to drop bail and lock keyboard)
Must be given within 15 ms of SNS I/O.
TAB or BKSP function performed
Begin approximately 100 ms programmed time-out.
100 ms?
No
Yes
A
A
SNS I/O to get status and data bytes.
Yes
Status byte bits 3 and 6 on?
No
TAB or BKSP key has been released. Typamatic is no longer in effect.
Error
No
Status byte bit 0 on and data field = 00.
Yes
SIO control (Bit 3=1, reset bit 0; bit 5=1, pick bail, unlock keyboard; bit 7=1, reset interrupt request)
Bit 6=1, enable interrupts if desired.
Exit to Mainline Program

Start I/O

Mnemonic: SIO

Op Code	Q Byte			Control Code
F3	1	M	N	

Operation: This instruction causes the keyboard and attachment circuitry to perform the operations specified by the control code.

The Q byte contains a device address of hexadecimal one in the high-order four bits. The low-order four bits of the Q byte are not used, but should be all zeros.

The individual bits of the control code specify individual operations as follows:

Bit	*Meaning*
0	Not used, may be any value.
1	Not used, may be any value.
2	1–set interrupt request if enabled (diagnostic use).
3	1–reset parity check. 0–do not reset parity check.
4	1–drop bail (lock keyboard). 0–do not drop bail.
5	1–pick up bail (unlock keyboard). 0–do not pick up bail.
6	1–enable interrupt request. 0–disable interrupt request.
7	1–turn off current interrupt request (the normal method for exiting interrupt level 1). 0–do not turn off current interrupt request.

The restore keyboard function necessary to prepare the keyboard for a succeeding key operation is accomplished by issuing a start I/O instruction with bits 4, 5, and 6 equal 1. All bit combinations of the control field are valid and all operations will be performed if the appropriate bits are on, except that bits 2 and 7 on is not guaranteed to result in interrupt request being set.

Example:

Instruction

F3	10	0F

The last key pressed is restored and the keyboard is prepared for the next keystroke. The keyboard interrupt request is reset, and interrupt level 1 is released.

Sense I/O

Mnemonic: SNS

Op Code	Q Byte	Operand Address
Y0	1 M N	

Operation: The two bytes stored in the keyboard attachment circuitry by a key operation are transferred to the two-byte field in storage specified by the operand address. The Q byte contains a device address of hexadecimal 1 in the high-order four bits. The low-order four bits are not used and can have any value. The low-order byte in storage is a status byte and is bit significant as follows:

Bit	*Meaning*
0	Parity check. This bit is turned on if an even number of bits are detected in the second byte of the two byte field.
1	Data character identifier. This bit will be 1 when a data key (graphic key) is pressed.
2	Command key identifier. This bit equals 1 when a command key is pressed.
3	Function character identifier. This bit will be 1 if any of the following keys is pressed: tab, backspace, program start, minus, erase, return, inquiry request, plus, space.
4	World Trade Corporation identifier. This bit will always equal 0 in systems in use in the USA.
5	Keyboard ready. This bit is used by the customer engineer for diagnostic purposes to indicate that the keyboard is unlocked and interrupts are enabled.
6	Typamatic. This bit will be on when either the tab key or backspace key is pressed.

The second byte (high-order) contains the unique bit configuration for the particular key pressed. In keys for which the upper shift is valid (numeric and special character keys on the typewriter) the bit pattern in this byte will also depend on whether the shift key is pressed.

Program Notes:

1. If a sense I/O instruction is executed before a key is pressed, the sense byte will contain the following:

   First Byte 1000 0000
   Second Byte 0000 0000

2. Data can be entered into the system at a rate of 900 key strokes per minute if the restore-keyboard start I/O instruction is issued within 15 milliseconds of the keyboard interrupt request.
3. Power-on reset and system reset (caused by pressing the system reset key) result in a locked keyboard with all indicators turned off and interrupts disabled. An initializing start I/O instruction unlocking the keyboard and enabling keyboard interrupts is required after one of these resets to allow the keyboard to operate.

Example:

Instruction

30	10	0F	CD

Two bytes of information from the keyboard are stored in storage locations 0FCC and 0FCD

Additional Program Notes:

1. An interrupt request should never be reset arbitrarily during initialization of the system due to the possibility of the operator activating the inquiry request switch (not under keyboard lock control).
2. The data and status bytes generated from all keyboard interrupts (including the inquiry request key) must be sensed in order to enable the keyboard for following interrupts. If the data is not sensed before the keyboard is restored, the SIO restore instruction will restore the keyboard and cause the KB ready light to come on. Although this light is on, the keys remain inoperative because data is still waiting to be sensed.
3. A sense I/O instruction to obtain the keyboard data should *always* precede the start I/O restore instruction. This is required because the keyboard data is reset approximately 20 ms after issuing the start I/O instruction.

CONSOLE CONFIGURATION

A console (Figure 3-2) contains the switches and indicators required by the operator for controlling the system.

Switch Panel

System Start Switch

When this switch is moved to the start position, the processor turns off the halt code lights and/or the stop indicator and resumes normal operation. When this switch is moved to the stop position, the processor halts at the end of the operation in process. This halt is indicated by turning on the stop indicator on the console. Pending I/O data transfers are completed without loss of information.

Inquiry Request Switch

This switch, mounted on the console, operates as though it were a key on the keyboard. Moving this switch to the on position causes the data and status bytes to be stored in the keyboard adapter circuitry. The status byte has the function key bit on and the data byte contains a unique data character code for the inquiry request key. Operating this switch also causes an interrupt request. Keyboard interrupt must be enabled for this switch to operate.

Data Recorder Switch

Moving this switch to the on-line position places the data recorder on-line and allows data to be entered from the data recorder reading station or punched at the data recorder punching station. Moving this switch to the off-line position places the data recorder in the off-line condition and allows it to function as a normal data recorder.

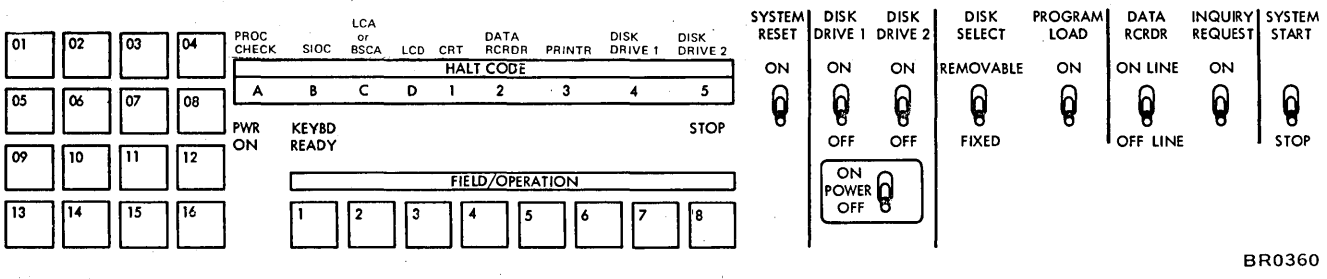


Figure 3-2. System Console

Program Load Switch

This switch initiates loading the program into main storage. The following actions occur when this switch is operated to the on position:

1. The following LSR's are reset: program IAR, DFDR, and program status register. No other LSR is reset. All other I/O and machine registers, controls, and status indicators are reset.
2. The record in cylinder 0, sector 0, on one of the disks in disk drive 1 is read into storage starting at location 0000. The disk that provides the first record is selected by a switch on the console.

When the program load switch is released, the processing unit executes the instructions read into storage by operating the program load key, starting at location 0000. If disk drive 1 is not ready, the I/O attention light lights when the program load switch is operated. It is unnecessary to operate the program load switch again; it is only necessary to make disk drive 1 ready to complete the program load function.

Disk Select Switch

This switch selects the disk from which the initial program load will be performed. When the switch is moved to the removable position, the removable disk is used for program loading. Similarly when the switch is in the fixed position, sector 0 of cylinder 0 of the fixed disk on disk drive 1 is used for program loading.

Disk Drive 1 and Disk Drive 2 Switches

These switches turn power on and off to the two disk drives.

Power Switch

This toggle switch controls the power to the system. When this switch is turned on, a partial system reset is performed in such a manner that no I/O operations are performed until explicitly directed. The system reset or program load switch must then be operated before any operation can take place. The integrity of data in storage is not guaranteed after this switch is operated.

System Reset Switch

Operating this switch causes all I/O and machine registers (not local storage registers) to be reset to 0. The instruction address register for the main program and the program status register are reset to 0. All other local storage registers are unaffected. A complete program restart is normally required after a system reset.

Indicator Panel

Processor Check Light

The processor check light turns on under any one of the following conditions:

1. When an invalid op code or parity error is detected in the CPU.
2. When an invalid Q code is detected.
3. When an I/O check occurs and the I/O check switch on the CE panel is in the stop position.

This light is turned off by system reset or by pressing the check reset key on the CE panel. Any of the errors causes the processing unit to come to an immediate stop. The clock is stopped, and input/output data may be lost. The specific error that causes the stop is displayed on the CE console display section.

I/O Attention Indicators

Any of these indicators on indicates that the corresponding I/O device has been issued a start I/O instruction when the I/O device is not ready to operate. A not-ready condition can be caused by power not being on or by some condition involving the paper or cards to be handled by the I/O device. The I/O attention indicators are: SIOC, BSCA, LCD, CRT, DATA RCRDR, PRINTR, DISK DRIVE 1, and DISK DRIVE 2. The specific conditions that cause each I/O indicator to turn on are discussed under the individual I/O devices.

Halt Code Indicators

These indicators are turned on by the individual bits in the halt identifier bytes of the halt program level instruction.

Power On Indicator

This indicator is turned on when system-power-on sequencing has been successfully completed and stays on until system power is turned off.

KEYBD Ready

This indicator is on when the keyboard has been enabled and unlocked.

Stop Indicator

This indicator is turned on when the system start switch is moved to the stop position. The indicator will come on immediately; however, any operation in progress will be completed with no loss of data. The indicator is turned off by moving the system start switch to the start position or by system reset.

Field/Operation Indicators

These indicators are turned on by a load I/O operation. The meaning of each indicator is determined by the program in use. A plastic overlay is provided for the field/operation indicators so that appropriate labels can be applied. Once turned on, the field/operation indicators remain on until another load I/O operation specifying turn off the field/operation indicators is executed, or until a system reset is performed.

Command Key Indicators

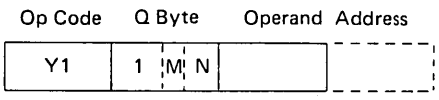
These lights are controlled by the load I/O instruction. Separate load I/O instructions are used for turning on or turning off the command indicators. Once turned on, command indicators remain on until a load I/O instruction turns them off or until a system reset.

Lamp Test Button

Pressing this button causes all the indicator bulbs on the keyboard-console to light. This button is located behind the indicator panel. It is accessed by pulling forward on the plastic tab located on the upper-left portion of the keyboard-console. The indicator panel will fold down and expose the lamp test button.

Load I/O

Mnemonic: LIO



Operation: The two-byte field located at the operand address is transferred to the keyboard-console adapter circuitry. The operand is addressed by its rightmost byte. The contents of the rightmost byte of the operand are used to turn on field/operation indicators or to turn command indicators on or off. The Q byte comprises a device address (always hexadecimal 1) in the high-order four bits, an M bit that can be either 0 or 1, and an N code. The N code is bit significant in the two low-order bits. The high-order bit of the N code (bit 5 of the Q byte) can be either 0 or 1.

When bit 6 of the Q byte equals 0, the load I/O instruction will either turn on or turn off the command indicators. Bit 7 of the Q byte determines which operation will be performed. When bit 7 of the Q byte equals 0, the command indicator specified by the rightmost byte of the operand will be turned off. When bit 7 of the Q byte equals 1, the command indicator specified by the rightmost byte of the operand will be turned on.

When bit 6 of the Q byte equals 1, the field indicator(s) specified by bits present in the rightmost byte of the operand will be turned on. Any field indicator not specified by the rightmost byte of the operand is turned off. When the field indicators are specified by bit 6 of the Q byte equal to 1, bit 7 of the Q byte can be either 0 or 1.

Command Light Control: The command light to be turned off or on is specified by the binary value of the bits in the rightmost byte of the operand. The light whose decimal label corresponds to the decimal value of the binary number in that rightmost byte of the operand is either turned on or turned off depending on the N code. If only the 8 basic command lights corresponding to the 8 basic command keys are installed, binary values more than 8 in the operand byte will be ignored.

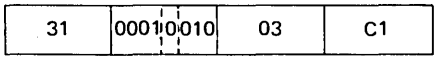
Field/Operation Indicator Control: The rightmost byte of the operand is bit significant as follows:

Bit	*Field/Operation Indicator Controlled*
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8

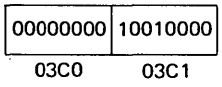
If a bit for a field indicator equals 1, the light is turned on. If a bit for the indicator equals 0, the light is turned off.

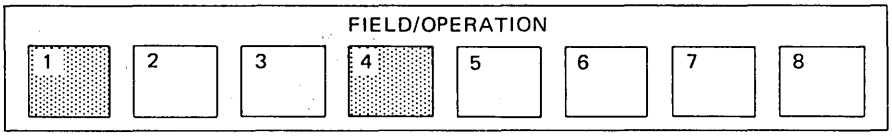
Example:

Instruction



Operand





I/O OPERATIONS

Altering Storage

Main storage can be altered in consecutive bytes without manually re-addressing storage by the following process:

1. Operate the system start switch to the stop position.
2. Set the storage test switch to STEP.
3. Turn the CE mode switch to ALTER SAR.
4. Set the data/address compare switches to the storage address where storage alteration is to begin.
5. Press the start key once.
6. Turn the CE mode switch to ALTER STOR.
7. Set the address increment switch to the on position.
8. Enter the data by pressing the keys corresponding to the hexadecimal value desired for each four bits (0-9 and A-F keys). Data will enter storage in the order in which the keys are pressed (the first key pressed will control the high-order four bits of the byte; the second key pressed will control the low-order four bits of the byte). As each second key is pressed, the hexadecimal value from that key and the preceding key is entered into storage as one byte. As each byte enters storage the storage address register is updated plus 1. (In this mode, any key position depressed other than 0 through 9 and A through F causes the keyboard to lock. A system reset is then required to unlock the keyboard. Therefore, when such a typing error is made, the contents of SAR should be noted prior to system reset, and then SAR reloaded and the byte in error retyped.)
9. Turn the CE mode switch to PROCESS.
10. Press the system reset switch, which resets the bail and locks the keyboard.
11. When all the desired data has been entered into storage, turn the CE mode switch to ALTER SAR.
12. Set the desired instruction address in the data/address compare switches.
13. Press the start key once.
14. Turn the CE mode switch to PROCESS.
15. Press the start key (the system begins processing).

Chapter 4. IBM 5213 and 2222 Serial Printers

The serial printer (Figure 4-1) produces printed data (readable from the console operator's position) for the System/3 Model 6. Five printers are available:

1. The IBM 5213 Printer Model 1 can print 132 characters per line at 85 characters per second. Forms are moved by a pin-feed platen with single or double spacing selected by the operator.
2. The IBM 5213 Printer Model 2 can print 132 characters per line at 85 characters per second. Forms movement is controlled by a pin-feed tractor with vertical forms control.
3. The IBM 5213 Printer Model 3 can print 132 characters per line at 85 characters per second with the printing element moving either from left to right or from right to left (bi-directional printing) under program control. Forms movement is controlled by a pin-feed tractor with vertical forms control. The 5213 Model 3 can also accommodate the enhanced print rate feature that increases the per-character print rate from 85 to a nominal 115 characters per second. High-speed printing occurs only for fields longer than five characters.
4. The IBM 2222 Printer Model 1 can print 220 characters per line at 85 characters per second. Forms are handled by dual pin-feed tractors with vertical forms control on the primary tractor only. The 2222 Printer also contains a ledger card device which allows feeding, printing on, and identifying ledger cards.

BR0363

Figure 4-1. IBM 5213 Printer

5. The IBM 2222 Printer Model 2 can print 220 characters per line at 85 characters per second with the printing element moving either left to right or right to left (bi-directional printing) under program control. Forms are handled by dual pin-feed tractors with vertical forms control on the primary tractor only.

Vertical forms control (on the 2222 and the 5213 Models 2 and 3) allows skipping from 2 to 256 lines. The 5213 Model 1 (without vertical forms control) can line space one or two lines by setting the manual control on the printer to the proper position. Forms motion for all printers is at 6 lines per inch. For a discussion of form designs available, refer to *Form-Design Considerations–System Printers,* Order Number GA24-3488

Printing Principle

Printing is done by a print head capable of making seven dots in a vertical arrangement. The print head is moved across the paper from left to right or from right to left at a constant velocity. As the print head moves through one character space, the head can produce dots in any of seven horizontal positions. A restriction is applied, however, that none of the vertical dot positions can produce a dot in two consecutive horizontal dot positions. Therefore, the maximum number of horizontal dots that can be produced by any dot position on the print head is four (in each character).

LEDGER CARD DEVICE

The ledger card device on the 2222 Printer allows the feeding, checking, and printing of data on ledger cards. The ledger card device is designed to interfere as little as possible with paper handling and readability of data printed on forms. Printing, feeding, and forms motion of the ledger card are under program control.

For identification purposes and checking that the proper document is being printed on, space is provided for printing coded numeric identification in the upper right edge of the card and for reading such coded notation.

The ledger card device also has a function that allows feeding the cards and locating the first available print line. Provision is also made for automatically signaling to the program when no print lines are available.

The right edge of the ledger card is always positioned at the farthest right printing positions of the 22 inch printer. The last six print positions of the ledger card (positions 215 through 220) are reserved for the line finder marks and two identification number (ID) code marks. These marks are printed in positions 216, 218, and 220. Except for the line finder mark code, no data should be printed beyond position 214.

The line finder marks are used for locating the next available print line. A line finder mark must be printed for each line printed to prevent over-printing when line posting of the same ledger card occurs again.

Ledger Card

Figure 4-2 shows the acceptable dimensions and printed area layout for the ledger card. The numbers on the ID coding correspond to the binary value of that bit when read into storage. The ID code and line finder marks are formed by over-printing the special LCD character to ensure proper read sensing. This printing and over-printing of ID code and line finder marks must be performed by the program; it is not performed automatically. The over-printing is accomplished by printing positions 215 to 220 first. The element must then be moved five positions to the left and positions 215 to 220 must be printed again. If bi-directional printing is installed, the over-printing can be accomplished on a print left operation.

The ID coding and line finder marks are sensed optically. This makes the contrast between the line finder mark and the ledger card background very important. IBM recommends that ledger cards be white, pink, buff, or yellow because these colors provide better reflectivity and contrast and, therefore, contribute to longer ribbon life. Blue ledger cards are not acceptable.

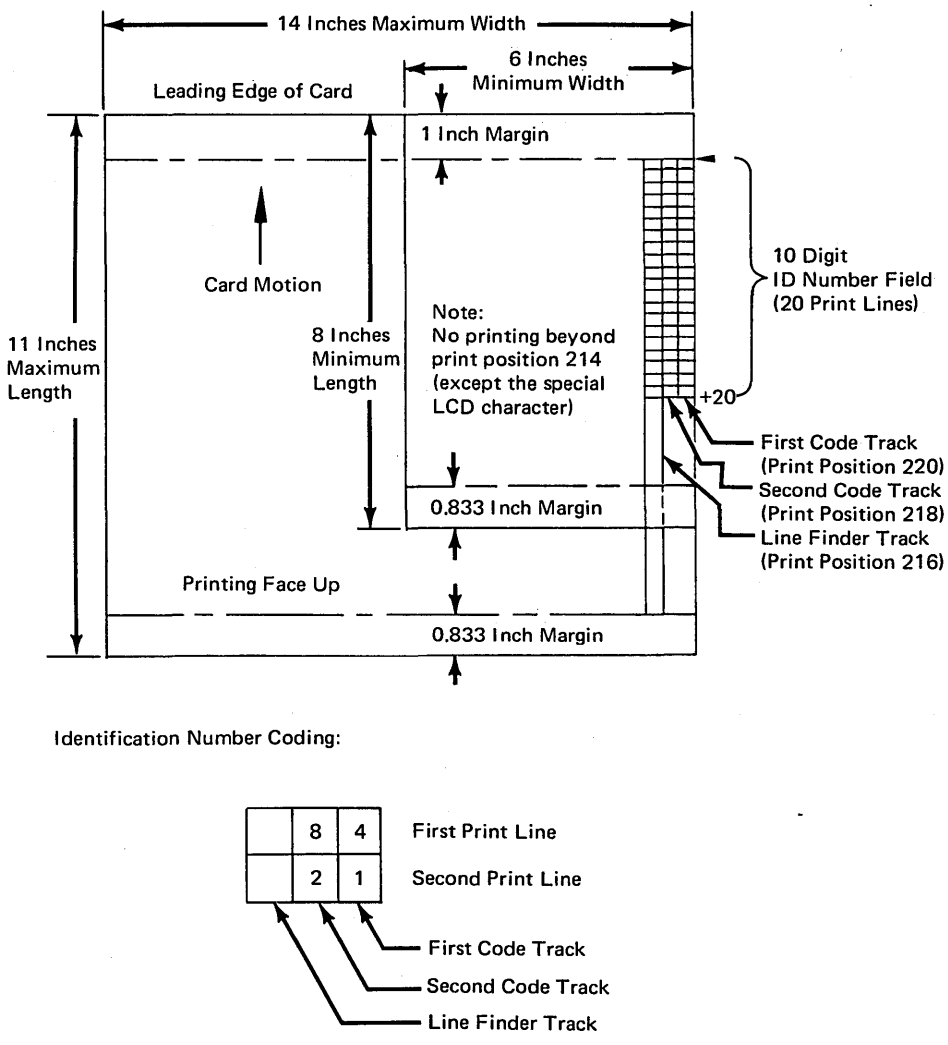


Figure 4-2. Ledger Card Format

BI-DIRECTIONAL PRINTING CAPABILITY

The bi-directional printing capability is a standard feature on the 5213 Model 3 and on the 2222 Model 2 printers. This feature allows printing with the print head moving either left to right or right to left under program control. By eliminating the requirement that the printing element be returned to the left margin before beginning each line of printing, this feature makes possible a higher lines-per-minute print rate. The lines-per-minute throughput depends upon the number of characters printed in each line regardless of whether printing is being performed in the uni-directional or bi-directional mode. Printing more characters results in a lower lines-per-minute throughput; printing fewer characters results in a higher throughput.

PRINTER FUNCTIONS

The serial printer (printing without the bi-directional printing feature) prints with the print head moving from left to right. The characters that can be printed and the bit patterns in storage that cause each character to be printed are shown in "Appendix B."

In addition to printing data the following functions can be performed:

1. Tab right
2. Tab left
3. Element return
4. Primary carriage index
5. Primary carriage skip (vertical forms control tractor only)
6. Secondary carriage single index (dual feed carriage)

PRINTER RESTRICTIONS

Two restrictions must be observed in serial printer operations:

1. Do not perform printing or element return operations when the fold in the forms is within 0.50 in. (12.7 mm) of the print line.
2. Do not overlap the forms in the primary and secondary forms tractors on the 2222 Printer.

PRINTER OPERATIONS

Printer operations are performed by executing a series of printer commands. These commands are stored in main storage and are addressed by the print command address register (PCAR). Execution of the commands is begun by a start I/O instruction. The commands can be either executed individually by issuing a separate start I/O instruction for each command to be executed, or can be chained to each other so that once execution has begun, the commands are executed one after the other until the end of the chain is reached.

Printer Commands

Printer commands are contained as individual bits within a command byte. The commands can be chained in any order. Some commands must be accompanied by a count byte one position directly to the right of the command byte. The commands available are:

1. Print
2. Tab right
3. Tab left
4. Primary vertical skip
5. Primary vertical index
6. Secondary vertical index
7. Element return

A printer command is issued by setting the appropriate bit in the command byte to 1. A command byte containing a command that requires a count byte must be followed immediately by that count byte. A command byte containing a command or commands that do not require count bytes is followed immediately by the next command byte if the commands are to be chained. The print, tab right, tab left, and primary vertical skip commands require a count byte. The primary vertical index, secondary vertical index, and element return commands do not require a count byte. Commands may be either chained or overlapped. Overlapped commands contain two or more commands within the same command byte. An example would be element return and primary vertical index within the same command, or tab right or left and primary vertical index within the same command. An overlapped command causes both operations to be performed at the same time.

Certain restrictions must be observed in overlapping commands:

1. Print commands cannot be overlapped with any other command.
2. No two count commands can be issued in the same command byte.
3. Tab commands and element return commands are mutually exclusive.
4. Primary vertical index command and primary vertical skip command are mutually exclusive. Any other combination of count and non-count commands is acceptable.
5. Primary and secondary carriage operations *should not* be overlapped.

Commands are chained by setting bit 0 of a command byte to 1. For each command byte with bit 0 equal to 1, the command or commands in that byte are executed; then the next command byte is retrieved from storage and the commands contained in that byte are executed. If one of the commands in the command byte is a command requiring a count byte, the second byte to the right of the command is chained to the command byte. If the command byte contains no count commands, the first byte to the right of the command byte is taken as the next command byte. The command byte bit significance is as follows:

Bit	*Meaning*
0	Command chain
1	Print data–command byte containing this bit equal to 1 must be followed by a count byte.
2	Horizontal tab right–command byte containing this bit equal to 1 must be followed by a count byte.
3	Horizontal tab left–command byte containing this bit equal to 1 must be followed by a count byte.
4	Primary carriage vertical forms skip–command byte containing this bit equal to 1 must be followed by a count byte.
5	Element return
6	Secondary carriage vertical forms index
7	Single primary carriage vertical index

Should an error occur while a chain of commands is being executed, execution of the command chain ends immediately, leaving the remaining commands unexecuted.

Performing Printer Operations

To perform printer operations certain initial operations must be performed. The printer command address register (PCAR) must be initialized to the leftmost byte of the command chain that is to be executed. The printer data address register (PDAR) must be initialized with the address of the leftmost byte of the print data area of storage. The print data area can be any length. If the printer commands are not chained, the printer command address register contains the address of the printer command to be executed.

When the printer command address register and printer data address register have been loaded, the program must execute a start I/O instruction. This instruction causes the printer command byte to be retrieved from storage, using the printer command address register, and the printer commands to be executed. A command byte containing the print command must be followed by a count byte. This count byte must contain, in a binary number, the number of positions to be printed minus 1. If the count byte equals 00, one character, addressed by the printer data address register, will be printed.

If the command byte specifies a tab right or tab left operation, it must be followed by a count byte. This count byte must contain, in binary numbering, the number of spaces to be moved minus 1. If the count byte equals 00, the print head will move one character space.

If the command byte specifies a primary vertical form skip operation, it must be followed by a count byte. This count byte must contain the number of vertical spaces to be skipped minus 1. If the count byte equals 00, a cycle check will occur and no forms movement will occur.

After a successful operation any of the count bytes equals FF.

For serial printer operations the printer prints only when the print head is moving from left to right. The printer will print only the number of characters specified by the count byte following the print command. Therefore, the print head must be moved to the correct horizontal position where the printing begins. This can be accomplished by horizontal tab right and horizontal tab left commands. The element return command is available for returning the print element to the left margin.

Printing Location Control

Six commands are available for controlling the location of printing on forms. Three of the commands are used for horizontal positioning of the print head before the printing begins. The other three commands are used for vertical positioning of the forms.

The element return command returns the printing element to the left margin. When the element is at the left margin a status bit is set in the status bytes to indicate this. This allows printing to start at the left margin.

Note that due to mechanical limitations, the print element must be initialized, preferably at the start of each job. A suggested method follows. A consecutive field in storage should be set up to contain:

Byte	*Hexadecimal Content*	*Description*
1	84	Chained element return
2	20	Tab right (not chained)
3	03	Tab count 4
4	04	Element return

Two start I/O instructions should then be issued consecutively to the printer attachment. The second start I/O is looped while the first is being executed, and is executed regardless of any previously encountered error. This procedure positions the element at true left margin. A test I/O and branch instruction can then be executed to test for element-at-left-margin. Refer to "Test I/O and Branch," in this printer section, for the N code for testing element-at-left-margin.

The tab right and the tab left commands allow moving the element the desired number of spaces either left or right. The number of spaces to be moved is specified by the count byte that follows the tab command byte. The movement of the printing element to the right can also be accomplished by printing blank characters in each of the print positions to be passed over. Printing blank characters is sometimes faster than tabbing. This occurs when the distance is less than 12 spaces.

Except for the element return command, which moves the element to the left margin, the printer attachment does not provide a counter to indicate print element location. The program must maintain a count of printing positions moved so that printing element position can be determined.

The three vertical forms position commands (vertical forms skip and primary and secondary carriage vertical forms index) position paper vertically for printing the correct line.

The printer attachment does not provide a counter for vertical forms location. The program must provide a count to indicate the vertical forms location.

Bi-Directional Printing

The primary use for the bi-directional printing feature is line printing, in which mode all lines are the same length and a complete line is printed from a single print command. For line printing operations in bi-directional mode the count byte for the print command must always contain the same number if all print lines are to be aligned at the left margin. The first character of each line is printed at the left margin. The first characters or last characters in each line can be blanks. Even though the printer functions with the printing element moving either from left to right or right to left, the print data address register always is initialized to the leftmost byte of the printing field.

Bi-directional printing operations do not require (but do not exclude) the use of the horizontal positioning commands other than perhaps an initial element return command. However, a special function has been included so that programs written for bi-directional printer functions can be compatible with systems that do not have the bi-directional printing capability. With the bi-directional printing feature installed, the element return command does not function if the start I/O instruction specifies bi-directional printing. If the bi-directional printing feature is not installed or the start I/O instruction does not specify bi-directional mode, the element return command is executed.

Bi-directional printing capability does not exclude serial printing capability on the same printer. Therefore, to initialize for bi-directional printing, an element return command can be issued to the printer in serial printing mode, and then printing can begin in bi-directional printing mode.

Vertical forms control in bi-directional printing mode has the same requirements and uses the same commands as vertical forms control in serial printing mode. The first print command should be issued with the printing element at the left margin. Printing commands issued with the printing element not at the left margin result in right to left printing motion. If the count byte specifies more characters than character spaces are available between the printing element position and the left margin, a margin check will occur when the printing element reaches the left margin.

Typical Programming Sequences

Serial printing in uni-directional mode and line printing in bi-directional mode require somewhat different programming approaches. A common application of serial printing might require printing of data entered from the console-keyboard, processing some of the data, printing the results of the processing on the same line, then returning to the left margin to print the data entered at the next keyboard entry. Such a programming sequence is illustrated in Figure 4-3.

1. The printer data address register need be loaded only for the first character of the entry (in this portion of the routine) because the register will be updated to the next higher address by each print operation.
2. This load operation is required for each character because the printer command address register is left with the address of the next higher order address following the print command and count byte addresses for the start I/O instruction that prints the character.
3. This decision indicates that the last character for this particular transaction has been entered. This indication is usually made by pressing the program start key or return key.
4. This block represents many steps, such as price-quantity extensions in invoices, inventory update, or back-order determination.
5. After the data has been processed, the results are stored in main storage. The address of the high-order byte of this data is loaded into the printer data address register to make it available to the printer.
6. The command chain performed by the start I/O is addressed by this address. The commands cause the previously processed data to be printed on the same line as the characters entered from the keyboard, in whatever print positions are desired. After printing is complete, the printing element is returned to the left margin and the paper is moved up to provide a new printing line.

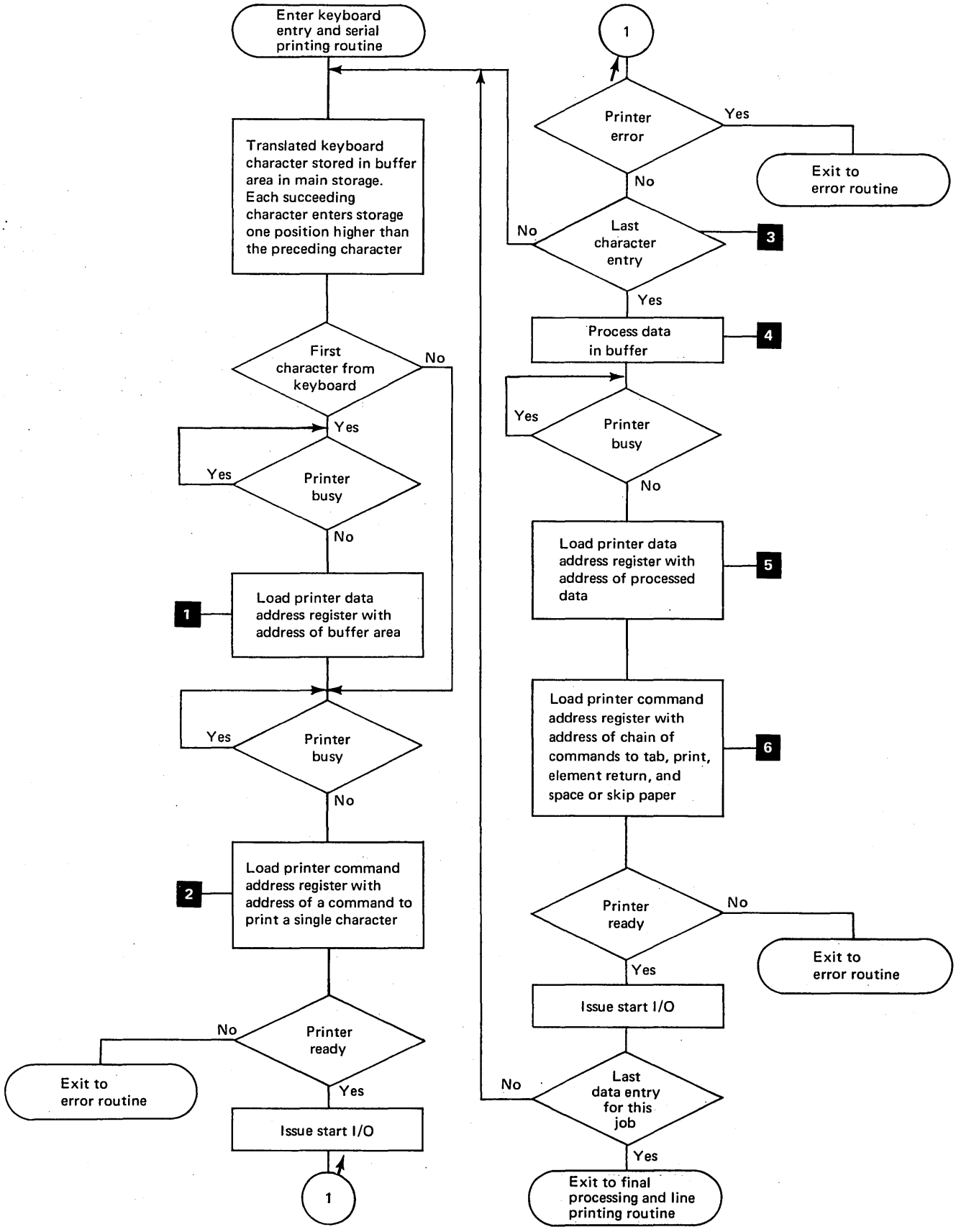


Figure 4-3. Serial Printing Routine

Line printing using the bi-directional printing feature is a somewhat simpler process than serial printing. All the data is assembled and processed, then printed in a single print instruction as illustrated in Figure 4-4.

1. This could be any typical data processing job. The results of the processing are assembled in an area of storage in such a manner that each print instruction can cause the same number of characters to print.
2. This operation is repeated for each line of print, unlike the serial printing function in which the address left in this register at the end of the single character print operation is used as the address of the next character.
3. The command chain addressed by loading the printer command address register in this operation contains an element return command to provide for compatibility with printers that do not have the bi-directional printing feature.
4. This start I/O instruction can be issued in either the uni-directional mode or the bi-directional mode. If the instruction is issued in the uni-directional mode or the printer does not have the bi-directional feature, the element return command will be effective and the printing element will return to the left margin after each line is printed. If the instruction is issued in the bi-directional mode to a bi-directional printer, the element return command will be ignored. If the element is at the left margin when the print command is encountered, printing will occur from left to right. If the element is not at the left margin, printing will occur from right to left.

   Note that printer data address register is loaded with the address of the high-order byte of the print data regardless of the direction of element travel.

Forms Recognition

The program is responsible for maintaining the printing within any boundaries established by the forms on which printing is being done. To this end, a counter must be established to count the number of line spaces that is used on the form. This counter can then be checked by compare instructions and branch on condition instructions can be used to cause any special operations required by an approach to the end of the form.

The 5213 printer carriage has a contact that recognizes an out-of-forms (called end-of-forms) condition. This contact produces a signal when the end of the last form in the stack has passed the contact. When this condition is recognized, the attachment responds "end-of-forms" to a test I/O instruction, and a status bit is provided by a sense I/O instruction.

Although the attachment responds end-of-forms to the test I/O, it accepts and performs print commands. The program is expected to finish printing on the last form and skip to eject the form. At this time the program should issue a load I/O instruction to turn on the printer I/O attention indicator and go to a programmed halt.

The 2222 Printer has dual forms control. It operates not only with the primary carriage but with a secondary carriage. The secondary carriage has its own end-of-forms contact. A special load I/O instruction causes the secondary carriage end-of-forms contact to function in the same manner as the primary carriage end-of-forms contact. The primary carriage must always contain paper to enable the printer to operate, even though only the secondary carriage may be in use at the time.

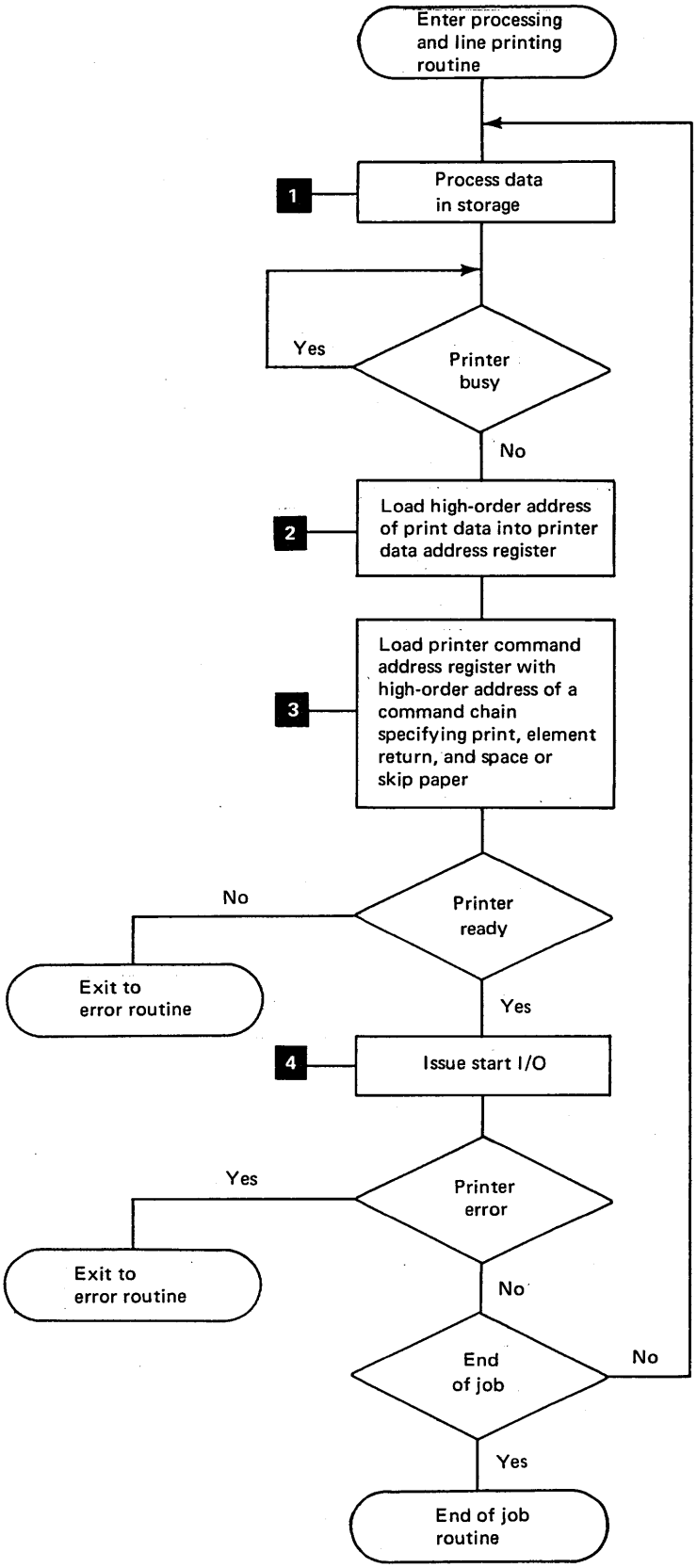


Figure 4-4. Line Printing Routine

Figure 4-5 illustrates a recommended programming sequence for forms control.

1. The program is responsible for ensuring that data is printed within the boundaries of the particular form. This counter is initialized with the number of line spaces between the top edge of the form and the bottom edge of the form.
2. Setting this latch causes the end-of-forms response to the test I/O instruction to occur when the secondary carriage end-of-forms contact closes as well as when the primary end-of-forms contact closes.
3. This operation enables the program to determine what forms control operations are required next by checking the remaining line count.
4. Skipping to a new form requires that the line space counter be set to a count that corresponds to the first line to be printed on the form.
5. Setting the end-of-forms latch when the end-of-forms contact is closed causes the printer to become not ready and turns on the printer I/O attention light on the keyboard console.
6. This halt instruction tells the operator to add paper to the printer.
7. The end-of-forms latch must be reset to enable the printer to accept start I/O instructions.
8. The end-of-forms contact closes to indicate the end-of-forms condition when about three inches of forms are left between the print line and the edge of the paper.

Printer I/O Attention Light

The printer I/O attention light on the keyboard console lights when:

1. The printer cover is not properly closed.
2. The N/6 lever is in the neutral (N) position (5213 models 2 and 3, and 2222 models 1 and 2 only).
3. The end-of-forms latch is set and the end-of-forms contact is closed. When paper is inserted, the end-of-forms contact opens and the light goes out, but the printer will remain not ready until the end-of-forms latch is reset.

Ledger Card Device Operations

The commands for operating the ledger card device are somewhat different from the printer commands in that the command fields are fixed. The operations that can be performed are:

1. Feed, read ID, and locate next print line
2. Feed, read ID, and eject
3. Eject
4. Index
5. Read mark and eject
6. Read all line finder marks

The ledger card device commands contain count bytes which are decremented to hexadecimal FF during the execution of the commands. To restore these count bytes to their correct values, the ledger card device command field in storage must be reloaded with the required command bytes before each start I/O instruction is issued to the ledger card device.

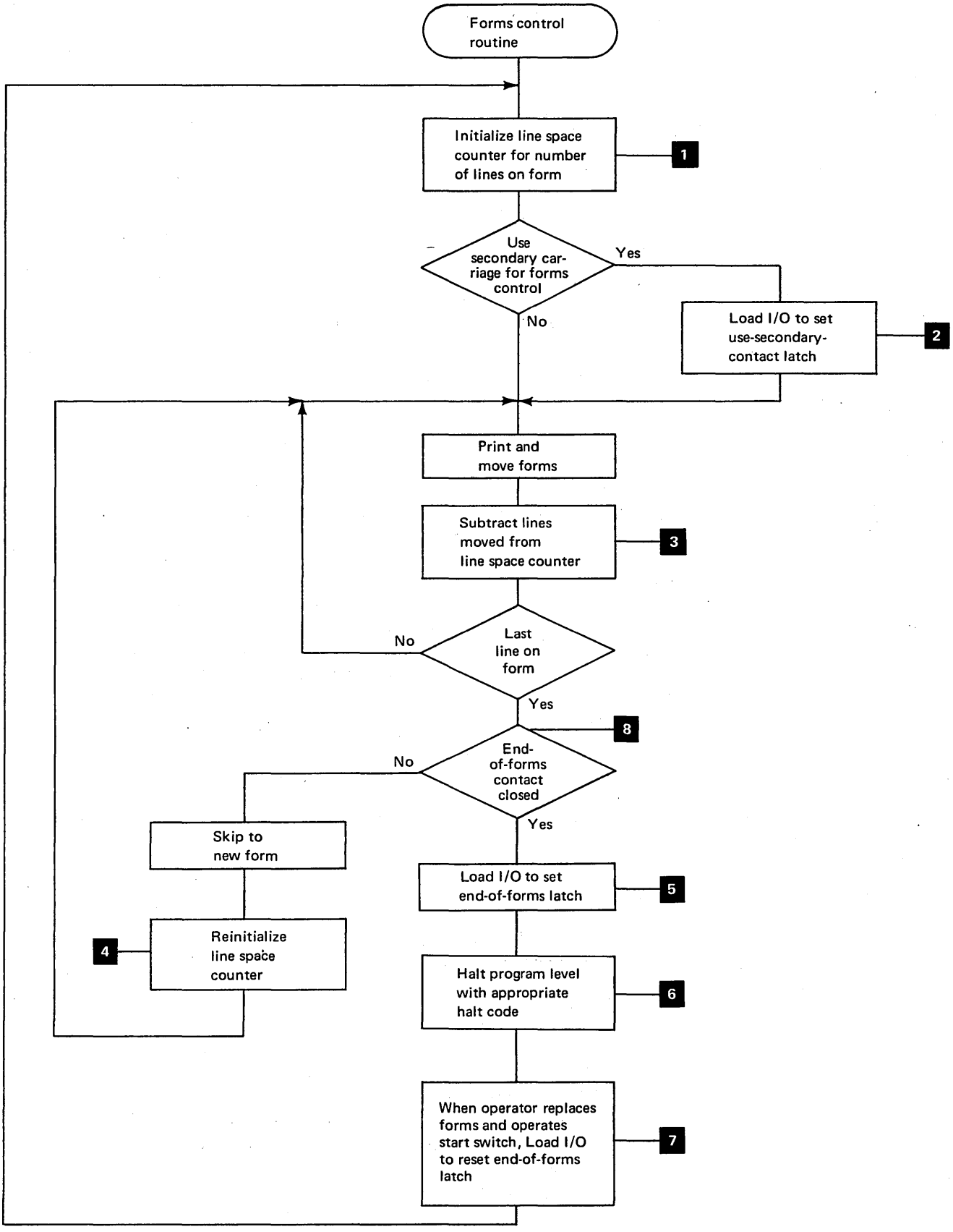


Figure 4-5. Forms Control Sequence

Feed, Read ID, and Locate Next Print Line

Command Field Format:

Byte Number	1	2	3	4	5	6	7	8
	03	05	04	00	05	04	06	13

This command causes the ledger card device to:

1. Feed a ledger card into the ledger card device.
2. Read the ID number into main storage beginning at the location specified by the address stored in the print data address register.
3. Move the ledger card until the next available print line is positioned for printing.

The print command address register contains the storage address of byte 1 of the command field. The locate line address register must contain the storage address of a two-byte field in which the first and second bytes must contain the hexadecimal values 11 and 0E respectively. The program does not have to wait until a command has been completely executed to examine the ID number even though the ledger card device is busy. Not read ID busy indicates that the ID number has already been read into storage.

If this command is given and the ledger card does not contain an ID number, the ID field in storage will be filled with blanks (hexadecimal 40), and the card will stop with the first print line positioned at the platen.

The read ID portion of the command causes the coded ID number to be optically sensed from the ledger card and assembled in storage. The ledger card device will always read 10 digits (20 coded lines) into storage. Two lines are required to assemble each digit. The data is assembled in storage in zoned decimal format. The numbered coding shown in Figure 16 in the bottom half of the figure indicates the bit position in the low-order 4 bits that each of the marks will turn on. For example, if the 8 block and the 1 block for any group of lines (for example print line 1 and print line 2) contain marks, bits 4 and 7 of the byte in storage will be set to 1. Two exceptions to this rule exist:

1. If the code position for a character contains marks in the 8-bit and 2-bit positions and not in the 4 and 1 positions the corresponding storage position will contain F0.
2. If none of the code positions contain print marks, the position in storage will contain hexadecimal 40.

Feed, Read ID, Eject

Command Field Format:

Byte Number	1	2	3	4	5	6	7	8
	03	05	04	00	05	04	07	13

This command causes the ledger card device to:

1. Feed the ledger card into the ledger card device.
2. Read the ID number into the locations specified in storage by the print data address register.
3. Eject the ledger card from the ledger card device.

The storage address of byte 1 of the command field is contained in the print command address register.

The program does not have to wait until the command has been completely executed to examine the ID number. Even though the ledger card device is busy, not-read-ID-busy indicates that the ID number has already been read into storage.

If this command is given and the ledger card has not yet been initialized, the ID field in storage will be filled with hexadecimal 40.

The data read into storage is in the same format as for the feed, read ID, and locate next print line command.

Eject

This command consists of a single byte containing hexadecimal 00. The eject command causes the ledger card to be ejected from the print station and out of the ledger card device. The storage address of the eject command byte is contained in the print command address register. This command should be issued to eject the ledger card after a unit check or when no check of the last printed line finder mark is desired.

Index

This command consists of a single byte containing hexadecimal 01. The index command moves the ledger card to the next sequential print line. The storage address of the command byte is contained in the print command address register. The card will eject automatically if an index command is issued after the last printable line has been detected.

Read Mark and Eject

This command consists of two bytes. The first contains 02 and the second contains 08. The command causes the following actions:

1. The card is ejected from the ledger card device.
2. As the card leaves the print station the ledger card device reads the last line finder mark written on the card.

The address of the first byte of the command is contained in the print command address register.

This command should be issued to eject the ledger card after a normal posting operation. This command checks the quality of the last printed line finder mark. An inadequate line finder mark causes a read mark check.

Read All Line Finder Marks

Command Field Format:

Byte Number	1	2	3	4	5	6	7	8
	03	05	04	00	05	04	07	55

This command is provided for diagnostic use to enable checking that all line finder marks have been printed and to allow determining what line is in error if any line finder mark has not been printed. This causes the ledger card device to:

1. Feed the ledger card into the device.
2. Read the contents of print position 216 (line finder mark column) from both read stations into storage locations specified by the print data address register.
3. Eject the card from the device.

The address of byte 1 of the command field is contained in the print command address register.

This command always enters 43 bytes of data into storage (provided that byte 8 of the command is exactly equal to hexadecimal 55) beginning at the location whose address is stored in the print data address register. The data read is stored in the low-order four bits (4 to 7) of the bytes in which the data is stored. The four high-order bits of the data bytes will be all ones.

One byte of data will be stored for each two line spaces that the ledger card advances. Bits 4 and 6 of the byte are controlled by the first read station (the read station before the printing position). Bits 5 and 7 of the byte are controlled by the second read station. Bit 4 of the byte is turned on by reading a line finder mark on an odd numbered line at the first read station. Bit six is turned on by reading a line finder mark on an even numbered line at the first read station. Bit 5 is turned on by reading a line finder mark on an odd numbered line at the second read station. Bit 7 is turned on by reading a line finder mark on an even numbered line at the second read station.

Because of the mechanical configuration of the machine, certain bits in the 43 bytes are meaningless. The maximum number of lines that can be printed on a maximum length card is 55. There are 29 line spaces between the first and second read stations. This provides a maximum of 85 possible print lines that could be read into the 43 bytes; therefore, bits 6 and 7 of byte 43 are meaningless. In addition, bits 5 and 7 of the first 14 bytes and bit 5 of byte 15 are meaningless because no lines are readable at the second read station for these bytes. Bits 4 and 6 become meaningless after all print lines possible on the ledger card have passed the first read station.

PRINTER INSTRUCTIONS

Start I/O

Mnemonic: SIO

Op Code	Q Byte	Control Code
F3	E M N	

Operation: This instruction causes the printer or ledger card device (LCD) attachment to begin cycle stealing to obtain commands located in main storage. The commands are addressed by the print command address register. The print command address register contains the address of the high-order byte of any chain of commands that is to be executed.

The Q byte contains a device address (always E for the printer or LCD) in the high-order 4 bits, an M bit, and an N code of all zeros. The M bit specifies whether the series of commands that are to be executed will affect the printer or the ledger card device. If the M bit equals 0 the printer is addressed by the instruction; if the M bit equals 1, the ledger card device is addressed.

The control code byte contains all zeros with two exceptions. For the printer, bit 7 is the printing mode control. When bit 7 of the control code equals 0, the printer operates in serial printing mode. When bit 7 of the control code equals 1, the printer operates in bi-directional printing mode (if the bi-directional printing feature is installed). For the LCD, bit 7 must be on only when a start I/O for a read-all-line-finder-marks command is issued.

The commands that are executed by the start I/O instruction are the commands listed in "Printer Commands" and "Ledger Card Device Operations."

Printer Functions: If the start I/O instruction is executed for the printer (M bit equals 0), the commands are executed beginning with the command addressed by the print command address register. If bit 0 of the command byte is 1, the following command is executed without requiring a new start I/O instruction. Printer commands will be executed from a single start I/O instruction as long as the command bytes have bit 0 equal to 1. When a command is executed with bit 0 of the command byte equal to 0, the execution of the printer command ends at the end of that command execution.

For the ledger card device start I/O instruction (M bit equal to 1), a separate start I/O instruction must be executed for each command that is to be executed.

If a start I/O instruction is executed with the M bit equal to 1 and the ledger card device feature is not installed, a processor check stop occurs with the invalid-Q-byte indicator lit on the CE rotary switch display.

Example:

Instruction

F3	E0	01

Commands

C0	5F	05

The printer prints a 96-character line, indexes 1 line space, and stops with the print head at either the left margin, or 96 characters away from the left margin, if the bi-directional printing feature is installed. If the bi-directional printing feature is not installed, the element return command in the third command byte causes the print head to return to the left margin.

Test I/O and Branch

Mnemonic: TIO

Op Code	Q Byte			Branch-To Address	
Z1	E	M	N		

Operation: The conditions specified in the Q byte are tested. If the condition or conditions exist, the next instruction is taken from the branch-to address and the address of the next sequential instruction is placed in the address recall register. If the condition or conditions do not exist, the next sequential instruction is taken and the branch-to address is placed in the address recall register. The address recall register will not then be changed until the next decimal, insert-and-test-characters, or branch instruction is executed. The Q byte comprises a device address (always E for the printer) in the high-order 4 bits, an M bit, and an N code. The M bit specifies whether conditions for the printer or for the ledger card device are to be tested. The N code specifies the conditions that are to be tested. Different conditions are tested for the printer and for the ledger card device. An M bit equal to 0 specifies testing the printer; an M bit equal to 1 specifies testing the ledger card device.

When the M bit equals 0 the N code specifies testing as follows:

Bits	*Printer Conditions*
5 6 7	
0 0 0	Unit check
0 0 1	End-of-forms (see "Forms Recognition")
0 1 0	Busy
1 0 0	Element-at-left-margin

Combinations of the above conditions may be tested. It is suggested, however, that only one condition at a time be tested. In combined testing, any one of the tested conditions causes the branch to occur.

The printer attachment becomes busy to the program after acceptance of any start I/O instruction. During this busy period, only sense I/O and test I/O instructions and the control load I/O instruction, which does not involve LSR's, are accepted. Busy drops after successful completion of an entire field of chained commands or on immediate detection of a machine error. This means that chaining is not completed if an error is detected.

Testing for element-at-left-margin indicates, when the branch is taken, that the print element is at the leftmost printing position. Normally, the print element should be programmed to the left margin at the start of a job. Refer to "Printing Location Control" for details.

When the M bit is equal to 1, the N code specifies testing as follows:

Bits	*LCD Conditions*
5 6 7	
0 0 0	Unit check
0 0 1	Last printable line
0 1 0	Ledger card device busy
0 1 1	LSR busy
1 0 0	Read ID busy
1 1 0	Card not aligned

Unit check for printer (M code equal to 0) consists of:

1. Cycle check (horizontal or vertical)
2. Data check or read only storage check
3. Margin check
4. Sync check
5. Invalid command check
6. Cover open
7. N/6 lever in neutral position

For further explanation refer to "Printer Error Recovery Procedures."

Unit check for the ledger card device consists of:

1. Sense amp check
2. Card skew check
3. Drive check
4. Read mark check
5. Line finder mark check
6. Invalid command check

Last-printable-line indicates that while performing either an index or feed, read ID, and locate next print line command, the last available print line on the ledger card has been located at the posting line. If a full ledger card is fed into the ledger card device by means of a feed read ID and locate command, the card is automatically ejected.

Ledger-card-device-busy occurs when the ledger card device is executing a start I/O instruction or during a manual eject. The ledger card device will become not busy when the instruction is completed, whether the instruction is ended normally or by a unit check condition.

LSR-busy is a signal which is activated by either the printer or the ledger card device. LSR-busy is activated by the following conditions:

1. A printer chained command is in process.
2. A printer count command is in process.
3. A ledger card device feed, read ID, and locate next print line is in process.
4. A ledger card device feed, read ID, and eject command is in process, and the ID number has not yet been read into storage.
5. A read all line finder marks command is in process, and all the line finder marks have not yet been read into storage.
6. A ledger card device read mark and eject command is in process, and the line finder mark has not yet been read.
7. A ledger card device index command is in process.

The ledger card device and the printer share both the print data address register and the print command address register. Care must be taken when overlapping printer and ledger card device commands. It is permissible to overlap printer and ledger card device commands only when LSR-busy is not active.

Read-ID-busy will be activated on the feed, read, and locate next print line, and feed, read, and eject commands. Read-ID-busy is activated until the ID number has been read into storage. When read-ID-busy is inactive, the program can examine the ID number on the ledger card while the ledger card is still being fed to the print line. Read-ID-busy is also activated while the read all line finder marks command is being executed until the 43 byte transfer has been completed.

Card-not-aligned indicates that the ledger card is not properly aligned at the pinch point of the first feed rolls of the ledger card device. The LCD I/O attention indicator is lighted and the card gate is raised when the command is issued to test for card-not-aligned. No card can be fed into the LCD until the card gate is raised. This condition must be tested for and found to be absent before issuing any of the following commands:

1. Feed, read ID, and locate next print line
2. Feed, read ID, and eject
3. Read all line finder marks

For ease of programming, the condition tested for should be card-not-aligned. A test I/O and branch instruction for card-not-aligned causes the LCD attention indicator on the console to be turned on if the card is not aligned. The indicator turns off when the card aligned condition is present.

Example:

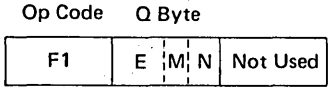
Instruction

	C1	E0	0C	21
Address	0123	0124	0125	0126

If unit check exists in the printer attachment, the next instruction is taken from 0C21. If unit check does not exist, the next instruction is taken from 0127.

Advance Program Level

Mnemonic: APL



Operation: The printer or ledger card device is tested for the condition specified by the Q byte. If the condition exists, the program loops on the advance program level instruction until the condition no longer exists. If the condition does not exist, the next sequential instruction is executed.

The Q byte comprises a device address (always E for the printer and ledger card device) in the high-order four bits, an M bit, and an N code. The M bit specifies whether the printer or ledger card device is to be tested, and the N code specifies what conditions are to be tested. Different conditions are tested for the printer than are tested for the ledger card device. An M bit equal to 0 specifies testing the printer; an M bit equal to 1 specifies testing the ledger card device.

When the M bit equals 0, the following conditions are tested:

Bits	*Printer Conditions*
5 6 7	
0 0 0	Unit check
0 0 1	End of forms (see "Forms Recognition")
0 1 0	Busy
1 0 0	Element at left margin

Combinations of conditions can be tested; it is suggested, however, that only one condition at a time be tested. In the combined tests, any of the conditions present causes the program to loop.

The printer attachment becomes busy to the program after acceptance of any start I/O instruction. During this busy period, only sense I/O and test I/O instructions and the control load I/O instruction, which does not involve LSR's, are accepted. Busy becomes inactive after successful completion of an entire field of chained commands or on immediate detection of a machine error. This means that chaining is not completed if an error is detected.

Element at left margin indicates that the print element is at the leftmost printing position. Normally, the print element should be programmed to the left margin at the start of a job. Refer to "Printing Location Control" for details. When the M bit equals 1 the following conditions in the ledger card device are tested:

Bits	*LCD Conditions*
5 6 7	
0 0 0	Unit check
0 0 1	Last printable line
0 1 0	Ledger card device busy
0 1 1	LSR busy
1 0 0	Read ID busy
1 1 0	Card not aligned

Unit check for the printer (M code equal to 0) consists of:

1. Cycle check (horizontal or vertical)
2. Data check or read only storage check
3. Margin check
4. Sync check
5. Invalid command check
6. Cover open
7. N/6 lever in neutral position

For further explanation refer to "Printer Error Recovery Procedures".

Unit check for the ledger card device (M bit equal to 1) consists of:

1. Sense amp check
2. Card skew check
3. Drive check
4. Read mark check
5. Line finder mark check
6. Invalid command check

Last-printable-line in the ledger card device indicates that while performing either an index command or a feed, read ID, and locate next print line command, the last available print line on the ledger card has been positioned for posting. When this condition occurs the ledger card device automatically ejects the card if another index command is issued. If a full card is fed into the ledger card device by a feed, read ID, and locate command, the card will be automatically ejected.

Ledger-card-device-busy occurs only when the ledger card device is executing a start I/O instruction. The ledger card device becomes not busy when the instruction is completed, either normally or by a unit check condition.

LSR-busy is a signal activated by either the printer or the ledger card device. It is activated under the following conditions:

1. A printer chained command is in process.
2. A printer count command is in process.
3. A ledger card device feed, read ID, and locate next print line is in process.
4. A ledger card device feed, read ID, and eject command is in process, and the ID number has not yet been read into storage.
5. A read all line finder marks command is in process, and all the line finder marks have not yet been read into storage.
6. A ledger card device read mark and eject command is in process, and the line finder mark has not yet been read.
7. A ledger card device index command is in process.

Because the ledger card device and printer share the same print data address register and print command address register, care must be taken when overlapping printer and ledger card device commands. It is permissible to overlap printer and ledger card device commands only when LSR-busy is not active.

Read-ID-busy is activated by the feed, read, and locate next print line, and feed, read, and eject commands. It remains active until the ID number has been read into storage. When read-ID-busy is no longer active, the program can determine the ID number on the ledger card before the ledger card has been completely fed to the print line. Read-ID-busy is also activated when executing the read all line finder marks command until the 43 byte transfer has been completed.

Card-not-aligned indicates that the ledger card is not properly aligned at the pinch point of the first feed rolls. The LCD I/O attention indicator is lighted and the card gate is raised when the command is issued to test for card-not-aligned. No card can be fed into the LCD until the card gate is raised. This condition must be tested for and found to be absent before issuing any of the following commands:

1. Feed, read ID, and locate next print line
2. Feed, read ID, and eject
3. Read all line finder marks

An advance program level instruction for card-not-aligned causes the LCD attention indicator to turn on if the card is not aligned. The indicator turns off when the card-not-aligned condition is absent.

Program Note. Do not use the advance program level instruction to test for unit check, end-of-forms, element at left margin, or last printable line. If any of these conditions is tested by advance program level and is found to be present, the system loops endlessly on the advance program level instruction with no indication to the operator that processing is not continuing in the normal manner.

Example:

Instruction

F1	EE	00

If a ledger card is aligned at the ledger card device alignment station, the system executes the next sequential instruction. If no card is aligned at the alignment station, the system repeats this instruction and turns on the LCD attention indicator.

Load I/O

Mnemonic: LIO

Op Code	Q Byte			Operand Address
Y1	E	M	N	

Operation: The two bytes in storage addressed by the operand address are loaded into the destination specified by the Q byte. The operand is addressed by its rightmost byte.

The Q byte comprises a device address (always E for the printer and the ledger card device) in the high-order four bits, an M bit, and an N code. An M bit of 0 specifies the printer; an M bit of 1 specifies the ledger card device. An M bit of 1 when the ledger card device is not installed causes a processor check stop with the invalid-Q-byte indicator on the CE rotary display turned on.

The N code specifies the destination to be loaded as follows:

Bits	*Destination*
5 6 7	
0 0 0	Locate line address register (ledger card device only)
0 1 0	Bit significant functions in the printer or ledger card device attachments
1 0 0	Print data address register
1 1 0	Print command address register

The bit significant functions for the printer or ledger card device are a single byte loaded from the low-order position of the operand field. The bits within the byte cause the following functions:

Bit	*Function*
0	Diagnostic block stepper high speed or ledger card device diagnostic reset (on = set, off = reset).
1	Force data bus out check.
2	Force diagnostic stepper emitter advance pulse or ledger card device emitter pulse (on = set, off = reset).
3	Force diagnostic print gate emitter vertical advance pulse or ledger card device sense cells off (on = set, off = reset).
4	Bit equal to 1—turn on diagnostic mode. Bit equal to 0—turn off diagnostic mode.
5	Integrated emitter or ledger card device card in switch (on = set, off = reset).
6	Bit equal to 1—set use secondary end-of-forms latch or ledger card device out switch. Bit equal to 0—reset end-of-forms latch or ledger card device out switch.
7	Bit equal to 1—set end-of-forms indicator latch. Bit equal to 0—reset end-of-forms indicator latch.

Bits 0 through 5 are for diagnostic use only and are used by the customer engineer. Bit 6 causes the secondary carriage end-of-forms switch to function in the same way as the primary carriage end-of-forms switch when the secondary carriage runs out of forms. Bit 7 causes the printer I/O attention light on the console keyboard to turn on.

Program Notes:

1. The load I/O instruction with an N code of 010 is executed even if the attachment is busy executing another command. The other load I/O instructions will not be executed if the attachment is busy.
2. A load I/O instruction specifying the locate line address register with an M bit equal to 0 and issued when the ledger card device feature is not installed is ignored, and the next sequential instruction is executed. The load instruction will appear to have been executed but the locate line address register will not be loaded.
3. If a load I/O instruction is given with an M bit equal to 1 and the ledger card device feature is not installed, the processor check stop occurs with the invalid-Q-byte indicator in the CE rotary display turned on.

Example:

Instruction

31	E6	3F	17

Operand

28	29
3F16	3F17

Print Command Address Register before Operation

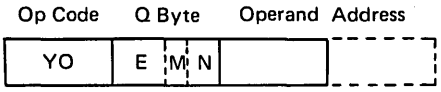
00	26

Print Command Address Register after Operation

28	29

Sense I/O

Mnemonic: SNS



Operation: Data from the specified area is stored in main storage at the location specified by the operand address. Two bytes of data are stored. The operand is addressed by its rightmost byte.

The Q byte contains a device address (always E for the printer and ledger card device) in the high-order four bits, an M bit, and an N code. The M bit specifies whether the sense data is to be taken from the ledger card device or from the printer. An M bit equal to 0 specifies sense data taken from the printer; an M bit equal to 1 specifies sense data taken from the ledger card device.

The N code specifies from where data will be taken for storage in the processing unit. The N code controls this as follows:

Bits	*Source*
5 6 7	
0 0 0	Locate line address register
0 0 1	Locate line address register
0 1 0	Status bytes
0 1 1	Diagnostic device interface signals
1 0 0	Print data address register
1 0 1	Print data address register
1 1 0	Print command address register
1 1 1	Print command address register

Local storage registers used as printer and ledger card device address registers are stored from their appropriate N codes whether the M bit is 0 or 1; however, the M bit controls what specific signals are stored for the status and diagnostic bytes.

When the M code equals 0 and the N code equals 010, status bytes are stored as shown in Figure 4-6. The conditions specified in status byte 1 are defined under "Printer Error Recovery Procedures." The conditions stored in status byte 2 are diagnostic bits only and of no interest to the problem programmer.

When the M bit equals 1 and the N code equals 010, the status and diagnostic bytes for the ledger card device are stored as shown in Figure 4-7. The conditions specified for status byte 1 are defined under "Printer Error Recovery Procedures." The conditions specified by status byte 2 are for diagnostic purposes and are of no interest to the problem programmer.

Bit	Status Byte 2 (High Order)	Status Byte 1 (Low Order)
0	Count End Latch	Horizontal Cycle Check
1	Print Left Command	Data Check
2	Matrix Counter Trigger 1	Margin Check
3	Matrix Counter Trigger 2	Sync Check
4	Matrix Counter Trigger 4	ROS Check
5	Printer Ready	Vertical Cycle Check
6	SS 2	Primary Carriage End of Forms
7	SS 1	Invalid Command Check

Figure 4-6. Printer Status Bytes

Bit	Status Byte 2 (High Order)	Status Byte 1 (Low Order)
0	Sense Amp 1	Sense Amp Check
1	Sense Amp 2	Card Skew Check
2	Sense Amp 3	Drive Check
3	Sense Amp 4	Read Mark Check
4	Timing Pulse	Line Finder Mark Check
5	Drive Check SS	Invalid Command Check
6	Active LCD Clutch	Card-in Switch On
7	Hold Busy SS	Card-out Switch On

Figure 4-7. Ledger Card Device Status Bytes

When the M bit equals 0, the diagnostic device interface signals stored when the N code equals 011 are for diagnostic use by the customer engineer and are of no interest to the problem programmer.

When the M bit equals 1 and the N code equals 011, the diagnostic device interface signals stored are those shown in Figure 4-8. The contents of status byte 1 in this case are for diagnostic use by the customer engineer and are of no interest to the problem programmer.

If the ledger card device is not installed, status byte 1 will contain hexadecimal 00. Bits 0 through 2 of status byte 2 can be used by the problem programmer to determine what printer features are installed on the system.

Example:

Instruction

30	E2	05	70

Two status bytes are stored in main storage at locations 056F and 0570.

Bit	Status Byte 2 (High Order)	Status Byte 1 (Low Order)
0	5213 Printer Attached	Skip Line SS1
1	Not Vertical Forms Control or Enhanced Print Rate Feature	Skip Line SS2
2	Not Bi-directional Feature	Late Mark
3	Secondary Carriage End of Forms	Special Tie Off
4	Not LM SW 2 and not RM SW 1	Card Alignment SS
5	Right Margin Switch 2 or Left Margin Switch 1	Spare
6	Primary or Secondary Forms Motion Contact	Spare
7	Primary Forms Emitter Advance	Stop SS

Figure 4-8. Diagnostic Device Interface Status Bytes

PRINTER OPERATION TIMING

Printing is at 85 characters per second in all models of the 5213 and 2222 Printers. However, printing is:

1. From left-to-right in the 5213 Models 1 and 2 and in the 2222 Model 1.
2. Bi-directional (left-to-right and right-to-left) in the standard 5213 Model 3 and in the 2222 Model 2.

Tab operations occur at a varying rate. For right-to-left tabs of less than 8 spaces, the printing element moves at 8-1/2 inches per second. For right-to-left tabs of 8 or more spaces, the printing element moves at greater than 15 inches per second until fewer than 8 spaces are left. At that time the print element slows to 8-1/2 inches per second.

Left-to-right tabs in the 5213 Model 1 and Model 2 are at 8-1/2 inches per second regardless of the length of the tab. In the 5213 Model 3 and the 2222 Model 1 and Model 2, all tabs are at 15 inches per second until less than 8 spaces are left.

Element return operations occur at greater than 15 inches per second in all printers.

Command execution time is always the greater of the following: either 65 milliseconds or 30 milliseconds plus the execution time. For example, for a print command printing one character, the time is:

$$30 \text{ ms} + \left(\frac{1 \text{ character}}{85 \text{ characters/sec}}\right) = 30 \text{ ms} + 1/85 \text{ sec} = 41.8 \text{ ms}$$

Since 41.8 milliseconds is less than 65 milliseconds, 65 milliseconds will be required to execute the instruction.

Another example is a tab left of 20 characters, which is 2 inches. The execution time is:

$$30 \text{ ms} + \left(\frac{1 \text{ sec}}{8\text{-}1/2 \text{ inches}}\right) \times 0.8 \text{ inches} + \left(\frac{1 \text{ sec}}{17 \text{ inches}}\right) \times 1.2 \text{ inches} = 194.6 \text{ ms}$$

Since 208.6 milliseconds is greater than 65 milliseconds, the execution time for the command is 208.6 milliseconds.

The enhanced-print-rate feature increases per-character print speed to 115 characters per second for fields longer than 5 characters. The appropriate print speed formula is:

$$30 \text{ ms} + \left(\frac{1}{85}\right) \times 5 \text{ characters} + \left(\frac{1}{115}\right) \times \text{number of characters exceeding 5.}$$

For example, for a 20-character field:

$$30 \text{ ms} + 5\left(\frac{1}{85}\right) + 15\left(\frac{1}{115}\right) = 218.7 \text{ ms.}$$

Operation timing for line printing operations using the bi-directional printing feature can be calculated from the formula: time (in milliseconds) = 11.8 x characters per line.

PRINTER ERROR RECOVERY PROCEDURES

Printer error conditions are latched in the I/O attachment and reset by the next printer start I/O instruction. The exact cause of the error can be detected by issuing a sense I/O instruction prior to the next start I/O instruction. The following errors cause unit check:

Horizontal Cycle Check

This check is caused by a failure of the printer to respond to a print, tab, or element-return command within 35 milliseconds. Figure 4-9 illustrates the recommended recovery procedure for a horizontal cycle check.

Programming Note: If a vertical forms motion command is overlapped with the horizontal command, the vertical command is executed if possible. If the commands are overlapped, a check must be performed to determine if a vertical cycle check occurred. If both horizontal and vertical cycle checks occur on the same printer operation, the program must determine whether any count byte belongs to the horizontal operation or to the vertical operation.

Data Check

This check indicates that a parity error was detected in the printer attachment data register. The character in error will not have been printed. It was not the last character to be printed; the character that was to follow the character in error will also have been spaced over by the time the print element stops. Figure 4-10 shows the recommended error recovery procedure for data check.

Margin Check

Margin check turns on and the print element stops moving if a margin switch is encountered in a horizontal count command before the count byte is reduced to zero.

The print element should be reoriented to the left margin. If the left margin status byte is not on, the element return command must be given. Margin check will not occur on element return commands. When margin check is detected, the print command address register may be pointing at the next command or at the count byte.

Sync Check

Sync check is caused by the timing element in the printer getting out of step with the timing elements in the printer attachment. When a sync check is detected, it is stored in a hardware latch. The attachment automatically corrects itself, and the operation continues until normal count end. A print command which resulted in sync check will be properly completed and possibly one character may not be printed correctly. Sync checks can occur on all horizontal commands except element return.

When sync check is detected, the processing unit should perform a program halt and give the operator the option to:

1. Visually check the printout and continue.
2. Use a program checkpoint to restart and reprint only the form in error.
3. Restart the complete job.

If the sync check occurs on a tab command, the operator should be encouraged to continue because the automatic correction function is successful almost 100% of the time.

ROS Check

This check indicates a parity check in one of the seven bytes that make up the print character that is being read from the attachment to be printed. Printing is immediately suppressed and the operation is terminated. If the character in error is not the last character to be printed, the next character following the character in error will also have been spaced over by the time the print element stops.

Error recovery for the ROS check is the same as error recovery for data check (Figure 4-10).

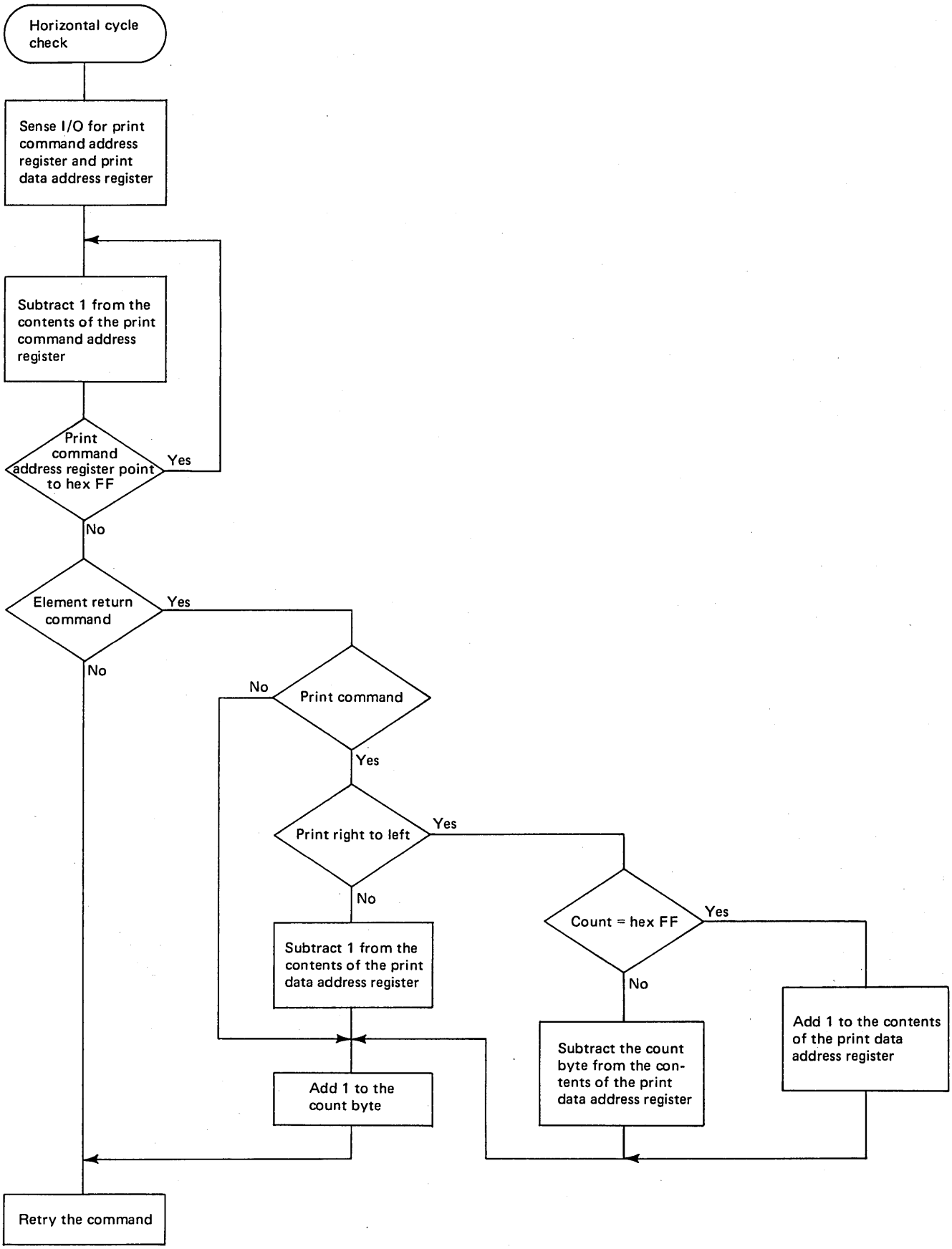


Figure 4-9. Horizontal Cycle Check Error Recovery

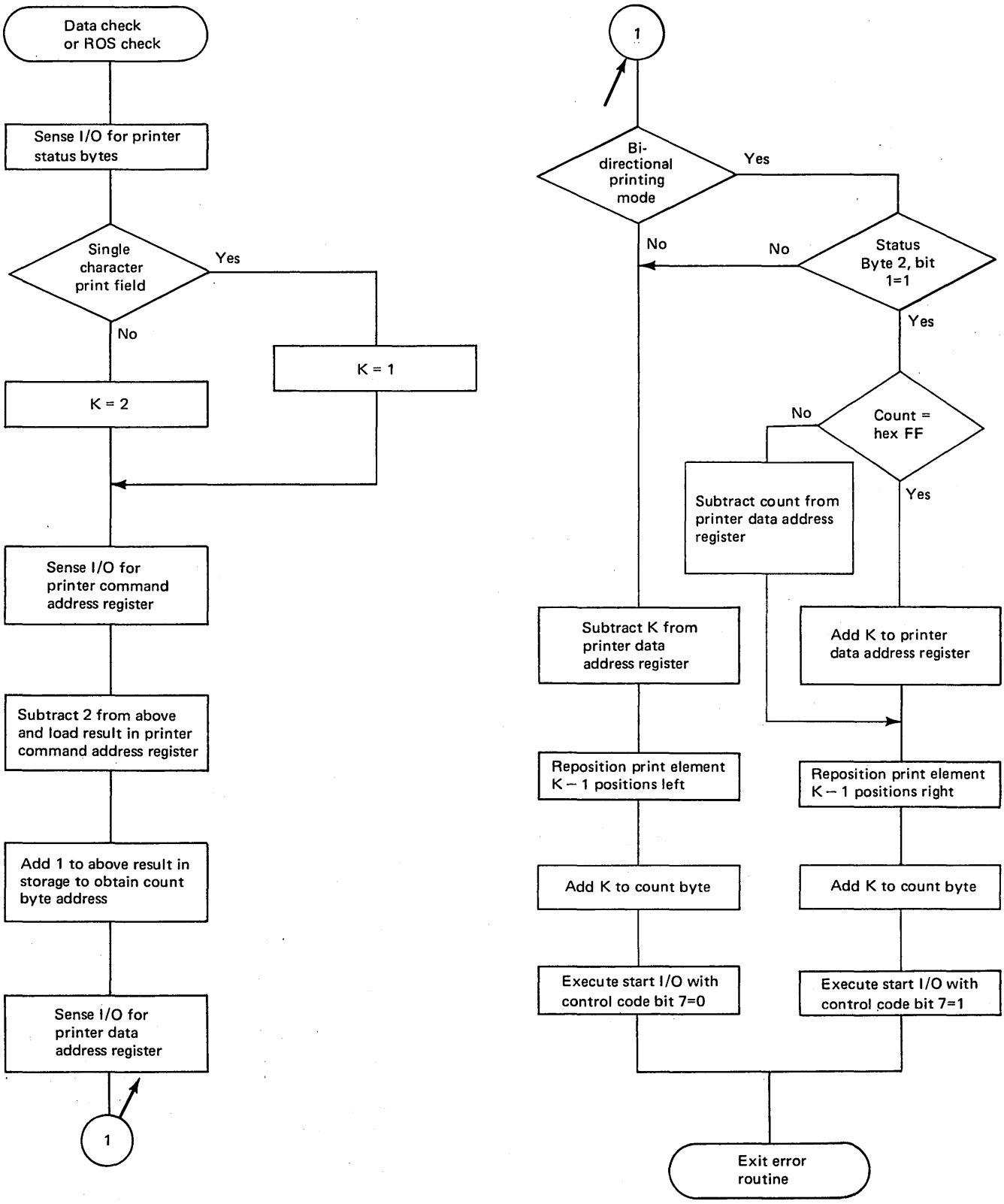


Figure 4-10. Data Check and ROS Check Error Recovery Procedure

Vertical Cycle Check

This check is caused by a failure of the printer to indicate motion within 35 milliseconds (120 milliseconds on printers with pin feed platen) after a vertical action command is initiated by the attachment. This includes primary index, primary skip, and secondary index commands. Figure 4-11 illustrates the recommended recovery procedure.

Program Notes.

1. Because either a secondary carriage or a primary carriage response will imply carriage motion, both primary and secondary carriage operations should not be issued in the same command.
2. If the command byte includes a horizontal motion command, the horizontal command will be executed, if possible. If both horizontal and vertical cycle checks occur during execution of the same command, a check must be made to determine which command, if any, is a count command. If a horizontal command that overlaps the vertical command executes correctly, the retry of the vertical command must mask off the vertical command bits.

Invalid Command

This check indicates that an invalid command was issued to the printer attachment. The invalid command will not be executed and a unit check will immediately occur. The print command address register will contain the address of the byte directly to the right of the command byte that failed. Execute a sense I/O instruction to obtain the contents of the print command address register. Subtract 1 from this and display contents for visual checking.

Data Bus Out Check

This check indicates that data with even parity was sent from the CPU to the printer attachment. It causes the CPU to stop with the processor check indicator turned on.

LEDGER CARD DEVICE ERROR RECOVERY PROCEDURES

Drive Check

This check indicates that the ledger card device was given a command and one of the following conditions occurred:

1. There was no card in the transport.
2. The card entered the transport but failed to continue to the card output tray.
3. No movement took place in the transport.

When a drive check is detected, the program must record the error in the error log and execute a halt program level instruction with an appropriate halt identifier.

The operator removes any card in the transport either manually or by pressing the eject key. After removing the card from the transport, the operator reinserts the card in the ledger card device feed chute and operates the system start key.

Operating this key should cause the program to reinitialize the command field and the printer-command, printer-data, and locate-line address registers, and to execute the start I/O instruction that resulted in the drive check. If repeated failures occur (the number of retries to be determined by the programmer), the program should execute a halt program level instruction with a halt identifier that specifies that the CE should be called.

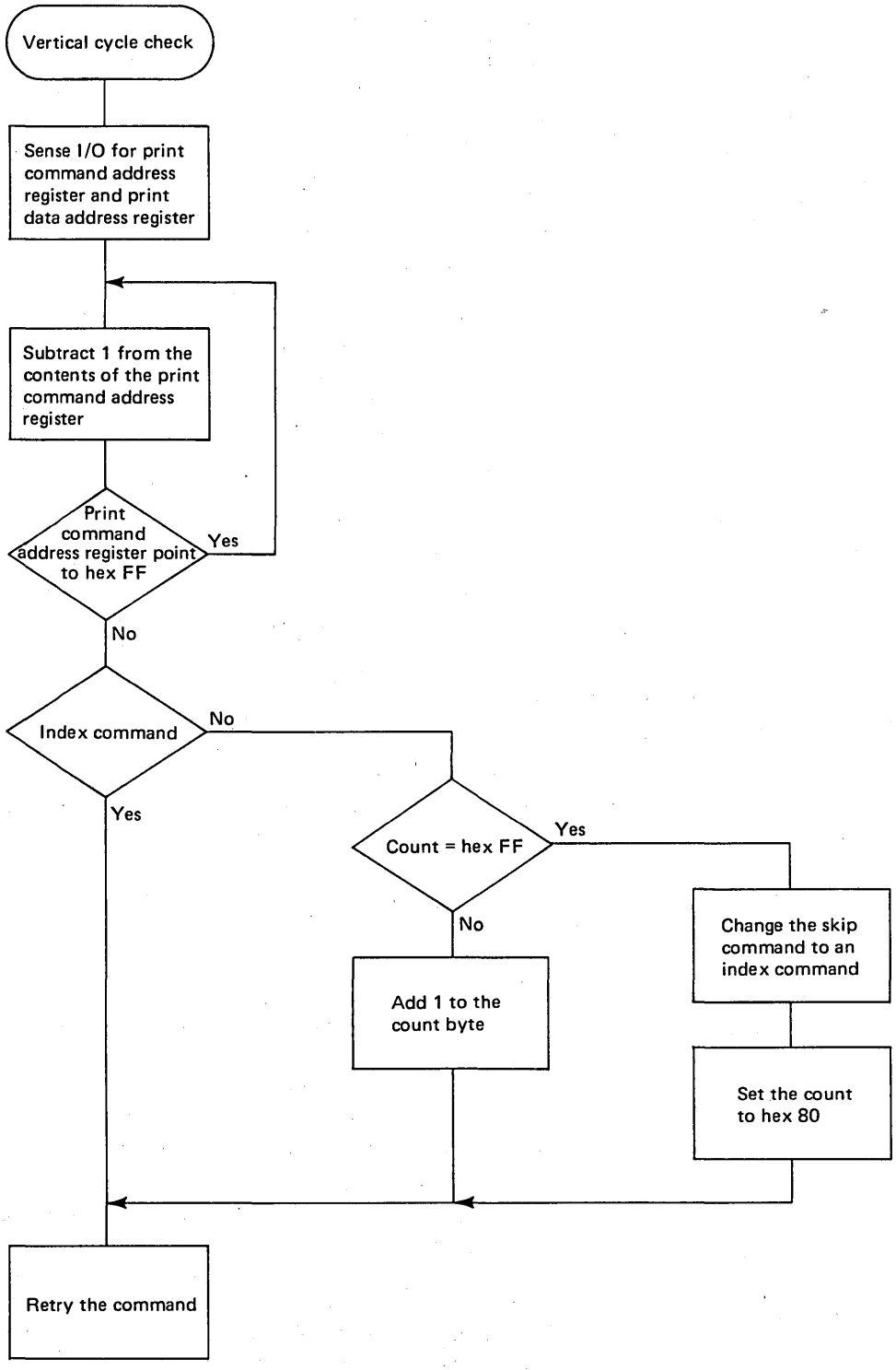


Figure 4-11. Vertical Cycle Check Error Recovery

Sense Amp Check

This check is caused by a failure in the optical sense cells associated with reading the ID number or line finder marks on the ledger card.

When this check is detected, the program must enter the error in the system log and come to a programmed halt with an appropriate halt identifier. An eject command must be issued when the system start key is operated.

Check that the pressure roll release lever is in the run position and that the left-hand card guide is adjusted properly. When these have been checked, eject the card by operating the system start key. Insert the card in the ledger card device feed chute.

When the card has been inserted, the program should reinitialize the command field and the printer-command, printer-data, and locate-line address registers, and retry the start I/O instruction in which the sense amp check occurred. If repeated failures occur, the program should issue a halt program level instruction with a halt identifier that specifies that the CE should be called.

Card Skew Check

This check indicates that the ledger card is out of registration with the feed rolls or the transport.

When this check is detected, the program must enter the error in the system log and come to a programmed halt with an appropriate halt identifier. An eject command must be issued when the system start key is operated.

The operator must check that the pressure roll release lever is in the run position and that the left-hand card guide is adjusted properly. When these have been checked, the card is ejected by operating the system start key. The card is then inserted in the ledger card device feed chute.

When the card has been inserted, the program should reinitialize the command field and the printer-command, printer-data, and locate-line address registers, and retry the start I/O instruction in which the card skew check occurred. If repeated failures occur, the program should issue a halt program level instruction with a halt identifier that specifies that the CE should be called.

Read Mark Check

This check occurs during a read mark and eject command. It indicates that the second read station did not read the last line finder mark printed on the card. The program should issue an eject command. The operator should inspect the line finder marks on the card that is ejected, mark over any weak line finder marks with an IBM ELECTROGRAPHIC ® pencil or equivalent, and put the card in the output tray. If consistent read mark checks occur, check that the printer ribbon is not worn out, preventing printing line finder marks of sufficient density to be read.

Note. Optical sensing is more critical than the human eye; therefore, an apparently "worn out" ribbon used in ledger card operations can still be acceptable for normal printing applications.

Line Finder Mark Check

This check occurs only during a feed, read ID, and locate next print line command. It indicates that after the next available print line was determined, a subsequent line was found to have a line finder mark. The program should eject the card and stop with an appropriate halt ID. The operator should remove any extraneous marks from the card, scribe over any weak line finder marks with an IBM ELECTROGRAPHIC ® pencil or equivalent, and reinsert the card. After the card is reinserted, the program should reinitialize the command field and the printer-command, printer-data, and locate-line address registers, and reissue the start I/O to reposition the card, and reissue the feed, read ID, and locate next print line command. If consistent line finder mark checks occur, check that the printer ribbon is not worn out, preventing printing line finder marks of sufficient density to be read.

Note. Optical sensing is more critical than the human eye; therefore, an apparently "worn out" ribbon used in ledger card operations can still be acceptable for normal printing applications.

Invalid Command

This check indicates that an invalid command was issued to the ledger card device. The invalid command will not be executed and a unit check will occur immediately. The program should reinitialize the command field and the printer-command, printer-data, and locate-line address registers, and come to a programmed halt with an appropriate halt code. The operator must eject the ledger card with the manual eject key and operate the system start key. When the system start key is operated, the program should test for "card aligned," then repeat the start I/O instruction in which the invalid command was encountered.

MAINTENANCE MONITOR

A maintenance monitor, used for diagnostic purposes only, indicates to the service representative when a print head should be replaced. This meter is located in the front center section of the printer and is visible when the printer cover is opened.

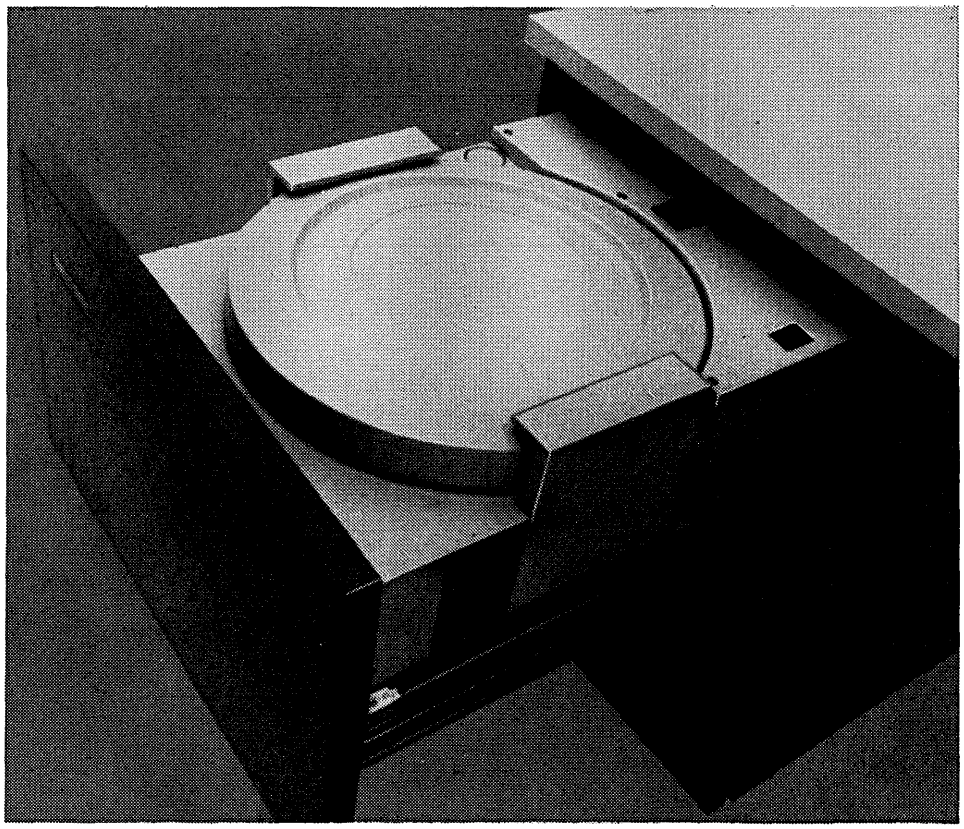
Chapter 5. IBM 5444 Disk Storage Drive

The IBM 5444 Disk Storage Drive (Figure 5-1) is an integral part of the system. It provides the system with direct access storage with capacities ranging from 2,457,600 bytes through 9,830,400 bytes.

The disk drive incorporates one fixed disk and one removable disk. The removable disk is contained in a cartridge (Figure 5-2) that protects the disk when the cartridge is removed from the drive.

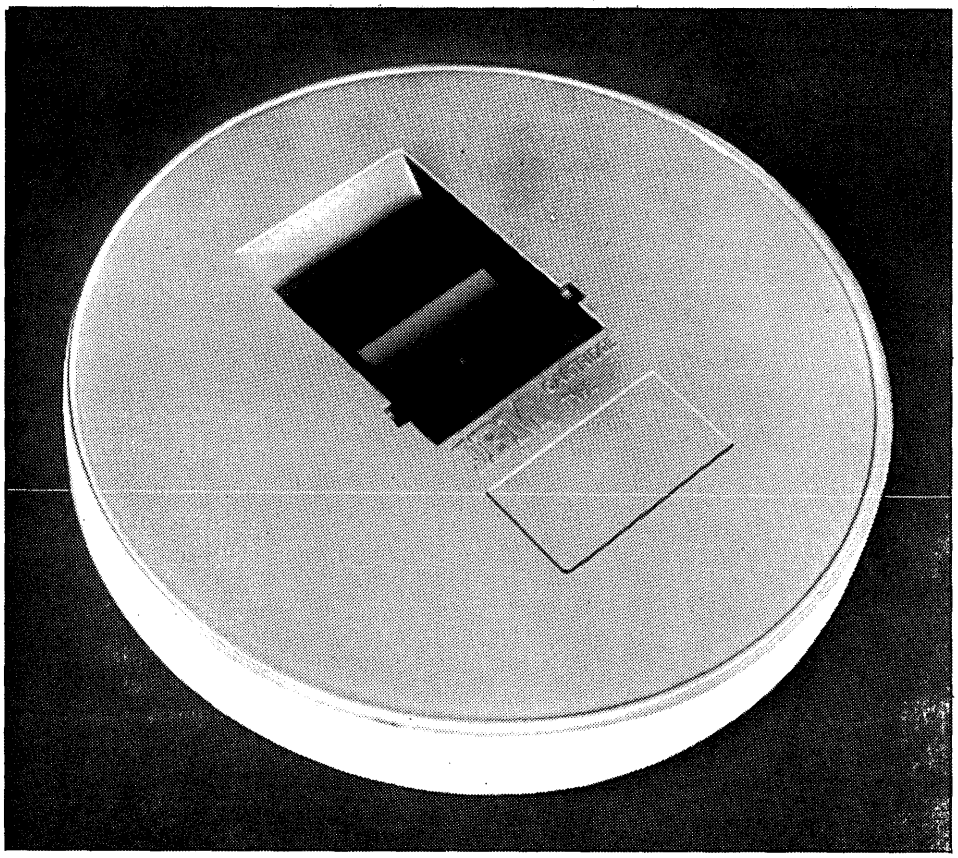
Four configurations can be obtained by combining the three disk drive models. Model 1 has 100 tracks on each surface of each disk for a total capacity of 2,457,600 bytes. Model 2 has 200 tracks on each surface of each disk for a total capacity of 4,915,200 bytes. Model 3 has 200 tracks on each surface of the removable disk only for a total capacity of 2,457,600 bytes. In each model there are 3 additional alternate tracks on each surface. The four possible configurations are:

1. One model 1
2. One model 2
3. One model 2 and one model 3
4. Two model 2s



BR0380

Figure 5-1. IBM 5444 Disk Storage Drive



BR0433

Figure 5-2. Disk Cartridge

FILE ORGANIZATION

Each surface of each disk contains 104 or 204 tracks. The tracks that are related to each other in the vertical plane on a single disk are considered to form a cylinder as shown in Figure 5-3. On drives with two disks (models 1 and 2) the corresponding cylinders on both disks have the same cylinder number. Two hundred of the cylinders on each disk (100 for Model 1) are used for data tracks, 3 are used as alternate tracks, and 1 is used for CE purposes.

Track Format

Each track is divided into 24 sectors as shown in Figure 5-4. Each sector has its own individual address. A sector is made up of an address marker, sector identifier, data field, and some gaps.

Index marker—a mark that is fixed for each disk and provides orientation information to the attachment. It is the starting point for every track.

AM (address marker)—a specially written group of bits used to indicate the start of a new sector.

ID (sector identifier)—a group of bytes containing the unique identification of that sector for that disk.

Data—area of the sector that contains 256 bytes of data and three bytes of check characters.

Gaps—specially written areas on the disk that are used to separate and define the other elements of the sector.

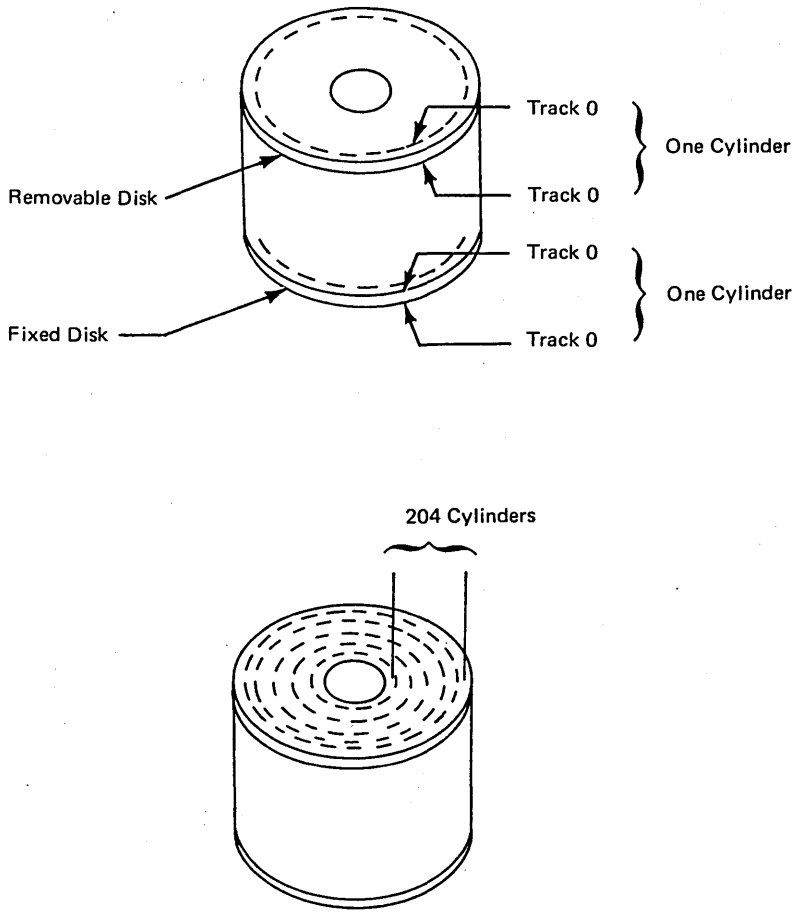


Figure 5-3. Cylinder Concept

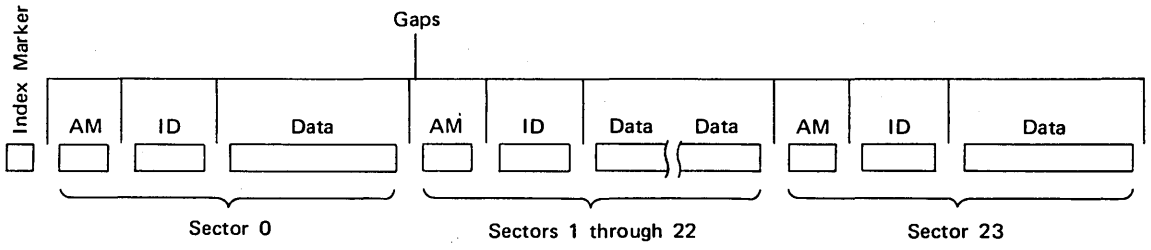
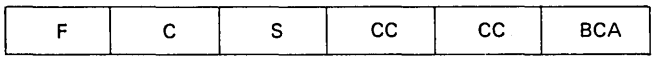


Figure 5-4. Sector Layout

SECTOR IDENTIFIER FORMAT AND ADDRESS

The identifier area of a sector contains 3 bytes of flags and address information and 3 bytes of check information:



F (flag byte)—contains the flagging information in bits 6 and 7. All other bits in this byte should be 0.

C (cylinder byte)—contains the binary number that corresponds to the physical location of the track on the disk.

S (sector byte)—contains the number of the sector. Sectors on top of the disk have sector numbers from 0 through 23. Sectors on the lower surface of the disk have sector numbers from 32 through 55.

CC (cyclic check)—automatically generated and used for checking purposes.

BCA (bit count appendage)—an automatically generated checking byte.

The address of any individual sector is contained in the first three bytes of the identifier. This address applies only to the disk on which the bytes are located. Sectors occupying the same physical location on a fixed disk and on all removable disks have identical binary numbers in the cylinder and sector bytes. Use of the sector requires that the drive and the disk containing the desired sector be specified. Cylinders are numbered 0 through 203. Cylinder 203 is reserved for CE use. Cylinders 1 through 3 are used as alternate cylinders if a track in cylinders 4 through 202 is found defective. Cylinders 0 through 202 are the normal data cylinders.

Sectors within a track are identified by their physical position on the track with relation to the index point and by the surface of the disk on which they reside. The sectors on the upper surface of the disk are numbered 0 through 23 starting with index, and the sectors on the lower surface are numbered 32 through 55. A specific sector, then, is addressed by specifying a drive number, fixed or removable disk, a cylinder number, and the sector number.

DISK OPERATING RESTRICTIONS

The disk file drives operate only when a removable disk is properly installed on the drive and the disk file drive drawer is in the closed position. The drawers cannot be opened unless power is first removed from the drive by moving the disk drive 1 or disk drive 2 switch to off while system power is on. Drawers should be kept closed at all times unless a disk cartridge is being inserted or removed. The 5440 Disk Cartridge must be stored in the same operating environment as that of the system for at least 2 hours before the cartridge is mounted for processing. This will assure that the disk cartridge will have time to adjust to the temperature and humidity conditions in which it will be used.

FILE OPERATIONS

Two operations must be performed to prepare for each file operation. The address of the file control field must be stored in the disk file control address register (one of the local storage registers) and the address of the high-order byte of the disk file data field must be stored in the disk file data address register.

The file control field consists of 4 bytes designated F byte, C byte, S byte, and N byte. The bytes are used as follows:

Byte	*Use*
F	This is the first byte in the field and the byte addressed by the disk file control address register. In seek operations this byte is not used. In other file operations it contains the flag bits in bits 6 and 7.
C	This second byte of the field contains a binary number that designates a cylinder number. This byte contains the same kind of information for all operations.
S	This third byte of the field contains a sector number (binary) in bits 0 through 5 and uses bit 7 as a seek control. In seek operations bit 0 of this byte determines whether, in succeeding read, write, or scan operations, the track on the upper surface of the disk is used or the track on the lower surface of the disk is used. When bit 0 is equal to 0, the upper track is used; bit 0 equal to 1 causes the lower track to be used. The access head that reads the upper track is called head 0; the access head for the lower track is head 1. For seek operations bit 7 of this byte specifies the direction of the seek; bit 7 equal to 1 specifies seek in the direction of increasing cylinder number; bit 7 equal to 0 specifies seek in the direction of decreasing cylinder number. For any operation other than seek, bit 7 of this byte must be 0.
N	This last byte in the field specifies either the number of cylinders to move the access mechanism for a seek operation or the number of sectors to operate on for any other operation. For operations other than seek, this binary number must be 1 less than the actual number of sectors desired.

Seek

The access mechanism of the selected drive is moved a specified number of cylinders and the upper or lower head for the specified disk is set for future read, write, or scan operations. The number of cylinders to be crossed and the head to be set are specified by the disk control field as described before.

The N byte specifies the number of cylinders the access mechanism will travel during the seek.

Bit 7 of the S byte specifies the direction of movement; forward (bit 7 equal to 1) is from cylinder 0 toward cylinder 202. The head is specified by bit 0 of the S byte.

The recalibration function is executed by specifying (1) a seek in the reverse direction and (2) a number of cylinders to be moved that is greater than or equal to 224. The recalibrate function causes the access mechanism to seek to cylinder 0, and select head 0, regardless of the S byte in the control field.

The cylinder 0 bit in the sense byte is set when the mechanism reaches cylinder 0 and can be interrogated with a sense I/O instruction after the seek is completed.

Seek operation is begun by issuing a start I/O instruction. A second start I/O instruction can be issued to the same disk drive if the start I/O specifies a read, write, or scan operation. The second instruction will be accepted provisionally and executed if no errors occur in the operation of the seek instruction.

No information in storage is changed by this operation. Test I/O for busy or advance program level on busy will not detect busy unless a read, write, or scan instruction has been provisionally accepted. The sense bit for seek busy will be on, however, for interrogation by the sense I/O instruction. A seek instruction to an access mechanism that is already seeking results in the processing unit performing a one-instruction loop on the seek instruction until the access mechanism is no longer busy seeking. An attempt to seek to the cylinder at which the access mechanism is located results only in the transfer of the disk control field to the attachment because no access mechanism motion is required.

In addition to positioning the heads over the desired track, the seek instruction provides the *only* means of selecting the head that will be used for succeeding read, write, or scan operations. If an operation is performed with one head, and the succeeding operation is to be performed by the other head in the same cylinder, a no-movement seek can be used to switch heads. A no-movement seek is a seek to the cylinder at which the access mechanism is located.

A typical application of this principle occurs when a write operation beginning with a sector address using head zero specifies a number of sectors that requires head switching and writing the last sectors with head one. A no-movement seek is required to switch heads back to head zero before a verify operation can begin at the first sector written.

Seek Time

Figure 5-5 shows the approximate time required to seek across any number of tracks from 10 through 200. Seek time can also be determined from the following formula:

Seek time for 1 track = 39 ms
Seek time for two or more tracks = 47 ms + 3.42 (N-2) ms
(where N = number of tracks to be crossed)

Note: Because of tolerances in the drive mechanisms, as much as 3.55 milliseconds may be required to cross each track seeking in one direction; however, a drive that requires 3.55 milliseconds in one direction will not require more than 3.29 milliseconds to cross each track in the opposite direction. Therefore, 3.42 milliseconds is the average maximum track crossing time after two tracks have been crossed.

Read Data

This instruction initiates the transfer of data from the selected disk to main storage. Data is transferred in multiples of 256 bytes (the contents of an individual file sector). The entire contents of a cylinder can be read if reading is started with sector 0. Only consecutive sectors are read when multiple sector reading is indicated.

Reading begins with the sector specified by the S byte of the disk control field in main storage. The data is transferred to main storage starting at the address stored in the disk file data address register, and succeeding bytes are transferred to progressively higher locations. The disk file data address register is continuously updated so that it points to the storage address where the next byte is to be stored.

When the disk control field specifies that more than one sector is to be read, the sector address is updated each time a sector is read. Disk sector addresses cannot overflow from disk to disk because each disk contains the same addresses on the same cylinders.

The disk control field in storage is compared with the sector identifier fields on the disk track to find the first sector to be read. The comparison is repeated for each additional sector to be read. If the disk control field and the sector identifier fail to match, the operation ends after the data portion of that sector is transferred to main storage even if other sectors remain to be read.

During a read operation, the attachment generates two cyclic check (CC) bytes and a one-byte bit count appendage from the data that has been read and compares these to the CC bytes and bit count appendage read back with the data. This comparison provides a data check for errors. During multiple sector reads, the operation will be terminated at the end of any sector in which an error is detected.

Two other abnormal conditions cause termination of the read operation. Reading is terminated at the end of any sector if the sector read is the last sector (sector 55) in the cylinder. Equipment check causes an immediate termination without waiting for the end of the sector.

The read operation ends when the N byte of the disk control field reaches FF and the data from that sector has been transferred. The number in the N byte is decremented by one at the beginning of each sector transferred.

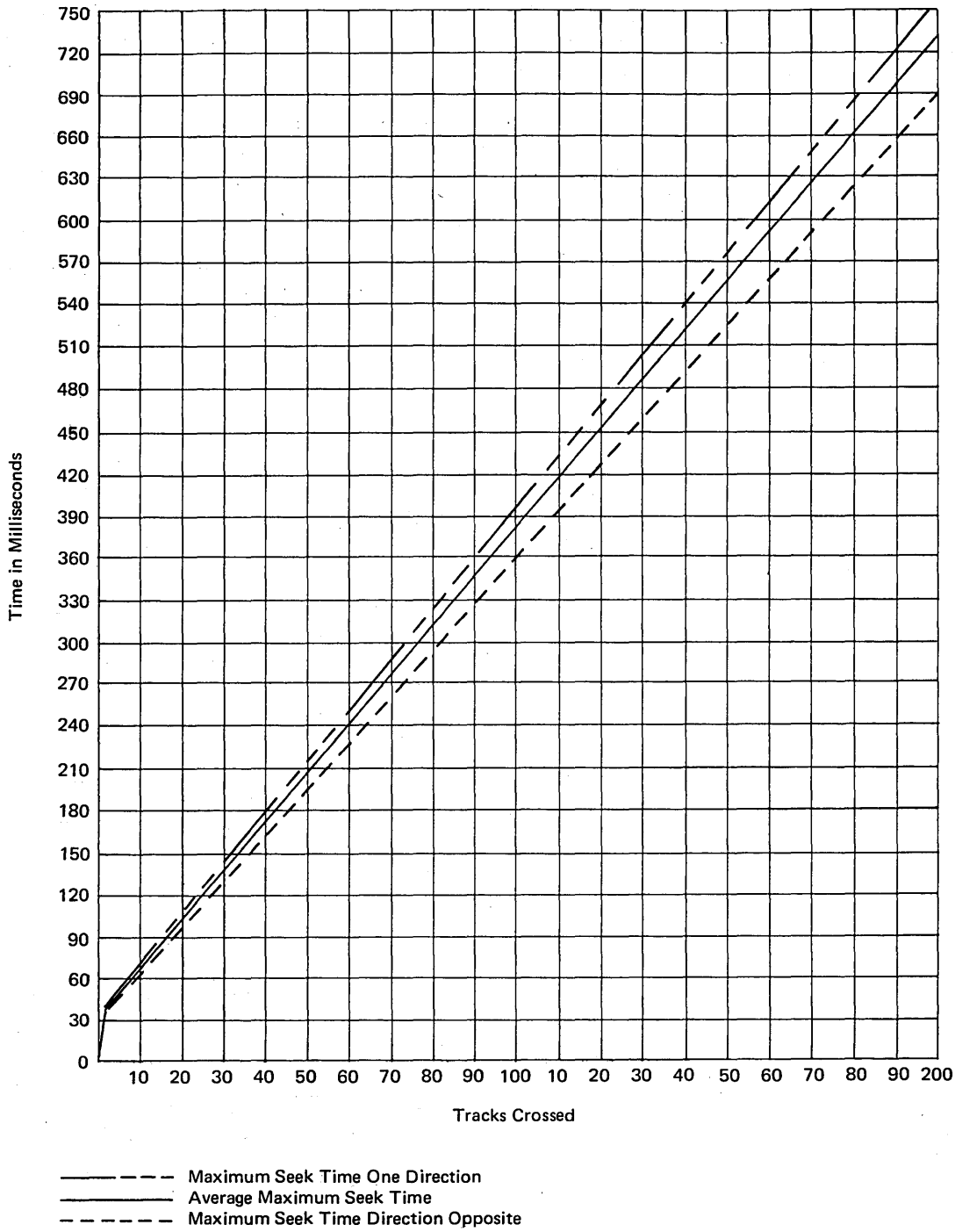


Figure 5-5. Seek Timing

At the end of the operation the four bytes of the disk control field contain information about the progress of operation. The number of sectors processed (or attempted to be processed) is equal to the original value of the N byte minus the value of N at the end of the operation unless all sectors requested have been processed. If all sectors have been processed, the value of N at the end of the operation is FF. At the end of the operation, the S byte of the disk control field contains the identifier of the last sector processed, unless there is a missing address marker on the disk or no sector could be found with an identifier that matched that in the disk control field. If no sector has been processed, the S byte in the disk control field is the S byte of the first sector desired. If an address marker is missing and a sector has been processed in a multi-sector operation, the S byte in the disk control field is that of the sector that lacks an address marker.

The file control unit is busy to all other operations except sense I/O during a read data operation.

Read Identifier

This operation transfers the sector identifier (F, C, and S bytes) from the selected disk to storage. The operation starts with the first identifier to come under the head after the instruction is executed. The operation transfers the first sector identifier found to the address designated by the disk file control address register. If an error is found in this identifier, the next sector identifier is read and transferred to storage starting at the original address contained in the disk file control address register. The operation is terminated by the transfer of the first sector identifier found without an error, by no record found, or by equipment check.

The file control unit is busy to any new operation except sense I/O while the read identifier operation is being performed.

The information contained in the disk control field at the beginning of this operation is not used but is destroyed by the information read in from the disk. At the end of this operation the first three bytes (F, C, and S) of the disk control field contain the last sector identifier read from the file. The last (N) byte of the disk control field is not changed. This operation cannot switch reading between the upper and lower surfaces of the disk.

At the end of the operation the disk file control address register contains the original address unless there is an equipment check.

Read Diagnostic

This operation is similar to a read data operation. Reading always begins at the index. Up to 48 sectors can be read (the entire contents of the cylinder) but no more than 24 sectors should be read. Exceeding the 24 sector limit increases the chances of reading the wrong data field into storage.

The data portion of the record is read and placed in storage beginning at the address specified in the disk file data address register. One is subtracted from the N byte and is added to the S byte of the disk control field for each sector read. The data address in the disk file data address register is returned to its original value at the beginning of each sector so that successive data fields overlay each other in storage.

The operation ends at the end of the sector in which the sector count is reduced to FF, the end of the cylinder is detected, or equipment check is detected. No other error conditions terminate the operation. When the operation is terminated, the contents of the data area of the last sector read are in the data area of main storage.

Read diagnostic operates with reduced address marker requirements so that data that cannot be read by a read data operation because of a missing address marker can be recovered.

The original sector identifier in storage (F, C, and S bytes of the disk control field) should be the identifier of the first record on the track, so that the identifier area in storage at the end of the operation contains the identifier of the last record read unless there is no record found without a data check. No record found without a data check or a track condition check indicates that an address marker is missing earlier on the track.

The error conditions indicated apply only to the last sector read. Any detected errors that do not cause the operation to end before the number of sectors requested has been read are reset at the end of the sector in which they are detected.

The number of sectors read can be determined by subtracting the N byte of the disk control field from the original value of the N byte unless all sectors have been read. If all sectors have been read, the N byte is set to FF.

The control unit is busy to any new operation, except sense I/O, while performing a read diagnostic operation.

Read IPL

This operation is initiated by operating the program load switch on the keyboard-console. Initial program loading will be performed from the disk on disk drive one specified by the position of the disk select switch on the keyboard-console. The read IPL operation causes the 256 bytes of data contained in the first record after the index mark on track zero of the selected disk to be transferred to storage starting with storage address 0000. Control is then passed to the processing unit and instruction execution begins, starting at address 0000.

No compare is made on the identifier of the first record; the first record found after the index mark is read and any error conditions are made available for program testing. If no record is found or the wrong record is read, the program will not start correctly. An unsuccessful IPL operation requires an operator retry.

A test I/O and branch instruction should be performed to test for errors before attempting the first start I/O instruction.

Verify

A start I/O instruction initiates this operation. The verify operation is performed for write checking. It must be performed after every write operation. This operation is performed in the same way as the read data operation except that no data is transferred to main storage and the disk file data address register is not updated. No cycle steals are required except for updating the sector and number bytes in the disk control field.

The function of write checking is done by generating the cyclic check and bit count appendage characters read from the disk.

At the end of the operation the disk control field contains information about the progress of the operation. The sector byte of the disk control field indicates the last sector verified. The number of sectors verified can be determined by subtracting the contents of the N byte of the disk control field from the original value of the N byte unless all sectors have been verified. If all sectors have been verified, the N byte contains FF.

Write Data

This operation transfers data from storage to the selected track on the disk. The data is transferred in multiples of 256 bytes. The entire data content of a cylinder can be written (48 sectors) if writing starts with head 0, sector 0. Only consecutive sectors can be written by multiple-sector write operations.

Writing begins with the sector specified by the identifier portion (F, C, and S bytes) of the disk control field located in storage and addressed by the disk file control address register. The identifier from the disk control field is compared with the sector identifiers read from the selected disk track. Comparing begins with the first sector identifier to come under the head. An equal condition between the disk control field identifier and the sector identifier enables the writing of the 256 bytes of sector data. The data is fetched from storage using the disk file data address register for addressing.

When multiple sectors are indicated, one is added to the S byte and subtracted from the N byte of the disk control field for each sector written (except when transferring heads, which requires that 9 be added to the S byte). This updated disk control field identifier is then compared to the next identifier read from the disk. An equal comparison of all succeeding addresses must occur before their corresponding data fields are written on the disk. The data field of a sector will not be written if an error is found before writing of data begins.

The write data operation is ended at the end of the sector in which the count byte (N byte) is reduced to FF, the end of the cylinder is reached, or a check condition occurs. An equipment check terminates the operation immediately. The presence of an error can be detected by a test I/O and branch instruction.

The file control unit is busy to any instruction except sense I/O while it is performing a write data operation.

During writing, the control unit generates two cyclic check and one bit count appendage characters for each data field. The three characters are recorded at the end of the data field. Write errors must be checked for with a verify operation in order to meet file performance specifications.

At the end of the operation the disk control field contains information about the progress of the operation. The identifier portion of the disk control field indicates the last sector written or where writing was attempted. The number of records processed can be determined by subtracting the contents of the N byte from the original value of the N byte unless all sectors were written. If all sectors were written, the N byte contains FF.

Write Identifier

This operation is initiated by a start I/O instruction. The operation writes 24 sector formats (address marker, sector identifier, gaps, and data) on the selected track beginning at the index marker. There is no identifier field compare on a write identifier instruction before writing.

The identifier portion of the disk control field is written as the sector identifier of the first sector after the index marker. The N byte of the disk control field is forced to a value of decimal 23 by this operation so that exactly 24 sectors are written on the track.

As each identifier is written on the disk, 1 is added to the S byte and subtracted from the N byte of the disk control field. The data field of each sector is filled with a character stored at the address contained in the disk file data address register. The disk file data address register is not updated during the operation so the same character is propagated in all data byte positions of the sectors.

During writing of each identifier and data field, the control unit generates two cyclic check and one bit count appendage bytes and automatically writes them as the last 3 bytes of both the identifier and the data fields. The check data for the identifier applies only to the identifier, and the check data appendage to the data applies only to the data.

At the end of the operation, the disk control field contains information about the progress of the operation. The identifier portion of the disk control field indicates the last sector written or where writing was attempted. The number of records processed can be determined by subtracting the contents of the N byte of the disk control field from the original value of the N byte unless all records have been processed. If all records have been processed, the N byte contains FF.

The file control unit is busy to all new operations except sense I/O during a write identifier operation.

A verify operation must check for write errors following each write identifier operation in order to meet file performance specifications.

Scan

Scan operations are initiated by a start I/O instruction. The scan operation searches the data fields on the disk to find one that meets certain specified conditions when compared to a sector-sized data field in storage. Up to one cylinder of data (48 sectors) can be scanned in one operation. The scan operation can specify one of the following conditions to satisfy the scan:

1. Equal
2. Low or equal
3. High or equal

The data in the sectors on the disk is compared with the 256 characters in the disk data field in storage. The disk data field is addressed by the disk field data register. The comparison of individual characters within the sector can be masked off by placing a mask character consisting of all bits (hexadecimal FF) in each non-compare byte in the disk data field in storage. If only 10 bytes are to be compared, the field must contain 246 mask characters in the byte positions of the characters that are not to be scanned.

Scanning of the data begins with the sector specified by the identifier portion of the disk control field. Comparing of sector addresses begins with the first sector identifier to come under the head. After the beginning sector is scanned the S byte is updated to the identifier of the next sector and the N byte is decreased by 1 for each sector scanned.

The operation ends under the following conditions:

1. When the data on the disk satisfies whichever condition is specified by the start I/O instruction—
   a. Equal to the storage data field
   b. Equal to or lower than the storage data field
   c. Equal to or higher than the storage data field
2. At the end of the sector in which the sector count in the N byte of the disk control field goes to FF.
3. When the end of the cylinder is reached.
4. At the end of any sector in which an error occurs after the first identifier specified by the disk control field has been found.

The control unit is busy to any new operations except sense I/O while performing a scan operation.

A scan found condition is indicated to a test I/O and branch or advance program level instruction. The appropriate bit in the status byte is also set by a scan equal condition.

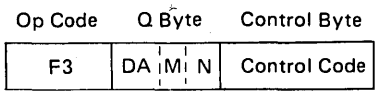
At the end of the operation the disk control field contains information about the progress of the operation. The identifier portion contains the sector identifier of the last sector scanned unless there is a missing address marker. If there is a missing address marker, the identifier portion indicates the sector with the missing address marker. If no sector has been scanned, the identifier portion indicates the first sector designated. The number of sectors scanned can be determined by subtracting the contents of the N byte from the original value of the N byte unless all sectors have been processed. If all sectors have been processed, the N byte is hexadecimal FF.

The disk file data address register contains the original address at the end of the operation unless equipment check occurs. The register contains the address of the last character processed in the event of an equipment check.

DISK FILE INSTRUCTIONS

Start I/O

Mnemonic: SIO



Operation: This instruction is used to select a drive and disk and to specify the operation that is to be performed by that drive and disk.

The DA portion (4 bits) of the Q byte specifies the drive that is to be used. Hexadecimal A specifies drive 1, and hexadecimal B specifies drive 2.

The M bit of the Q byte specifies the disk on the specified drive that is to be used. M bit equals 0 specifies the removable disk; M bit equals 1 specifies the fixed disk.

The N code of the Q byte and bits 6 and 7 of the control code byte specify the operation to be performed. Bits 0 through 5 of the control code byte are ignored and can be anything. Bits 0 through 5 of the control byte should be 0's. The operations that can be specified are:

N Bits	*Control Bits 6 and 7*	*Operations*
000	00	Seek
001	00	Read data
001	01	Read identifier
001	10	Read diagnostic
001	11	Verify
010	00	Write data
010	01	Write identifier
011	00	Scan equal
011	01	Scan low or equal
011	10	Scan high or equal

Any N code other than those specified causes the processing unit to stop with a processor check and an invalid-Q-byte indication.

Issuing a start I/O instruction to a control unit that is busy causes the program to loop on the start I/O instruction until the unit is no longer busy. Issuing a seek start I/O instruction to a drive that is seeking, or issuing a seek start I/O instruction to a drive that is not ready causes the program to loop on the start I/O instruction until the condition is corrected.

A single start I/O specifying a read, write, or scan will be provisionally accepted by the control unit for later execution if the selected drive is executing a seek. If error conditions are set at the end of the seek when a read, write, or scan has been provisionally accepted, the read, write, or scan is no-oped and the no-op bit in the status bytes is set.

Note: If for any reason the seek ends before a read, write, or scan start I/O is issued, the start I/O instruction will reset the error conditions. This may happen when a seek is terminated prematurely by a seek check.

A seek instruction on one drive can be overlapped with a seek on another drive. A read, write or scan on one drive can be overlapped with a seek on the other drive if the seek is issued first. Overlap will not occur if the seek is issued during a read, write, or scan.

The start I/O instruction uses the contents of the disk file data address register as the initial address of all sector data fields. It uses the contents of the disk file control address register as the address of the disk control field.

A start I/O addressed to an unsafe drive is no-oped and the no-op bit in the status byte is set.

Any start I/O that is executed resets any previously generated device status except seek check, equipment check (unsafe), cylinder 0, and no-op. Seek check is reset by start I/O only if it is associated with the drive that is addressed. Equipment check is not reset by any instruction (it can be reset by turning the disk drive power off for a moment). No-op is reset by a sense I/O instruction that enters this status bit into storage. Cylinder 0 is reset when the access is no longer on cylinder 0.

Example:

Instruction

F3	A1	00

Disk File Identifier Address Register

02	00

Disk File Data Address Register

0F	00

Disk Control Field

00	07	A0	01
0200	0201	0202	0203

512 bytes of data will be transferred to storage and placed in locations 0F00 through 10FF.

Load I/O

Mnemonic: LIO

Op Code	Q Byte	Operand	Address
Y1	DA M N		

Operation: This instruction loads the two bytes of data contained in the operand addressed by the operand address into a local storage register specified by the Q byte. The operand is addressed by its low-order byte.

The Q byte specifies the drive that is to operate in the first four bits (device address DA) and the local storage register to be loaded in the last three bits (N code). The M bit is not used.

The device address portion of the Q byte can take either of two values: A for drive 1 or B for drive 2.

The N code can specify only three values:

1. An N code of 011 is reserved for CE use.
2. An N code of 100 specifies the disk file data address register.
3. An N code of 110 specifies the disk file control address register.

Any N code other than the ones specified causes the processing unit to stop with a processor check and an invalid-Q-byte indication.

A load I/O instruction issued to a busy control unit causes the program to loop on the load I/O instruction until the control unit is no longer busy.

Load I/O does not set any file status conditions.

The load I/O instruction is executed if the addressed drive is executing a seek or recalibrate operation and a read, write, or scan has not been accepted or provisionally accepted. The load I/O instruction is executed if the addressed drive is not ready.

Example:

Instruction

31	B4	2F	B1

Operand

0F	00
2FB0	2FB1

Disk File Read/Write Address Register before Operation

0B	20

Disk File Read/Write Address Register after Operation

0F	00

Test I/O and Branch

Mnemonic: TIO

Op Code	Q Byte			Operand	Address
Z1	DA	M	N		

Operation: This instruction tests for the conditions specified in the Q byte. If the condition tested for is present, the next instruction is taken from the storage address specified by the operand address and the address of the next sequential instruction is placed in the address recall register. If the condition is not present, the next sequential instruction is executed and the address contained in the operand address is stored in the address recall register. The information stored in the address recall register remains there until the next decimal, insert-and-test-characters, or branch instruction is executed.

The Q byte specifies the drive to be tested and the conditions to be tested for. The device address (DA) portion of the Q byte specifies the drive to be tested and can take on two values: hexadecimal A indicating drive 1 and hexadecimal B indicating drive 2.

The N code of the Q byte can specify testing for any of three conditions:

1. N code 000 (not ready/check)—indicates that the drive is not in condition to operate or that a check condition has been detected. A check condition is indicated when either drive is addressed if the following device status is present.

   Data check
   Track condition check
   Missing address marker
   End of cylinder
   No record found
   Equipment check not caused by unsafe
   No-op
   Overrun

   A check condition is also indicated if a seek check or unsafe exists for the addressed drive. A seek check or an unsafe for the drive not addressed will not be indicated. The drive that has the check condition can be determined from the status byte.
2. N code 010 (busy)—control unit either is executing a read, write, or scan operation or has provisionally accepted one of these operations for execution at the end of the seek operation in progress.
3. N code 100 (scan found)—indicated when either drive is addressed. The sense byte indicates which drive contains the scan found condition. The scan found indication is reset by the next start I/O instruction.

Any N code other than those listed causes the processing unit to come to processor check stop with an invalid-Q-byte indication.

The M bit is used when addressing the second drive because it is possible that the fixed disk is not installed. Addressing an uninstalled fixed disk results in a not ready indication.

Example:

Instruction

C1	A4	02	00
0100	0101	0102	0103

Status Byte 1

10000000

Instruction Address Register before Operation

01	04

Address Recall Register before Operation

2F	C7

Instruction Address Register after Operation

02	00

Address Recall Register after Operation

01	04

Advance Program Level

Mnemonic: APL

Op Code	Q Byte			
F1	DA	M	N	Not Used

Operation: This instruction tests for the conditions specified in the Q byte. If the condition tested for is present, the system loops on the advance program level instruction until the condition no longer exists. If the condition is not present, the system takes the next sequential instruction.

The Q byte specifies the drive to be tested and the condition to be tested for. The device address (DA) portion of the Q byte specifies the drive to be tested and can assume either of two values: hexadecimal A indicating drive 1 or hexadecimal B indicating drive 2.

The N code of the Q byte can test for any of the following three conditions:

1. N code 000 (not ready/check)–indicates that the drive is not in condition to operate or that a check condition has been detected. A check is indicated when either drive is addressed and the following device status is present.

   Data check
   Track condition check
   Missing address marker
   End of cylinder
   No record found
   Equipment check not caused by unsafe
   No-op
   Overrun

   Check condition is also indicated if seek check or unsafe exists for the addressed drive. Seek check or unsafe for the drive not addressed will not be indicated. The drive with the check condition can be determined from the status byte.
2. N code 010 (busy)–one of the drives either is executing or has accepted provisionally for later execution a read, write, or scan operation.
3. N code 100 (scan found)–indicates when either drive is addressed and a scan has been matched in one of the drives. The sense byte indicates which drive contained the scan found condition. Scan found indication is reset by the next start I/O instruction.

Any N code other than those listed causes a processor check stop with an invalid-Q-byte indication.

The M bit of the Q byte is used when addressing the second drive because it is possible that the fixed disk is not installed. Addressing an uninstalled fixed disk results in a not ready indication. The third byte of the instruction is not used.

Example:

Instruction

F1	A2	00

The program continues to execute this instruction until drive 1 is no longer busy.

Sense I/O

Mnemonic: SNS

Op Code	Q Byte			Operand	Address
Y0	DA	M	N		

Operation: This instruction causes the two bytes contained in the specified local storage register or the specified two bytes of status information to be transferred to the two-byte field in storage addressed by the operand address. The operand is addressed by the low-order byte.

The Q byte specifies the drive to be sensed and the register or status bytes to be transferred. The device address (DA) portion of the Q byte specifies the drive to be sensed. The device address can be either of two hexadecimal values: A specifies drive 1, and B specifies drive 2. The N code specifies what is to be transferred to storage as follows:

1. N code 010–status bytes 0 and 1
2. N code 011–status bytes 2 and 3
3. N code 100–disk file data address register
4. N code 110–disk file control address register

Any N code other than those specified above causes a processor check stop with an invalid-Q-byte indication.

The status bytes are bit significant as illustrated in Figure 5-6. The higher numbered status byte is stored in the low-order position of the field. An explanation of each status bit is provided in "Check Conditions and Status."

The M bit is used in this instruction to determine if the intervention required status bit will be set if the second drive is addressed and the fixed disk is not installed.

The sense I/O instruction will be accepted by the file control unit no matter what other operations are in progress at the time.

Some bits of the status bytes are drive sensitive to the sense I/O instruction. Equipment check caused by unsafe, cylinder 0, seek check, seek busy, intervention required, unsafe, head settling, and index are set in the status bytes only when they apply to the drive addressed by the sense I/O instruction.

All remaining status bits are presented with the status bytes to a sense I/O for either drive. All status bits except no-op are reset by the next start I/O instruction issued to either drive. No-op is reset by the sense I/O instruction to either drive that transfers it to storage.

Example:

Instruction

30	A2	05	FF

Status Bytes at File before Operation

81	00

Operand before Operation

00	AB
05FE	05FF

Operand after Operation

81	00
05FE	05FF

Status Bytes at File after Operation

81	00

CHECK CONDITIONS AND STATUS

All file check conditions as well as general status information about the file are conveyed to the processing unit as bits in status bytes. Each bit in each status byte has a special significance.

Status Byte 0

Bit 0—No-op

This status indicates that the last file instruction issued was not executed. It is caused by the selected file being unsafe or by a check condition occurring during a seek on a drive that has provisionally accepted a read, write, or scan instruction. This bit is reset by check reset, system reset, or the sense I/O instruction that transfers the bit to storage.

Bit	Byte 0	Byte 1	Byte 2
0	No-Op	Scan Equal Hit	Unsafe
1	Intervention Required	Cylinder Zero	Tap Line A
2	Missing Address Marker	End of Cylinder	Tap Line B
3	Equipment Check	Seek Busy	Tap Line C
4	Data Check	100 Cylinder	Index
5	No Record Found	Overrun	Head Settling
6	Track Condition Check	Status Address A	Jumpered CE Bit
7	Seek Check	Status Address B	Model 6

Byte 3 is for CE diagnostic use and has no meaning to the I/O control program.

Figure 5-6. File Sense Byte Information

Bit 1—Intervention Required

This bit indicates that the addressed drive is not ready (removable disk not installed, power not on, drawer not closed, etc.). Addressing drive 2 in a system with only one drive or addressing the fixed disk on drive 2 when only the removable disk is installed also causes this indication. This bit is reset by correcting the condition that causes the file to be not ready or by system reset.

Bit 2—Missing Address Marker

This bit is set on any multiple-sector operation when the first sector has been found and any two following sequential sectors read from the disk have identical bits in bit position 5 of the S byte of the sector identifier read from the disk. If this condition is detected before the first sector is found or on a single-sector operation, it will be indicated after the control unit has determined that the record cannot be found on the track. This bit is also set if no address marker is found and index has been passed twice while looking for an address marker. This bit is not set if a data check is detected in one of the two identifier fields. The bit is reset by the next start I/O instruction.

Bit 3—Equipment Check

This bit indicates that the control unit has detected a hardware failure or the selected drive is unsafe.

Bit 4—Data Check

This status indicates that a cyclic check or bit count appendage check revealed an inconsistency between the bits written on the disk and the bits read from the disk while reading the identifier or the data fields.

Bit 5—No Record Found

This bit indicates that the first sector called for by a read, write, or scan instruction could not be found on the track or that after the first sector was found one of the succeeding sectors in a multiple-sector operation had a sector identifier that did not match the identifier in the disk control field. This bit, after a read identifier operation, indicates that no identifier without an error was found on the track. This "no record found" bit is also set by a track condition check.

Bit 6—Track Condition Check

This bit indicates that bits 6 and 7 of the flag byte in the disk control field do not match bits 6 and 7 of the flag byte on the track in a read, write, or scan operation.

Bit 7—Seek Check

This bit is set when the control unit detects a seek error or an attempt is made to seek to a cylinder outside the capacity of the files installed.

Byte 1

Bit 0—Scan Equal Hit

This bit indicates that equal condition has been satisfied whenever a scan instruction is executed.

Bit 1—Cylinder 0

This bit indicates that the *selected drive's* access mechanism is positioned at cylinder 0.

Bit 2—End of Cylinder

This bit turns on when a multiple-sector operation specifies a sector count greater than the number of sectors available between the first sector specified and the end of the cylinder. All sectors up to and including the last one on the cylinder were successfully processed. This can occur if head 1 addresses are written by head 0 and an attempt is made to read beyond the end of the track (switch heads). This can occur on alternate tracks.

Bit 3—Seek Busy

This bit indicates that the drive addressed by the sense I/O instruction is seeking.

Bit 4—100 Cylinder

This bit indicates that the drive installed in the system has one hundred cylinders available to the customer.

Bit 5—Overrun

This bit is set when the processing unit fails to allow a cycle steal to the file unit in time to transfer data before it is lost. This status occurs during processor check stop in the processing unit that stops the processing unit clock or by a disk defect that causes noise to interfere with correct read timing.

Bits 6 and 7—Status Address A and Status Address B

These two bits indicate the drive that was specified in the last read, write, or scan instruction. This provides the number of the drive that pertains to attachment-dependent status bits. When both bits are 0, drive 1 is specified. When bit 7 is 1, drive 2 is specified. This address is reset when a start I/O instruction is accepted by either drive.

Byte 2

Bit 0—Unsafe

This bit indicates that one of the following checks has been detected by the file:

1. Read and write are selected together.
2. Write is selected and both head 0 and head 1 are selected or both the fixed and removable disks are selected.
3. Write is selected but the write circuits are not operating.
4. Write is not selected but the write circuits are operating.
5. Drive is accessing and the write circuits are operating.

Unsafe also causes equipment check and must have a unique program halt indicator.

Bits 1, 2, and 3—Tap Lines A, B, and C

These three bits are used by CE diagnostic programs and have no meaning to the I/O control program.

Bit 4—Index

This bit is active only during the time the index mark passes a sensing head. The duration is approximately 43 microseconds.

Bit 5—Head Settling

This bit indicates that a head settling operation is occurring following a seek operation.

Bit 6—CE Sense Bit

This bit can be used by the CE in diagnostic programs.

Bit 7—Model 6

This line will always be active on this system.

Byte 3

Bits 0-7—CE Sense Bits

These bits are for CE diagnostic use and have no meaning to the I/O control program.

FLAGGING

Defective recording areas are handled by track flagging. The flagging procedure included in the file is used to identify defective tracks and their alternates. Alternate tracks can be assigned under program control at the time that a track in cylinders 4 to 202 is found to be defective. Cylinders 1 to 3 are provided for assignment as alternate tracks.

The flagging procedure uses flag byte bits 6 and 7 of each sector identifier recorded on the file. Bit 6 alone indicates that the track is defective, and bit 7 alone indicates that the track is an alternate. When both bits equal 0, the track is an original good track. Both bits set to 1 indicates a defective alternate track. A defective track on cylinder 0 will require disk repair.

A track with a bad spot is marked defective and an alternate track assigned to replace the whole track. When a track is found to be defective, a write identifier operation must be performed on its corresponding alternate track to write the flag bytes with bit 7 equal to 1 and the C and S bytes of the identifiers from the defective track. Then the recoverable data from the defective track must be written on the corresponding sectors of the alternate track. Finally the defective track must be written with a write identifier operation to write flag bytes with bit 6 equal to 1 and the C and S bytes of the identifiers from the alternate track on the defective track. A defective alternate track should be rewritten with a write identifier operation to write flag bytes with both bits 6 and 7 equal to 1 and the C and S bytes to contain their own original addresses.

When bits 6 and 7 of the F byte in storage and the F byte on the disk do not agree, track condition check is set as the device status. Track condition check causes an error indication to test-I/O-and-branch or advance-program-level instructions that test not ready/check.

The identifier fields of the tracks are:

1. Good–bits 6 and 7 of the F byte are both 0 and the C and S bytes contain the cylinder, head, and sector numbers that are correct for that track.
2. Defective–bit 6 is 1 and bit 7 is 0 in the F byte. The C and S bytes contain the cylinder, head, and sector address from the alternate track.
3. Alternate–bit 6 is 0 and bit 7 is 1 in the F byte. The C and S bytes contain the cylinder, head, and sector addresses from the defective track.
4. Defective Alternate–bits 6 and 7 of the F byte are both 1. The C and S bytes contain the cylinder, head, and sector addresses of the respective sectors.

TRACK INITIALIZATION PROCEDURES

The following procedures must be followed by track initialization programs for the 5444 disk file. They analyze the condition of the surface and format the tracks.

1. Read identifier to determine that the track has not been previously flagged. This step need not be performed when initializing a previously unused disk.
2. Write identifier with a data field of hexadecimal 55. The flag bytes for tracks 1, 2, and 3 should be written as hexadecimal 01.
3. Verify all the sectors to insure that data can be recovered correctly. If an error occurs go to step 10.
4. Repeat step 2 with a data field of hexadecimal 00.
5. Repeat step 3.
6. Seek to the next track and repeat steps 1 through 5.
7. Repeat steps 2 through 6 until all tracks have been processed.
8. Read identifier on all tracks to check for seek errors. During this operation, if a seek error on a writing operation is detected, initialization must repeat steps 2 through 7. A seek error on the writing operation causes two different tracks to contain the same identifiers or the identifiers for one track to be missing.
9. Perform steps 1 through 8 at least once.

10. If an error occurs, the device status must be analyzed. If a missing address marker or data check occurs, retry a verify instruction at least 10 times. On the first unsuccessful retry that indicates missing address marker or data check, flag the track as defective and go to step 11. If all 10 retries are successful, proceed with the initialization procedure from the point at which it was interrupted.

    For any error other than missing address marker or data check, follow the suggested error recovery procedures.
11. Assign an alternate track.
12. Write identifier on the defective track with the address of the alternate track in the identifier and a hexadecimal value of 02 in the flag byte. A defective alternate track should contain its own address and a hexadecimal value of 03 in the flag byte.
13. Set the flag byte in the disk control field to hexadecimal 02. Perform a read identifier operation. If the address of the alternate track is not recoverable, the disk must be repaired unless this is an alternate track.
14. Seek to the alternate track.
15. Set the flag byte in the disk control field to hexadecimal 01. Write identifier on the alternate track with the identifier of the defective track in the disk control field. Alternate tracks must be proved reliable by steps 2 through 5 before they are used as alternates.
16. Continue with initialization on the next track.

The basic requirement is for one pass through steps 1 through 8. An option must be provided to allow any number of passes up to 255.

No program should change the flagging of previously flagged tracks except as follows:

1. Initialization programs must have additional capabilities.
    a. The option to ignore all previously flagged tracks.
    b. The option to unconditionally flag or unflag any individual track.
2. Operating programs that have provision for dynamic flagging must perform steps 11 through 15 of this procedure.

SUGGESTED ERROR RECOVERY PROCEDURES (FILE AND ATTACHMENT)

Minimum error recovery procedures are defined for the file and attachment in this section. Error conditions are indicated as a response to a test I/O and branch or advance program level command. The following table specifies the indicators to be tested, and the priority or order in which they should be tested. For example, priority 1 (error/not ready) should be tested before any other indicator. The action column in the following table relates to Figure 5-7. The action to be taken is that action which should follow the failure of a start I/O instruction.

	Device Status			
Priority	*Byte*	*Bit*	*Condition*	*Action*
1			Error/not ready	III
2	0	3	Equipment check	II
3	0	1	Intervention required	VIII
4	0	5	No record found	IV
5	0	2	Missing address marker	IV
6	0	4	Data check	IV
7	0	6	Track condition check	IV
8	1	5	Overrun	IV
9	0	7	Seek check	VI
10	1	2	End of cylinder	V

Step	Action I
1	If there is no additional error recovery procedure, perform an operator message and stop.
2	If there is an additional recovery procedure, exit to it.
3	If the additional error recovery procedure fails, perform an operator message and stop.
	Action II
	Retry the original operation or sequence of operations once. On the second occurrence of this error, do Action I.
	Action III
	Perform a sense I/O command and check the conditions as indicated.
	Action IV
1	Perform a read ID command. If there is an error, do Action VII for the original operation.
2	Check to determine if positioned on correct track. If on correct track, do Action VII for the original operation.
3	Check to determine if head switching from an alternate track has just taken place. If switching has taken place, set flag byte to hex 00, determine next track, and do Action V2.
4	Check to determine if present track is defective (flag byte from read ID is hex 02). If track is defective, set flag to hex 01, determine alternate track from ID sealed and do Action V2.
5	Do Action VI.
	Action V
1	Decrease original number of sectors by the number of sectors successfully completed. Calculate new sector address, new data address, and new track.
2	Seek to desired track.
3	Retry operation on new track for the proper number of sectors.
	Action VI
1	Issue a recalibrate.
2	Seek to the original track.
3	Do Action VII, if the original action was not a seek.
4	If the error persists repeat steps 1-3 sixteen times. After sixteen unsuccessful reseeks, do Action I.
	Action VII
1	If original operation was a verify-of-a-write or a write, repeat the original sequence eight times. After eight unsuccessful retries, do Action I.
2	If original operation was not a verify-of-a-write or a write, repeat the original operation sixteen times.
3	If error persists do Action VI. After sixteen unsuccessful retries of Actions VI and VII, do Action I.
	Action VIII
	Perform an operator message and stop. After restart (push start), repeat the original operation or sequence of operations.

Figure 5-7. Error Recovery Actions

Chapter 6. IBM 5496 Data Recorder

The IBM 5496 Data Recorder Model 1 with the On-Line Reader/Punch Feature for System/3 Model 6 (Figure 6-1) operates on-line with the system to provide card input and output functions. The data recorder is connected to the system by a cable and can read or punch and print cards at a rate of 22 cards per minute. Reading and punching are under control of the system program. Printing is under control of the print switch on the data recorder. Only data that has been punched by the program can be printed.

Data is entered in the cards in the six bit card code. This code is illustrated in "Appendix B".

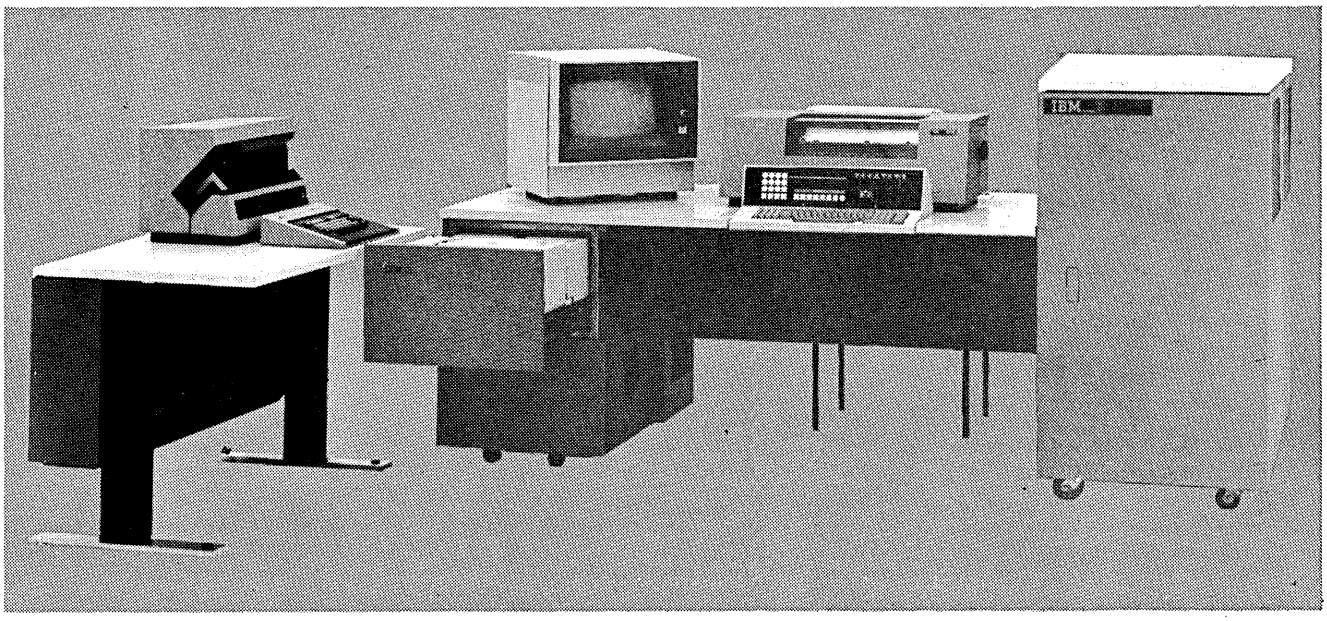
Operation of the data recorder is controlled by the data recorder switch on the keyboard-console. When the data recorder switch is in the on-line position, the data recorder is under the control of the system program. When the data recorder switch is in the off-line position, the data recorder functions as a normal off-line data recorder.

DATA RECORDER KEYS AND INDICATORS

Certain keys and indicators on the data recorder function in the on-line condition as well as when the data recorder is being used off line.

Release Key

The release key on the data recorder keyboard can be used when the data recorder is on line for clearing hopper jams and transport jams.



BR0397

Figure 6-1. IBM 5496 Data Recorder Model 1

Power Switch and Indicators

The power switch is used for applying power to the data recorder. When this power switch is turned on, the machine is ready for immediate use. The column indicators on the data recorder indicate the next column of data to be read or punched. These indicators also provide a visual indication that the power switch is on.

Stacker Full Indicator

This lamp turns on when the stacker is full. The lamp turns off when cards are removed from the stacker and the release key is pressed.

Feed Check Indicator

The feed check lamp indicates a card misfeed or a card jam. The indicator is turned off when the data recorder transport is cleared and the release key is pressed.

Print Switch

This switch is used to control printing on the card. When the switch is on, any data that is punched into the card from the system will be printed on the cards.

Program Switch

This switch must be in the off position when the data recorder is used on line.

Auto Record Release Switch

This switch must be in the auto position when operating the data recorder on line.

Punch/Verify Switch

This switch must be in the punch position when operating the data recorder on line.

DATA RECORDER CONTROLS ON THE KEYBOARD-CONSOLE

One control and one indicator for the data recorder are mounted on the system keyboard-console. The data recorder switch has been discussed. The indicator is the data recorder I/O attention indicator. This indicator is turned on by:

1. A feed check in the data recorder
2. Issuing a start I/O command to the data recorder when the data recorder switch is in the off-line position.

An I/O attention indication caused by a feed check can be corrected by clearing the data recorder transport and pressing the data recorder release key. An I/O attention caused by issuing a data recorder start I/O command with the data recorder off line can be corrected by system reset or by moving the data recorder switch to the on-line position.

DATA RECORDER OPERATIONS

Card Reading

This operation is used to read data from cards into the processing unit. Data is read from the cards into the data recorder and then is transferred to the processing unit by cycle stealing. Data enters storage in character format with the character from column 1 entering the storage location addressed by the data recorder address register at the time that the read command is issued. Data from successively higher columns is entered in successively higher storage addresses. Ninety-six columns of data will enter storage from all read operations. When a read instruction is issued to the data recorder, the data recorder becomes busy and remains busy, if no jams occur, until after the card is read and all jam checks are completed. If a jam occurs in the hopper, the card is neither read nor punched and busy ends as soon as the jam is detected.

If a jam occurs in the transport, invalid data has possibly been transferred to processing unit storage. The data recorder becomes not busy as soon as the jam is detected.

To obtain rated throughput, read commands succeeding either read or punch commands must be issued within 50 milliseconds after the data recorder becomes not busy after executing the previous command.

When reading cards, the read information should be checked to determine whether the card is a data file delimiter. A data file delimiter card, such as /* or a card punched with the words READ KEY, should always follow the cards containing the read information.

Punching Operations

This operation is begun by a start I/O instruction. If the print switch is turned on, any data that is punched into the card will be printed on the card. The punch operation takes data from 96 storage locations in the processing unit and transfers it to the data recorder. When all the data has been transferred to the data recorder, the data recorder feeds a card and punches the data. The data is located in storage with column 1 at the address contained in the data recorder address register. Ninety-six bytes of data are always transferred to the data recorder. For a punch operation, the data recorder becomes busy when the punch command is issued and becomes not busy, if no jams occur, after the card is punched and all jam checks are completed. If a transport jam occurs, invalid data may have been punched. The data recorder becomes not busy as soon as the jam has been detected. To obtain rated throughput, the punch command succeeding a read or punch command must be issued within 50 milliseconds after the data recorder becomes not busy from the preceding command.

DATA RECORDER INSTRUCTIONS

Start I/O

Mnemonic: SIO

Op Code	Q Byte	Control Code
F3	F 0 N	

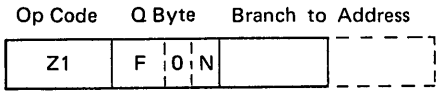
Operation: The data recorder performs the read or punch operation specified by the Q byte. Data to be transferred between the processing unit and the data recorder is addressed by the data recorder address register beginning with the data to be transferred to or from column 1.

The Q byte comprises a device address (always F for the data recorder) in the high-order four bits, an M bit of 0 and an N code. The N code is in bits 5, 6, and 7. For the data recorder, N code bit 5 is not used and should be 0. Bits 6 and 7 control the operation to be performed by the data recorder. When bit 6 is 0 and bit 7 is 1, the operation to be performed is read a card. When bit 6 is 1 and bit 7 is 0, the operation to be performed is punch a card (and print if the printing switch is on). The control code byte is not used by the data recorder and should be all zeros.

If a start I/O instruction is issued to the data recorder when the data recorder is on line and busy, the program loops on the start I/O instruction until the data recorder becomes not busy.

Test I/O and Branch

Mnemonic: TIO

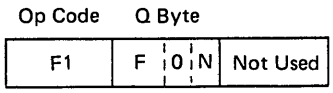


Operation: The data recorder is tested for the conditions specified by the Q byte. If the condition exists, the next instruction is taken from the branch-to address and the address of the next sequential instruction is placed in the address recall register. If the condition does not exist, the next instruction is taken from the next sequential instruction and the branch-to address is placed in the address recall register. The address recall register will not then be changed until a decimal, branch, or insert-and-test-characters instruction is executed.

The Q byte comprises a device address (always F for the data recorder) in the high-order four bits, an M bit of 0, and an N code in bits 5, 6, and 7. Bits 5 and 7 are not used for this instruction but should be zero. Bit 6 specifies what condition will be tested for the branch. If bit 6 equals 0, the I/O check or not ready conditions will be tested. If bit 6 equals 1, busy condition will be tested. The data recorder will always accept a test I/O and branch instruction.

Advance Program Level

Mnemonic: APL



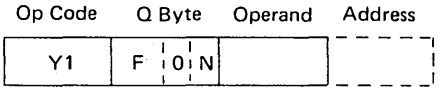
Operation: The data recorder is tested for the condition specified by the Q byte. If the condition exists, the program loops on the advance program level instruction until the condition no longer exists, then takes the next sequential instruction. If the condition does not exist, the program takes the next sequential instruction.

The Q byte comprises a device address (always F for the data recorder) in the high-order four bits, an M bit of 0, and an N code in bits 5, 6, and 7. Bits 5 and 7 of the N code are not used and should always be zero. Bit 6 specifies the condition to be tested. If bit 6 equals 0, the data recorder is tested for I/O check or not ready. If bit 6 equals 1, the data recorder is checked for busy.

Program Note: Advance program level should not be used to check for I/O check or not ready. If this condition is present, the program enters a one-instruction loop with no indication to the operator that the program is not progressing as it should.

Load I/O

Mnemonic: LIO



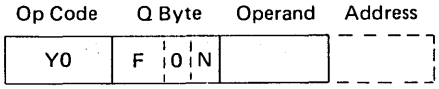
Operation: This instruction causes the two bytes of data located at the operand address and the next lower storage address to be loaded into the data recorder address register. The operand is addressed by its rightmost byte.

The Q byte comprises the device address (always F for the data recorder) in the high-order four bits, an M bit of 0, and an N code. In this instruction the N code is not used and should always be zero.

If the data recorder is busy when the load I/O instruction is executed, the program will loop on the load I/O instruction until the data recorder is not busy. If the data recorder is not ready or an I/O check occurs when the load I/O instruction is issued, the load I/O instruction will execute and the program will proceed to the next sequential instruction. When the data recorder is not busy and ready, the load I/O instruction will also be executed.

Sense I/O

Mnemonic: SNS



Operation: The data specified by the Q byte is loaded into the two-byte field in storage addressed by the operand address. The operand is addressed by its rightmost byte.

The Q byte comprises a device address (always F for the data recorder) in the high-order four bits, an M bit of 0, and an N code in bits 5, 6, and 7. Bits 5 and 7 do not enter into the operation and should be zero. Bit 6 specifies the data that is to be loaded into the storage locations. When bit 6 equals 0, the contents of the data recorder address register are loaded into the storage locations specified by the operand address. When bit 6 equals 1, two status bytes are loaded into storage.

Byte 2 (high-order byte) of the status bytes is a diagnostic byte and is of no interest to the problem programmer. Status byte 1 (the low-order status byte stored in the operand address) is bit significant as follows:

Bit	*Meaning*
0	Off line
1	Transport jam
2	Stacker full, hopper empty, or hopper jam
3	Not used
4	Incorrect card code
5	Compare error on read or punch I/O cycle, or failure to take read cycle steal cycles
6	0 = 129 attached to system 1 = 5496 attached to system
7	Reserved

Off line is caused by any one of the following:

1. The data recorder switch on the keyboard-console is in the off-line position.
2. The program switch on the 5496 operator console is in the on position, and the data recorder switch on the keyboard-console is in the on-line position.
3. The auto record release switch on the 5496 operator console is off, and the data recorder switch on the keyboard-console is in the on-line position.
4. The punch/verify switch is in the verify position, and the data recorder switch on the keyboard-console is in the on-line position.

Hopper jam or hopper empty is reset when the hopper is cleared, cards are placed in the hopper, and the release key on the data recorder is pressed.

Transport jam is reset when the jam is cleared and the release key on the data recorder is pressed.

Stacker full is removed when cards are taken from the stacker and the release key is pressed. Stacker full causes a not ready condition only when the data recorder is busy. If a command is issued to the data recorder when the stacker is full, the command will be accepted and executed, then the data recorder will become not ready for any further commands.

The incorrect card code status bit is set during a punch-a-card command if a byte not shown in "Appendix B" as a preferred card code graphic is sent to the data recorder. Data, as shown in "Appendix B," will be punched in the card. The data will be printed on the card if the print switch is on. The command will be completed. There may not be an error, depending on the program, so the status bit is set. The status bit is reset by the next start I/O instruction or by system reset.

Compare error on a read or punch I/O cycle occurs under the following conditions. On punch operations, the data transferred from storage to the data recorder is read back from the data recorder to the data recorder attachment and compared with the data sent from storage. On a read operation, the data sent from the data recorder to storage is read back from the attachment to the data recorder and compared with the data read from the card. If the data being transferred between the data recorder and storage does not match the read-back data in either of these operations, the compare error occurs. This error sets a status bit. The status bit is reset by either the next start I/O instruction or by system reset.

ERROR RECOVERY PROCEDURES

The following error recovery procedures are used during punching or reading cards on the data recorder:

Compare Error on Read or Punch I/O Cycle

This error indicates that incorrect data was punched into the card. This error should have its own unique halt indicator. The card must be removed from the stacker and either replaced in the hopper (if reading) or discarded (if punching) and the command must be retried.

Incorrect Card Code

This may or may not be an error depending upon the program. If it is an error, the card contains incorrect data. The card must be removed from the stacker and examined to determine what column contains the incorrect data; then the operation should be retried.

Transport Jam

To recover from a transport jam, clear the jam, reissue the command in which the jam occurred, and press the release key. This procedure will recover from a single card jam. If a jam involves more than one card, the job should be restarted. If a transport jam occurred during card reading and the card is damaged, a new card can be repunched. Move the data recorder switch on the keyboard-console to OFF LINE without first depressing the release key. Press the release key insert a blank card, and punch a new card in the off-line card punch mode. Insert the new card in front of the hopper; turn the data recorder switch on the keyboard-console to ON LINE; press the release key.

Chapter 7. IBM 129 Card Data Recorder

The IBM 129 Card Data Recorder Models 1, 2, and 3 with the Card Input/Output Feature operates on line with the System/3 to provide the system with 80-column card input and output in IBM 80-column card code. The 129 is connected to the system by a cable. It can read cards at a rate of up to 50 cards per minute or punch and print cards at a rate of 12 to 50 cards per minute depending on the number of consecutive blank columns in the card. Reading and punching are under control of the system program. Printing is under control of the print switch on the card data recorder. Only data that has been punched by the program can be printed.

Data is entered in the cards in the 12-bit IBM 80-column card code. Data read from the card enters storage in EBCDIC. Only 64 card codes are readable when the 129 is on line. The codes that are readable and their corresponding EBCDIC values are shown in "Appendix B." All EBCDIC values cause some character to be punched into the card, but not necessarily the desired value. The codes punched in the card for each EBCDIC value are shown in "Appendix B."

Operation of the card data recorder is controlled by the data recorder switch on the keyboard-console. When the data recorder switch is in the on-line position, the 129 is under control of the system program. When the switch is in the off-line position, the 129 functions as a normal off-line data recorder.

129 CARD DATA RECORDER SWITCHES AND INDICATORS

Certain switches and indicators on the card data recorder function in the on-line condition as well as when the card data recorder is being used off line.

Verify Reset Key

The verify reset key on the 129 keyboard is used to clear the following conditions when the 129 is on line: transport jam, hopper jam, hopper empty, and stacker full.

Power Switch and Indicators

The power switch applies power to the card data recorder. When this switch is turned on, the machine is ready for immediate use unless the data recorder switch on the keyboard-console is in the on-line position. Turning on the power switch when the 129 is in on-line mode causes a transport jam condition. Transport jam is corrected by clearing the transport of cards, pressing the feed key twice, and pressing the verify reset key. A column indicator on the 129 provides a visual indication that the power switch is on.

Card Data Recorder Column Indicator

In addition to the power-on indication, the column indicator signals the following error conditions:

1. If a transport jam occurs, the column indicator contains 8A.
2. If a hopper jam, hopper empty, or stacker full condition occurs, the column indicator contains 88.

Print Switch

This switch controls printing on the card. When the switch is on, any data that is punched into the card from the system will be printed on the card.

Punch/Verify Switch

This switch must be in the punch position when the 129 is used on line.

Program Rotary Switch

This switch must be in the data read position when the 129 is used on line.

Record Advance/Card Feed Switch

This switch must be in the auto position when operating the 129 on line.

DATA RECORDER CONTROLS ON THE 5406 KEYBOARD-CONSOLE

One control and one indicator for the data recorder are mounted on the system keyboard-console. The data recorder switch has been discussed. The data recorder I/O attention indicator is turned on by:

1. A transport jam in the 129.
2. Issuing a start I/O command to the 129 when the 129 is not ready. Not ready is caused by an erroneous switch setting on the 129 or by the data recorder switch on, the keyboard-console being in the off-line position.

An I/O attention indication caused by a transport jam can be corrected by clearing the card data recorder transport and pressing the card data recorder verify reset key. An I/O attention caused by issuing a card data recorder start I/O command with the 129 off line or not ready can be corrected by moving the data recorder switch to the on-line position, by correcting the switch settings on the 129, or by a system reset.

CARD DATA RECORDER OPERATIONS

Before any card data recorder operations can be performed, cards must be in both the register and preregister stations on the unit. This occurs when the feed key is pressed and held until the first card is fed and registered and the second card is fed to the preregister station.

Card Reading

Card reading is begun by a start I/O instruction. This operation reads data from cards into the processing unit. Data is read from the cards into the card data recorder and then is transferred to the processing unit by cycle stealing. Data enters storage in character format with the character from column 1 entering the storage location addressed by the data recorder address register at the time that the read command is issued. Data from successively higher columns is entered in successively higher storage addresses. Eighty columns of data will enter storage from all read operations. When a read instruction is issued to the card data recorder, it becomes busy and remains busy, if no jams occur, until after the card is read and all jam checks are completed. If a hopper jam, hopper empty, or stacker full condition occurs, the card is neither read nor punched. When the condition is corrected and the verify reset key is pressed, the card is read or punched.

If a jam occurs in the transport, invalid data has possibly been transferred to main storage. The 129 becomes not busy as soon as the jam is detected.

To obtain rated throughput, read commands succeeding either read or punch commands must be issued within 105 milliseconds after the unit becomes not busy after executing the previous command.

When reading cards, the read information should be checked to determine whether the card is a data file delimiter. A data file delimiter card, such as /*, or a card punched with the words READ KEY, should always follow the cards containing the read information.

Punching Operations

Card punching is begun by a start I/O instruction. If the print switch is turned on, any data that is punched into the card will be printed on the card. The punch operation takes data from 80 storage locations in the processing unit and transfers it to the card data recorder. When all the data has been transferred to the card data recorder, it punches the data. The data is located in main storage with column 1 at the address contained in the data recorder address register. Eighty bytes of data are always transferred to the card data recorder.

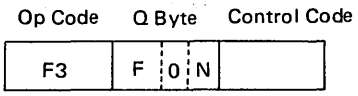
For punch operations, the 129 becomes busy when the punch command is issued; it becomes not busy after the card is punched and all jam checks are completed, unless a jam occurs. If a transport jam occurs, invalid data may have been punched. The data recorder becomes not busy as soon as the jam has been detected.

To obtain rated throughput, the punch command must be issued within 105 milliseconds of the time the 129 becomes not busy from the preceding read or punch command. When two or more successive columns are not punched (there are blanks in main storage for those column positions), the card is transported at read speed until a nonblank column is encountered.

129 CARD DATA RECORDER INSTRUCTIONS

Start I/O

Mnemonic: SIO



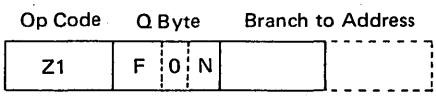
Operation: The card data recorder performs the read or punch operation specified by the Q byte. Data to be transferred between the processing unit and the 129 is addressed by the data recorder address register beginning with the data to be transferred to or from column 1.

The Q byte comprises a device address (always F for the 129) in the high-order four bits, an M bit of 0, and an N code. The N code is in bits 5, 6, and 7. For the 129, N code bit 5 is not used and should be 0. Bits 6 and 7 control the operation to be performed by the card data recorder. When bit 6 is 0 and bit 7 is 1, a card read operation is performed. When bit 6 is 1 and bit 7 is 0, a card punch operation (and print if the printing switch is on) is performed. The control code byte is not used by the 129 and should be all zeros.

If a start I/O instruction is issued to the 129 when it is on line and busy, the program loops on the start I/O instruction until the 129 becomes not busy.

Test I/O and Branch

Mnemonic: TIO

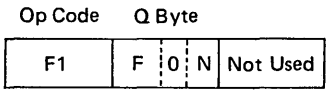


Operation: The card data recorder is tested for the condition specified by the Q byte. If the condition exists, the next instruction is taken from the branch-to address, and the address of the next sequential instruction is placed in the address recall register. If the condition does not exist, the next instruction is taken from the next sequential instruction and the branch-to address is placed in the address recall register. This register will not then be changed until a decimal, branch, or insert-and-test-characters instruction is executed.

The Q byte comprises a device address (always F for the 129) in the high-order four bits, an M bit of 0, and an N code in bits 5, 6, and 7. Bits 5 and 7 are not used for this instruction but should be 0. Bit 6 specifies what condition will be tested for the branch. If bit 6 equals 0, the I/O check or not ready condition will be tested. If bit 6 equals 1, the busy condition will be tested. The 129 will always accept a test I/O and branch instruction.

Advance Program Level

Mnemonic: APL



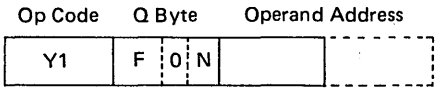
Operation: The card data recorder is tested for the condition specified by the Q byte. If the condition exists, the program loops on the advance program level instruction until the condition no longer exists, then takes the next sequential instruction. If the condition does not exist, the program takes the next sequential instruction.

The Q byte comprises a device address (always F for the 129) in the high-order four bits, an M bit of 0, and an N code in bits 5, 6, and 7. Bits 5 and 7 of the N code are not used and should always be 0. Bit 6 specifies the condition to be tested. If bit 6 equals 0, the 129 is tested for the I/O check or not ready condition. If bit 6 equals 1, the 129 is checked for the busy condition.

Program Note: Advance program level should not be used to test for I/O check or not ready. If this condition is present, the program enters a one-instruction loop with no indication to the operator that the program is not progressing as it should.

Load I/O

Mnemonic: LIO



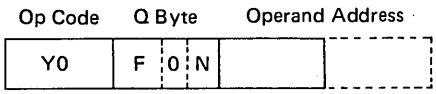
Operation: This instruction causes the two bytes of data located at the operand address and the next lower storage address to be loaded into the data recorder address register. The operand is addressed by its rightmost byte.

The Q byte comprises the device address (always F for the 129) in the high-order four bits, an M bit of 0, and an N code. In this instruction the N code is not used and should always be 0.

If the 129 is busy when the load I/O instruction is executed, the program will loop on the load I/O instruction until the 129 is not busy. If the 129 is not ready or an I/O check occurs when the load I/O instruction is issued, the load I/O instruction will execute and the program will proceed to the next sequential instruction. When the 129 is not busy and ready, the load I/O instruction will also be executed.

Sense I/O

Mnemonic: SNS



Operation: The data specified by the Q byte is loaded into the two-byte field in storage addressed by the operand address. The operand is addressed by its rightmost byte.

The Q byte comprises a device address (always F for the 129) in the high-order four bits, an M bit of 0, and an N code in bits 5, 6, and 7. Bits 5 and 7 do not enter into the operation and should be 0. Bit 6 specifies the data that is to be loaded into the storage locations. When bit 6 equals 0, the contents of the data recorder address register are loaded into the storage locations specified by the operand address. When bit 6 equals 1, two status bytes are loaded into storage.

Byte 2 (high-order byte) of the status bytes is a diagnostic byte and is of no interest to the problem programmer.

Status byte 1, (the low-order status byte stored in the operand address) is bit significant as follows:

Bit	*Meaning*
0	Off line
1	Transport jam
2	Stacker full, hopper empty, or hopper jam
3	Not used
4	Incorrect card code
5	Compare error on read or punch I/O cycle
6	0 = 129 attached to the system 1 = 5496 attached to the system
7	Reserved

An off-line condition is caused by any one of the following:

1. The data recorder switch on the keyboard-console is in the off-line position.
2. The punch/verify switch on the 129 keyboard is not in the punch position.
3. The program switch on the 129 keyboard is not in the read data position.
4. The record advance switch on the 129 keyboard is in the manual position.

Hopper jam or hopper empty is reset when the hopper is cleared, cards are placed in the hopper, the feed key is pressed twice, and the verify reset key is pressed.

Transport jam is reset when the jam is removed from the transport, the feed key is pressed twice, and the verify reset key is pressed.

Stacker full is reset by removing the cards from the stacker, pressing the feed key, and pressing the verify reset key.

The incorrect card code status bit is set during a punch-a-card command if a byte not shown in "Appendix B" as a preferred card code graphic is sent to the 129. Data, as shown in "Appendix B," will be punched in the card and printed if the print switch is on. The command will be completed. There may not be an error, depending on the program, so the status bit is set. The status bit is reset by the next start I/O instruction or by a system reset.

Compare error on a read or punch I/O cycle occurs under the following conditions:

1. On punch operations, the data transferred from main storage to the 129 is read back from the 129 to the card data recorder attachment and compared with the data sent from main storage.
2. On a read operation, the data sent from the 129 to main storage is read back from the attachment to the 129 and compared with the data read from the card.

If the data being transferred between the 129 and main storage does not match the read-back data in either of these operations, the compare error occurs. This error also occurs if any combination of punches not in the standard 64-character set is read from the card. The 64-character set and the punch combinations that produce each character are shown in "Appendix B." This error sets a status bit (bit 5) that is reset by either the next start I/O instruction or by a system reset.

129 ERROR RECOVERY PROCEDURES

The following error recovery procedures are used during punching or reading cards on the card data recorder.

Compare Error on Read or Punch Cycle

This error indicates that incorrect data may have been punched or read. It should have its own unique halt indicator. If this error occurs during a read operation, it indicates either that the card was read incorrectly or that characters outside the 64-character EBCDIC set are punched in the card. Remove the last card in the stacker and use the clear switch to remove any cards in the transport. In read mode, check the cards for incorrect characters and place the correct cards in correct sequence in the hopper. In punch mode, discard the incorrectly punched card. Press the feed key twice, then retry the command.

Incorrect Card Code

This may be an error depending upon the program. If it is an error, the card contains incorrect data. Remove the card from the stacker and retry the operation.

Transport Jam

A transport jam during or after punching should be cleared as follows:

1. Remove the cards from the transport and the register and preregister stations.
2. Discard the last card punched.
3. Return the unpunched and undamaged cards to the hopper.
4. Press the feed key twice.
5. Press the verify reset key.
6. Retry the operation.

A transport jam during or after reading a card is cleared as follows:

1. Remove the cards from the transport and the registration and preregistration stations.
2. Replace the last card read and the cards from the registration and preregistration stations in the hopper in proper order.
3. Press the feed key twice.
4. Press the verify reset key.
5. Retry the operation.

If a transport jam or hopper jam occurs during reading, and the card is damaged, a new card can be punched as follows:

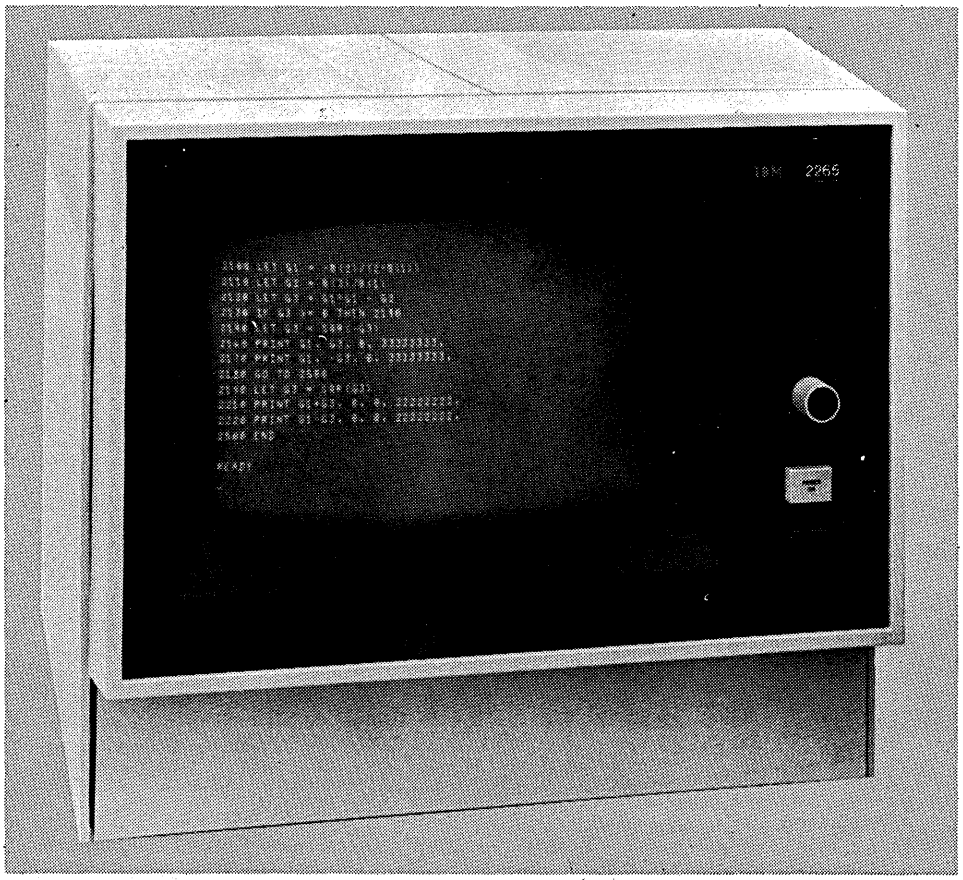
1. Move the keyboard-console data recorder switch to OFF LINE.
2. Clear the 129 transport, register station, and preregister station by operating the clear switch.
3. Punch the new card in the normal manner with the 129 off line.
4. Place the new card and the cards from the register and preregister stations in the hopper in proper order.
5. Press the feed key twice.
6. Move the keyboard-console data recorder switch to ON LINE.
7. Press the verify reset key and retry the read operation.

Chapter 8. IBM 2265 Display Station Model 2

The 2265 Display Station Model 2 (Figure 8-1) displays alphameric characters on a screen similar to that of a television set. 960 characters can be displayed in 15 lines of 64 characters per line. A character position marker called a cursor indicates the position the next character entered into the display area will occupy on the screen. The display of characters on the screen of the display station is under control of the system program.

Display Unit Manual Controls and Indicators

Two controls on the display unit itself and one switch and one indicator on the keyboard-console are used for controlling the display unit. The controls on the display unit are a power on-off switch, which controls the primary power to the display unit, and a character brightness control, which controls the brightness of the display. The system reset switch on the keyboard-console causes the display unit to stop displaying and resets the display unit to its initial state. A CRT I/O attention indicator on the keyboard-console indicates that operator attention is required if an I/O instruction is issued to the display unit when the display unit power switch is off.



BR0403

Figure 8-1. IBM 2265 Display Station Model 2

DISPLAY UNIT OPERATIONS

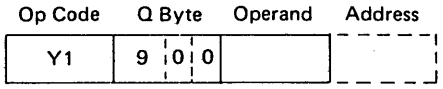
The display unit performs just one function, that of displaying data stored in main storage. This display of data is initiated by a start I/O instruction and is stopped by a start I/O instruction. Issuing a start I/O instruction to the display unit causes the display unit to display the contents of a 960-byte area of storage. This 960-position block of storage must be located at an address of XX01, where X is any hexadecimal digit valid for that particular system as a high-order hexadecimal address digit, and where (XX01 + 03C0) must be less than the maximum core address.

The characters displayed on the display are the standard graphic characters for the system plus a cursor character, which is a horizontal mark beneath the character in the position displayed. The character codes for the characters displayed on the display unit are shown in "Appendix B". The attachment must be prepared for data display by loading the high-order address of the data block in storage into the CRT address register.

DISPLAY UNIT INSTRUCTIONS

Load I/O

Mnemonic: LIO



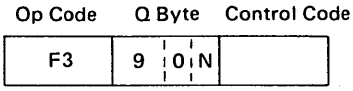
Operation: This instruction causes the two-byte field located at the operand address to be loaded into the CRT address register. The operand is addressed by its rightmost byte.

The Q byte contains a device address of 9 in the high-order four bits, an M bit of 0, and an N code of 0.

Executing this instruction causes the display unit to begin its next display operation at the upper left hand corner of the screen.

Start I/O

Mnemonic: SIO

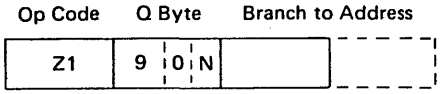


Operation: This instruction causes the display unit to attempt to perform the operation specified by the Q byte.

The Q byte comprises the device address of hexadecimal 9 in the high-order four bits, an M bit of 0, and an N code. The N code specifies the operation to be performed. Bit 5 of the N code is not used. Bits 6 and 7 specify the operation as follows. Bits 6 and 7 equal to 00 or 01 causes the display unit to stop displaying and blank the screen. N bits 6 and 7 equal to 10 or 11 is a display command. This causes the display unit to begin displaying characters taking the characters from the address specified in the CRT address register. 960 characters will be displayed on the screen. If a load I/O instruction is not issued before a display instruction, the attachment will display beginning with the character that would have been displayed next. The display unit will continue to display the characters in storage until issued a halt command, the system stop key is activated, or system reset occurs. If the system stop key was activated, the display unit will begin displaying again by subsequently activating the system start key.

Test I/O and Branch

Mnemonic: TIO

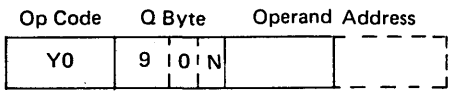


Operation: The condition specified by the Q byte is tested. If the condition exists, the next instruction is taken from the branch-to address and the address of the next sequential instruction is placed in the address recall register. If the condition does not exist, the next sequential instruction is taken and the branch-to address is stored in the address recall register.

The Q byte contains a device address of 9 in the high-order four bits, an M bit of 0, and an N code in bits 5, 6, and 7. Bits 5 and 7 are not used by this instruction and should be zero. Bit 6 specifies the condition to be tested. If bit 6 equals 0, the condition tested is CRT check. If bit 6 equals 1, the condition tested is CRT busy. The display unit will accept a test I/O command at any time.

Sense I/O

Mnemonic: SNS



Operation: Two bytes of data are moved to the two-byte field addressed by the operand address. The operand is addressed by its rightmost byte. The Q byte specifies the data that is to be moved into storage.

The Q byte contains a device address of hexadecimal 9 in the high-order four bits, an M bit of 0, and an N code. Bits 5 and 7 of the N code are not used and should be zero. Bit 6 of the N code controls the data that will be moved to storage. When bit 6 equals 0, the contents of the CRT address register are moved into the operand field. When bit 6 of the N code equals 1, two status bytes are moved into the operand field.

The byte stored in the high-order position of the operand field contains the contents of the display station data register, where each byte of data is stored waiting its turn at display on the screen. The byte in the low-order position in the operand field is the status byte that is bit-significant as follows:

Bit	*Meaning*
0	Write op (diagnostic only)
1	Start char gen (diagnostic only)
2	Step-display (diagnostic only)
3	Cycle steal request (diagnostic only)
4	Display reset (diagnostic only)
5	Data register parity check
6	Display not ready
7	Cycle steal acknowledged (diagnostic only)

DISPLAY STATION CHECK CONDITIONS

Three check conditions are indicated for the display station. A parity check from storage to the display station attachment causes a processor check stop with a DBO parity indication. Ordinarily a complete program restart is required to recover from this error.

A data register parity check occurs when data sent to the attachment is stored with a parity error in the data register. This error can be recovered from by issuing another start I/O instruction and restarting the display operation.

Not ready is caused by the power switch on the display station being off when a start I/O instruction is issued to the display station. This error also causes the CRT I/O attention light on the keyboard-console to turn on. The not ready condition can be reset and operation can resume by turning on the power switch on the display station.

Cursor Control

Setting bit 1 of the data byte for the display equal to 0 causes the cursor to be displayed along with the contents of that particular byte in storage.

The serial input/output channel adapter (SIOC) provides a means for attaching additional input/output devices for which attachment circuitry is not incorporated in the system. It also provides a means for attaching special units that may be requested by the customer. The control unit for any I/O unit that is to attach to the SIOC must be designed to be compatible with the SIOC. Only one control unit can be physically attached to the SIOC at any one time, although more than one I/O device can be controlled by that control unit. If the control unit is controlling more than one device, only one device can operate at any one time. The SIOC accepts data in the form of an 8-bit byte (plus parity). Data is accepted one byte at a time, parallel by bit. If the I/O device attached presents data in any other form, some translation must be performed by the program.

The SIOC provides an intermediate control unit between the system I/O channel and the device control unit. This intermediate control unit produces the necessary signals to control the device control unit from information furnished to the SIOC by instructions from the processing unit, control bytes stored in registers in the SIOC by the processing unit, and information supplied by the device control unit.

SIOC Registers

Data Transfer Register

A nine-bit data transfer register is provided in the SIOC to temporarily store one byte of data (8 bits plus parity) that is to be transferred between the I/O device and main storage. Data transfer is normally on a cycle steal basis, but the contents of this register can be moved between the register and storage with load I/O and sense I/O instructions when this is required by the characteristics of the I/O device involved or for diagnostic purposes. The register is tested for correct parity with a sense bit set by incorrect parity.

Length Count Register

Because data transfer occurs on a cycle steal basis, the adapter must keep track of the number of bytes transferred. A length count register is provided to perform this function. This counter limits the number of bytes to be transferred to 256 bytes per record. A load I/O instruction is used to place the number of bytes to be transferred in the length count register. The number that is placed in the length count register is the binary representation of a number equal to 256 minus the number of bytes to be transferred. Normally the I/O device signals when enough bytes have been transferred, but the length count register signals when the correct number of bytes has been exceeded and stops the data transfer. The contents of this register and the count exceeded condition can be placed in storage with a sense I/O instruction.

I/O Select Register

This register is used for issuing up to 16 separate I/O device control signals. It is loaded with a start I/O instruction. The functions that these control signals perform in the I/O device are determined by that device, and the data that must be placed in the register for different conditions will be defined by the I/O device. In general, all the signals will not be used by any one I/O device.

I/O Transfer Lines

This is not a hardware register but a set of 11 signal lines from the device to the SIOC that can communicate information to the processing unit. These lines can be tested with a sense I/O instruction and used for program decisions based on information received from the I/O unit. The conditions that will be conveyed on these lines are defined by the I/O devices and will be specified in manuals or sections of this manual relating to the I/O device.

Function Register

This register defines the mode of operation of the I/O device. It must be loaded before attempting to execute the program operating the device. It can be loaded by the program before any device operations are attempted. The specific bits that must be stored in this register by a load I/O instruction are defined by the I/O device.

SIOC Data Address Register

This is one of the local storage registers that is used to store the address of the data field that is to be used by the I/O device. The register is loaded by a load I/O instruction.

SIOC Operation

The operation of the SIOC requires that certain I/O instructions be performed to prepare the program and adapter for operation. A means of identifying the individual I/O devices that are attached to the SIOC has been provided. The identification is established at the time the I/O device is designed. These identification lines (four) are stored on a sense I/O operation that specifies the byte that contains their sense bits. The following procedures should be performed to operate the SIOC.

1. Sense the I/O identification byte.
2. Test that an I/O device is attached to the SIOC.
3. Load the function register with the appropriate bytes to control the particular I/O device.

I/O operations require that certain instructions be performed before the instruction that transfers the data is executed. Before each data transfer operation, the length count register must be loaded with a count equal to 256 minus the number of bytes to be transferred by that operation. The SIOC data address register must be loaded with the address of the first byte of the data field to be operated on. Then the start I/O instruction that actually transfers data can be issued. Testing and sensing operations should be included in the operating program but can be inserted at the discretion of the programmer in accordance with good programming practice.

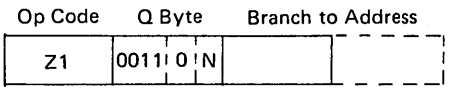
The SIOC operates in interrupt mode on interrupt level 4. Each time the I/O device requires some special service from the processing unit, such as processing in time for stacker selection, it interrupts the processing unit. Interrupts must be enabled for the I/O device before the SIOC can interrupt the processing unit.

INSTRUCTIONS

The commands for all I/O devices attached to the SIOC are the same. The interpretation given to some of the commands by the I/O devices may be different. The interpretations are discussed in the I/O device sections.

Test I/O and Branch

Mnemonic: TIO



Operation: This instruction tests for the conditions specified in the Q byte. If the condition is present, the next instruction is taken from the address specified by the branch-to address, and the address of the next sequential instruction is placed in the address recall register. If the condition is not present, the next sequential instruction is executed, and the address specified by the branch-to address is placed in the address recall register. The address placed in the address recall register remains there until the next branch, insert-and-test-characters, or decimal instruction.

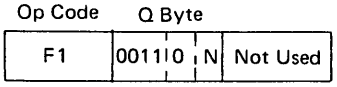
The Q byte contains the device address (always 0011 for the SIOC), an M bit of 0, and an N code. The N code specifies the condition that is to be tested as follows:

N Code	*Condition Tested*
000	SIOC not ready/check
001	Invalid
010	SIOC busy
011	Invalid
100	Invalid
101	Invalid
110	Invalid
111	Invalid

Issuing a test I/O and branch instruction with any of the invalid N codes causes a processor check stop with an invalid-Q-byte indication.

Advance Program Level

Mnemonic: APL



Operation: This instruction tests for the condition specified in the Q byte. If the condition exists, the processing unit loops on this instruction until the condition no longer exists.

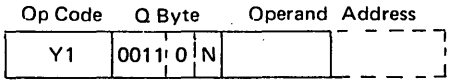
The Q byte contains the device address (always 0011 for SIOC), an M bit of 0, and an N code. The N code specifies the condition that is to be tested as follows:

N Code	*Condition Tested*
000	SIOC not ready/check
001	Invalid
010	SIOC busy
011	Invalid
100	Invalid
101	Invalid
110	Invalid
111	Invalid

Issuing an advance program level instruction with any of the invalid N codes causes a processor check stop with an invalid-Q-byte indication.

Load I/O

Mnemonic: LIO



Operation: This instruction transfers the contents of the two-byte field addressed by the operand address to the register designated by the Q byte. The operand is addressed by its low-order byte. If the SIOC is busy when this instruction is executed, the system loops on the load I/O instruction until the SIOC becomes not busy.

The Q byte contains a device address (always 0011 for the SIOC), an M bit of 0, and an N code. The N code specifies the register to be loaded as follows:

N Code	*Register*
000	Invalid
001	I/O function register
010	SIOC length count register
011	Invalid
100	SIOC data address register
101	Data transfer register
110	Invalid
111	Invalid

The bytes loaded into the function register are bit significant as follows:

High-Order Byte

Bit	*Meaning*
0	Write mode set service response
1	Reset service response after 6 microseconds
2	Transfer line 2 EOT
3	Transfer line 1 EOT
4	Even parity
5	Decrement DAR
6	Latch I/O 1 select
7	Slave (transfer line 6 and 7 latch control)

Low-Order Byte

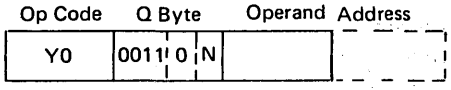
Bit	*Meaning*
0	Diagnostic mode (used only for CE diagnostic testing)
1	Spare
2	Latch transfer line 4
3	Latch transfer line 3
4	Latch transfer line 1
5	Transfer line 3 reset DISC latch
6	Reset latch after 6 microseconds
7	Transfer line 5 reset latch

The various bits in these two bytes that are set are determined by the I/O device attached to the SIOC at the time and will be specified by the instructions for programming that device.

Specifying an invalid N code results in a processor check stop with an invalid-Q-byte indication.

Sense I/O

Mnemonic: SNS



Operation: This instruction causes the two bytes of sense data specified by the Q byte to be transferred to the two-byte field specified by the operand address. The operand is addressed by the low-order byte. This instruction is executed even though the SIOC is busy or has a not ready/check condition.

The Q byte contains a device address (always 0011 for the SIOC), an M bit of 0, and an N code. The N code specifies the bytes to be sensed as follows:

N Code	*Function*
000	Invalid
001	I/O function register
010	Length count register and status byte
011	I/O transfer lines
100	Data address register
101	Data transfer register and diagnostic byte
110	Invalid
111	Invalid

Specification of an invalid N code causes a processor check stop with an invalid-Q-byte indication.

The status byte and the diagnostic byte are the high-order bytes of their respective sense operations. They are bit significant as follows:

Status Byte

Bit	*Meaning*
0	Spare
1	End request
2	Interrupt pending
3	I/O attention
4	Data transfer register parity check
5	No-op
6	Length count register overflow
7	I/O ready

Diagnostic Byte

Bit	*Meaning*
0	SIOC request latch
1	Service request
2	Service response
3	Interrupt enable
4	I/O disconnect
5	Write call
6	Read call
7	I/O select

Bits 0 and 1 of the status byte and all of the bits of the diagnostic byte are for CE diagnostic use and have no meaning to the I/O control program.

The transfer lines are bit-significant as follows:

Low-Order Byte

Bit	*Meaning*
0	I/O transfer line 8
1	I/O transfer line 7
2	I/O transfer line 6
3	I/O transfer line 5
4	I/O transfer line 4
5	I/O transfer line 3
6	I/O transfer line 2
7	I/O transfer line 1

High-Order Byte

Bit	*Meaning*
0	I/O identifier bit 8
1	I/O identifier bit 4
2	I/O identifier bit 2
3	I/O identifier bit 1
4	I/O device attached
5	I/O transfer line 11
6	I/O transfer line 10
7	I/O transfer line 9

The meaning of each of the I/O transfer lines (check condition, device status, etc.) is determined by each individual I/O device control unit and will be specified by manuals discussing that I/O device. Not all I/O transfer lines will necessarily be used by any one unit.

Start I/O

Mnemonic: SIO

Op Code	Q Byte	Control Code
F3	0011 0 N	

Operation: The start I/O instruction is used to control the mode of operation of the SIOC adapter and to issue control signals (I/O select lines) to the attached I/O device. A start I/O read or write instruction will electronically attach the adapter to the I/O device by setting it in the read or write mode respectively. The SIOC adapter must be placed in either one of these modes for the transfer of data to occur. This instruction is also used to enable or disable the ability of the adapter to request an interrupt priority, if required by the attached I/O device. The interrupt requests in the SIOC adapter are also reset with the start I/O instruction.

The start I/O instruction with an N code of 000 will always be accepted and executed by the adapter. Start I/O instructions with valid N codes other than 000 will be accepted and executed unless I/O attention or busy conditions exist. If I/O attention or busy exists and the processing unit is not servicing an SIOC interruption, a start I/O instruction will not be accepted. The processing unit will loop on the start I/O instruction until the condition is corrected. When the adapter becomes not busy or the I/O attention is corrected, the instruction will be accepted and normal instruction sequencing will continue. If a start I/O instruction is issued when the device is not ready because a device is not attached, the instruction cannot be executed. In this situation the no-op status bit is set, and the start I/O instruction is ignored. If a start I/O instruction with an N code other than 000 is issued when the I/O attention condition exists and the processing unit is servicing an SIOC interruption, the no-op status bit is set and the instruction is ignored. The no-op status bit can be sensed and reset with a sense I/O instruction.

The Q byte comprises a device address (always 0011 for the SIOC) in the first four bits, an M bit of 0, and an N code. The N code operates in conjunction with the control code byte to select the operation that the I/O device or the adapter is to perform.

N Code	*Control Code*	*Function*
	01234567	
000	00000001	Reset interrupt request
000	00000010	Enable interrupt
000	00000100	Disable interrupt
000	00001000	Reset SIOC adapter
001	00000000	Read I/O device
010	00000000	Write I/O device
011		I/O control 1
100		I/O control 2

The I/O control N codes cause the select register to be set with the contents of the control code byte. The control code byte is bit significant as follows:

I/O Control Byte 1

Bit	*Meaning*
0	I/O 8 select
1	I/O 7 select
2	I/O 6 select
3	I/O 5 select
4	I/O 4 select
5	I/O 3 select
6	I/O 2 select
7	I/O 1 select

I/O Control Byte 2

Bit	*Meaning*
0	I/O 14 select
1	I/O 13 select
2	I/O 12 select
3	I/O 11 select
4	I/O 10 select
5	I/O 9 select
6	I/O unit 2 select
7	I/O unit 1 select

CHECKING

The contents of the data transfer register and the I/O channel data are tested for parity errors during data transfer operations and whenever instructions and data are being transmitted over the I/O channel to the SIOC. Adapter detected parity errors on data coming from the processing unit result in a processor check stop with a parity error indication. Parity errors detected in the data transfer register set a data-transfer-register-parity-check sense bit that can be tested by a sense I/O instruction.

Chapter 10. IBM 1255 Magnetic Character Reader

The IBM 1255 Magnetic Character Reader provides the capability of entering data inscribed with magnetic ink characters on paper documents. The 1255 is available in these models:

Model	*Font Read*	*Maximum Throughput***	*Number of Stackers*
1	E–13B	500/min	6
2	E–13B	750/min	6
3	E–13B	750/min	12
21	CMC 7*	500/min	6
22	CMC 7*	750/min	6
23	CMC 7*	750/min	12

* Character font used outside the United States
** Measured with 6-inch documents

A discussion of the capabilities, characteristics, and operations of the magnetic character reader can be found in *IBM 1255 Magnetic Character Reader Components Description,* Order No. GA24-3542.

OPERATION

The 1255 attaches to the system SIOC and operates through the instructions issued to the SIOC. The exact form of these instructions is discussed in the SIOC chapter of this manual.

General Programming Requirements

In addition to the instructions that actually control reader functions, the following items must be handled in a specific manner in order for the 1255 to operate with the SIOC:

1. Before executing the instructions that cause the reader to operate, the function register of the SIOC must be loaded by a load I/O instruction. The two bytes loaded must contain a 1 in bits 1 and 5 of the high-order byte and a 1 in bit 6 of the low-order byte. All other bits in these bytes must be zero.
2. The length count register must be loaded by a load I/O instruction issued to the SIOC. The number to be loaded into the register is 256 minus the number of bytes to be read from the 1255. This operation must be performed before each read instruction.
3. The SIOC data address register must be loaded with an address before reading occurs for each read operation. This address designates where in storage the data read from the document is to be stored. The address must be that of the low-order byte of the data field. This register is loaded with a load I/O instruction.
4. The device identification assigned to the 1255 is 0011. The fact that the 1255 is the device attached to the SIOC can be detected by the sense I/O instruction sensing the I/O transfer lines. Bits 0 through 3 of the high order sense bytes stored by this instruction contain the identification for the 1255. Bits 0 and 1 will be 0, and bits 2 and 3 will be 1.
5. A start I/O instruction must be issued to enable interrupts for the SIOC. The 1255 requires that processing of the documents be performed within specified periods of time to provide correct processing. The 1255 causes an interrupt at the end of every document, and this interrupt must be enabled to allow processing to commence.

Feeding Documents

Documents do not begin to feed immediately after the start key is pressed on the 1255 (when it is attached to the system). The engage command is necessary to start documents feeding. A disengage command from the processing unit is required to stop document feeding under program control. The engage command is issued by executing a start I/O instruction for the SIOC with an N code of 100 and a control code of 00000001. The disengage command is issued by a start I/O instruction with an N code of 100 and a control code of 00000010. Once engaged, the reader continues to feed documents as long as documents are available in the hopper to be fed, as long as no malfunction of either the reader or the program occurs, and until a disengage command is given or the reader stop key is pressed.

Retrieving Data from Documents

Data is obtained from documents passed through the 1255 by issuing a start I/O command specifying read. A read command must be issued for each document before that document reaches the read head. Failure to issue the necessary read command results in the document's being rejected and a signal being sent to the processing unit.

For data to be transferred from the 1255 the validity-check-and-readout switches for the desired field must be pressed.

The 1255 generates an end of transmission (EOT) signal at the end of each document. The EOT signals the SIOC to request an interrupt. The first character transferred from the 1255 enters storage at the address designated by the SIOC data address register. Subsequent characters enter successively lower storage locations.

Directing the Disposition of Documents

Documents are directed to the stackers in the 1255 by stacker select commands. These commands are generated by start I/O instructions that load the I/O select register. For 500 documents per minute models, the stacker select command must be issued within 24 milliseconds of the time a document leaves the read head (signaled by an interrupt request) if the document is to be stacked in the first (lowest) stacker, or within 50 milliseconds of the document's leaving the read head if it is to be stacked in any other stacker. For 750 documents per minute models, the stacker select command must always be issued within 24 milliseconds after the document leaves the read head. If the stacker select command is not issued within these limits, the document is rejected and the 1255 stops after all documents in the transport are directed to the reject stacker. The fact that the reader is stopped is conveyed to the processing unit.

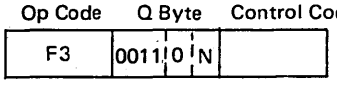
Obtaining Information about the Condition of the Reader

Indications about the condition of the reader are obtained by issuing a sense I/O command. The sense command is required to determine if the read command was issued in time, if the fields read from the document are valid, where documents are located in the transport, and if the reader is operating.

INSTRUCTIONS

Start I/O

Mnemonic: SIO



Operation: The reader performs the operation specified by the N code and the control code.

The Q byte comprises a device address (always 0011 for the reader) in the first four bits, an M bit of 0, and an N code. The N code in conjunction with the control code specifies the operation to be performed. The operations performed are:

N Code	*Control Code*	*Operation*
000 or 001	00000001	Reset interrupt request (performed by the SIOC)
000 or 001	000000010	Enable interrupt (performed by the SIOC)
000 or 001	000000100	Disable interrupt (performed by the SIOC)
000 or 001	000001000	Reset SIOC adapter (performed by the SIOC)
001	000000000	Read I/O device
010	0000000000	Invalid for 1255
011		Control I/O
100		Control I/O

The control I/O operations set the I/O select register to produce the desired operation. The following operations can be performed by each N code:

N Code 011

Control Code Bit	*Models 1, 2, 21, 22 Operation*	*Models 3 and 23 Operation*
0	Not used	Select stacker 7
1	Select stacker 6**	Select stacker 6
2	Not used	Select stacker 5
3	Select stacker 4	Select stacker 4
4	Select stacker 3*	Select stacker 3
5	Select stacker 2	Select stacker 2
6	Select stacker 1*	Select stacker 1
7	Select stacker 0	Select stacker 0

N Code 100

Control Code Bit	*Models 1, 2, 21, 22 Operation*	*Models 3 and 23 Operation*
0	Not used	Not used
1	Select reject stacker	Select reject stacker
2	Not used	Not used
3	Not used	Select stacker A
4	Not used	Select stacker 9
5	Select stacker 8	Select stacker 8
6	Disengage feed	Disengage feed
7	Engage feed	Engage feed

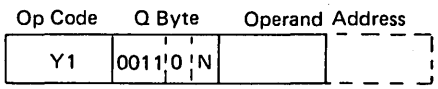
Stackers on the 12 stacker readers are arranged in two vertical rows of six stackers each. Stackers on the left bank are numbered from bottom to top: 0, 1, 2, 3, 4, R. Those on the right bank are numbered 5, 6, 7, 8, 9, A.

---

* Invalid code for standard (even/odd) sort pattern readers
** Invalid code for optional (0-4/5-9) sort pattern readers

Load I/O

Mnemonic: LIO



Operation: The two bytes contained in the two-byte field addressed by the operand address are placed in the register designated by the Q byte. The operand is addressed by the low-order byte.

The Q byte comprises the device address (always 0011) in the high-order four bits, an M bit of 0, and an N code. The N code specifies the register into which the contents of the operand field are to be loaded.

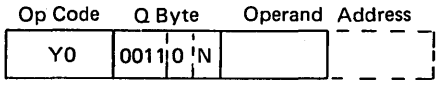
N Code	*Destination*
000	Invalid
001	I/O function register
010	SIOC length count register
011	Invalid
100	SIOC data address register
101	Data transfer register
110	Invalid
111	Invalid

Specification of an invalid N code results in a processor check stop with an invalid-Q-byte indication. The I/O function register must be loaded with the following:

High-Order Byte	*Low-Order Byte*
00000100	00000010

Sense I/O

Mnemonic: SNS



Operation: The two bytes specified by the Q byte are placed in the two-byte field addressed by the operand address. The operand is addressed by the low-order byte.

The Q byte comprises the device address (always 0011) in the high-order four bits, an M bit of 0, and an N code. The N code specifies the sense bytes or registers that are to be sensed.

N Code	*Senses*
000	Invalid
001	I/O function register
010	Length count register and status byte
011	I/O transfer lines
100	Data address register
101	Data transfer register and diagnostic byte
110	Invalid
111	Invalid

Specification of an invalid N code causes a processor check stop with an invalid-Q-byte indication.

The status byte and diagnostic byte are stored as the high-order bytes of their respective sense operations. They are bit-significant as follows:

Status Byte

Bit	*Meaning*
0	Spare
1	End request
2	Interrupt pending
3	I/O attention
4	Data transfer register parity check
5	No-op
6	Length count register overflow
7	I/O ready

The I/O transfer lines are bit-significant as follows:

High-Order Byte

Bit	*Meaning*	
0	0	(these bits are the device ID)
1	0	
2	1	
3	1	
4	Not used	
5	Not used	
6	Not used	
7	Sorter is stopped	

Low-Order Byte

Bit	*Meaning*
0	Auto reject
1	Serial number field valid
2	Transit routing field valid
3	Account number field valid
4	Process control field valid
5	Amount field valid
6	Document under read head
7	Document to be read

Sorter-is-stopped is conditioned by the main motor being stopped. A main motor stop is caused by a jam, a late stacker select, an empty feed hopper, or the reader stop key being pressed. This line is deconditioned (bit turned off) by clearing the stop condition and restarting the reader.

All field valid indicators are conditioned when their respective fields, including bracketing symbols, are read without error and deconditioned when the leading edge of the next document is sensed at the read head.

The auto reject indication turns on for any document that is rejected automatically by the reader. This occurs if a read command is not issued for a document before the document reaches the read head, a short document, an overly long document, or when a document spacing error occurs. The indicator turns on when the error condition is detected and stays on until the next document arrives at the read head. Except for a document spacing error, the indicator stays on through the second document because both documents are rejected. A stacker select command other than reject must not be issued for an auto-reject document to prevent missorting.

The document-under-read-head bit comes on when a document passes under the read head and turns off when the document leaves the read head. It can be used to determine if a document cleared the read head. If the read command has been terminated before the end of the document, a stacker select command must not be given for the document until the document leaves the read head.

The document-to-be-read bit turns on when there is a document in motion between the separator and the read head. The bit turns off when the document passes under the read head unless there is another document between the separator and the read head. The bit also turns off from a jam condition between the separator and the read head.

Test I/O and Advance Program Level

These instructions operate on the SIOC even though they must be used when operating the 1255. See the SIOC chapter for a discussion of these instructions. The test I/O busy indication means that the 1255 is performing an operation.

FEATURES

Account Number Checking

For a description of the manner in which account number checking is performed see the 1255 Components Description manual. If an incorrect account number is found when this feature is installed, the account-number-field-valid indicator is turned off. No special programming is involved with the account number checking feature.

51-Column Sort Feature

This feature allows the 1255 to handle documents shorter than the standard documents. These documents lack a transit-routing field. This fact could be used by a program to distinguish 51-column documents from others.

Dash Symbol Transmission

This feature allows the 1255 to transmit the dash symbol from the transit-routing field. Because different nations of the world use the dash symbol in different positions of their transit-routing fields, this fact can be used by programming to distinguish between checks from different countries.

High-Order Zero and Blank Selection

This feature has no effect on programming System/3 Model 6 for operating the 1255.

Chapter 11. Binary Synchronous Communication Adapter

The binary synchronous communication adapter feature (BSCA) provides System/3 with the capability of communicating with other computers or with data terminals at remote locations. The system can both transmit and receive during any communication but can do only one or the other at any one time (half duplex operation). Communication can be performed over two-wire, four-wire, or broad-band facilities in point-to-point or multi-point networks. The operation of the communication adapter is controlled by the system program.

Point-to-Point Communications Networks

The system functions in either a switched or a non-switched point-to-point network. In a non-switched network, only two stations are connected in the network. In a switched network more than two stations are available in the network and connections between any of the stations are accomplished by switching (usually through a telephone dial network). In any point-to-point network connection, one station is specified as the primary station and the other is designated as the secondary station. The stations are allowed to contend for the line (both request permission to transmit at the same time) but the primary station will always gain control of the line in any case of contention. Normally contention cannot occur in the switched network because one of the stations must be ready to receive before a call can be completed. System/3 can be designated as either the primary or secondary station. Figure 11-1 illustrates a non-switched point-to-point network.

Multi-Point Communications Network

A multi-point network has more than two stations connected to the communications lines at any one time. One station is designated as the control station for the network and all the other stations are designated as tributary stations. The control station is responsible for preventing more than one station from attempting to use the transmission lines at any one time. It does this by giving each station on the line permission to transmit in turn. The control station can transmit to any of the tributary stations at any time that one of the tributary stations is not transmitting. System/3 can serve only as a tributary station in a multi-point network. Figure 11-2 illustrates a multi-point network.

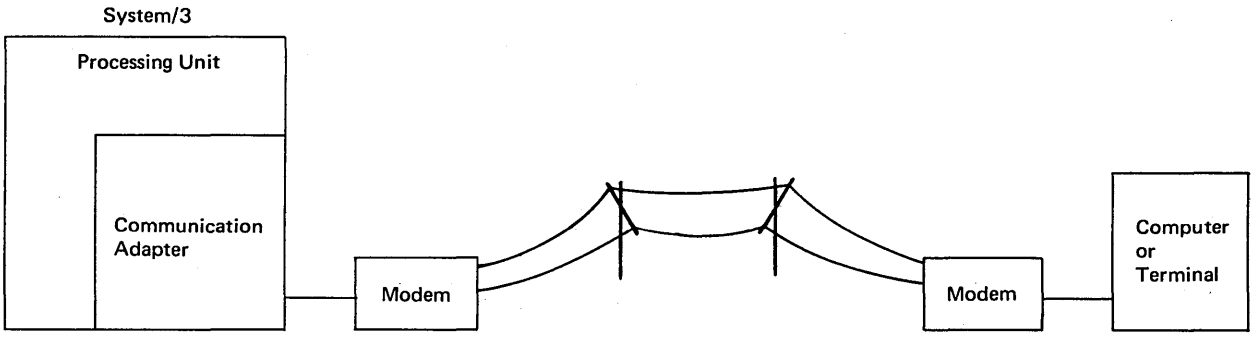


Figure 11-1. Point-to-Point Communications Network

Modems

In order to place data on a transmission line and retrieve data at the other end of the line, a unit called a modulator-demodulator (modem) is used. The modem (also called a data set) receives characters presented by the communication adapter during transmission and places them on the communications lines. During receiving, the modem accepts characters from the lines and presents them to the communication adapter. Modems operate at different rates of character transmission. The modem that is to be used with the system must be specified. (See your IBM marketing representative for information concerning what modems, or their equivalent, can be used.)

Transmission Codes

Data can be transferred in either of two codes, Extended Binary Coded Decimal Interchange Code (EBCDIC) or USA Standard Code for Information Interchange (USASCII). Either but not both codes can be specified for any particular communication adapter.

EBCDIC is the standard, 8-bit plus parity, internal binary code of System/3 Model 6. This code is illustrated in "Appendix B." USASCII is a 7-bit plus parity code illustrated in Figure 11-3. Unlike EBCDIC, which numbers its bits 0 through 7 starting at the high-order bit, USASCII numbers its bits 1 through 7 starting at the low-order bit. The 8-bit equivalent coding shown in Figure 11-3 is the code that must be in main storage for the USASCII character shown to be transmitted and is the code that is stored when the USASCII character is received. These equivalent codes are not the EBCDIC codes for the equivalent characters.

EBCDIC and USASCII not only have different numbers of bits but use different coding structures to represent characters. When USASCII is used with the System/3 Model 6 communication adapter, it must be translated from EBCDIC before transmission and translated to EBCDIC after reception. This translation is not performed by the communication adapter. The program must translate the data before and after communications operations.

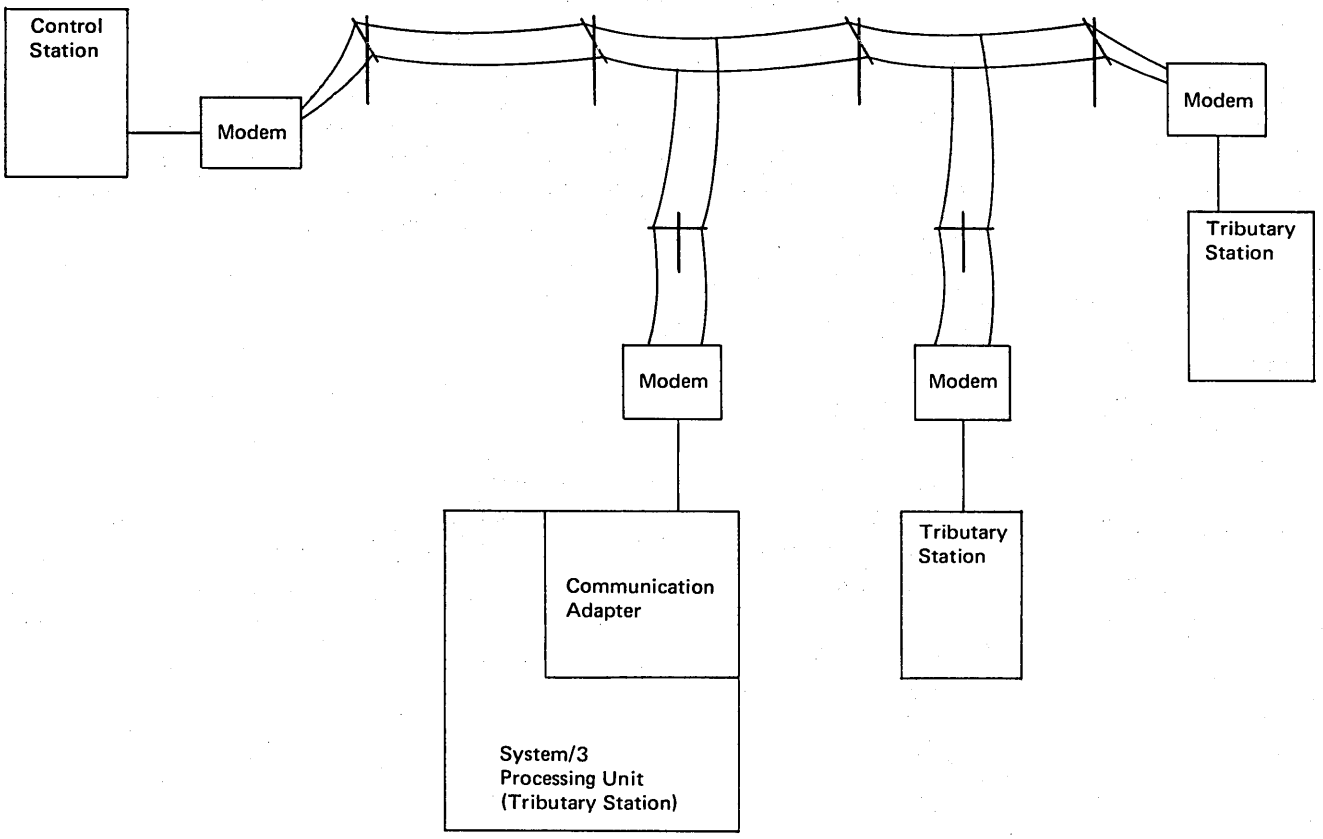


Figure 11-2. Multipoint Communications Network

USASCII Code	8-Bit Equivalent	Symbol	USASCII Code	8-Bit Equivalent	Symbol
0000000	00000000		0100100	00100100	$
0000001	00000001	SOH	0100101	00100101	%
0000010	00000010	STX	0100110	00100110	&
0000011	00000011	ETX	0100111	00100111	'
0000100	00000100	EOT	0101000	00101000	(
0000101	00000101	ENQ	0101001	00101001	)
0000110	00000110		0101010	00101010	*
0000111	00000111		0101011	00101011	+
0001000	00001000		0101100	00101100	,
0001001	00001001		0101101	00101101	—
0001010	00001010		0101110	00101110	.
0001011	00001011		0101111	00101111	/
0001100	00001100		0110000	00110000	0
0001101	00001101		0110001	00110001	1
0001110	00001110		0110010	00110010	2
0001111	00001111		0110011	00110011	3
0010000	00010000	DLE	0110100	00110100	4
0010001	00010001		0110101	00110101	5
0010010	00010010		0110110	00110110	6
0010011	00010011		0110111	00110111	7
0010100	00010100		0111000	00111000	8
0010101	00010101	NAK	0111001	00111001	9
0010110	00010110	SYN	0111010	00111010	:
0010111	00010111	ETB	0111011	00111011	;
0011000	00011000		0111100	00111100	<
0011001	00011001		0111101	00111101	=
0011010	00011010		0111110	00111110	>
0011011	00011011		0111111	00111111	?
0011100	00011100		1000000	01000000	°
0011101	00011101		1000001	01000001	A
0011110	00011110		1000010	01000010	B
0011111	00011111	US	1000011	01000011	C
0100000	00100000	Space	1000100	01000100	D
0100001	00100001	!	1000101	01000101	E
0100010	00100010	"	1000110	01000110	F

Figure 11-3 (Part 1 of 2). 7-Bit USASCII Code

USASCII Code	8-Bit Equivalent	Symbol	USASCII Code	8-Bit Equivalent	Symbol
0100011	00100011	#	1100011	01100011	c
1000111	01000111	G	1100100	01100100	d
1001000	01001000	H	1100101	01100101	e
1001001	01001001	I	1100110	01100110	f
1001010	01001010	J	1100111	01100111	g
1001011	01001011	K	1101000	01101000	h
1001100	01001100	L	1101001	01101001	i
1001101	01001101	M	1101010	01101010	j
1001110	01001110	N	1101011	01101011	k
1001111	01001111	O	1101100	01101100	l
1010000	01010000	P	1101101	01101101	m
1010001	01010001	Q	1101110	01101110	n
1010010	01010010	R	1101111	01101111	o
1010011	01010011	S	1110000	01110000	p
1010100	01010100	T	1110001	01110001	q
1010101	01010101	U	1110010	01110010	r
1010110	01010110	V	1110011	01110011	s
1010111	01010111	W	1110100	01110100	t
1011000	01011000	X	1110101	01110101	u
1011001	01011001	Y	1110110	01110110	v
1011010	01011010	Z	1110111	01110111	w
1011011	01011011	[	1111000	01111000	x
1011100	01011100	\	1111001	01111001	y
1011101	01011101	]	1111010	01111010	z
1011110	01011110	^	1111011	01111011	{
1011111	01011111	–	1111100	01111100	|
1100000	01100000	`	1111101	01111101	}
1100001	01100001	a	1111110	01111110	~
1100010	01100010	b	1111111	01111111	

Figure 11-3 (Part 2 of 2). 7-Bit USASCII Code

Data Rates

The communication adapter can control data transmission at rates from 600 bits per second to 50,000 bits per second. To provide these different rates, two different modem interfaces are used. The data set interface is a medium-speed interface that makes possible data rates from 600 bits per second to 7,200 bits per second. The data station interface is an optional feature, high-speed interface that makes possible data rates between 19,200 bits per second and 50,000 bits per second.

To control data going into the communication facility through the modem, a timing device known as a clock is used. This clock determines the data transmission rate for the system. The communication adapter does not furnish clocking to the modem except as a special feature. For the high-speed interface the modem must furnish the clock. External clocking supplied by the modem may be used with the medium-speed interface. A special feature is available that provides an internal clock in the communication adapter.

Data Transfer and Processing

The communication adapter functions as a terminal between the communication facility and the processing unit. In performing this function it is subject to the same instructions and procedures as the other input/output devices attached to the system. Data is transferred by cycle stealing in the same manner as for other I/O devices.

The communication adapter operates on interrupt level 2. An interrupt occurs at the end of each I/O operation and between data blocks during a communication. The interrupt routines should follow the form shown in Figure 11-4. Note that the routine must first determine whether the interrupt is for an intermediate transmission block (ITB) or for the end of an operation. The intermediate block check must precede the end of operation check because end of operations can follow close behind the intermediate block interrupt. If the end of operation interrupt is serviced before the intermediate block interrupt, the intermediate block interrupt may be lost.

SUBFEATURES OF THE COMMUNICATION ADAPTER

Two subfeatures of the communication adapter are standard: intermediate block checking and auto answer. The auto answer feature enables the communication adapter to respond to a telephone request for data communications automatically without operator intervention. The intermediate block checking feature allows transmission and reception of checking characters for checking the accuracy of communication without interrupting the steady flow of information from the transmitting station to the receiving station.

In addition to the two standard subfeatures, certain optional subfeatures are offered to enhance the capabilities of the communication adapter.

Station Selection

This feature allows the system to operate as a tributary station in a multi-point communications network. This feature excludes the auto call feature and is not available with the high-speed interface.

Internal Clock

This feature provides an internal clocking system in the communication adapter to allow operation with modems that do not provide clocking to the adapter. The internal clock feature provides the following transmission rates:

600 bits per second
1200 bits per second
2000 bits per second
2400 bits per second.

Only one of the above transmission rates can be specified for each communication adapter. This feature excludes the high-speed interface.

High-Speed Interface

This feature enables the communication adapter to interface with modems that provide data rates between 19,200 bits per second and 50,000 bits per second. This feature excludes the internal clock feature, so the modem must furnish data clocking when this feature is installed. The high-speed interface feature is not available through World Trade Corporation.

Auto Call

This feature permits automatic connection with a remote station to be established by means of a program instruction. An auto calling unit (ACU), not supplied by IBM, must be used with this feature to enable connection to occur. This feature excludes the station selection feature. The auto call feature is not available through World Trade Corporation.

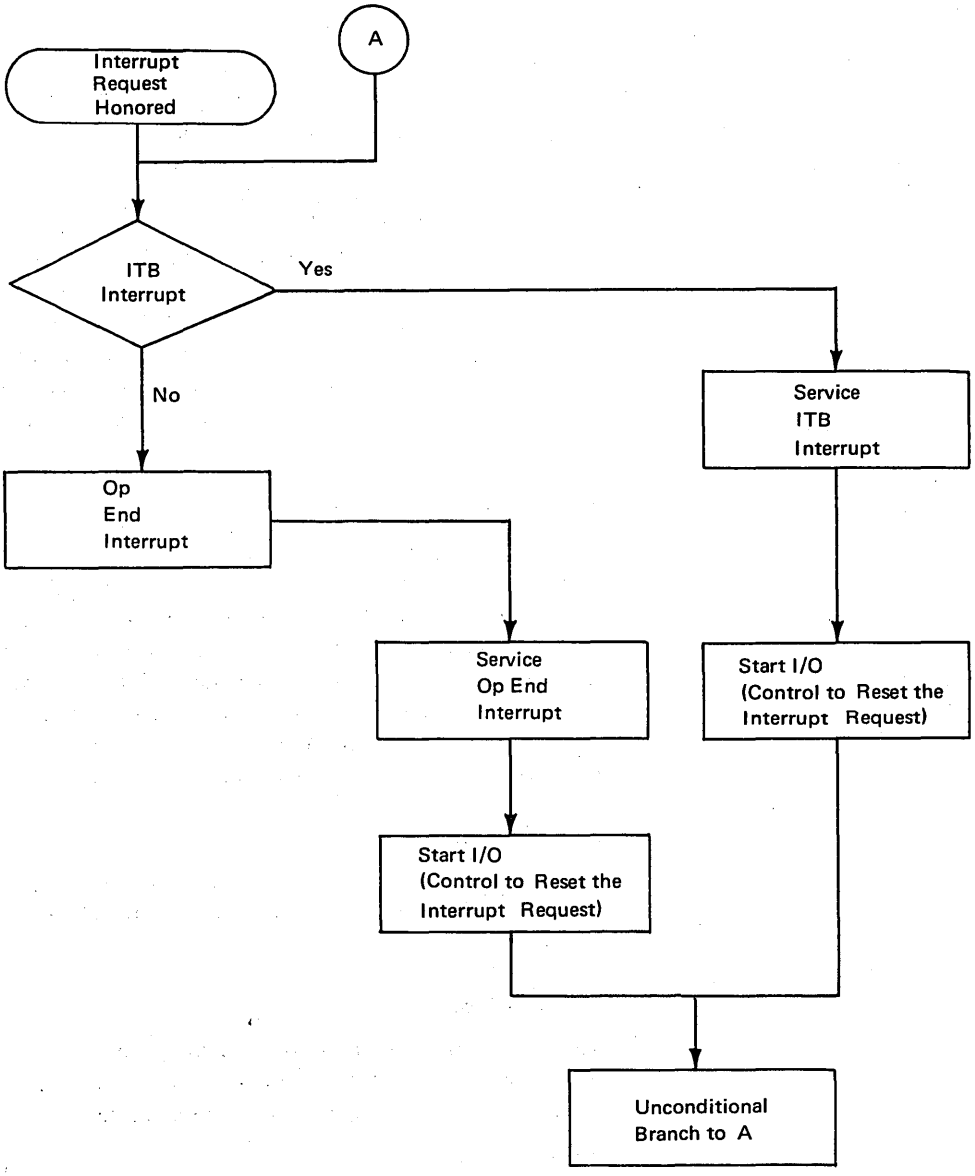


Figure 11-4. Generalized Communication Adapter Interrupt Routine

Transparency

This feature allows all of the 256 possible bit combinations available in the EBCDIC to be transmitted through the communication adapter as data. Certain EBCDIC characters are control characters that cause the communication adapter to perform a function. These characters cannot be used as data in nontransparent operations. The transparency feature allows these characters to be handled as data. This feature excludes the USASCII option.

Data Rate Selection

This feature allows switch selection of the data transmission rate. The switch provides either 600-bit-per-second or 1200-bit-per-second data transmission and reception. The data rate selection feature is available only through World Trade Corporation.

LOCAL COMMUNICATIONS ADAPTER (LCA)

The local communications adapter feature allows direct attachment (no data set/modem) of an IBM 3741 Data Station Model 2 to an IBM System/3 Model 6. The LCA is installed in the IBM 5406 Processing Unit. The external (data set/modem) cable furnished with the attached 3741-2 is plugged directly into a connector provided with the LCA feature.

Only one device may be physically attached to the LCA at a time. The LCA provides clocking at a rate of 2400 bits per second for the attached device and operates in a point-to-point, non-transparent mode using extended binary coded decimal interchange code (EBCDIC).

The LCA cannot be installed on a system with an installed BSCA, and only one LCA can be installed. None of the BSCA subfeatures can be used with the LCA feature.

Registers, indicators, and programming used for operation of the BSCA feature are used for the LCA feature.

LOCAL STORAGE REGISTERS USED BY COMMUNICATIONS ADAPTERS

Three local storage registers are provided for the communication adapter: the current-address register, the transition-address register, and the stop-address register. These registers hold the storage addresses of data or control bytes at which certain actions are to occur, or the address of the next byte to be transmitted or received.

The current-address register contains the address of the next byte to be operated on. When data is being transmitted, this register is used to address storage for each byte that is to be transmitted. When data is being received, this register is used to address storage for storing each byte as it is received from the line.

The transition-address register stores the address at which transition is desired between transmitting and receiving in a transmit-and-receive operation. When the address in the current-address register equals the address in the transition-address register, the adapter stops taking data from storage on cycle steals and begins stealing cycles to store the characters received from the communications line.

The stop-address register stores the address at which communication adapter operation is to stop. When the address in the current-address register equals the address in the stop-address register, the communication adapter ends its operation and generates an interrupt request.

COMMUNICATION ADAPTER MODES AND STATES

The communication adapter must be in either of two modes when it is on line with power on. These two modes are data mode and control mode. The adapter is in data mode at any time when data is being passed between the adapter and some other station in the network. When the adapter is not in data mode, it is in control mode. Examples of control mode are those times when the adapter is attempting to gain control of the communications line in order to transmit data and those times when the adapter is monitoring the line waiting to be called or polled.

When the communication adapter is in the data mode, it must operate in one of two states: master or slave. The communication adapter is in the master state when it is the station transmitting data and waiting for responses from the remote station. The adapter is in the slave state when it is receiving data from the remote station and responding with characters that tell the remote station the status of the data received. During an exchange of communications, the adapter can shift from master state to slave state and back to master state as many times as necessary to complete the exchange of data.

COMMUNICATION ADAPTER CHECKING

Each unit of data that is transmitted between the system and another station is checked for accuracy of data reception. This is necessary because electrical noise on the communications lines can cause extra bits to appear at the receiving end of the line. Two different schemes for developing the block check characters are used by the EBCDIC and by the USASCII systems.

The EBCDIC system develops a two-byte block check character by a cyclic redundancy code. These two bytes are transmitted following certain adapter control characters. The block check characters do not enter storage but are compared to the block check characters developed at the receiving station from the data received. If the block check characters received at the receiving station match the block check characters developed by the receiving station from the data received, the receiving station acknowledges the receipt of the data. If the block check characters do not match, the receiving station signals the transmitting station that an error occurred during the transmission.

The USASCII system uses a combined longitudinal/vertical redundancy check to develop a single-byte block check character. This block check character is used in the same manner as the EBCDIC block check characters.

In addition to the block check characters, each byte received or transmitted is checked by the processing unit circuitry for correct parity. If a character with even parity is detected, the processing unit stops with the processor check light on.

CONTROL CODES

Certain of the characters in both the EBCDIC and USASCII codes have been designated as line control characters. The USASCII line control characters are shown with the rest of the USASCII code in Figure 11-3. The EBCDIC line control characters are shown in "Appendix B." These control characters must surround the data and constitute the initialization sequences necessary to establish contact between stations and control sequences necessary for correct data handling. The System/3 Model 6 program is responsible for framing data to be transmitted with the correct control characters. When these control characters are received from the remote station, they cause the adapter to perform some function. In addition, the control characters are stored in main storage just as they are received from the line. (For an exception, refer to "Synchronous Idle-Syn.")

The individual line control characters provide control to the communication adapter. In addition, other control functions are obtained by using sequences of control characters or by using control characters followed by characters for which graphics (printable characters) have been defined. Each control function (whether initiated by a line control character or a line control sequence) has a mnemonic. These mnemonics are used in the following discussions of communication adapter operations. Figure 11-5 shows the control characters and sequences.

CONTROL CHARACTER USES

Each of the control characters, whether used alone or with another character, has its own special use. Because of the characteristics of these usages, certain rules must be followed in the sequencing of the characters.

Start of Heading (SOH)

This character is used to begin the heading of the message to be transmitted. Not all messages will have a heading, but in systems where heading data is required for message routing this character is used to indicate that the data following the SOH character is heading data. The SOH character causes the communication adapter to leave control mode and enter data mode. If the heading is broken into parts by ITB characters, the SOH character is used to indicate that each of the parts is part of the heading. The use of the SOH character following a block of data ended by the ITB character is valid only if the preceding block of data was a heading block delimited by a SOH character. The use of the SOH character within a heading is invalid unless the heading has been broken by an ITB character.

Name	Mnemonic	EBCDIC	USASCII
Start of Heading	SOH	SOH	SOH
Start of Text	STX	STX	STX
End of Transmission Block	ETB	ETB	ETB
End of Text	ETX	ETX	ETX
End of Transmission	EOT	EOT	EOT
Enquiry	ENQ	ENQ	ENQ
Negative Acknowledge	NAK	NAK	NAK
Synchronous Idle	SYN	SYN	SYN
Data Link Escape	DLE	DLE	DLE
Intermediate Block Character	ITB	IUS	US
Even Acknowledge	ACK 0	DLE (70)	DLE 0
Odd Acknowledge	ACK 1	DLE /	DLE 1
Wait before Transmit -- Positive Acknowledge	WACK	DLE ,	DLE ;
Mandatory Disconnect	DISC	DLE EOT	DLE EOT
Reverse Interrupt	RVI	DLE @	DLE
Temporary Text Delay	TTD	STX ENQ	STX ENQ
Transparent Start of Text	XSTX	DLE STX	
Transparent Intermediate Block	XITB	DLE IUS	
Transparent End of Text	XETX	DLE ETX	
Transparent End of Transmission Block	XETB	DLE ETB	
Transparent Synchronous Idle	XSYN	DLE SYN	
Transparent Block Cancel	XENQ	DLE ENQ	
Transparent Temporary Text Delay	XTTD	DLE STX DLE ENQ	
Data DLE in Transparent Mode	XDLE	DLE DLE	

Figure 11-5. Control Characters and Sequences

Start of Text (STX)

This character is used to indicate the beginning of text in a message. This character can follow data blocks that were begun with the SOH character or can introduce the text in messages that contain no heading. If no heading preceded it, the STX character causes the communication adapter to leave control mode and enter data mode. The STX character can be used, but is not necessary, to indicate that a new block of text is beginning after a block ended with the ITB character. The STX character *must* be used to begin new blocks of text following blocks of text ended by ETB or ETX characters.

End of Intermediate Data Block (ITB)

Placing this character in a message causes the communication adapter to perform a block check character operation (transmit the block check characters if the adapter is the master station or receive and compare the block check characters if the adapter is the slave station) and clear the accumulated block check characters. If the block check characters received at the slave station match the characters accumulated at the slave station, the adapter stores the address of the next higher storage location in the transition-address register and generates an interrupt request. The adapter then continues to receive the data transmitted by the remote station. If the block check characters do not match, no interrupt is generated, and the adapter stops accepting data. However, the remote station continues to transmit.

End of Transmission Block (ETB)

This character is used to signal the end of a block of data in a message that is subdivided into parts for better transmission control. The ETB character signals the end of a transmission block that is not the end of the message. The ETB signal demands a response from the receiving station.

When the communication adapter is transmitting, the ETB signal must be in storage at a position one less than the position in which the first character received will be stored. This prepares storage to receive the response to the block check characters. The block check characters are sent automatically by the communication adapter immediately following the transmission of the ETB character. When the communication adapter is receiving, the ETB character signals that the block check characters will follow immediately. As soon as the block check characters are received and compared, an interrupt request is generated. At this time the program must sample the communication adapter to determine if all data was received correctly. As a result of this determination the program must frame a reply and begin an I/O operation to signal the original transmitting station that the data was received either correctly or incorrectly. The ETB character can be used in either text or heading data.

End of Text (ETX)

This character is used to signal the end of a message. The ETX signal demands a response from the receiving station.

When the communication adapter is transmitting, the ETX character must be in storage at a position one less than the position in which the first character received will be stored. This prepares storage to receive the response to the block check characters. The block check characters are transmitted automatically by the communication adapter immediately following the transmission of the ETX character. When the communication adapter is receiving, the ETX character signals that the block check characters will follow immediately. As soon as the block check characters are received and compared, an interrupt request is generated. At this time the program must sample the communication adapter to determine if all data was received correctly. As a result of this determination the program must frame a reply and begin an I/O operation to signal the original transmitting station that the data was received either correctly or incorrectly.

Enquiry (ENQ)

This character is used in both control mode and data mode. When detected in data mode this character causes a shift to control mode. In control mode this character is used to establish contact between stations. In data mode this character is used to request a reply or to signal that the block of data just transmitted is to be ignored. The ENQ character always calls for a response from the station that received the ENQ. The transmitting station must have the ENQ characters stored one character before the location at which the first character of the reply will be stored. The receiving station communication adapter generates an op end interrupt request when the ENQ character is received. The receiving station must then analyze the interruption, frame a reply to the remote station, and initiate an I/O instruction to transmit the reply.

Negative Acknowledge (NAK)

This character is used as a negative response signal in both control mode and data mode. The control mode NAK indicates that the terminal is not in condition to receive when a control signal instructing the communication adapter to prepare to receive is detected. In data mode, NAK indicates that the data transmitted since the last acknowledgement character was sent was not received correctly. Modifying information specifying the reason for the negative acknowledge can be included in the response. In such an operation the NAK character is the last character in the sequence.

The NAK character is always followed by four contiguous 1 bits. These 1 bits are stored in the low-order four bits of the byte immediately following the byte where the NAK character is stored. The high-order four bits of this byte should be ignored. Receiving a NAK response causes the communication adapter to generate an interrupt request.

End of Transmission (EOT)

This character signals the end of an exchange of data between two stations and does not require any reply. This character is also used in multi-point networks as the first character in the polling or selection sequence to insure that all stations are in control mode and (by the tributary station) as the negative reply to the polling sequence. The EOT character causes the communication adapter to leave data mode and to enter control mode. EOT sent when the adapter is in control mode causes the system to revert to the conditions that apply when the network first enters control mode. No characters other than SYN can immediately precede the EOT character.

The EOT character is followed immediately by four contiguous 1 bits. These bits will be stored in the next higher position of main storage after the EOT character. Reception of an EOT character causes an interrupt request to be generated by the communication adapter.

Synchronous Idle (SYN)

This control character is generated automatically by the communication adapter and used to keep the sending and receiving stations in synchronism with each other. This character is also used by the communication adapter as a time-fill character when the adapter must wait for data characters from main storage. SYN characters received from the line are not sent to main storage but used only to maintain synchronism. If SYN is one of the characters in storage in the transmitting field at the transmitting station, it will be transmitted but will be stripped from the message at the receiving station.

Data Link Escape (DLE)

This character is used to provide additional communication signals by combining with other signals in sequences of two or more characters. See "Data DLE in Transparent Mode."

Even Acknowledge (ACK 0)

This two-character control sequence is a positive acknowledgement by the receiving station of the previous transmission block. In data mode it indicates that the block check characters that have been received since the last acknowledgement sequence have matched the block check characters accumulated by the communication adapter. In control mode it is used as a signal that the communication adapter is ready to receive. ACK 0 always requires a response by the station that receives it. Receiving an ACK 0 sequence causes the adapter to generate an interrupt request.

Odd Acknowledge (ACK 1)

This control sequence is used only in data mode. It is the positive acknowledgement of correct receipt of the odd numbered data blocks of the message. The first block of data transmitted by the master station is always the first odd block. Receiving an ACK 1 causes the communication adapter to generate an interrupt request.

Wait before Transmit—Positive Acknowledgement (WACK)

This two-character control sequence is used to indicate that the last data block was received correctly but that the receiving station is not ready to continue receiving. It can also indicate that a station called on to receive is temporarily unable to receive but will be able to receive in a short time. Receiving WACK from the other station causes the communication adapter to generate an interrupt request.

Mandatory Disconnect (DISC)

This two-character control sequence is used only in switched point-to-point networks. The system transmits this signal to tell the remote station to disconnect from the transmission lines. When the system receives this signal from the remote station, the communication adapter generates an interrupt request.

Transparent Start of Text (XSTX)

This two-character control sequence is used in the same manner as the STX control character. When XSTX is transmitted or received, it causes the transparency option to become active and disables the non-transparent control functions that are valid data mode control characters in non-transparent operation. The XSTX sequence must begin every block of transparent text because any of the transparent text block delimiter characters returns the communication adapter to the non-transparent operating condition.

Transparent End of Intermediate Transmission Block (XITB)

This two-character control sequence is used only in transparent operation to delimit the end of a transparent text block for error checking purposes without requiring that the receiving station respond immediately with an indication of correctness of reception. It operates in the same manner as the ITB control character and causes the communication adapter to enter the non-transparent operating condition.

Transparent End of Transmission Block (XETB)

This two-character control sequence is used only in transparent mode. It performs the same functions as ETB performs in non-transparent operations. XETB causes the communication adapter to enter the non-transparent operating condition.

Transparent End of Text (XETX)

This two-character control sequence is used only in transparent operation. It performs the same functions as ETX performs in non-transparent operation. XETX causes the communication adapter to enter the non-transparent operating condition.

Transparent Synchronous Idle (XSYN)

This two-character control sequence is used in transparent operation to maintain synchronism between the transmitting station and the receiving station. It can also be inserted as a time-fill character when the communication adapter must wait for data from main storage. The XSYN is not stored. XSYN does not cause the communication adapter to enter the non-transparent operating condition.

Reverse Interrupt (RVI)

This two-character control sequence is used by the slave station to request that the master station end its transmission and allow the slave to transmit a priority message. RVI is sent as a response to a block of text instead of one of the positive acknowledgement characters. RVI implies a positive acknowledgement of the last text block. The master station is expected to terminate the transmission immediately or at the earliest time possible after transmitting one or more additional transmission blocks. Successive RVI responses are not allowed except as responses to ENQ. Receiving RVI causes the adapter to generate an interrupt.

Temporary Text Delay (TTD)

This two-character control sequence is used by the master station to inform the slave station that there will be a delay that exceeds two seconds in transmitting the next data block or that the master station wishes to abort and cancel the transmission in progress. The response of the slave station to TTD is NAK. When the communication adapter receives a TTD sequence it generates an interrupt request. The forward abort sequence is TTD, NAK, EOT.

Transparent Temporary Text Delay (XTTD)

This signal is never transmitted by the system communication adapter. It can be received, however, from a station that is placed in the transparent operating condition by a switch setting. The functions of XTTD are the same as those of TTD.

Transparent Block Cancel (XENQ)

This two-character control sequence serves the same function as the ENQ signal serves in non-transparent data mode. Ending a text block (which must be in transparent mode) with XENQ causes the slave station to discard the text block. When the communication adapter receives XENQ, it will generate an interrupt request. Sending or receiving XENQ causes the communication adapter to enter non-transparent mode.

Data DLE in Transparent Mode

The two-character sequence of two successive DLE characters in transparent mode is interpreted as conveying the second DLE as a binary data character rather than as a control extension character. The first DLE is discarded and the second is stored in main storage as a character.

Change of Direction Characters

All of the control characters and sequences except the ITB and XITB cause the communication facility to expect a change in direction of transmission. When the system is transmitting, it must transmit a change of direction character as the last character of its transmission to allow the slave station to change direction and respond. When the system is receiving, receiving a change of direction character causes the communication adapter to generate an interrupt request.

COMMUNICATIONS OPERATIONS

A communication system performs two basic functions: transmitting messages and receiving messages. Each of these functions requires several subfunctions. For transmitting messages the subfunctions are:

1. Establish communication with the remote station.
2. Transmit the message.
3. Ascertain that the remote station received the message.

The subfunctions for receiving are:

1. Answer when the remote station attempts to contact you.
2. Receive the message.
3. Assure the remote station that you received the message or indicate that the message was not received correctly.

Two operations must be performed before either of the transmit or receive functions can be performed. The communication adapter must be enabled and communication adapter interruptions must be enabled.

Message Transmission Operations

Establishing Contact between Stations—Point-to-Point—Non-Switched

Establishing contact between stations involves transmitting an ENQ character and waiting for a reply from the other station. The three communication adapter local storage registers must be loaded with the proper addresses to enable this operation to occur. The current-address register is loaded with the storage address of the ENQ character. The transition-address register is loaded with the address of the next higher storage address from the ENQ address. The stop-address register is loaded with the next higher address from that in the transition-address register.

After the registers are prepared, the start I/O transmit-and-receive instruction is issued. The communication adapter performs a cycle steal operation and transmits the ENQ character obtained from storage. The current-address register is updated and because the current-address register equals the transition-address register, the communication adapter leaves the transmit condition and enters the receive condition.

When the communication adapter is in the receive condition, the called station may or may not respond to the ENQ signal. If the called station responds with a valid control character, the communication adapter stores the character in main storage at the location specified by the current-address register. After the character is stored, the communication adapter generates an interrupt request. The interrupt routine must check for a valid response character (ACK 0, NAK, WACK). If the response character is ACK 0, the called station is ready to receive and the system can proceed with the message transmission phase of the transmit operation. If the response is NAK, the called station is not ready to receive and the remote operator must be signaled to prepare the called station for receiving. If the response is WACK, the called station is temporarily unable to receive but will be able to receive shortly. In this case, the system may retransmit the ENQ signal by reloading the address of the ENQ character in the current-address register. The system may also choose to cancel the attempt at communications at this time and try again later. If the called station fails to respond with a valid control character, the communication adapter generates a receive time-out at the end of three seconds. This time out causes an interrupt request. The procedure followed by the interrupt routine depends upon whether the system is the primary station in the network or the secondary station. The primary station will normally repeat the transmission of the ENQ character immediately. Secondary stations, in order to allow the primary station to repeat its signal and be heard if both the primary and secondary stations transmitted the ENQ at the same time, will normally perform a start I/O receive instruction before transmitting the ENQ a second time. If no response is received to several ENQ signals, the remote operator must be advised to prepare the remote station for receiving.

Establishing Contact between Stations–Point-to-Point–Switched

If the system does not have the auto call feature installed, the operator must dial the number of the remote station with which communication is desired. The auto call feature allows a circuit to be established by a program without intervention by the operator. For auto call operation, the auto call instruction should be issued as soon as possible after the communication adapter is enabled to reduce the possibility of receiving a call before the desired call is initiated. The communication adapter local storage registers must be loaded before the auto call start I/O instruction. The current-address register must be loaded with the storage address of the first digit of the number to be called. The stop-address register must be loaded with the address in the current-address register plus the number of digits to be dialed. When the call is answered or the auto calling unit determines that the call cannot be completed, the communication adapter generates an interrupt request.

The interruption routine then begins the process of establishing contact between the system and the remote station or abandons the call if the remote station cannot be dialed.

Whether the circuit between the system and the remote station is established by operator intervention or by the auto call facility, the following procedure is used to establish the transmission contact between the two stations. The system loads the current-address register with the address of the first character of the calling sequence, loads the transition-address register with the address of the next higher position of storage after the last character of the calling sequence, and loads the stop-address register with the address in the transition address register plus the number of characters in the response sequence. A transmit-and-receive start I/O instruction is then issued. The normal calling sequence is a station identification for the calling station followed by an ENQ character or the ENQ character alone. The normal response is the station identification of the called station followed by ACK 0. This is used to insure that the correct station is contacted.

Issuing the transmit-and-receive start I/O instruction causes the communication adapter to begin cycle stealing and transmitting the characters obtained from storage. The current-address register is increased by 1 for each character transmitted. When the current-address register equals the transition-address register, the communication adapter stops transmitting and prepares to receive. When the remote station sends its response to the system's transmission, the communication adapter steals cycles and stores the response in main storage at the addresses specified by the current-address register. When the current-address register equals the stop-address register, when the communication adapter detects a valid control character in the response, or when no response is received for 3 seconds, the communication adapter generates an interrupt request.

The interruption routine must determine whether the interruption is caused by the normal 'go ahead' (station identification followed by ACK 0), a negative response (NAK), too many characters in the response (no control character before the current-address register equals the stop-address register), or no response.

If the interruption is caused by the positive acknowledgement, the program can proceed to transmit the message. If the interruption is caused by the NAK response, the initial contact sequence should be retried. If retrying the sequence results in NAK, the remote operator must be directed to prepare the remote station for receiving. Too many characters in the response indicates that the operation should be retried. The stop address should be checked to insure that it allows for the correct number of characters in the response.

No response to the transmission indicates that either the remote station is not in condition to respond or that the remote station transmitted a request for line control (station identification followed by ENQ) at the same time as the system. The action taken by the system depends upon whether the system is the primary station or the secondary station. If the system is the primary station, it should immediately transmit another bid for line control to signal the remote station (which will listen longer for a response than will a primary station). If the system is the secondary station, it should perform a receive operation to allow the primary station more time to bid for the line. If two or more successive line bids result in no response, it must be assumed that the remote station is not in condition to receive. An error recovery procedure must be executed that allows getting the remote station on line and ready to receive.

Establishing Contact between Stations—Multi-Point

In multi-point operations the system can operate only as a tributary station. Tributary stations do not originate contact between stations but do have responsibilities as contact is established. In a multi-point system the tributary stations are allowed to transmit only when they are asked if they have messages to transmit. This inquiry for messages by the control station is referred to as polling. In order to be available when the control station signals that the system will be allowed to transmit, the system should issue a receive-initial start I/O instruction. In this operation the communication adapter monitors the data and control sequences transmitted over the line. An EOT character transmitted over the line places the adapter in control mode and allows it to interpret its own address as an address. The local storage register must be prepared for this operation by loading the current-address register with the address of the location in storage where the first character received will be stored, loading the stop-address register with the address in the current-address register plus one less than the number of characters to be received in the initialization sequence, and loading the low-order byte of the transition-address register with the single-character address of the communication adapter. The station address is transmitted as two identical characters. Bit 2 of the character (bit 6 of USASCII characters) is ignored in coding the address so that, for example, an address BB and an address SS both will operate the same communication adapter. The high-order byte of the transition-address register is not used. When the communication adapter detects its address as the first two characters after an EOT character, it begins storing the characters in main storage. When a valid control character is received, when the current-address register equals the stop-address register, or when the adapter does not receive a character for three seconds, the adapter stops receiving and generates an interrupt request.

The interrupt routine must determine what caused the interruption. The address characters will indicate if the address sequence carries permission to transmit (polling) or indicates that the control station wishes to transmit (selection). If the interruption is caused by a receive time-out or by the current-address equal to the stop-address without a control character's being detected, the interruption routine must determine the cause of the improper sequence and correct it.

If the communication adapter has received a valid polling sequence, the system begins to transmit immediately; if the system has nothing to transmit, it responds with EOT.

Data Transmission

After communications between the stations is established, the system can begin transmitting. Transmission is begun by a transmit-and-receive start I/O instruction. This instruction uses the current-address register for cycle stealing data bytes for transmission by the communication adapter. The transition-address register is used to determine when all of the data bytes have been transmitted and when to wait for a reply from the remote station. The stop-address register is used to determine when the correct number of characters has been received from the remote station.

Contact between stations is established with both stations in control mode. The messages to be transmitted must be framed by control characters to signal the remote station that the characters that follow are message characters, not control characters. Either of two control characters, depending upon the requirements of the communication system, can be used to start the message. If handling or addressing information that is not to be processed as part of the text is required for proper message handling, a heading must be included. This heading should be preceded by an SOH character to place the communication adapter in data mode and to signal that the information that follows is heading data and not part of the text. If the heading is very long, it may be desirable to break it up with ITB characters to allow heading processing decisions to be made while heading data is still being received. At the end of the heading, an STX character signals the beginning of the text proper. Long text can be broken up for checking purposes and to allow processing while more data is received by inserting ITB or ETB characters. If the text does not require breaking up, it should be ended with an ETX.

Sending an ITB causes the communication adapter to transmit its two block check characters, but does not require a reply from the remote station. As soon as the block check characters have been transmitted, the adapter resumes stealing cycles and transmitting the data obtained from storage. If block checking is performed with ITB, no STX is required to start the next text block.

If ETB is used to break up long text, it must be the last character preceding the transition address. When the current-address register equals the transition-address register, the communication adapter stops transmitting and enters receive operation. The remote station must check the block check characters sent automatically following the ETB character and respond with either a positive acknowledgement character or a negative acknowledgement character. A positive acknowledgement character is ACK 0 for even-numbered text or heading blocks and ACK 1 for odd-numbered text and heading blocks. The negative acknowledgement character is NAK. If the remote station requires more than two seconds to frame a reply, it may send WACK. If no reply is received within three seconds, a receive time-out occurs and causes the communication adapter to generate an interrupt request. Reception of a valid control character causes the adapter to store that character in main storage and generate an interrupt request.

The interrupt routine must check to determine the cause of the interrupt. If the reply is a positive acknowledgement character, it must be checked to determine that it is the correct acknowledgement character. Reception of an odd acknowledgement character when an even one is expected or vice versa indicates that the remote station missed an entire block or missed a control sequence. If the positive acknowledgement character is correct, the system must issue another transmit-and-receive instruction either to transmit the next block of text or another message, or to transmit an EOT character to indicate that no more messages will be transmitted.

If the positive acknowledgement character is incorrect, the system must retransmit all of the message that was transmitted since the last correct positive acknowledgement character was received.

A negative acknowledgement response requires that the message be retransmitted from the last valid positive acknowledgement.

A WACK response requires the system to wait for a short period (the two-second time-out function is useful for this) and transmit an ENQ character. The transmit-and-receive instruction is used for this so that the system can receive the remote station response. This can be a positive acknowledgement character, a negative acknowledgement character, or another wait before transmitting character. The actions following this response are the same as those detailed previously. If no response is received in three seconds and the receive time-out occurs, the system must issue a transmit-and-receive start I/O instruction to transmit an ENQ character and receive a reply.

If negative responses, wait responses, or lack of responses are repeated, the program must count the number of such responses. After a fixed number of such responses the program should abort the transmission, by disabling the communication adapter, and try again. Such an error recovery procedure can also include an alert to the remote station operator that trouble is being encountered and that he should check that his equipment is conditioned for operation.

If more than one message is to be transmitted, each message must end with an ETX character. This character signals the end of a message and allows the next block of data to begin with an SOH character.

When all messages have been sent and acknowledged, the system must transmit an EOT character to signal that it is relinquishing control of the line. This character does not require a reply from the remote station. It does, however, give the remote station permission to transmit. The system should leave itself in a condition to receive any messages from the remote station.

Communications Assurance

As the system transmits its messages to the remote station, the communication adapter accumulates checking characters as a function of the number of bits in each character transmitted. Certain of the control characters or sequences cause the communication adapter to transmit these checking characters. The remote station takes one of two actions depending upon the character transmitted.

If the control character is ITB, the remote station compares the block check characters to the block check characters it has accumulated from the same data. If the block check characters from both stations match, the remote station notes this and continues to receive data from the system. If the characters do not match, the remote station notes this condition and the system continues to transmit. When the system transmits a control character requiring a response, the remote station transmits either a positive acknowledgement, if all blocks have been received without errors, or a negative response, if an error has been detected in any block. If the remote station is equipped to signal to the system which block was in error, the system may retransmit all data beginning with that block. Otherwise, the system must retransmit all data from the last correct positive acknowledgement.

If the control character is ETX or ETB, the remote station compares the block characters received from the system with its own block check character. If the characters match, the remote station transmits either ACK 0, if the block is an even-numbered block, or ACK 1, if the block is an odd-numbered block. The first block transmitted after communications is established is always an odd-numbered block. The system must compare the acknowledgement character to its own record of odd and even blocks and determine that the correct response was received. An incorrect response requires the system to perform some error recovery routine to determine what information was lost between it and the remote station. This information must then be retransmitted. If the block check characters transmitted by the system do not match the block check characters accumulated by the remote station, the remote station sends a NAK response to an ETB or ETX character. The system must retransmit the data block for which the block check characters did not compare.

Message Reception Operations

Establishing Contact between Stations—Point-to-Point—Non-Switched

In a receiving operation, contact between stations is initiated by the remote station. In order to be listening when the remote station attempts to establish contact, the system must have the communication adapter in the receive condition. For this purpose a receive-initial start I/O instruction is used. For this instruction the current-address register is loaded with the storage address of the location where the first character of the initialization sequence is to be stored. The stop-address register is loaded with a storage address equal to the current-address register plus the number of characters to be received in the initialization sequence. In a non-switched network this will be one character.

After the receive-initial start I/O instruction is issued, the communication adapter waits for a signal from the remote station. The first signal from the remote station (after synchronizing characters are received) should be an ENQ character. This character causes the communication adapter to generate an interrupt request. In honoring the interrupt request the system issues a transmit-and-receive start I/O instruction. The transmit portion (addressed by the current-address register) is normally the two-character sequence ACK 0. The transition-address register must be loaded with the address of the main-storage location one higher than the address of the location of the second character of the ACK 0 sequence. The stop-address register must be loaded with the address from the transition-address register plus the number of characters expected in the message.

If the system is not ready to service the message, the reply to the ENQ character can be sent as a transmit-only operation with the character NAK. Transmit-only operation occurs when the transition-address register equals the stop-address register in a transmit-and-receive start I/O instruction. If the system can service the message within two seconds, the WACK character can be used to signal the remote station to try again after a two-second time-out.

Establishing Contact between Stations—Point-to-Point—Switched

In a switched network, the contact between stations is also initiated by the remote station. The same receive-initial start I/O instruction is used to monitor the communication facility for a call. When the communication adapter detects a call, the characters following the synchronization characters are placed in main storage starting at the address contained in the current-address register and continuing until either a change of direction character is received or the current-address register equals the stop-address register. The stop-address register must be loaded with the address of the first character storage location plus the number of characters to be received in the initialization sequence. The normal initialization sequence is a station identification (for the remote station) followed by ENQ.

When the ENQ is detected, the current-address register equals the stop-address register, or if no characters are received for three seconds after a call is recognized, the received character is stored and the communication adapter generates an interrupt request. The interruption routine must determine the cause of the interruption and generate the proper response. If the interrupt is caused by the current-address register's equaling the stop-address register before the ENQ is detected or failure to receive a character within three seconds, the interrupt routine must perform an error recovery procedure to obtain the correct initialization procedure.

If the interrupt request is caused by the ENQ character (the normal procedure), the system must execute a transmit-and-receive start I/O instruction to transmit the response to the line bid and receive the incoming messages. Normally the transmission consists of the system identification followed by ACK 0. If the system is not ready to service the message, either NAK or WACK can be substituted for ACK 0 in the sequence. For the start I/O operation the current-address register is loaded with the storage location of the first character of the station identification, the transition-address register is loaded with the next higher storage address after the address of the last character of the response sequence, and the stop-address register is loaded with the transition-address register contents plus the number of characters expected in the message. After sending the response, the communication adapter immediately assumes the receiving condition.

Establishing Contact between Stations—Multi-Point

As in the other receiving operations and in multi-point transmitting operations, contact between stations must be initiated by the control station in multi-point receiving operations. To be ready to receive the initialization sequence, the system executes a start I/O receive-initial instruction. For this operation the current-address register must be loaded with the address where the first character of the initialization sequence will be stored. The transition-address register must be loaded with the station address of the system in the low-order byte and any character in the high-order byte. The stop-address register must be loaded with the contents of the current-address register plus one less than the number of characters in the polling/selection sequence. The polling/selection sequence contains two identical characters as a station address which can be followed by other identifying characters as necessary. For a station address, the 2 bit in the EBCDIC and the 6 bit in the USASCII are ignored when comparing the transition-address register to the addressing character from the line. Thus, if the address character in the low-order byte of the transition-address register is B, both station addresses BB and SS will be recognized as the station address. This characteristic can be used to distinguish between polling sequences and selection sequences.

When the communication adapter recognizes its own station address, it stores a sequence of characters beginning with the second address character. The sequence is normally two address characters followed by ENQ. When the ENQ character is received, the current-address register equals the stop-address register, or if no character is received for three seconds after the communication adapter recognizes its own address, the communication adapter stops receiving and generates an interrupt request.

If the interrupt request is caused by too many characters in the polling/selection sequence or by failure to receive a character within three seconds, the interrupt routine must enter an error recovery routine to attempt to obtain a correct polling/selection sequence.

When the interrupt request is caused by receiving ENQ, the interrupt routine must determine if the sequence is polling or selection. If the system is being selected, the correct response is ACK 0 if the system is ready to receive; NAK or WACK is the response if the system is not ready to receive. This response is transmitted with a transmit-and-receive start I/O instruction. For this instruction the current-address register must be loaded with the address of the response character or characters, the transition-address register must be loaded with the next higher address after the response characters, and the stop-address register must be loaded with the contents of the transition-address register plus the number of characters expected in the message.

Message Reception

The receive portion of the transmit-and-receive operation that follows the initialization sequence is used to receive the first message or text block. The message can start with SOH followed by heading data or with STX followed by the text proper. If the message contains a heading, the heading must be ended by STX. Long text may be broken up by ITB or ETB characters. The end of the text of any message is signaled by ETX. When any of the text breaking or ending characters is received, the communication adapter compares the block check characters that follow such a character to the block check characters the adapter has accumulated from the bits received from the line. Then the communication adapter generates an interrupt request.

The interrupt routine must first check for the cause of the interruption. If the interrupt is caused by an ITB character, the adapter has stored the address of the location in storage next higher from the location where the ITB character is stored and is continuing to store characters in storage as they are received from the line. This enables the system to store the transition-address register in order to remember where the first character of the next transmission block is stored.

If the interruption is caused by ETB or ETX, the interrupt routine must frame a reply depending upon whether the text was received correctly. This reply is then transmitted to the remote terminal with a transmit-and-receive start I/O instruction.

When the remote terminal has transmitted its last message, it responds EOT to the system's positive acknowledgement character response to the remote terminal ETX. This signals the system that the control station is finished and that another receive-initial instruction should be issued to the communication adapter to enable it to monitor the line for another polling/selection sequence.

Data Checking

As the remote station transmits messages, it accumulates block check characters from the count of the bits transmitted. As these bits are received at the system communication adapter, the communication adapter accumulates block check characters from the bits it receives. Each time the remote station transmits an ITB, ETB, or ETX character, it also transmits its block check characters. The communication adapter compares these block check characters that it receives from the line with the block check characters that it has accumulated from the data it has received from the line. If the block check characters accumulated by the communication adapter do not match the block check characters received from the line, a status bit is set. During servicing of the interrupt resulting from an ITB, ETB, or ETX character, the program must sample the status bits and determine if the block check is correct.

If the interruption is a result of an ETB or ETX character, the result of the block check is transmitted immediately. The positive acknowledgement characters are alternated: ACK 0 being transmitted in response to even-numbered blocks and ACK 1 being transmitted in response to odd-numbered blocks. The program is responsible for transmitting the correct positive acknowledgement. The first block of text transmitted is always considered an odd-numbered block. If the wrong acknowledgement character is returned, the remote station assumes that a block of data or control characters was missed and initiates an error recovery procedure.

When block checking is initiated by ITB, the result of the block check is not transmitted immediately. Instead, if the block check is satisfactory, the communication adapter continues to receive and store characters. If the block check is incorrect, no more data is stored, no more ITB interruptions are generated, and the block check status bit is set to indicate that a block check error occurred. When the next ETB or ETX interruption is generated, the status bits are checked to determine if all data was received correctly.

Transparent Operation

Transparent operation allows transmitting and receiving any of the byte structures possible in EBCDIC. Transparent operation is entered by preceding the block of text for which transparent operation is desired with the XSTX control sequence.

In transmitting transparent data, the communication adapter inserts a DLE character preceding each DLE character that is obtained from storage. This DLE is stripped by the receiving adapter and does not enter storage.

Any character that is to end a block of text (ITB, ETB, ETX, or ENQ) must be placed at a location one less than the transition-address register points to. When the transmitting communication adapter detects a block delimiting character followed by the transition address, it inserts a DLE character ahead of the block delimiter to signal that the delimiter is a control character. The receiving communication adapter discards the DLE after using it to identify the control sequence.

The requirement that the block delimiting character be located one position before the transition address is normal operation for ETB, ETX, and ENQ, but it requires some changes in ITB operation. Because the ITB character requires no response from the receiving station, the normal transmit-and-receive operation is not used. Instead, the transition-address register is set equal to the stop-address register for the transmit-and-receive start I/O instruction. This results in an instruction that causes a transmit-only operation. Unlike the non-transparent mode of operation, though, it requires a new transmit-only operation for each transparent data block ending with ITB.

Limited Conversational Operation

In addition to the responses detailed in the sections on transmitting and receiving data, the program may accept a message from the remote terminal as a valid positive acknowledgement of a transmission. As the receiving station, the system may send a message as a valid positive acknowledgement of a message. In effect, the stations converse with each other. The conversation is limited somewhat, though, by a requirement that no conversational reply message can be answered with a conversational reply without at least one intervening positive-acknowledgement-character acknowledgement. A message used as a positive acknowledgement is counted as an odd or even acknowledgement in the same manner as any positive acknowledgement character. Figure 11-6 illustrates a limited conversational sequence.

Receive-Only Operation

The receive-only operation is defined for those times when no response character should precede the receive operation. Such an operation might be the attempt to complete reception of a message when a receive time-out occurs. The receive-only operation could also be used to eliminate contention in a point-to-point network when both stations want to transmit at the same time. The secondary station could perform a receive-only instruction after its receive time-out in order to listen for the primary station.

For the receive-only operation the local storage registers are loaded as follows. The current-address register is loaded with the address of the storage location where the first received character is to be stored. The stop-address register is loaded with the contents of the current-address register plus the number of characters expected in the message. For recovery from a receive time-out situation, the local storage registers may not require loading because the time-out should have left them pointing at the correct storage locations.

The receive operation begins when the start I/O receive instruction is executed. Characters received from the line are stored in main storage at the location specified by the current-address register. The operation ends and an interrupt request is generated when an ITB, ETB, ENQ, or ETX character is received, when the current-address register is equal to the stop-address register, or when no character is received from the line for three seconds. The actions and responses required in servicing a receive-only operation interruption are the same as those specified for receiving data during a transmit-and-receive operation.

Disconnect Operation

The communication adapter can be disconnected from the line and operations discontinued by issuing a disable BSCA start I/O instruction. The remote station should be notified of the system's intention to disconnect by a transmit-and-receive instruction transmitting the DISC control sequence. The communication adapter is automatically disconnected if no heading, text, response, or control characters are transmitted or received for 20 seconds.

Time must be allowed, especially in switched networks, for the disconnection to be completed before a new enable BSCA start I/O instruction is issued. The two-second time-out function can be used for this purpose.

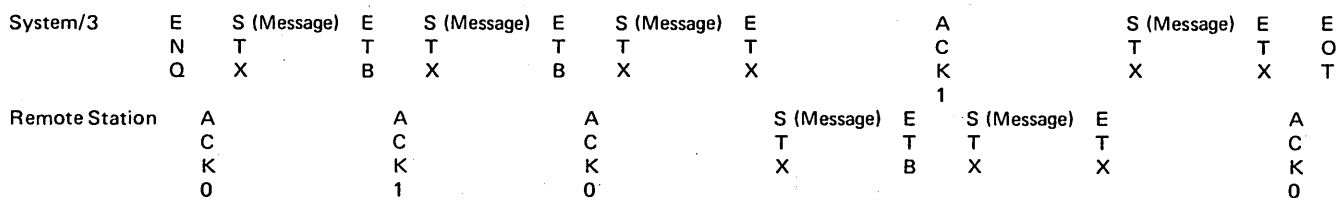


Figure 11-6. Point-to-Point Conversational Exchange

Start I/O

Mnemonic: SIO

Op Code	Q Byte	Control Code
F3	8 0 N	

Operation: The start I/O instruction is used to initiate all communication adapter operations. If the communication adapter is busy or is not ready for any reason except for unit check, the program will loop on the start I/O instruction until the communication adapter becomes not busy or is made ready. If the start I/O instruction is issued when the communication adapter is in the not ready condition, the BSCA I/O attention light on the keyboard-console will light. Correcting the not ready condition causes the instruction to be executed.

The Q byte specifies the communication adapter as the I/O unit that is to operate and specifies the function to be performed. Bits 0 through 3 of the Q byte are the device address for the BSCA. This is always 8. Bit 4 is a modifier bit that is always 0 for the communication adapter.

Bits 5 through 7 of the Q byte are called the N code. The N code specifies the operation to be performed as follows:

Bits *5 6 7*	*Operation*
0 0 0	Control
0 0 1	Receive only
0 1 0	Transmit and receive
0 1 1	Receive initial
1 0 0	Auto call
1 0 1	Invalid
1 1 0	Loop test
1 1 1	Invalid

Specification of an invalid N code causes the processing unit to stop with the processor-check and invalid-Q indicators on.

The third byte of the instruction is a control code. It is used to cause communication adapter control functions as follows:

Control Code			*Function*
Bit 7 = 1			Reset interrupt request.
Bit 7 = 0			None.
Bit 6 = 1			Enable interrupt.
Bit 6 = 0			Disable interrupt.
Bit 5 = 1			Start two-second time-out.
Bit 5 = 0			Cancel two-second time-out.
Bit 4			Reserved.
Bit 0 = 0			None.
Bit 0 = 1	AND	Bit 3 = 1	Enable step mode.
		Bit 3 = 0	Disable step mode.
		Bit 2 = 1	Enable test mode.
		Bit 2 = 0	Disable test mode.
		Bit 1 = 1	Enable BSCA.
		Bit 1 = 0	Disable BSCA.

Control Function: The N code that specifies the control function provides only the functions specified by the control code. This is the only instruction that can initiate the two-second time-out function.

Receive-Only Function: This operation accepts characters from the line and places them in storage at the location designated by the current-address register. The current-address register is updated plus one each time a character is stored. Receive-only operation ends when a change of direction character is received from the line, when the current-address register equals the stop-address register, or when no character is received from the line for three seconds.

Any of the control functions except start or cancel two-second time-out functions can be initiated by this instruction. In a non-switched network, the enable function must have been issued at least two seconds before the receive instruction.

Transmit-and-Receive Function: This function takes characters from storage at the location designated by the current-address register and transmits them on the line to the remote station. The current-address register is updated plus one for each character transmitted. The last character to be transmitted must be a change of direction character and must be at an address one less than that contained in the transition-address register.

When the current-address register has been updated to equal the transition-address register, the communication adapter stops transmitting and begins receiving characters from the line and storing the characters in main storage at the location specified by the current-address register. The current-address register is updated plus one for each character stored.

The operation ends and an interrupt request is generated when a change of direction character is received, the current-address register equals the stop-address register, or no character is received for three seconds.

Any of the control functions except start or cancel two-second time-out can be initiated by this instruction. In a non-switched network, the enable BSCA should be performed at least two seconds before the transmit-and-receive instruction is issued.

The transmit-and-receive instruction can be used as a transmit-only instruction by setting the transition-address register equal to the stop-address register. The transmit-and-receive instruction with a zero length transmit field (initial value of the current-address register and the transition-address register the same) is not allowed. At least one character must be used in the transmit field to provide the change of direction in the line for receiving.

The transmit-and-receive function is provided so that short line-turnaround times can be accommodated. The transmit-and-receive instruction should be used in all transmit initial-contact sequences because the remote station can respond too quickly to be heard if an interrupt routine must be processed before the receive operation begins.

Receive-Initial Function: This instruction is used as a standby to allow the remote station to establish contact in order to transmit a message. This function is the only one that can be used for establishing contact in multi-point networks. In this operation the communication adapter monitors the line until an initialization sequence is begun. When the communication adapter receives the initialization sequence, it stores the characters received in the location specified by the current-address register. The current-address register is updated plus one for each character stored. The operation ends and an interrupt request is generated when a change of direction character is received, the current-address register equals the stop-address register, or no character is received for three seconds after the initialization sequence is begun. Any of the control functions except start or cancel two-second time-out can be combined with this instruction. In a non-switched network the enable BSCA function should precede the receive-initial instruction by at least two seconds.

Auto Call: This function is provided as an optional feature in the communication adapter. In operation, the communication adapter takes the number to be called, one byte at a time, from storage locations specified by the current-address register. These numbers are sent to an auto calling unit that dials the number of the remote station. The current-address register is updated plus one for each byte transferred. When the current-address register equals the stop-address register, the communication adapter stops sending bytes to the auto calling unit and waits for an indication of the line connection's having been established or of the call's having been cancelled. If the connection is established, the adapter is signaled to end the operation. If the call is aborted, the time-out status bit is set and the operation is ended. At the end of the operation an interrupt request is generated. Any of the control functions except start or cancel two-second time-out or enable BSCA can be combined with this operation.

Loop Test Function: The loop test function is used by the CE to test the functioning of the communication adapter. It is of no use to a problem programmer.

Reset Interrupt Request, Enable Interrupt, and Disable Interrupt Control Functions: These control functions are used to allow or disallow the communication adapter to interrupt the main program stream and to end the interruption once it has occurred. The communication adapter operates on interrupt level 2. Two kinds of interruptions can occur from the communication adapter: an ITB interruption and an operation-end interruption. The interruption routine must determine with the test-I/O-and-branch instruction which kind of interruption occurred.

The ITB interruption occurs only when an ITB character is received. The interruption will not occur unless the block check characters indicate that everything to that point was received correctly. When the ITB interrupt occurs, the transition-address register must be stored to indicate the point at which the new data begins in storage. All the data up to but not including this address is the data that is to be processed. The status bytes are not sensed during an ITB interrupt because the bits in the status bytes apply to the data being received rather than the data that has been received (for ITB operation only).

Op-end interruptions occur at the end of all the functions controlled by the N code. In addition, the two-second time-out causes an interruption two seconds after the start I/O two-second time-out instruction is executed. Op-end interruption routines normally sense the status bytes to determine the status of the last transmission. The status bytes are valid for op-end interrupts because no data is transferred between the interrupt request and the operation of the interrupt routine.

Certain timing considerations must be considered when handling communications adapter interruptions. Because the communication adapter continues to receive data from the remote station during ITB interrupt routine servicing, the interrupt routine must be timed so that all processing is completed and the interrupt routine completed before the next ITB character is received. The processing time available is a function of the data rate of the modem used. Extra time must be allowed in the interrupt routine to account for any time that may be required by cycle steals from the data being received and from other I/O units.

Executing a start I/O disable interrupt (control code bit 6 = 0) when an interruption routine is in progress resets the interrupt request and ends the interruption routine.

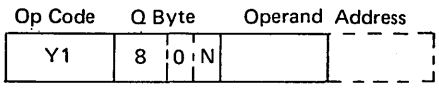
Start and Cancel Two-Second Time-Out Functions: These two control functions are the only ones that cannot be combined with other functions of the start I/O instruction. The start function causes the communication adapter to measure two seconds of time then cause an op-end interruption. The cancel function causes the two-second time-out to be ended when the start I/O instruction is executed.

Enable–Disable Step and Test Mode Functions: These are diagnostic functions useful to the customer engineer but of no interest to the problem programmer.

Enable–Disable BSCA Control Functions: The enable BSCA function causes the communication adapter to become operable and allows it to connect to the modem and perform data handling functions. The disable BSCA function deconditions the adapter and disconnects it from the data set.

Load I/O

Mnemonic: LIO



Operation: The contents of the two-byte field addressed by the operand address are placed in the register specified by the Q byte. The operand is addressed by its rightmost byte.

The Q byte contains a device address (always 8 for the communication adapter) in the high-order four bits, an M bit of 0, and an N code. The N code specifies the register to be loaded as follows:

N Bits	*Register*
5 6 7	
0 0 0	Invalid
0 0 1	Stop-address register
0 1 0	Transition-address register
0 1 1	Invalid
1 0 0	Current-address register
1 0 1	Invalid
1 1 0	Current-address buffer (diagnostic function)
1 1 1	Invalid

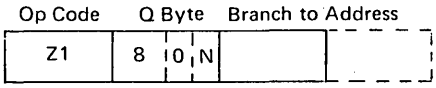
Specification of an invalid N code or an M bit of 1 results in processor check stop with an invalid-Q-byte indication.

The current-address buffer specified by N code 110 is not a true register but a diagnostic function that is used by the CE to repeat a character n times for testing purposes.

If a load I/O instruction is issued when the communication adapter is busy, the processing unit loops on the load I/O instruction until the busy condition no longer exists.

Test I/O and Branch

Mnemonic: TIO



Operation: The conditions specified by the Q byte are tested. If the specified condition exists, the next instruction is taken from the address specified by the branch-to address and the next sequential instruction address is placed in the address recall register. If the condition specified does not exist, the next sequential instruction is taken and the branch-to address is placed in the address recall register. The address recall register will not be changed until the next decimal, insert-and-test-characters, or branch instruction is executed.

The Q byte contains a device address (always 8 for the communication adapter) in the high-order four bits, an M bit of 0, and an N code. The N code specifies the condition to be tested as follows:

N Bits	*Condition Tested*
5 6 7	
0 0 0	Not ready/unit check
0 0 1	Op-end interrupt
0 1 0	Busy
0 1 1	ITB interrupt
1 0 0	Interrupt pending
1 0 1	Invalid
1 1 0	New data
1 1 1	Invalid

Specification of an invalid N code or an M bit of 1 results in a processor-check stop with an invalid-Q-byte indication.

The current-address buffer specified by N code 110 is not a true register but a diagnostic function that is used by the CE to repeat a character n times for testing purposes.

If a load I/O instruction is issued when the communication adapter is busy, the processing unit loops on the load I/O instruction until the busy condition no longer exists.

1. In a point-to-point non-switched network the adapter becomes busy as soon as the adapter is synchronized with a remote station.
2. In a point-to-point switched network the adapter becomes busy as soon as the modem indicates that it has received a call.
3. In a multi-point network the adapter becomes busy when it recognizes its own address in control mode.

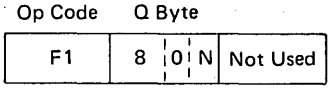
Unit check means that one of the status bits in status byte 2 is on.

New data is a diagnostic condition of no interest to the problem programmer.

Program Note: Testing op-end interrupt or ITB interrupt with the test I/O and branch instruction resets the interruption identifier bits in the communication adapter.

Advance Program Level

Mnemonic: APL



Operation: The conditions specified by the Q byte are tested. If the specified condition exists, the program loops on the advance program level instruction until the specified condition no longer exists. If the specified condition does not exist, the next sequential instruction is executed.

The Q byte contains a device address (always 8 for the communication adapter) in the high-order four bits, an M bit of 0, and an N code. The N code specifies the condition to be tested as follows:

N Bits	*Condition Tested*
5 6 7	
0 0 0	Not ready/unit check
0 0 1	Op-end interrupt
0 1 0	Busy
0 1 1	ITB interrupt
1 0 0	Interrupt pending
1 0 1	Invalid
1 1 0	New data
1 1 1	Invalid

Specification of an invalid N code or an M bit of 1 causes a processor-check stop with an invalid-Q-byte indication.

New data is a diagnostic condition that occurs only in test mode and is of no interest to the problem programmer.

The communication adapter becomes busy under different conditions, depending upon the kind of operation that is being performed. For all operations except receive initial, the adapter becomes busy as soon as the start I/O instruction is accepted. It remains busy until the operation ends. For receive initial operations, the following conditions cause busy.

1. In a point-to-point non-switched network the adapter becomes busy as soon as the adapter is synchronized with the remote station.
2. In a point-to-point switched network the adapter becomes busy as soon as the modem indicates that it has received a call.
3. In a multi-point network the adapter becomes busy when it recognizes its own address in control mode.

Unit check means that one of the bits in status byte 2 is on.

Program Notes:

1. Testing op-end interrupt or ITB interrupt with the advance program level instruction resets the interruption identifier bits in the communication adapter.
2. The advance program level instruction should not be used to test for not ready/check or new data. If either of the conditions exists when the instruction is executed, the system loops on the advance program level instruction with no indication to the operator that no data is being processed.

Sense I/O

Mnemonic: SNS

Op Code	Q Byte			Operand	Address
Y0	8	0	N		

Operation: The contents of the register specified by the Q byte are stored in the two-byte field addressed by the operand address. The operand is addressed by the rightmost byte.

The Q byte contains a device address (always 8 for the communication adapter) in the high-order four bits, an M bit of 0, and an N code. The N code specifies the register to be stored as follows:

N Bits	*Register*
5 6 7	
0 0 0	Diagnostic
0 0 1	Stop-address register
0 1 0	Transition-address register
0 1 1	Status bytes
1 0 0	Current-address register
1 0 1	Invalid
1 1 0	CRC/LRC buffer
1 1 1	Invalid

Specification of an invalid N code or an M bit of 1 causes the processor-check stop with an invalid-Q-byte indication.

The diagnostic and CRC/LRC buffer functions are used by the customer engineer for servicing the adapter. They are of no interest to the problem programmer.

The status bytes are bit significant as illustrated in Figure 11-7. Byte 1 is stored in the storage location addressed by the operand address; byte 2 is stored in the next lower storage location.

The time-out bit is turned on by two conditions:

1. No data is received during a receiving operation for three seconds.
2. An auto call operation is terminated by an abandon-call-and-retry signal from the auto calling unit. This indicates that the call was not answered.

The time-out bit is reset off by any non-control start I/O instruction.

The CRC/LRC/VRC status bit is set on during a receive operation with the adapter in the busy state and synchronized when:

1. The block check character received from the line does not match the block check character generated by the adapter.
2. A USASCII character is received with incorrect vertical redundancy. USASCII characters with VRC checks are stored with a 1 bit in the high-order bit position of the byte. Characters with a VRC check are never recognized as control characters by the adapter.

The CRC/LRC/VRC bit is turned off by any non-control start I/O instruction.

The adapter-check-on-transmit bit is turned on during a communication adapter transmit operation by incorrect parity detected in certain of the adapter circuits or by the processing unit's allowing a cycle steal before the preceding character is transmitted by the adapter. The bit is turned off by any non-control start I/O instruction. Adapter-check-on-transmit causes operation to end and causes an immediate op-end interrupt.

The adapter-check-on-receive bit turns on during a communication adapter receive operation if incorrect parity is detected in the adapter circuits or if a cycle steal is not allowed in time to store a character before it is destroyed by an incoming character. Adapter-check-on-receive does not end the receive operation. The bit is turned off by any non-control start I/O instruction.

The invalid-USASCII character bit is turned on if an adapter with USASCII code detects a 1 bit in the high-order position of a byte obtained from storage during a transmit operation. The bit is turned off by any non-control start I/O instruction.

The abortive-disconnect bit is turned on in a switched network if the data-set-ready line comes on then goes off when the BSCA is enabled. Turning on and off the data-set-ready line indicates the release of the connection and causes the data terminal ready line to turn off. The abortive-disconnect bit must be reset by issuing a disable BSCA start I/O instruction. Sufficient time must be allowed by the program for a forced disconnect to occur before the communication adapter is enabled for a retry operation. The two-second time-out function can be used to assure this.

The disconnect time-out bit turns on if no heading, text, response, or control transmission occurs from either station for 20 seconds. This bit is reset off by a start I/O disable BSCA instruction.

The data-set-ready condition bit is set on when the data set ready signal is detected and latched on. The bit is turned off if data set ready comes on and turns off when the communication adapter is enabled or the BSCA is disabled.

The data-line-occupied bit turns on when the auto calling unit signals that the data line is occupied. When this bit is on, a start I/O auto call instruction or start I/O receive-initial instruction will loop on itself until the line is unoccupied. No start I/O auto call or receive-initial instruction should be issued in an interrupt routine when this bit is on.

Bit	Byte 2 High Order	Byte 1 Low Order
0	Timeout	Not Assigned
1	CRC/LRC/VRC	Not Assigned
2	Adapter Check on Transmit	Not Assigned
3	Adapter Check on Receive	Not Assigned
4	Invalid USASCII Character	Not Assigned
5	Abortive Disconnect	Not Assigned
6	Disconnect Timeout	Data Set Ready Condition
7	Not Assigned	Data Line Occupied

Figure 11-7. Status Bit Significance

SUGGESTED ERROR RECOVERY PROCEDURES

At the end of every transmit and/or receive operation a test I/O for unit check should be performed. If unit check is detected, the sense I/O for status bytes should be performed. Testing of the status bits and the procedures for recovering from the errors should be performed in the order given.

Invalid USASCII Character, Byte 2 Bit 4

This bit need not be tested if the adapter is not operating with USASCII code. If the bit is on, the operation must be restarted by operator intervention.

Lost Connection or Disconnect Time-Out, Byte 2 Bits 5 and 6

When either of these bits is on, the operation must be restarted by the operator.

Adapter Check on Transmit or Receive, Byte 2 Bits 2 and 3

When either of these bits is on, the operation must be restarted by the operator.

Time-Out, Byte 2 Bit 0

Recovery procedure when this bit is on depends on the mode of operation of the adapter at the time.

1. If the adapter is in control mode (attempting to establish communications or listening for the other station to call) or performing an auto call operation, it should retry the operation at least seven times. If no correct response is received, the operator must intervene to establish communications.
2. If the adapter is operating as a slave (receiving a message), the program must issue the receive portion of the previous operation at least 7 times. If the message cannot be received completely, the operator must intervene to obtain correct operation.
3. If the adapter is operating as the master (transmitting the message), the program must execute a start I/O transmit-and-receive instruction transmitting the ENQ character and attempting to receive a response. If no response is received during at least seven retries, the operator must intervene to obtain the correct operation.

CRC/LRC/VRC, Byte 2 Bit 1

Recovery procedures when this bit is on depend upon the mode of operation of the adapter at the time.

1. If the adapter is in the control mode, the program should retry the last operation at least seven times. If no correct response can be obtained, the operator must intervene to obtain the correct operation.
2. If the adapter is operating as the slave, the program should issue a transmit-and-receive start I/O instruction transmitting the NAK character and receiving the last block of data again. If the data cannot be received correctly in at least seven retries, the operator must intervene to obtain correct operation.
3. If the adapter is operating as the master, the program must issue a transmit-and-receive start I/O instruction transmitting the ENQ character and receiving a response character. If a response character without an error cannot be received, the operator must intervene to obtain the correct operation.

Program-Detected Errors

After the program has tested for adapter-detected errors, it must check for correct response characters. The correct response characters and the error recovery procedure, if the response is incorrect, depend upon the mode of operation of the adapter.

1. When the adapter is the slave, the abnormal response is an absence of an initial STX character to any block of text ended by an ETB or ETX. Another abnormal response is the absence of an ETB or ETX or ENQ character at the end of a block of text signaled by an op-end interrupt. This can occur if the current-address register reaches the value of the stop-address register before the ETX or ETB or ENQ character is received. The program must issue the receive portion of the last operation at least seven times. If no correct operation can be accomplished after seven retries, the operator must intervene to correct the error.
2. When the adapter is operating as the master, and the recovery procedure for the time-out bit is performed with a response character received, and the response character is not the correct ACK sequence (ACK 0 or ACK 1), perform the following:
   a. Transmit and receive, transmitting the last text sent and listening for a response.
   b. If the response is not received before time-out occurs, the time-out procedure is repeated and the response character checked.
   c. Continued incorrect response characters after time-out result in actions a and b until the number of retries specified for the time-out recovery attempt has been reached. The error then requires operator intervention.
3. If the adapter is operating as the master and any response other than the correct ACK sequence or the EOT character is received:
   a. Transmit and receive the ENQ character and the response once. If the invalid response recurs, the operator must intervene. If the response is NAK, do step b.
   b. Transmit and receive the last text and response at least seven times. If the correct response is not received in seven retries, the operator must intervene.

INSTRUCTION REJECT AND ATTENTION CONDITIONS

Certain conditions cause the attention lights on the operator console to light and/or instructions to be rejected. Rejecting an instruction causes the system to loop on the rejected instruction until the instruction reject condition is no longer present.

DBO Parity Check

This condition affects all communication adapter instructions. It causes the processing unit to stop with a processor-check stop indication and a DBO check indication.

Invalid N Code

This condition affects all communication adapter instructions. It causes the processor-check stop with an invalid-Q indication.

Data Set Ready Latch Off

This condition affects the receive, transmit-and-receive, and receive-initial operations for non-switched point-to-point and multi-point networks. The condition can be tested with a test-I/O-and-branch-on-not-ready instruction. This condition sets status byte 1 bit 6 on. If one of the affected instructions is given with this condition present, the instruction is rejected, the I/O attention indicator on the keyboard console is turned on, and the BSCA attention indicator is turned on.

ACU Power Off or Data Line Occupied On

These conditions affect auto-call and receive-initial instructions in switched networks. The conditions can be tested by the test-I/O-and-branch-on-not-ready instruction. Data line occupied also turns on status byte 1 bit 7. If one of the affected instructions is given with this condition present, the instruction is rejected and the BSCA I/O attention indicator on the keyboard-console is turned on.

Busy

This condition affects all load I/O instructions except current-address buffer and all start I/O instructions except control. This condition can be tested with the test-I/O-and-branch-on-busy instruction. If an affected instruction is issued with this condition present, the instruction is rejected.

BSCA Disabled or External Test Switch On and Test Mode Disabled

These conditions affect all start I/O instructions except control. The conditions can be tested with the test-I/O-and-branch-not-ready instruction. If the affected instructions are issued with this condition present, the instruction is rejected and the BSCA I/O attention indicator on the keyboard-console is turned on.

Data Set Ready Latch On and Data Set Ready Off

This condition does not reject any instructions. It turns on the BSCA I/O attention indicator on the keyboard-console.

BSCA OPERATOR'S PANEL (Figure 11-8)

This panel is installed only when the binary synchronous communication adapter special feature is installed. It is installed at the front of the left side of the processing unit and is visible to the operator through a cut-out in the cover.

BSCA ATTN

This light (on the keyboard-console) turns on when I/O attention is caused by a condition in the communication adapter. The indicator on the BSCA panel is nonfunctional.

Unit Check

This light turns on when any bit in the status bytes is on.

DT TERM Ready

This light indicates that the modem is ready for use.

Clear to Send

This light indicates that the clear-to-send line from the modem is on and that the adapter may now transmit.

Receive Trigger

This light indicates the status of the receive trigger. The light is on when the trigger is at a binary zero state.

TSM Trigger

This light indicates the status of the transmit trigger. The light is on when the trigger is at a binary zero state.

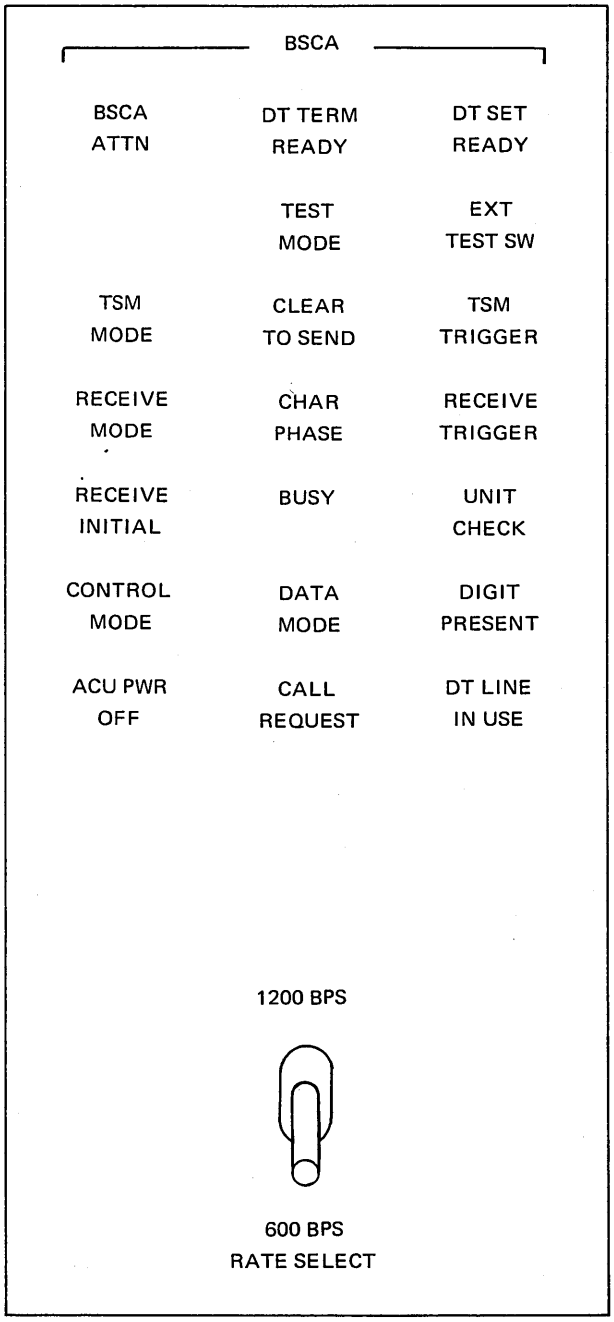


Figure 11-8. BSCA Operator's Panel

Receive Mode

This light indicates that the adapter has been instructed to perform a receive operation.

TSM Mode

This light indicates that the adapter has been instructed to perform a transmit operation.

Receive Initial

This light is turned on by a receive-initial instruction. It is turned off at the end of the receive-initial operation.

Busy

This light indicates that the communication adapter is executing a receive-initial, transmit-and-receive, auto call, receive, or loop test instruction.

CHAR Phase

This light indicates that the adapter has established character sync with the transmitting station by receiving two successive SYN characters. The light is turned off at the end of the receive operation.

Data Mode

This light is turned on by the decode of an SOH or STX during a transmit or receive operation. It is turned off at the end of the transmit or receive operation.

Digit Present

This light indicates that a digit has been obtained from storage when the auto call feature is installed.

Control Mode

This light is turned on by an EOT sequence during a transmit, receive, or a receive-initial monitor operation when the station select feature is installed. It is turned off by decode of an SOH or STX.

ACU PWR Off

This light indicates that the auto calling unit has power off when the auto call feature is installed.

Call Request

This light indicates that the communication adapter has received an auto call instruction and is performing an auto call operation when the auto call feature is installed.

DT Line in Use

This light indicates that the data-line-occupied line from the auto calling unit is on.

Test Mode

This light indicates that the program has placed the adapter in a test mode of operation.

EXT Test SW

This light indicates that the switch at the modem end of the low-speed modem cable is in the test position. For high-speed modems this indicator will be active when the local loop switch is in the test position.

DT Set Ready

This light indicates that the data-set-ready line from the modem is on and the modem is ready for use.

Rate Select Switch

This switch is present only on the World Trade Corporation systems with the rate selection optional feature installed.

LCA OPERATOR'S PANEL (Figure 11-9)

This panel is installed only when the local communications adapter special feature is installed. It is installed at the front of the left side of the processing unit and is visible to the operator through a cut-out in the cover.

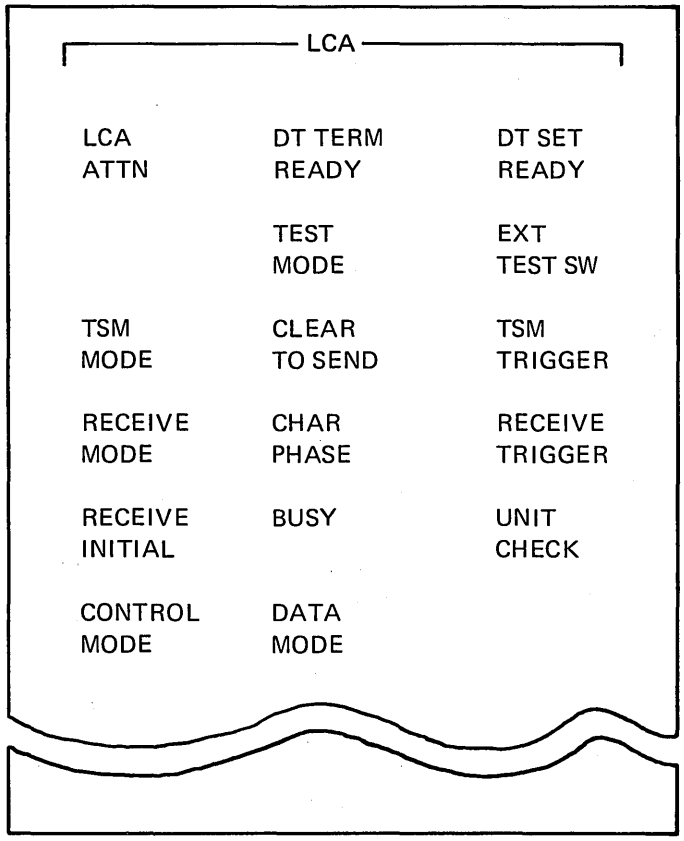


Figure 11-9. LCA Operator's Panel

LCA ATTN

This light (on the keyboard-console) turns on when I/O attention is caused by a condition in the communication adapter. The indicator on the LCA panel is nonfunctional.

Unit Check

This light turns on when any bit in the status bytes is on.

DT TERM Ready

This light indicates that the LCA is ready for use.

Clear to Send

This light indicates that the clear-to-send line from the LCA is on and that the adapter may now transmit.

Receive Trigger

This light indicates the status of the receive trigger. The light is on when the trigger is at a binary zero state.

TSM Trigger

This light indicates the status of the transmit trigger. The light is on when the trigger is at a binary zero state.

Receive Mode

This light indicates that the adapter has been instructed to perform a receive operation.

TSM Mode

This light indicates that the adapter has been instructed to perform a transmit operation.

Receive Initial

This light is turned on by a receive-initial instruction. It is turned off at the end of the receive-initial operation.

Busy

This light indicates that the communication adapter is executing a receive-initial, transmit-and-receive, auto call, receive, or loop test instruction.

CHAR Phase

This light indicates that the adapter has established character sync with the transmitting station by receiving two successive SYN characters. The light is turned off at the end of the receive operation.

Data Mode

This light is turned on by the decode of an SOH or STX during a transmit or receive operation. It is turned off at the end of the transmit or receive operation.

Control Mode

This light is turned on by an EOT sequence during a transmit, receive, or a receive-initial monitor operation. It is turned off by decode of an SOH or STX.

Test Mode

This light indicates that the program has placed the adapter in a test mode of operation.

EXT Test SW

This light indicates that the switch at the LCA end of the cable is in the test position.

DT Set Ready

This light indicates that the device attached to the LCA feature is ready for operation.

BSCA/LCA CE Controls

The following two switches are located on the CE panel.

BSCA Step Key

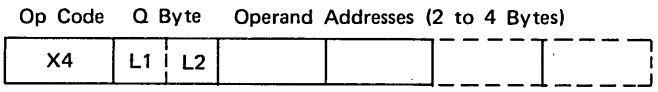
This key is effective only when the BSCA or LCA is in test mode and step mode simultaneously. It causes the communication adapter to advance one bit time for each key depression.

Local Loop Test Switch

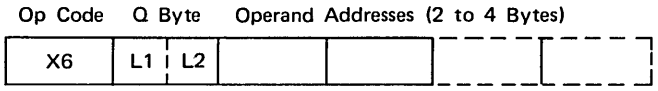
This toggle switch sets the high-speed modem into local test mode and causes data to be wrapped around through the modem with the start I/O loop test instruction. (This switch is not used with the LCA feature.)

Appendix A. Instruction Formats

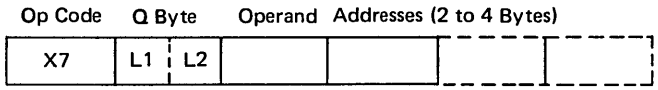
Zero and Add (ZAZ)



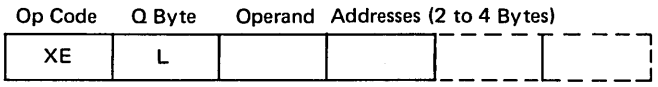
Add Zoned Decimal (AZ)



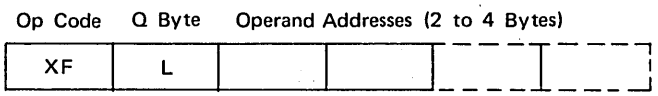
Subtract Zoned Decimal (SZ)



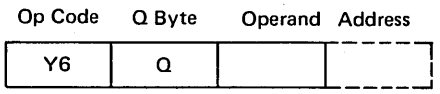
Add Logical Characters (ALC)



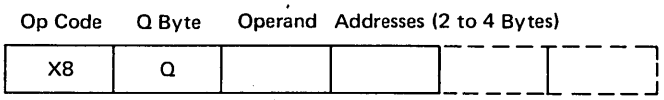
Subtract Logical Characters (SLC)



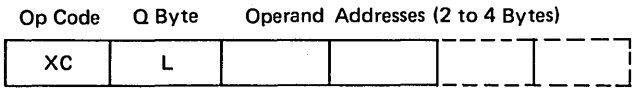
Add to Register (A)



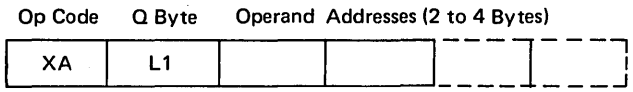
Move Hex Character (MVX)



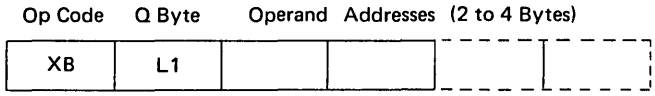
Move Characters (MVC)



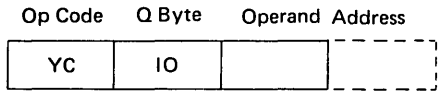
Edit (ED)



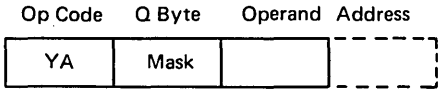
Insert and Test Characters (ITC)



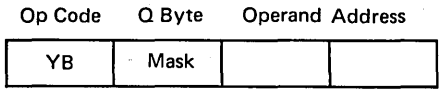
Move Logical Immediate (MVI)



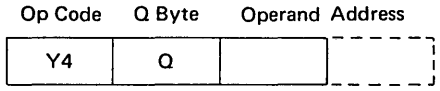
Set Bits On Masked (SBN)



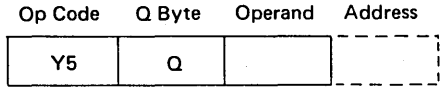
Set Bits Off Masked (SBF)



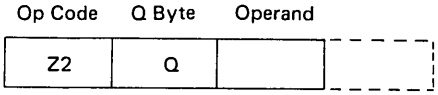
Store Register (ST)



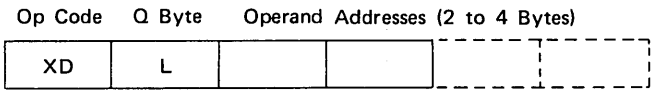
Load Register (L)



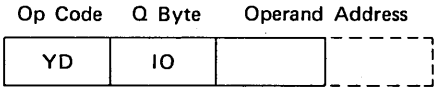
Load Address (LA)



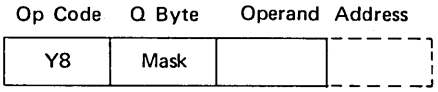
Compare Logical Characters (CLC)



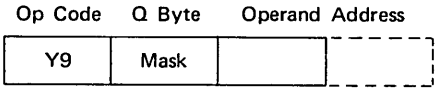
Compare Logical Immediate (CLI)



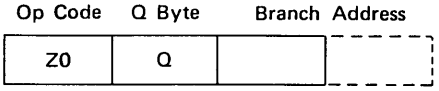
Test Bits On Masked (TBN)



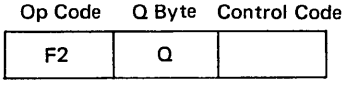
Test Bits Off Masked (TBF)



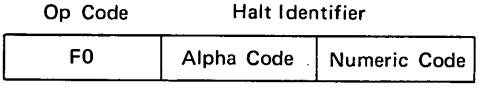
Branch On Condition (BC)



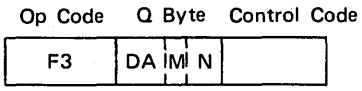
Jump On Condition (JC)



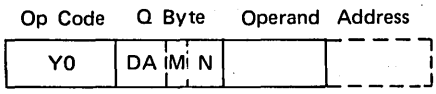
Halt Program Level (HPL)



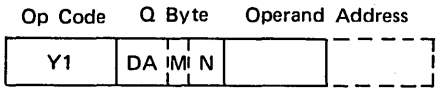
Start I/O (SIO)



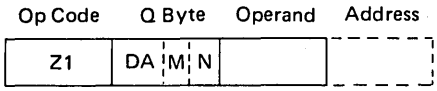
Sense I/O (SNS)



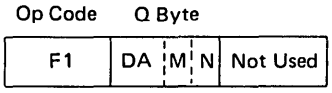
Load I/O (LIO)



Test I/O and Branch



Advance Program Level



Appendix B. System/3 Model 6 Codes

Hex Value	Binary Value	Card Code (Note 3)	Printer Graphic (Note 1)	Card Graphic (Note 3)	Keyboard Key (Note 2)	Display Station Graphic	Communication Adapter Character	Decimal Value	80-Column Code (Note 3)
00	00000000	No punches	Blank	Blank	T-0,A-0			0	No punches
01	00000001	BA 1	A	A	T-1,A-1, C-01,	A	SOH	1	12 1
02	00000010	BA 2	B	B	T-2,A-2, C-02, Enter–	B	STX	2	12 2
03	00000011	BA 21	C	C	T-3,A-3, C-03, Erase	C	ETX	3	12 3
04	00000100	BA 4	D	D	T-4,A-4, C-04	D		4	12 4
05	00000101	BA 4 1	E	E	T-5,A-5, C-05,Tab	E		5	12 5
06	00000110	BA 42	F	F	T-6,A-6, C-06	F		6	12 6
07	00000111	BA 421	G	G	T-7,A-7, C-07	G		7	12 7
08	00001000	BA8	H	H	T-8,A-8, C-08	H		8	12 8
09	00001001	BA8 1	I	I	T-9,A-9, C-09	I		9	12 9
0A	00001010	BA8 2	Ç	Ç	T-A,C-10	Ç		10	12 8 2
0B	00001011	BA8 21	.	.	T-B,C-11	.		11	12 8 3
0C	00001100	BA84	<	<	T-C,C-12	≤		12	12 8 4
0D	00001101	BA84 1	(	(	T-D,C-13	(		13	12 8 5
0E	00001110	BA842	+	+	T-E,C-14	+		14	12 8 6
0F	00001111	BA8421	|	|	T-F,C-15	|		15	12 8 7
10	00010000	A8 2	&	&	),C-16	&	DLE	16	12
11	00010001	B 1	J	J	↑,Inquiry Request	J		17	11 1
12	00010010	B 2	K	K	@	K		18	11 2
13	00010011	B 21	L	L	#	L		19	11 3
14	00010100	B 4	M	M	$	M		20	11 4
15	00010101	B 4 1	N	N	%,Return	N		21	11 5
16	00010110	B 42	O	O	Ç,Backspace	O		22	11 6
17	00010111	B 421	P	P	&	P		23	11 7
18	00011000	B 8	Q	Q	'	Q		24	11 8
19	00011001	B 8 1	R	R	(	R		25	11 9
1A	00011010	B 8 2	↑	!	G	↑		26	11 8 2

Hex Value	Binary Value	Card Code (Note 3)	Printer Graphic (Note 1)	Card Graphic (Note 3)	Keyboard Key (Note 2)	Display Station Graphic	Communication Adapter Character	Decimal Value	80-Column Code (Note 3)
1B	00011011	B 8 21	$	$	H	$		27	11 8 3
1C	00011100	B 84	*	*	I	*		28	11 8 4
1D	00011101	B 84 1	)	)	J	)		29	11 8 5
1E	00011110	B 842	;	;	K	;		30	11 8 6
1F	00011111	B 8421	¬	¬	L	¬	IUS	31	11 8 7
20	00100000	B	–	–	M	–		32	11
21	00100001	A 1	/	/	N	/		33	0 1
22	00100010	A 2	S	S	O	S		34	0 2
23	00100011	A 21	T	T	P	T		35	0 3
24	00100100	A 4	U	U	Q	U		36	0 4
25	00100101	A 4 1	V	V	R	V		37	0 5
26	00100110	A 42	W	W	S	W	ETB	38	0 6
27	00100111	A 421	X	X	T	X		39	0 7
28	00101000	A8	Y	Y	U	Y		40	0 8
29	00101001	A8 1	Z	Z	V	Z		41	0 9
2A	00101010	BA	(Note 4)	(Note 5)	W	_		42	11 0
2B	00101011	A8 21	,	,	X	,		43	0 8 3
2C	00101100	A84	%	%	Y	%		44	0 8 4
2D	00101101	A84 1	_	_	Z	_	ENQ	45	0 8 5
2E	00101110	A842	>	>	–	>		46	0 8 6
2F	00101111	A8421	?	?	=	?		47	0 8 7
30	00110000	A	0	0	+	0		48	0
31	00110001	1	1	1	N =	1		49	1
32	00110010	2	2	2	;	2	SYN	50	2
33	00110011	21	3	3	*	3		51	3
34	00110100	4	4	4	,	4		52	4
35	00110101	4 1	5	5	T =	5		53	5
36	00110110	42	6	6	/	6		54	6
37	00110111	421	7	7	?	7	EOT	55	7
38	00111000	8	8	8	|	8		56	8
39	00111001	8 1	9	9		9		57	9
3A	00111010	8 2	:	:	:	:		58	8 2
3B	00111011	8 21	#	#	≠	#		59	8 3

Hex Value	Binary Value	Card Code (Note 3)	Printer Graphic (Note 1)	Card Graphic (Note 3)	Keyboard Key (Note 2)	Display Station Graphic	Communication Adapter Character	Decimal Value	80-Column Code (Note 3)
3C	00111100	84	@	@	<	@		60	8 4
3D	00111101	84 1	'	'	>	'	NAK	61	8 5
3E	00111110	842	=	=	_	=		62	8 6
3F	00111111	8421	≠	"	¬	≠		63	8 7
40	01000000	No Punches	Blank	Blank	Space	Blank	Blank	64	No Punches
41	01000001	BA 1	A	A		A		65	12 1
42	01000010	BA 2	B	B		B		66	12 2
43	01000011	BA 21	C	C		C		67	12 3
44	01000100	BA 4	D	D		D		68	12 4
45	01000101	BA 4 1	E	E		E		69	12 5
46	01000110	BA 42	F	F		F		70	12 6
47	01000111	BA 421	G	G		G		71	12 7
48	01001000	BA8	H	H		H		72	12 8
49	01001001	BA8 1	I	I		I		73	12 9
4A	01001010	BA8 2	¢	¢		¢	¢	74	12 8 2
4B	01001011	BA8 21	.	.		.	.	75	12 8 3
4C	01001100	BA84	<	<		<	<	76	12 8 4
4D	01001101	BA84 1	(	(		(	(	77	12 8 5
4E	01001110	BA842	+	+		+	+	78	12 8 6
4F	01001111	BA8421	|	|		|	|	79	12 8 7
50	01010000	A8 2	&	&		&	&	80	12
51	01010001	B 1	J	J		J		81	11 1
52	01010010	B 2	K	K		K		82	11 2
53	01010011	B 21	L	L		L		83	11 3
54	01010100	B 4	M	M		M		84	11 4
55	01010101	B 4 1	N	N		N		85	11 5
56	01010110	B 42	O	O		O		86	11 6
57	01010111	B 421	P	P		P		87	11 7
58	01011000	B 8	Q	Q		Q		88	11 8
59	01011001	B 8 1	R	R		R		89	11 9
5A	01011010	B 8 2	↑	!		↑	↑	90	11 8 2
5B	01011011	B 8 21	$	$		$	$	91	11 8 3

Hex Value	Binary Value	Card Code (Note 3)	Printer Graphic (Note 1)	Card Graphic (Note 3)	Keyboard Key (Note 2)	Display Station Graphic	Communication Adapter Character	Decimal Value	80-Column Code (Note 3)
5C	01011100	B 84	*	*		*	*	92	11 8 4
5D	01011101	B 84 1	)	)		)	)	93	11 8 5
5E	01011110	B 842	;	;		;	;	94	11 8 6
5F	01011111	B 8421	¬	¬		¬	¬	95	11 8 7
60	01100000	B	-	-		-	–	96	11
61	01100001	A 1	/	/		/	/	97	0 1
62	01100010	A 2	S	S		S		98	0 2
63	01100011	A 21	T	T		T		99	0 3
64	01100100	A 4	U	U		U		100	0 4
65	01100101	A 4 1	V	V		V		101	0 5
66	01100110	A 42	W	W		W		102	0 6
67	01100111	A 421	X	X		X		103	0 7
68	01101000	A8	Y	Y		Y		104	0 8
69	01101001	A8 1	Z	Z		Z		105	0 9
6A	01101010	BA	(Note 4)	} (Note 5)		Blank		106	11 0
6B	01101011	A8 21	,	,		,	,	107	0 8 3
6C	01101100	A84	%	%		%	%	108	0 8 4
6D	01101101	A84 1	_	_		_	_	109	0 8 5
6E	01101110	A842	>	>		>	>	110	0 8 6
6F	01101111	A8421	?	?		?	?	111	0 8 7
70	01110000	A	0	0		0	(70)	112	0
71	01110001	1	1	1		1		113	1
72	01110010	2	2	2		2		114	2
73	01110011	21	3	3		3		115	3
74	01110100	4	4	4		4		116	4
75	01110101	4 1	5	5		5		117	5
76	01110110	42	6	6		6		118	6
77	01110111	421	7	7		7		119	7
78	01111000	8	8	8		8		120	8
79	01111001	8 1	9	9		9		121	9
7A	01111010	8 2	:	:		:	:	122	8 2
7B	01111011	8 21	#	#		#	#	123	8 3

Hex Value	Binary Value	Card Code (Note 3)	Printer Graphic (Note 1)	Card Graphic (Note 3)	Keyboard Key (Note 2)	Display Station Graphic	Communication Adapter Character	Decimal Value	80-Column Code (Note 3)
7C	01111100	(84)	(@)	(@)		@	@	124	(8 4)
7D	01111101	(84 1)	(')	(')		'	'	125	(8 5)
7E	01111110	(842)	(=)	(=)		=	=	126	(8 6)
7F	01111111	(8421)	(≠)	(")		≠	≠	127	(8 7)
80	10000000	No Punches	Blank	Blank		_		128	No Punches
81	10000001	BA 1	A	A	Program-Start	A		129	12 1
82	10000010	BA 2	B	B		B		130	12 2
83	10000011	BA 21	C	C		C		131	12 3
84	10000100	BA 4	D	D		D		132	12 4
85	10000101	BA 4 1	E	E		E		133	12 5
86	10000110	BA 42	F	F		F		134	12 6
87	10000111	BA 421	G	G		G		135	12 7
88	10001000	BA8	H	H		H		136	12 8
89	10001001	BA8 1	I	I		I		137	12 9
8A	10001010	BA8 2	Ç	Ç		Ç		138	12 8 2
8B	10001011	BA8 21	.	.		.		139	12 8 3
8C	10001100	BA84	<	<		<		140	12 8 4
8D	10001101	BA84 1	(	(		(		141	12 8 5
8E	10001110	BA842	+	+		+		142	12 8 6
8F	10001111	BA8421	|	|		|		143	12 8 7
90	10010000	BA	&	} (Note 5)		&		144	11 0
91	10010001	B 1	J	J	Enter+	J		145	11 1
92	10010010	B 2	K	K		K		146	11 2
93	10010011	B 21	L	L		L		147	11 3
94	10010100	B 4	M	M		M		148	11 4
95	10010101	B 4 1	N	N		N		149	11 5
96	10010110	B 42	O	O		O		150	11 6
97	10010111	B 421	P	P		P		151	11 7
98	10011000	B 8	Q	Q		Q		152	11 8
99	10011001	B 8 1	R	R		R		153	11 9
9A	10011010	B 8 2	↑	!		↑		154	11 8 2
9B	10011011	B 8 21	$	$		$		155	11 8 3

Hex Value	Binary Value	Card Code (Note 3)	Printer Graphic (Note 1)	Card Graphic (Note 3)	Keyboard Key (Note 2)	Display Station Graphic	Communication Adapter Character	Decimal Value	80-Column Code (Note 3)
9C	10011100	B 84	*	*		*		156	11 8 4
9D	10011101	B 84 1	)	)		)		157	11 8 5
9E	10011110	B 842	;	;		;		158	11 8 6
9F	10011111	B 8421	¬	¬		¬		159	11 8 7
A0	10100000	B	–	–		–		160	11
A1	10100001	A 1	/	/		/		161	0 1
A2	10100010	A 2	S	S		S		162	0 2
A3	10100011	A 21	T	T		T		163	0 3
A4	10100100	A 4	U	U		U		164	0 4
A5	10100101	A 4 1	V	V		V		165	0 5
A6	10100110	A 42	W	W		W		166	0 6
A7	10100111	A 421	X	X		X		167	0 7
A8	10101000	A8	Y	Y		Y		168	0 8
A9	10101001	A8 1	Z	Z		Z		169	0 9
AA	10101010	A8 2	(Note 4)	&		–		170	12
AB	10101011	A8 21	,	,		,		171	0 8 3
AC	10101100	A84	%	%		%		172	0 8 4
AD	10101101	A84 1	–	–		–		173	0 8 5
AE	10101110	A842	>	>		>		174	0 8 6
AF	10101111	A8421	?	?		?		175	0 8 7
B0	10110000	A	0	0		0		176	0
B1	10110001	1	1	1		1		177	1
B2	10110010	2	2	2		2		178	2
B3	10110011	21	3	3		3		179	3
B4	10110100	4	4	4		4		180	4
B5	10110101	4 1	5	5		5		181	5
B6	10110110	42	6	6		6		182	6
B7	10110111	421	7	7		7		183	7
B8	10111000	8	8	8		8		184	8
B9	10111001	8 1	9	9		9		185	9
BA	10111010	8 2	:	:		:		186	8 2
BB	10111011	8 21	#	#		#		187	8 3

Hex Value	Binary Value	Card Code (Note 3)	Printer Graphic (Note 1)	Card Graphic (Note 3)	Keyboard Key (Note 2)	Display Station Graphic	Communication Adapter Character	Decimal Value	80-Column Code (Note 3)
BC	10111100	84	@	@		@		188	8 4
BD	10111101	84 1	'	'		'		189	8 5
BE	10111110	842	=	=		=		190	8 6
BF	10111111	8421	≠	"		≠		191	8 7
C0	11000000	No Punches	Blank	Blank		Blank		192	No Punches
C1	11000001	BA 1	A	A		A	A	193	12 1
C2	11000010	BA 2	B	B		B	B	194	12 2
C3	11000011	BA 21	C	C		C	C	195	12 3
C4	11000100	BA 4	D	D		D	D	196	12 4
C5	11000101	BA 4 1	E	E		E	E	197	12 5
C6	11000110	BA 42	F	F		F	F	198	12 6
C7	11000111	BA 421	G	G		G	G	199	12 7
C8	11001000	BA8	H	H		H	H	200	12 8
C9	11001001	BA8 1	I	I		I	I	201	12 9
CA	11001010	BA8 2	¢	¢		¢	¢	202	12 8 2
CB	11001011	BA8 21	.	.		.	.	203	12 8 3
CC	11001100	BA84	<	<		<	<	204	12 8 4
CD	11001101	BA84 1	(	(		(	(	205	12 8 5
CE	11001110	BA842	+	+		+	+	206	12 8 6
CF	11001111	BA8421	|	|		|	|	207	12 8 7
D0	11010000	BA	&	}(Note 5)		&		208	11 0
D1	11010001	B 1	J	J		J	J	209	11 1
D2	11010010	B 2	K	K		K	K	210	11 2
D3	11010011	B 21	L	L		L	L	211	11 3
D4	11010100	B 4	M	M		M	M	212	11 4
D5	11010101	B 4 1	N	N		N	N	213	11 5
D6	11010110	B 42	O	O		O	O	214	11 6
D7	11010111	B 421	P	P		P	P	215	11 7
D8	11011000	B 8	Q	Q		Q	Q	216	11 8
D9	11011001	B 8 1	R	R		R	R	217	11 9
DA	11011010	B 8 2	↑	!		↑		218	11 8 2
DB	11011011	B 8 21	$	$		$		219	11 8 3

Hex Value	Binary Value	Card Code (Note 3)	Printer Graphic (Note 1)	Card Graphic (Note 3)	Keyboard Key (Note 2)	Display Station Graphic	Communication Adapter Character	Decimal Value	80-Column Code (Note 3)
DC	11011100	B 84	*	*		*		220	11 8 4
DD	11011101	B 84 1	)	)		)		221	11 8 5
DE	11011110	B 842	;	;		;		222	11 8 6
DF	11011111	B 8421	¬	¬		¬		223	11 8 7
E0	11100000	B	–	–		–		224	11
E1	11100001	A 1	/	/		/		225	0 1
E2	11100010	A 2	S	S		S	S	226	0 2
E3	11100011	A 21	T	T		T	T	227	0 3
E4	11100100	A 4	U	U		U	U	228	0 4
E5	11100101	A 4 1	V	V		V	V	229	0 5
E6	11100110	A 42	W	W		W	W	230	0 6
E7	11100111	A 421	X	X		X	X	231	0 7
E8	11101000	A8	Y	Y		Y	Y	232	0 8
E9	11101001	A8 1	Z	Z		Z	Z	233	0 9
EA	11101010	A8 2	(Note 4)	&		_		234	12
EB	11101011	A8 21	,	,		,		235	0 8 3
EC	11101100	A84	%	%		%		236	0 8 4
ED	11101101	A84 1	–	–		–		237	0 8 5
EE	11101110	A842	>	>		>		238	0 8 6
EF	11101111	A8421	?	?		?		239	0 8 7
F0	11110000	A	0	0		0	0	240	0
F1	11110001	1	1	1		1	1	241	1
F2	11110010	2	2	2		2	2	242	2
F3	11110011	21	3	3		3	3	243	3
F4	11110100	4	4	4		4	4	244	4
F5	11110101	4 1	5	5		5	5	245	5
F6	11110110	42	6	6		6	6	246	6
F7	11110111	421	7	7		7	7	247	7
F8	11111000	8	8	8		8	8	248	8
F9	11111001	8 1	9	9		9	9	249	9
FA	11111010	8 2	:	:		:		250	8 2
FB	11111011	8 21	#	#		#		251	8 3

Hex Value	Binary Value	Card Code (Note 3)	Printer Graphic (Note 1)	Card Graphic (Note 3)	Keyboard Key (Note 2)	Display Station Graphic	Communication Adapter Character	Decimal Value	80-Column Code (Note 3)	
FC	11111100	84	@	@		@		252	8	4
FD	11111101	84 1	'	'		'		253	8	5
FE	11111110	842	=	=		=		254	8	6
FF	11111111	8421	≠	≠		≠		255	8	7

Notes:

1. The printer will print a character no matter what binary value is contained in the data byte. The circled graphics indicate the preferred EBCDIC value for the particular character.

2. Certain binary values are generated by more than one key on the keyboard, and the decimal key on the numeric section of the keyboard generates a different value from that generated by the key on the typewriter section. To differentiate between these keys in those instances when it is necessary, the following code is used: T = typewriter section, A = numeric section, and C = command key section.

3. The data recorder or card data recorder will punch a character no matter what binary value is contained in the data byte. The circled graphics and card codes indicate the preferred EBCDIC value for the particular character and the EBCDIC that is stored when that character is read from the card.

4. This is a special character used only to produce line finder marks and ID numbers for the ledger card device.

5. This character does not print on the IBM 129 Card Data Recorder.

Appendix C. Powers of Two Table

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125

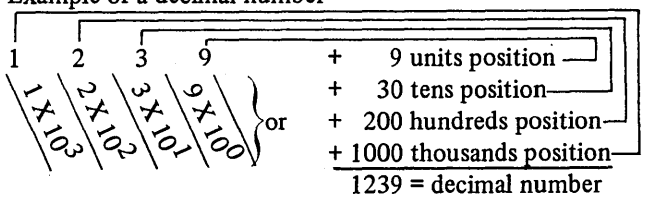
Appendix D. Binary and Hexadecimal Number Notations

Binary Number Notation

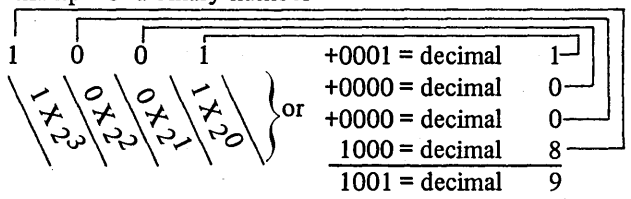
A binary number system, such as is used in System/3 uses a base of two. The concept of using a base of two can be compared with the base of ten (decimal) number system.

Decimal Number		*Binary Number*
0	=	0
1	=	1
2	=	10
3	=	11
4	=	100
5	=	101
6	=	110
7	=	111
8	=	1000
9	=	1001

Example of a decimal number



Example of a binary number



The decimal number system allows counting to ten in each position—from units to tens to hundreds to thousands etc. The binary system allows counting to two in each position. Register displays in the System/3 are in binary forms: a bit light on is a "one"; a bit light off is a "zero."

Hexadecimal Number System

It has been noted that binary numbers require about three times as many positions as decimal numbers to express the equivalent number. This is not much of a problem to the computer; however, in talking and writing or in communicating with the computer, these binary numbers are bulky. A long string of 1's and 0's cannot be effectively transmitted from one individual to another. Some shorthand method is necessary.

The hexadecimal number system fills this need. Because of the simple relationship of hexadecimal to binary, numbers can be converted from one system to another by inspection. The base or radix of the hexadecimal system is 16. This means there are 16 symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. The letters A, B, C, D, E, and F represent the 10-base system values of 10, 11, 12, 13, 14, and 15, respectively.

Four binary positions are equivalent to one hexadecimal position. The following table shows the comparable values of the three number systems.

Decimal	*Binary*	*Hexadecimal*
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

At this point all 16 symbols have been used, and a carry to the next higher position of the number is necessary. For example:

Decimal	*Binary*	*Hexadecimal*
16	0001 0000	10
17	0001 0001	11
18	0001 0010	12
19	0001 0011	13
20	0001 0100	14
21	0001 0101	15

–and so on–

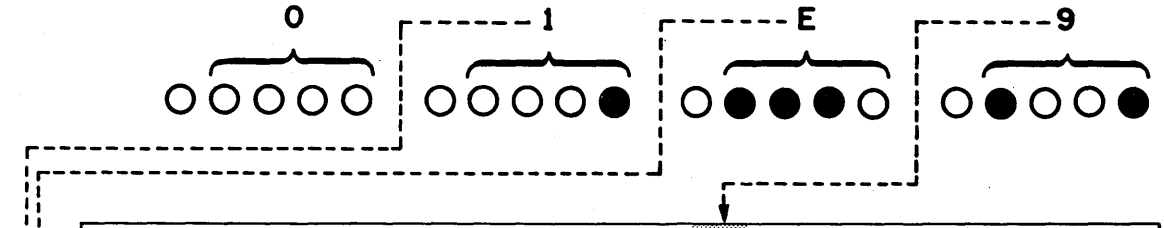
Remember that as far as the internal circuitry of the computer is concerned, it only understands binary. But an operator can look at a series of lights on the computer console showing binary 1's and 0's, for example: 0001 1110 0001 0011, and say that the lights represent the hexadecimal value 1E13 which is easier to state than the string of 1's and 0's.

The tables in this appendix provide for direct conversion of decimal and hexadecimal numbers in these ranges:

Hexadecimal	*Decimal*
000 to FFF	0000 to 4095

For numbers outside the range of the tables, add the following values to the table figures:

Hexadecimal	*Decimal*
1000	4096
2000	8192
3000	12288
4000	16384
5000	20480
6000	24576
7000	28672
8000	32768



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 _	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015
01 _	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031
02 _	0032	0033	0034	0035	0036	0037	0038	0039	0040	0041	0042	0043	0044	0045	0046	0047
03 _	0048	0049	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	0060	0061	0062	0063
04 _	0064	0065	0066	0067	0068	0069	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079
05 _	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	0090	0091	0092	0093	0094	0095
06 _	0096	0097	0098	0099	0100	0101	0102	0103	0104	0105	0106	0107	0108	0109	0110	0111
07 _	0112	0113	0114	0115	0116	0117	0118	0119	0120	0121	0122	0123	0124	0125	0126	0127
08 _	0128	0129	0130	0131	0132	0133	0134	0135	0136	0137	0138	0139	0140	0141	0142	0143
09 _	0144	0145	0146	0147	0148	0149	0150	0151	0152	0153	0154	0155	0156	0157	0158	0159
0A _	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	0173	0174	0175
0B _	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0186	0187	0188	0189	0190	0191
0C _	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0202	0203	0204	0205	0206	0207
0D _	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0218	0219	0220	0221	0222	0223
0E _	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0234	0235	0236	0237	0238	0239
0F _	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0250	0251	0252	0253	0254	0255
10 _	0256	0257	0258	0259	0260	0261	0262	0263	0264	0265	0266	0267	0268	0269	0270	0271
11 _	0272	0273	0274	0275	0276	0277	0278	0279	0280	0281	0282	0283	0284	0285	0286	0287
12 _	0288	0289	0290	0291	0292	0293	0294	0295	0296	0297	0298	0299	0300	0301	0302	0303
13 _	0304	0305	0306	0307	0308	0309	0310	0311	0312	0313	0314	0315	0316	0317	0318	0319
14 _	0320	0321	0322	0323	0324	0325	0326	0327	0328	0329	0330	0331	0332	0333	0334	0335
15 _	0336	0337	0338	0339	0340	0341	0342	0343	0344	0345	0346	0347	0348	0349	0350	0351
16 _	0352	0353	0354	0355	0356	0357	0358	0359	0360	0361	0362	0363	0364	0365	0366	0367
17 _	0368	0369	0370	0371	0372	0373	0374	0375	0376	0377	0378	0379	0380	0381	0382	0383
18 _	0384	0385	0386	0387	0388	0389	0390	0391	0392	0393	0394	0395	0396	0397	0398	0399
19 _	0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	0410	0411	0412	0413	0414	0415
1A _	0416	0417	0418	0419	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	0430	0431
1B _	0432	0433	0434	0435	0436	0437	0438	0439	0440	0441	0442	0443	0444	0445	0446	0447
1C _	0448	0449	0450	0451	0452	0453	0454	0455	0456	0457	0458	0459	0460	0461	0462	0463
1D _	0464	0465	0466	0467	0468	0469	0470	0471	0472	0473	0474	0475	0476	0477	0478	0479
1E _	0480	0481	0482	0483	0484	0485	0486	0487	0488	0489	0490	0491	0492	0493	0494	0495
1F _	0496	0497	0498	0499	0500	0501	0502	0503	0504	0505	0506	0507	0508	0509	0510	0511

BR0432

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20_	0512	0513	0514	0515	0516	0517	0518	0519	0520	0521	0522	0523	0524	0525	0526	0527
21_	0528	0529	0530	0531	0532	0533	0534	0535	0536	0537	0538	0539	0540	0541	0542	0543
22_	0544	0545	0546	0547	0548	0549	0550	0551	0552	0553	0554	0555	0556	0557	0558	0559
23_	0560	0561	0562	0563	0564	0565	0566	0567	0568	0569	0570	0571	0572	0573	0574	0575
24_	0576	0577	0578	0579	0580	0581	0582	0583	0584	0585	0586	0587	0588	0589	0590	0591
25_	0592	0593	0594	0595	0596	0597	0598	0599	0600	0601	0602	0603	0604	0605	0606	0607
26_	0608	0609	0610	0611	0612	0613	0614	0615	0616	0617	0618	0619	0620	0621	0622	0623
27_	0624	0625	0626	0627	0628	0629	0630	0631	0632	0633	0634	0635	0636	0637	0638	0639
28_	0640	0641	0642	0643	0644	0645	0646	0647	0648	0649	0650	0651	0652	0653	0654	0655
29_	0656	0657	0658	0659	0660	0661	0662	0663	0664	0665	0666	0667	0668	0669	0670	0671
2A_	0672	0673	0674	0675	0676	0677	0678	0679	0680	0681	0682	0683	0684	0685	0686	0687
2B_	0688	0689	0690	0691	0692	0693	0694	0695	0696	0697	0698	0699	0700	0701	0702	0703
2C_	0704	0705	0706	0707	0708	0709	0710	0711	0712	0713	0714	0715	0716	0717	0718	0719
2D_	0720	0721	0722	0723	0724	0725	0726	0727	0728	0729	0730	0731	0732	0733	0734	0735
2E_	0736	0737	0738	0739	0740	0741	0742	0743	0744	0745	0746	0747	0748	0749	0750	0751
2F_	0752	0753	0754	0755	0756	0757	0758	0759	0760	0761	0762	0763	0764	0765	0766	0767
30_	0768	0769	0770	0771	0772	0773	0774	0775	0776	0777	0778	0779	0780	0781	0782	0783
31_	0784	0785	0786	0787	0788	0789	0790	0791	0792	0793	0794	0795	0796	0797	0798	0799
32_	0800	0801	0802	0803	0804	0805	0806	0807	0808	0809	0810	0811	0812	0813	0814	0815
33_	0816	0817	0818	0819	0820	0821	0822	0823	0824	0825	0826	0827	0828	0829	0830	0831
34_	0832	0833	0834	0835	0836	0837	0838	0839	0840	0841	0842	0843	0844	0845	0846	0847
35_	0848	0849	0850	0851	0852	0853	0854	0855	0856	0857	0858	0859	0860	0861	0862	0863
36_	0864	0865	0866	0867	0868	0869	0870	0871	0872	0873	0874	0875	0876	0877	0878	0879
37_	0880	0881	0882	0883	0884	0885	0886	0887	0888	0889	0890	0891	0892	0893	0894	0895
38_	0896	0897	0898	0899	0900	0901	0902	0903	0904	0905	0906	0907	0908	0909	0910	0911
39_	0912	0913	0914	0915	0916	0917	0918	0919	0920	0921	0922	0923	0924	0925	0926	0927
3A_	0928	0929	0930	0931	0932	0933	0934	0935	0936	0937	0938	0939	0940	0941	0942	0943
3B_	0944	0945	0946	0947	0948	0949	0950	0951	0952	0953	0954	0955	0956	0957	0958	0959
3C_	0960	0961	0962	0963	0964	0965	0966	0967	0968	0969	0970	0971	0972	0973	0974	0975
3D_	0976	0977	0978	0979	0980	0981	0982	0983	0984	0985	0986	0987	0988	0989	0990	0991
3E_	0992	0993	0994	0995	0996	0997	0998	0999	1000	1001	1002	1003	1004	1005	1006	1007
3F_	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
40_	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039
41_	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055
42_	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071
43_	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087
44_	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103
45_	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119
46_	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135
47_	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151
48_	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167
49_	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183
4A_	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199
4B_	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
4C_	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231
4D_	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247
4E_	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263
4F_	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279
50_	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295
51_	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311
52_	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327
53_	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343
54_	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359
55_	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375
56_	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391
57_	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407
58_	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423
59_	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439
5A_	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455
5B_	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471
5C_	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487
5D_	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503
5E_	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519
5F_	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535

BR0432

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
60_	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551
61_	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567
62_	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583
63_	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
64_	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615
65_	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
66_	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647
67_	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	1662	1663
68_	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679
69_	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695
6A_	1696	1697	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1709	1710	1711
6B_	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721	1722	1723	1724	1725	1726	1727
6C_	1728	1729	1730	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743
6D_	1744	1745	1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756	1757	1758	1759
6E_	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775
6F_	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787	1788	1789	1790	1791
70_	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804	1805	1806	1807
71_	1808	1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820	1821	1822	1823
72_	1824	1825	1826	1827	1828	1829	1830	1831	1832	1833	1834	1835	1836	1837	1838	1839
73_	1840	1841	1842	1843	1844	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855
74_	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871
75_	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887
76_	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903
77_	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919
78_	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935
79_	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951
7A_	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967
7B_	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983
7C_	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
7D_	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
7E_	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031
7F_	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
80_	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063
81_	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079
82_	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095
83_	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111
84_	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127
85_	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143
86_	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159
87_	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175
88_	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191
89_	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207
8A_	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223
8B_	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239
8C_	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255
8D_	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271
8E_	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287
8F_	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303
90_	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319
91_	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335
92_	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351
93_	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367
94_	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383
95_	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399
96_	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415
97_	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431
98_	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447
99_	2448	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463
9A_	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479
9B_	2480	2481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495
9C_	2496	2497	2498	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510	2511
9D_	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527
9E_	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543
9F_	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557	2558	2559

BR0432

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A0 _	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572	2573	2574	2575
A1 _	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586	2587	2588	2589	2590	2591
A2 _	2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605	2606	2607
A3 _	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618	2619	2620	2621	2622	2623
A4 _	2624	2625	2626	2627	2628	2629	2630	2631	2632	2633	2634	2635	2636	2637	2638	2639
A5 _	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652	2653	2654	2655
A6 _	2656	2657	2658	2659	2660	2661	2662	2663	2664	2665	2666	2667	2668	2669	2670	2671
A7 _	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687
A8 _	2688	2689	2690	2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2701	2702	2703
A9 _	2704	2705	2706	2707	2708	2709	2710	2711	2712	2713	2714	2715	2716	2717	2718	2719
AA _	2720	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732	2733	2734	2735
AB _	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749	2750	2751
AC _	2752	2753	2754	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765	2766	2767
AD _	2768	2769	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780	2781	2782	2783
AE _	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795	2796	2797	2798	2799
AF _	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813	2814	2815
B0 _	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2831
B1 _	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843	2844	2845	2846	2847
B2 _	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860	2861	2862	2863
B3 _	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879
B4 _	2880	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894	2895
B5 _	2896	2897	2898	2899	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909	2910	2911
B6 _	2912	2913	2914	2915	2916	2917	2918	2919	2920	2921	2922	2923	2924	2925	2926	2927
B7 _	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943
B8 _	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959
B9 _	2960	2961	2962	2963	2964	2965	2966	2967	2968	2969	2970	2971	2972	2973	2974	2975
BA _	2976	2977	2978	2979	2980	2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991
BB _	2992	2993	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005	3006	3007
BC _	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022	3023
BD _	3024	3025	3026	3027	3028	3029	3030	3031	3032	3033	3034	3035	3036	3037	3038	3039
BE _	3040	3041	3042	3043	3044	3045	3046	3047	3048	3049	3050	3051	3052	3053	3054	3055
BF _	3056	3057	3058	3059	3060	3061	3062	3063	3064	3065	3066	3067	3068	3069	3070	3071

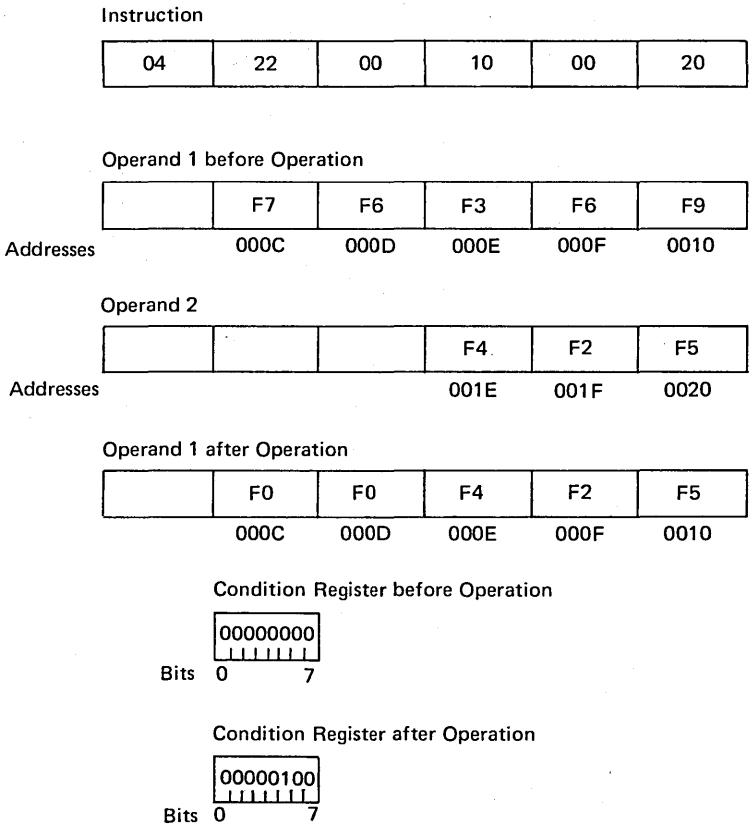
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C0 _	3072	3073	3074	3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086	3087
C1 _	3088	3089	3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103
C2 _	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114	3115	3116	3117	3118	3119
C3 _	3120	3121	3122	3123	3124	3125	3126	3127	3128	3129	3130	3131	3132	3133	3134	3135
C4 _	3136	3137	3138	3139	3140	3141	3142	3143	3144	3145	3146	3147	3148	3149	3150	3151
C5 _	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	3163	3164	3165	3166	3167
C6 _	3168	3169	3170	3171	3172	3173	3174	3175	3176	3177	3178	3179	3180	3181	3182	3183
C7 _	3184	3185	3186	3187	3188	3189	3190	3191	3192	3193	3194	3195	3196	3197	3198	3199
C8 _	3200	3201	3202	3203	3204	3205	3206	3207	3208	3209	3210	3211	3212	3213	3214	3215
C9 _	3216	3217	3218	3219	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231
CA _	3232	3233	3234	3235	3236	3237	3238	3239	3240	3241	3242	3243	3244	3245	3246	3247
CB _	3248	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	3259	3260	3261	3262	3263
CC _	3264	3265	3266	3267	3268	3269	3270	3271	3272	3273	3274	3275	3276	3277	3278	3279
CD _	3280	3281	3282	3283	3284	3285	3286	3287	3288	3289	3290	3291	3292	3293	3294	3295
CE _	3296	3297	3298	3299	3300	3301	3302	3303	3304	3305	3306	3307	3308	3309	3310	3311
CF _	3312	3313	3314	3315	3316	3317	3318	3319	3320	3321	3322	3323	3324	3325	3326	3327
D0 _	3328	3329	3330	3331	3332	3333	3334	3335	3336	3337	3338	3339	3340	3341	3342	3343
D1 _	3344	3345	3346	3347	3348	3349	3350	3351	3352	3353	3354	3355	3356	3357	3358	3359
D2 _	3360	3361	3362	3363	3364	3365	3366	3367	3368	3369	3370	3371	3372	3373	3374	3375
D3 _	3376	3377	3378	3379	3380	3381	3382	3383	3384	3385	3386	3387	3388	3389	3390	3391
D4 _	3392	3393	3394	3395	3396	3397	3398	3399	3400	3401	3402	3403	3404	3405	3406	3407
D5 _	3408	3409	3410	3411	3412	3413	3414	3415	3416	3417	3418	3419	3420	3421	3422	3423
D6 _	3424	3425	3426	3427	3428	3429	3430	3431	3432	3433	3434	3435	3436	3437	3438	3439
D7 _	3440	3441	3442	3443	3444	3445	3446	3447	3448	3449	3450	3451	3452	3453	3454	3455
D8 _	3456	3457	3458	3459	3460	3461	3462	3463	3464	3465	3466	3467	3468	3469	3470	3471
D9 _	3472	3473	3474	3475	3476	3477	3478	3479	3480	3481	3482	3483	3484	3485	3486	3487
DA _	3488	3489	3490	3491	3492	3493	3494	3495	3496	3497	3498	3499	3500	3501	3502	3503
DB _	3504	3505	3506	3507	3508	3509	3510	3511	3512	3513	3514	3515	3516	3517	3518	3519
DC _	3520	3521	3522	3523	3524	3525	3526	3527	3528	3529	3530	3531	3532	3533	3534	3535
DD _	3536	3537	3538	3539	3540	3541	3542	3543	3544	3545	3546	3547	3548	3549	3550	3551
DE _	3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565	3566	3567
DF _	3568	3569	3570	3571	3572	3573	3574	3575	3576	3577	3578	3579	3580	3581	3582	3583

BR0432

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E0_	3584	3585	3586	3587	3588	3589	3590	3591	3592	3593	3594	3595	3596	3597	3598	3599
E1_	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612	3613	3614	3615
E2_	3616	3617	3618	3619	3620	3621	3622	3623	3624	3625	3626	3627	3628	3629	3630	3631
E3_	3632	3633	3634	3635	3636	3637	3638	3639	3640	3641	3642	3643	3644	3645	3646	3647
E4_	3648	3649	3650	3651	3652	3653	3654	3655	3656	3657	3658	3659	3660	3661	3662	3663
E5_	3664	3665	3666	3667	3668	3669	3670	3671	3672	3673	3674	3675	3676	3677	3678	3679
E6_	3680	3681	3682	3683	3684	3685	3686	3687	3688	3689	3690	3691	3692	3693	3694	3695
E7_	3696	3697	3698	3699	3700	3701	3702	3703	3704	3705	3706	3707	3708	3709	3710	3711
E8_	3712	3713	3714	3715	3716	3717	3718	3719	3720	3721	3722	3723	3724	3725	3726	3727
E9_	3728	3729	3730	3731	3732	3733	3734	3735	3736	3737	3738	3739	3740	3741	3742	3743
EA_	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755	3756	3757	3758	3759
EB_	3760	3761	3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774	3775
EC_	3776	3777	3778	3779	3780	3781	3782	3783	3784	3785	3786	3787	3788	3789	3790	3791
ED_	3792	3793	3794	3795	3796	3797	3798	3799	3800	3801	3802	3803	3804	3805	3806	3807
EE_	3808	3809	3810	3811	3812	3813	3814	3815	3816	3817	3818	3819	3820	3821	3822	3823
EF_	3824	3825	3826	3827	3828	3829	3830	3831	3832	3833	3834	3835	3836	3837	3838	3839
F0_	3840	3841	3842	3843	3844	3845	3846	3847	3848	3849	3850	3851	3852	3853	3854	3855
F1_	3856	3857	3858	3859	3860	3861	3862	3863	3864	3865	3866	3867	3868	3869	3870	3871
F2_	3872	3873	3874	3875	3876	3877	3878	3879	3880	3881	3882	3883	3884	3885	3886	3887
F3_	3888	3889	3890	3891	3892	3893	3894	3895	3896	3897	3898	3899	3900	3901	3902	3903
F4_	3904	3905	3906	3907	3908	3909	3910	3911	3912	3913	3914	3915	3916	3917	3918	3919
F5_	3920	3921	3922	3923	3924	3925	3926	3927	3928	3929	3930	3931	3932	3933	3934	3935
F6_	3936	3937	3938	3939	3940	3941	3942	3943	3944	3945	3946	3947	3948	3949	3950	3951
F7_	3952	3953	3954	3955	3956	3957	3958	3959	3960	3961	3962	3963	3964	3965	3966	3967
F8_	3968	3969	3970	3971	3972	3973	3974	3975	3976	3977	3978	3979	3980	3981	3982	3983
F9_	3984	3985	3986	3987	3988	3989	3990	3991	3992	3993	3994	3995	3996	3997	3998	3999
FA_	4000	4001	4002	4003	4004	4005	4006	4007	4008	4009	4010	4011	4012	4013	4014	4015
FB_	4016	4017	4018	4019	4020	4021	4022	4023	4024	4025	4026	4027	4028	4029	4030	4031
FC_	4032	4033	4034	4035	4036	4037	4038	4039	4040	4041	4042	4043	4044	4045	4046	4047
FD_	4048	4049	4050	4051	4052	4053	4054	4055	4056	4057	4058	4059	4060	4061	4062	4063
FE_	4064	4065	4066	4067	4068	4069	4070	4071	4072	4073	4074	4075	4076	4077	4078	4079
FF_	4080	4081	4082	4083	4084	4085	4086	4087	4088	4089	4090	4091	4092	4093	4094	4095

BR0432

Zero and Add Zoned (ZAZ)



Add Zoned Decimal (AZ)

Instruction

06	22	00	10	00	20

Operand 1 before Operation

	F7	F6	F3	F6	F9
Addresses	000C	000D	000E	000F	0010

Operand 2

			F4	F2	F5
Addresses			001E	001F	0020

Operand 1 after Operation

	F7	F6	F7	F9	F4
	000C	000D	000E	000F	0010

Condition Register before Operation

00000000

Bits 0 7

Condition Register after Operation

00000100

Bits 0 7

Subtract Zoned Decimal (SZ)

Instruction

07	22	00	10	00	20

Operand 1 before Operation

	F7	F6	F3	F6	F9
Addresses	000C	000D	000E	000F	0010

Operand 2

			F4	F2	F5
Addresses			001E	001F	0020

Operand 1 after Operation

	F7	F5	F9	F4	F4
	000C	000D	000E	000F	0010

Condition Register before Operation

00000000
Bits 0 7

Condition Register after Operation

00000100
Bits 0 7

Add Logical Characters (ALC)

Instruction Index Register 1 = 0CC0

5E	03	00	10

Operand 1 before Operation

		00110101	11001011	11101101	01100100
Addresses		0CBD	0CBE	0CBF	0CC0

Operand 2

		01011011	01010101	01111000	11001101
Addresses		0CCD	0CCE	0CCF	0CD0

Operand 1 after Operation

		10010001	00100001	01100110	00110001
		0CBD	0CBE	0CBF	0CC0

Condition Register before Operation

00000000

Bits 0 7

Condition Register after Operation

00000010

Bits 0 7

Subtract Logical Characters (SLC)

Instruction Index Register 2 = 0CC0

AF	03	00	10

Operand 1 before Operation

		10010110	01011010	01110111	10111111
Addresses		0CBD	0CBE	0CBF	0CC0

Operand 2

		01110100	10000110	01100010	10100100
Addresses		0CCD	0CCE	0CCF	0CD0

Operand 1 after Operation

		00100001	11010100	00010101	00011011
		0CBD	0CBE	0CBF	0CC0

Condition Register before Operation

0000 0000

Bits 0 7

Condition Register after Operation

00000 1 00

Bits 0 7

Add To Register (A)

Instruction

36	00000010	00	04

Operand 1

01001000	00100000
0003	0004

Index Register 2

00110101	01101010

Before Operation

01111101	10001010

After Operation

Condition Register after Operation

00 000 0 10

Bits 0 7

Move Hex Character (MVX)

Instruction

98	01	A0	65

Index Register 1 = 2B15
Index Register 2 = 1F20

Operand 1 before Operation

2F

1FC0

Operand 2

4C

2B7A

Operand 1 after Operation

CF

1FC0

Move Characters (MVC)

Instruction

0C	06	1A	06	2B	5A

Operand 1 before Operation

	D1	C1	D4	C5	E2	
Addresses	1A01	1A02	1A03	1A04	1A05	1A06

Operand 2

	D9	D6	C2	C5	D9	E3
Addresses	2B55	2B56	2B57	2B58	2B59	2B5A

Operand 1 after Operation

D9	D6	C2	C5	D9	E3
1A01	1A02	1A03	1A04	1A05	1A06

Edit (ED)

Instruction

0A	0A	00	BF	00	07

Operand 1 before Operation

$	20	,	20	20	20	.	20	20	ъ	*
00B5	00B6	00B7	00B8	00B9	00BA	00BB	00BC	00BD	00BE	00BF

Operand 2

0	1	0	8	0	R
0002	0003	0004	0005	0006	0007

Note: "R" represents "-9"

Operand 1 after Operation

$	0	,	1	0	8	.	0	9	ъ	*
00B5	00B6	00B7	00B8	00B9	00BA	00BB	00BC	00BD	00BE	00BF

Note: Location 00BD contains a 9 because the zone bits of all replaced characters (zeros) in the edit pattern are set to all ones.

Condition Code

00000010

Bits 0 7

Insert and Test Characters (ITC)

Instruction

0B	09	00	B6	00	10

Operand 1 before Operation

$	0	,	1	0	8	.	0	9	ъ	*
00B5	00B6	00B7	00B8	00B9	00BA	00BB	00BC	00BD	00BE	00BF

Operand 2

*
0010

Operand 1 after Operation

$	*	*	1	0	8	.	0	9	ъ	*
00B5	00B6	00B7	00B8	00B9	00BA	00BB	00BC	00BD	00BE	00BF

Note: Address 00B5 was not included in the first operand.

Address Recall Register after Operation

00	B8

Move Logical Immediate (MVI)

Instruction

3C	AF	2F	CB

Operand before Operation

00

2FCB

Operand after Operation

AF

2FCB

Set Bits On Masked (SBN)

Instruction

3A	01011010	00	20

Operand before Operation

00001100

0020

Operand after Operation

01011110

0020

Set Bits Off Masked (SBF)

Instruction

3B	10000001	00	30

Operand before Operation

01111001

0030

Operand after Operation

01111000

0030

Store Register (ST)

Instruction

34	00001000	32	BB

Address Recall Register

0A	CD

Operand before Operation

2F	C2
32BA	32BB

Operand after Operation

0A	CD
32BA	32BB

Load Register (L)

Instruction

35	00000100	00	11

Operand

00000000	00000000
0010	0011

Program Status Register before Operation

00001100	00110001

Program Status Register after Operation

00000000	00000100

Condition Register after Operation

00000100
Bits 0 7

Load Address (LA)

Instruction

D2	02	05

Index Register 1

2A	15

Index Register 2 after Operation

2A	1A

Compare Logical Characters (CLC)

Instruction

0D	02	00	12	00	02

Operand 1

27	FA	26
0010	0011	0012

Operand 2

23	FA	26
0000	0001	0002

Condition Register

00000100

Bits 0 7

Compare Logical Immediate (CLI)

Instruction

3D	7F	00	21

Storage Operand

7F
0021

Condition Register after Operation

00000001

Bits 0 7

Test Bits On Masked (TBN)

Instruction

38	00010110	00	21

Storage Operand

10010101

0021

Condition Register after Operation

00010000

Bits 0 7

Test Bits Off Masked (TBF)

Instruction

39	01101100	00	25

Storage Operand

10010100

0025

Condition Register after Operation

00010000

Bits 0 7

Branch On Condition (BC)

Instruction

C0	10001000	02	BF

0BCC 0BCD 0BCE 0BCF

Condition Code before Operation

00011001

Instruction Address Register after Operation

02	BF

Address Recall Register after Operation

0B	D0

Condition Register after Operation

00010001

Bits 0 7

Jump On Condition (JC)

Instruction

F2	00110000	0F
0BBD	0BBE	0BBF

Condition Register before Operation

00001001

Instruction Address Register after Operation

0B	CF

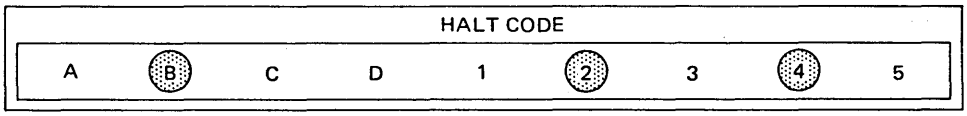
Condition Register after Operation

00001001

Bits 0 7

Halt Program Level (HPL)

F0	20	28



Start I/O for Keyboard Console (SIO)

Instruction

F3	10	0F

The last key pressed is restored and the keyboard is prepared for the next keystroke. The keyboard interrupt request is reset, and interrupt level 1 is released.

Sense I/O for Keyboard Console (SNS)

Instruction

30	10	0F	CD

Two bytes of information from the keyboard are stored in storage locations 0FCC and 0FCD

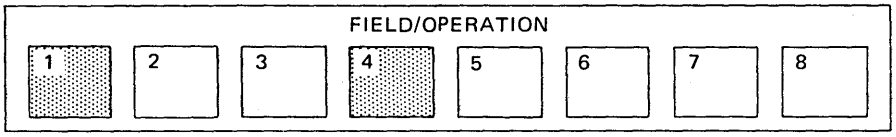
Load I/O for Keyboard Console (LIO)

Instruction

31	00010010	03	C1

Operand

00000000	10010000
03C0	03C1



Start I/O for Serial Printer (SIO)

Instruction

F3	E0	01

Commands

C0	5F	05

The printer prints a 96-character line, indexes 1 line space, and stops with the print head at either the left margin, or 96 characters away from the left margin, if the bi-directional printing feature is installed. If the bi-directional printing feature is not installed, the element return command in the third command byte causes the print head to return to the left margin.

Test I/O and Branch for Serial Printer (TIO)

Instruction

	C1	E0	0C	21
Address	0123	0124	0125	0126

If unit check exists in the printer attachment, the next instruction is taken from 0C21. If unit check does not exist, the next instruction is taken from 0127.

Advance Program Level for Serial Printer (APL)

Instruction

F1	EE	00

If a ledger card is aligned at the ledger card device alignment station, the system executes the next sequential instruction. If no card is aligned at the alignment station, the system repeats this instruction and turns on the LCD attention indicator.

Load I/O for Serial Printer (LIO)

Instruction

31	E6	3F	17

Operand

28	29
3F16	3F17

Print Command Address Register before Operation

00	26

Print Command Address Register after Operation

28	29

Sense I/O for Serial Printer (SNS)

Instruction

30	E2	05	70

Two status bytes are stored in main storage at locations 056F and 0570.

Start I/O for Disk File (SIO)

Instruction

F3	A1	00

Disk File Identifier Address Register

02	00

Disk File Data Address Register

0F	00

Disk Control Field

00	07	A0	01
0200	0201	0202	0203

512 bytes of data will be transferred to storage and placed in locations 0F00 through 10FF.

Load I/O for Disk File (LIO)

Instruction

31	B4	2F	B1

Operand

0F	00
2FB0	2FB1

Disk File Read/Write Address Register before Operation

0B	20

Disk File Read/Write Address Register after Operation

0F	00

Test I/O and Branch for Disk File (TIO)

Instruction

C1	A4	02	00
0100	0101	0102	0103

Status Byte 1

10000000

Instruction Address Register before Operation

01	04

Address Recall Register before Operation

2F	C7

Instruction Address Register after Operation

02	00

Address Recall Register after Operation

01	04

Advance Program Level for Disk File (APL)

Instruction

F1	A2	00

The program continues to execute this instruction until drive 1 is no longer busy.

Sense I/O for Disk File (SNS)

Instruction

30	A2	05	FF

Status Bytes at File before Operation

81	00

Operand before Operation

00	AB
05FE	05FF

Operand after Operation

81	00
05FE	05FF

Status Bytes at File after Operation

81	00

Appendix G. Instruction Timings

The following abbreviations are used in the instruction timing formulas:

N = Instruction length in bytes.
L1 = Length of destination field (two-address instruction) in bytes. The destination field is that field addressed by operand 1.
L2 = Length of source field (two-address instruction) in bytes. Source field is that field addressed by operand 2.
R = Length of operand 1 when recomplementing is necessary.

Instruction and Mnemonic	*Time in Microseconds*
Zero and add zoned (ZAZ)	1.52(N + L1 + L2) + 1.52(R)
Add zoned decimal (AZ)	1.52(N + L1 + L2) + 1.52(R)
Subtract zoned decimal (SZ)	1.52(N + L1 + L2) + 1.52(R)
Add logical characters (ALC)	1.52(N + 2L)
Subtract logical characters (SLC)	1.52(N + 2L)
Add to register (A)	1.52(N + 2)
Move hex character (MVX)	1.52(N + 2)
Move characters (MVC)	1.52(N + 2L)
Edit (ED)	1.52(N + L1 + L2)
Insert and test characters (ITC)	1.52(N + 1 + L1)
Move logical immediate (MVI)	1.52(N + 1)
Set bits on masked (SBN)	1.52(N + 1)
Set bits off masked (SBF)	1.52(N + 1)
Store register (ST)	1.52(N + 2)
Load register (L)	1.52(N + 2)
Load address (LA)	1.52(N)
Compare logical characters (CLC)	1.52(N + 2L)
Compare logical immediate (CLI)	1.52(N + 1)
Test bits on masked (TBN)	1.52(N + 1)
Test bits off masked (TBF)	1.52(N + 1)
Branch on condition (BC)	1.52(N)
Jump on condition (JC)	4.56
Halt program level (HPL)	4.56
Start I/O (SIO)	4.56
Sense I/O (SNS)	1.52(N + 2)
Load I/O (LIO)	1.52(N + 2)
Test I/O and branch (TIO)	1.52(N)
Advance program level (APL)	4.56

ALU. Arithmetic and logic unit.

AND. A logic operator having the property that if P is a statement, Q is a statement, R is a statement, . . ., then the AND of P, Q, R, . . ., is true if all statements are true, false if any statement is false.

Base-Displacement Address. An address formed by adding a base value to a displacement value.

Binary Cell. A storage cell of one binary digit capacity, for example, a single bit register or a single bit in a byte.

Binary Overflow. An indication that an overflow occurred during a binary arithmetic operation.

Bit. Contraction of "binary digit", the smallest unit of information in a binary system. A bit may be either a 1 or a 0.

Broadband. Communication channel having a band width greater than a voice-grade channel, and therefore capable of higher-speed data transmission.

Byte. A sequence of adjacent binary digits operated upon as a unit.

Cathode Ray Tube Display. (1) A device that presents data in a visual form by means of a controlled electron beam. (2) The data display produced by the device as in (1).

Chain, Chaining. A function of linking commands so as to execute the commands in sequence without requiring a new instruction for each command.

Change of Direction Character. A character that causes the communication adapter to change from transmit mode to receive mode or from receive mode to transmit mode.

Command. (1) A control signal. (2) Loosely, an instruction in machine language. (3) Loosely, a mathematical or logic operator.

Command Chain. A group of commands chained together (see also Chain, Chaining).

Complement. A number derived by subtracting the true value of the desired number from the base number. For example, the nines complement of 02735 is 99999-02735 = 97264. To indicate a complement number, the storage or accumulator that must contain the number must have one high-order digit position more than is required to contain the true value of the number.

Control Character. A character whose occurrence in a particular context initiates, modifies, or stops a control action—for example, a character that controls carriage return, or a character that controls transmission of data over communication networks. A control character may be recorded for use in a subsequent action. It may have a graphic representation in some circumstances.

Conversational Mode. Communication between a terminal and the computer in which each entry from the terminal elicits a response from the computer and vice versa.

CRT. (see Cathode Ray Tube Display)

Cycle. The time interval in which the processing unit reads one byte from storage and writes one byte into storage.

Cycle Steal. The interruption of processing unit instruction execution to allow an I/O unit one or more cycles for access to storage.

Decimal Overflow. An indication that an overflow occurred on a decimal arithmetic operation.

Defined Graphic. A graphic character for which an EBCDIC code has been defined for the system.

Digit Portion. The low-order four bits of a byte.

Direct Address. An address that specifies the location of an operand.

E-Phase. (see Execution Phase)

EBCDIC. (see Extended Binary Coded Decimal Interchange Code)

Execution Phase. That portion of a System/3 operation in which the operands specified by an instruction are operated on as specified by the instruction.

Extended Binary Coded Decimal Interchange Code. The eight bit code used within System/3 to represent printable characters.

Field. In a record, a specified area used for a particular category of data, for example, a group of card columns used to represent a wage-rate period.

Flag. (1) Any of various indicators used for identification, for example, a word mark. (2) A character that signals the occurrence of some condition.

Four-Wire Circuit. A communication path in which four wires (two for each direction of transmission) are presented to the station equipment.

Graphic. A symbol produced by a process such as handwriting, drawings, or printing.

Graphic Character. A character normally represented by a graphic.

Half Duplex. In communication, pertaining to an alternate, one way at a time, independent transmission.

Halt Identifier. A code displayed by the halt indicator lamps on the keyboard console.

High-Order. Pertaining to the leftmost and, usually, lowest numbered position of a byte or a field.

I-Phase. (see Instruction Phase)

Initial Program Load, Initial Program Loading (IPL). The procedure that causes the initial part of an operating system or other program to be loaded such that the program can then proceed under its own control.

Instruction Phase. The portion of a System/3 operation in which an instruction is retrieved from storage and analyzed for the operation to be performed.

Interrupt. (1) To stop a process in such a way that it can be resumed. (2) Also as "interruption", a system of input/output operations in which the input/output units are allowed to interrupt the main program stream to perform subroutines.

Interrupt Level. A system of priorities for input/output units which allows higher priority units to interrupt the subroutines of lower priority units.

IPL. (see Initial Program Load)

Line Control Character. (see Control Character)

Line Printer. A device that prints all characters of a line as a unit.

Loop. A sequence of instructions that is executed repeatedly until a terminal condition prevails.

Low-Order. Pertaining to the rightmost and, usually, highest numbered position of a field or byte.

Modem. Contraction of modulator-demodulator. A device which modulates and demodulates signals transmitted over communications facilities.

Multi-point. Line or circuit interconnecting several stations.

No-op. An instruction that causes no operation to be performed. Also "no-oping" and "no-oped", causing an instruction to perform as a no-op instruction.

Op-Code. That portion of a System/3 instruction that specifies what operation is to be performed.

Operand. That which is operated upon. An operand is usually identified by an address part of an instruction.

Operation. (1) A defined action, namely the act of obtaining a result from one or more operands in accordance with a rule that completely specifies the result for any permissible combinations of operands. (2) The set of such acts specified by such a rule, or the rule itself. (3) The act specified by a single computer instruction. (4) A program step undertaken or executed by a computer, for example, addition, multiplication, extraction, comparison, shift, transfer. The operation is usually specified by the operator part of an instruction.

OR. (1) A logic operator having the property that if P is a statement, Q is a statement, R is a statement, . . ., then the OR of P, Q, R, . . ., is true if at least one statement is true, false if all statements are false.

Overflow. (1) That portion of the result of an operation that exceeds the capacity of the intended unit of storage. (2) Pertaining to the generation of overflow as in (1).

Parity Bit. A binary digit appended to an array of bits to make the sum of all the bits always odd or always even.

Parity Check. A test that checks whether the number of 1's or 0's in an array of binary digits is odd or even.

Point-to-Point Transmission. Transmission of data directly between two points without the use of any intermediate terminal or computer.

Q Byte. The second byte of a System/3 instruction, the function of which is determined by the instruction being performed.

Read. To acquire or interpret data from a storage device, a data medium, or any other source.

Recomplement. The operation of converting a complement number to its true form.

Register. A device capable of storing a specified amount of data, such as one byte.

Reset. (1) To restore a storage device to a prescribed initial state, not necessarily denoting 0. (2) To place a binary cell into the state denoting 0.

Scan. To examine sequentially, part-by-part.

Sector. An addressable portion of a magnetic disk track.

Seek. To position the access mechanism of a direct access device at a specified location.

SER-DES. Contraction of serializer-deserializer. A device that converts parallel by bit data to serial by bit data and converts serial by bit data to parallel by bit data.

Serial Printer. A device that prints characters one at a time in order across the line. A serial printer does not require that a complete line be printed for each print operation.

Subroutine. A routine that can be part of another routine.

Track. The portion of a moving storage medium such as a drum, tape or disk that is accessible to a given read head position.

Two-Wire Circuit. A metallic circuit formed by two conductors insulated from each other. It is possible to use the two conductors as a one-way transmission path, a half-duplex path, or a duplex path.

Typamatic Key. A keyboard key that performs its function repeatedly when held pressed as opposed to a keyboard key that must be released and pressed again for each function desired.

USASCII. USA Standard Code for Information Interchange. The Standard Code, using a coded character set consisting of seven-bit coded characters (eight bits including parity check), used for information interchange among data processing systems, communications systems, and associated equipment. The USASCII set consists of control characters and graphic characters.

Write. To record data in a storage device or a data medium. The recording need not be permanent, such as on a cathode ray tube display device.

Zone, Zone Portion. The high-order four bits of a byte.

Where more than one page reference is given, the major reference is first.

READER'S COMMENT FORM

IBM System/3 Model 6
Components Reference Manual

GA34-0001-3

YOUR COMMENTS, PLEASE . . .

Your comments assist us in improving the usefulness of our publications; they are an important part of the input used in preparing updates to the publications. All comments and suggestions become the property of IBM.

Please do not use this form for technical questions about the system or for requests for additional publications; this only delays the response. Instead, direct your inquiries or requests to your IBM representative or to the IBM branch office serving your locality.

Corrections or clarifications needed:

Page *Comment*

Please include your name and address in the space below if you wish a reply.

GA34-0001-3

Cut Along Line

Fold

Fold

FIRST CLASS
PERMIT NO. 387
ROCHESTER, MINN.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM Corporation
General Systems Division
Development Laboratory
Publications, Dept. 245
Rochester, Minnesota 55901

Fold

Fold

IBM

International Business Machines Corporation
General Systems Division
875 Johnson Ferry Road N. E.
Atlanta, Ga. 30342
(USA Only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017

GA34-0001-3

IBM System/3 Model 6 Components Reference Manual (File No. S3-01) Printed in U.S.A. GA34-0001-3

International Business Machines Corporation
General Systems Division
875 Johnson Ferry Road N. E.
Atlanta, Ga. 30342
(USA Only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)