SYSTEM/32

IBM System/32
Functions
Reference Manual

IBM System/32
Functions
Programming Information

GA21-9176-1
File No. S32-01

IBM System/32
Functions
Reference Manual

Preface

This reference manual is intended for persons interested in the operation and characteristics of the IBM System/32 at the machine code level. Readers should be familiar with data processing techniques and understand stored program control at the machine code level.

Similar reference manuals present mnemonics for all instructions; this manual does not include mnemonics because there is no assembler language for System/32.

Readers familiar with IBM System/3 programming may notice a similarity between many System/32 and System/3 instructions. Some instructions are identical; however, there are distinct differences in some areas. For example: in System/3, all I/O operations are initiated by issuing start I/O instructions; in System/32, some I/O instructions use the familiar hex F3 operation code and other I/O operations are initiated by branching to a location in main storage.

Related Publications

- *IBM System/32 System Control Programming Reference Manual,* GC21-7593
- *IBM System/32 Operator's Guide,* GC21-7591
- *General Information: Binary Synchronous Communications,* GA27-3004

Titles and abstracts of other related publications are listed in the *IBM System/32 Bibliography,* GC20-0032.

Second Edition (May 1975)

This is a major revision of and replaces GA21-9176-0. A chapter has been added to describe the binary synchronous communications adapter (BSCA) functions on System/32, and numerous miscellaneous changes have been made throughout the manual. Additions and changes are denoted by a vertical line at the left of the addition or change.

Changes to the information herein are made periodically; before using this publication in connection with the operation of an IBM System/32, consult the latest *IBM System/32 Bibliography,* GC20-0032, for the editions that are applicable and current.

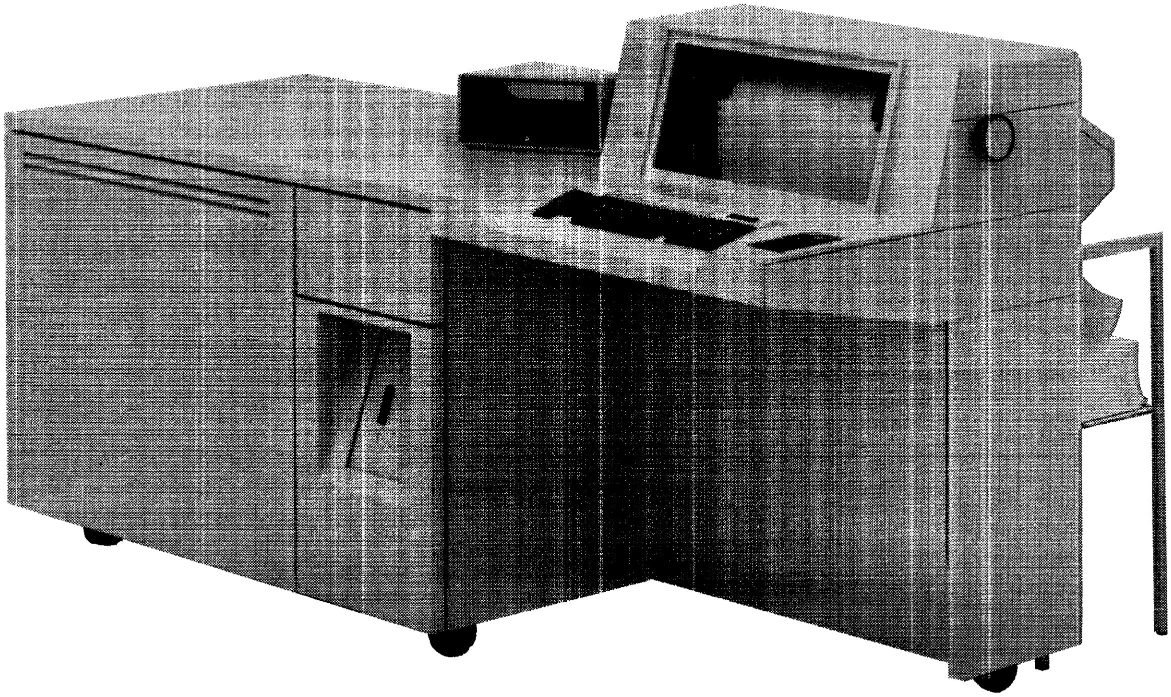
Request for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form is missing, comments may be addressed to IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901.

Contents

Abbreviations and Acronymns

µs	microseconds
AM	address mark
ARR	address recall register
BCC	block check character
BDE	basic data entry (keyboard operation)
BPS	bits per second
BSCA	binary synchronous communications adapter
CAR	control address register
CC	cylinder number bytes
CE	customer engineer
CHK	check
CHRNX	bytes in diskette control field
CRC	cyclic redundancy check
CSDE	controlled sequential data entry (keyboard operation)
DAR	data address register
DBI	data bus in
DBO	data bus out
Disp	displacement
DPLY	display
DTF	define the file
EBCDIC	extended binary coded decimal interchange code
ERMAP	error map
EWG	end write gap
FCCHS	identifier field in logical disk address
H	one of the bytes in the diskette control field
hex	hexadecimal
IAR	instruction address register
INSN	instruction
IMPL	initial microprocessor program load
IOB	input/output block
IOS	input/output system
IPL	initial program load
IRPT	interrupt
I/O	input/output
K	1,024 bytes
KEYBD	keyboard
LS	logical sector
LSR	local storage register
MAR	microprocessor address register (not used by program)
ms	millisecond
NFCCHS	bytes in disk control field
ns	nanosecond
op code	operation code
op end	operation end
op 1	operand 1
op 2	operand 2
PCR	processor condition register (not used by program)
PROC	processor
PS	physical sector
PSR	program status register
PWR	power
RDY	ready
RIB	request indicator byte
RPM	rotations per minute
S	byte in disk control field
SCP	system control program
SD	seek displacement, in tracks
SDE	sequential data entry (keyboard operation)
SDLC	synchronous data link control
SS	relative sector address
STOR	storage
SWG	start write gap
sync	synchronous, synchronization, synchronize
SYS	system (on CE panel, SYS INSN STEP means a single instruction step from your control program)
TH	thermal
TIO	test input/output instruction
VTAM	virtual telecommunications access method
XR	index register



System/32

Chapter 1. Introduction

The IBM System/32 is an operator-oriented, desk-size data processing system. The system has an operator console through which the operator enters data to the system, controls the operation of the system, and communicates with the system program. Integral parts of the console are the keyboard and display screen. The primary output unit is a system printer. Programs and data files reside on a nonremovable disk. Diskettes, which are removable, serve as a load/dump medium for creating backup files from information on the disk, as a data interchange medium for exchanging data with other systems, and as a medium for offline preparation of data and programs.

The system operates under control of programs stored in main storage and under control of a microprocessor. The microprocessor serves as a control unit and assists in control of system input and output functions. The microprocessor has a dedicated storage area called control storage.

The system overlaps input/output operations of most I/O devices with each other and with processor operations.

The models of System/32 differ in printing speeds and disk storage capacities. Figure 1-1 lists the available System/32 models. Though each model is listed with a main storage capacity of 16,384 bytes (16K – K = 1,024 bytes), main storage capacities of 24,576 bytes (24K) and 32,768 bytes (32K) are available for all models.

System Model	Main Storage Capacity	Serial Printer Speed	Line Printer Speed	Disk Data Storage Capacity
A12	16,384 bytes	40 chars/sec	– –	5,053,440 bytes
A13	16,384 bytes	40 chars/sec	– –	9,169,920 bytes
A22	16,384 bytes	80 chars/sec	– –	5,053,440 bytes
A23	16,384 bytes	80 chars/sec	– –	9,169,920 bytes
B12	16,384 bytes	– –	50 lines/min	5,053,440 bytes
B13	16,384 bytes	– –	50 lines/min	9,169,920 bytes
B22	16,384 bytes	– –	100 lines/min	5,053,440 bytes
B23	16,384 bytes	– –	100 lines/min	9,169,920 bytes
B32	16,384 bytes	– –	120 or 155 lines/min[1]	5,053,440 bytes
B33	16,384 bytes	– –	120 or 155 lines/min[1]	9,169,920 bytes

[1] Print speed is 120 lines per minute for 64-character print belts, 155 lines per minute for 48-character print belts.

Figure 1-1. System Models

ELEMENTS OF THE SYSTEM

Processor and Main Storage

Main storage holds 16,384 eight-bit data bytes. The processor, with an integrated microprocessor, provides all the arithmetic, logical, and input/output control functions for the entire system.

Keyboard

The system keyboard contains a set of alphameric keys (arranged in the standard typewriter format), a set of adding machine keys (arranged in the 10-key keyboard format), and a set of function keys used by the operator to select system functions.

Display Screen

The display screen displays data and messages. Through programming, characters can be stored in main storage as they are keyed, or up to 240 characters can be held in the display screen (and altered as required) before they are sent to main storage. Under program control or operator control, main storage data and the contents of registers can be displayed and, if desired, altered by use of the keyboard.

Printer

Depending on the model, the system has one of two different integrated printers:

- The *serial printer* uses a wire matrix to print characters serially. It operates at either 40 or 80 characters per second (maximum), depending on the model specified. The serial printer has a standard 64-character set of graphics. The standard serial printer processes both continuous forms and single form/ledger cards.
- The *line printer* operates at either 50, 100, or 155 lines per minute (maximum), depending on the model. It has standard character sets of 48 and 64 characters. The line printer processes continuous forms only.

Disk Storage

Every model has a disk drive with a nonremovable disk. Depending on the model, the disk stores either 5,053,440 or 9,169,920 bytes. Like the other units discussed, the disk drive unit is an integral part of the system.

Diskette Drive

Each model has a diskette drive unit. The system uses the IBM standard data interchange diskette (or equivalent). This allows the system to read diskettes written by IBM 3741 Data Stations and similar devices and to exchange data with other systems. Data can also be written on diskettes and stored offline for backup data and programs. Data on diskettes that will not be used on other systems need not be formatted like data on standard data interchange diskettes.

Additional Main Storage Capacity

The system can be equipped with either 8,192 or 16,384 positions of additional main storage as a special feature, bringing the total main storage capacity to either 24K or 32K bytes.

Telecommunications Feature

Each model of the system can be equipped with either the binary synchronous communications adapter (BSCA) as a special feature or with the synchronous data link control (SDLC) feature. The BSCA allows communication between System/32 and a remote system or terminal at data rates of from 600 to 7200 bits per second.

SDLC allows communication between System/32 and a System/370 operating with the virtual telecommunications access method (VTAM). Data rates can be from 600 to 7200 bits per second. Detailed information about SDLC will be contained in a future revision of this publication.

Keylock Power Switch

This feature is a key-operated switch that replaces the standard POWER ON/OFF switch on the operator panel. When the key is inserted and turned, power is supplied to the system. The feature helps restrict system operation to key-holders.

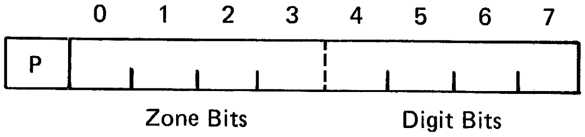
DATA FORMATS

Data resides in main storage in 8-bit (plus parity) bytes. The instruction the system is executing determines how the data is interpreted. A byte holds either decimal, alphabetic, or special characters; or binary numbers (logical data).

The system uses EBCDIC (extended binary coded decimal interchange code) for storing characters in main storage and for processing data.

Character Format

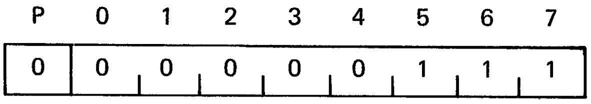
In character format, each byte of data is divided into two groups of four bits each. Bits 0-3 compose the zone portion, and bits 4-7 compose the digit portion. The character format represents a decimal digit, a special control character, or one of the characters that can be printed or displayed by the system (these characters are called *graphics*). The illustration below shows the byte as interpreted for character format.



For decimal arithmetic operations, the zone bit of the rightmost byte in the field indicates the sign of the number. (The system ignores the zone bits in all other bytes during the operation.) Zones containing hex B or D (binary 1011 or 1101) designate a negative number. Any other hex digit in the zone designates a positive number.

Binary Format (Logical Data)

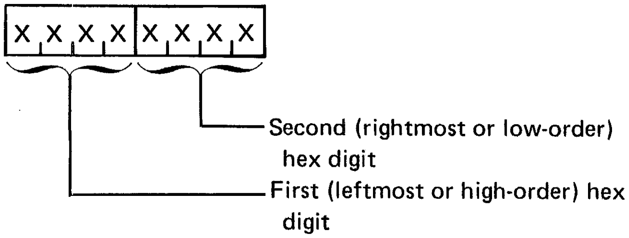
In binary format, bits in a byte define a digit, and the entire byte is an unsigned binary number (a binary integer). Binary bits are said to be on if 1, off if 0. The illustration below shows decimal 7 as a binary integer. Notice that the parity bit is set to 0 (see *Parity* in this chapter).



Unsigned Binary Integer

Hex Code

Each byte can be divided into two groups of four bits, and each of these groups can be represented as a single hex digit, as shown below:


X X X X X X X X
Second (rightmost or low-order) hex digit
First (leftmost or high-order) hex digit

The hex value of each combination of binary bits is:

Binary Bits	Hex Digit	Binary Bits	Hex Digit
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Throughout this manual, values stored in bytes are often expressed in hex notation.

Parity

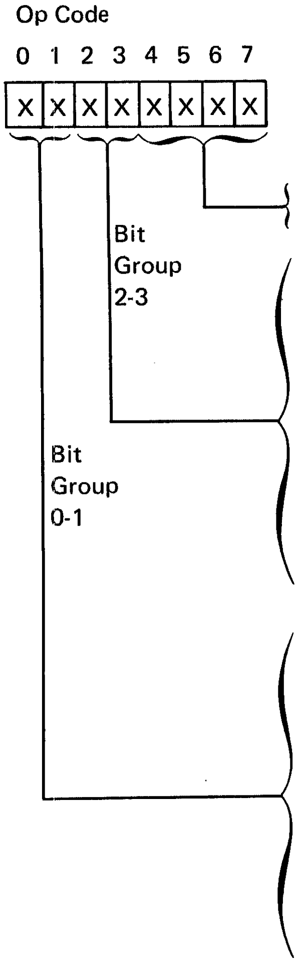
Each byte contains a parity bit that is developed by the system (and checked by the system during various operations). This bit ensures that the number of bits set to 1 in each byte is an odd number. (If the binary number or data code developed by the system has an even number of bits that are 1, the system sets the parity bit to 1 to make the entire byte contain an odd number of 1-bits. If the binary number or data code has an odd number of bits, the system sets the parity bit to 0 to maintain an odd number of bits in the entire byte.)

ADDRESSING

Main storage is addressed in binary, using hex notation. Its positions are consecutively numbered from hex 0000 to the upper limit of storage. The location of any field or group of bytes is specified by the address of the rightmost (low-order or highest-numbered storage position) byte in the field. The exception is the insert and test character instruction, which specifies the leftmost byte.

[illegible] the control address register
holds address of leftmost byte of control field
[illegible]

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 0-3 of the first byte (the operation code) of the instruction. These four bits are treated as two groups of two bits each: group 0-1 and group 2-3. Bits 0 and 1 control addressing for operand 1; bits 2 and 3 control addressing for operand 2. When bits for group 0-1 = 11, operand 1 is not used; when bits for group 2-3 = 11, operand 2 is not used. Figure 1-2 is an explanation of op code functions in addressing main storage.



Defines type of operation to be performed.

Specifies operand 2 main storage addressing activity for instruction:

00 – Operand 2 of the instruction contains a 2-byte address (direct addressing).
01 – Content of XR1 (index register 1) is added to the single byte in the operand 2 section of the instruction. The result specifies the storage location being addressed (base-displacement addressing).
10 – Content of XR2 (index register 2) is added to the single byte in the operand 2 section of the instruction. The result specifies the storage location being addressed (base-displacement addressing).
11 – Operand 2 portion of the instruction is not used to address main storage in this instruction.

Specifies operand 1 main storage addressing activity for instruction:

00 – Operand 1 of the instruction contains a 2-byte address (direct addressing).
01 – Content of XR1 is added to the single byte in the operand 1 section of the instruction. The result specifies the storage location being addressed (base-displacement addressing).
10 – Content of XR2 is added to the single byte in the operand 1 section of the instruction. The result specifies the storage location being addressed (base-displacement addressing).
11 – Operand 1 portion of the instruction is not used to address main storage in this instruction.

Note: When bits 0, 1, 2, and 3 = 1111 (hex F), the instruction is a command-type instruction and does not address main storage.

Figure 1-2. Op Code Function in Addressing Main Storage

Direct Addressing

When either or both of bit groups 0-1 or 2-3 = 00, the specified operand uses direct addressing.

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

Base-Displacement Addressing

A specified operand uses base-displacement addressing when either or both of the bit groups have one bit = 1 and the other bit = 0.

In base-displacement addressing, the contents of the 1-byte address in the instruction are added to the contents of a 2-byte address in an index register. The index register to be used is determined by the bit of the bit group that is 1 (Figure 1-2). Both bit groups can use the same index register during the execution of an instruction.

Any one value of an index register allows access to 256 storage positions.

INSTRUCTION FORMATS

Instruction formats are distinguished by their ability to address storage. The length of each instruction is determined by the type of addressing being performed.

All instruction formats have two elements in common: the op code and the Q-byte. Each of these elements is one byte. The op code determines the type of addressing (thereby the length of the instruction) and the operation to be performed. The function of the Q-byte is determined by the instruction and is discussed with each individual instruction.

Command-Type Instructions

Command-type instructions are always three bytes long. In a command-type instruction, the Q-byte contains the following information, depending on the instruction:

Device address and function specification
Jump condition

The command instruction is distinguished by bits 0-3 of the op code being all 1's.

Command Instruction

Op Code 1111	Q-Byte	Command

0 3
Bits

One-Address Instructions

One-address instructions can be either three or four bytes long. These instructions are distinguished by having either bits 0-1 or bits 2-3 of the op code byte both 1's. The two bits that are not both 1 (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, addressing is base-displacement; the instruction is three bytes long; and XR1 (01) or XR2 (10) is used. The Q-byte of a one-address instruction can contain:

An immediate operand
A mask
A branch condition
A data selection

One-Address Instruction—Base-Displacement Addressing

Op Code 1110 1101 1011 0111	Q-Byte	Displacement Operand

0 3
Bits

One-Address Instruction—Direct Addressing

Op Code 0011 1100	Q-Byte	(High-Order Byte of Address) Operand	(Low-Order Byte of Address) Operand

0 3
Bits

Two-Address Instructions

Two-address instructions can be four, five, or six bytes long. This instruction type is distinctive in that *neither* bit group 0-1 *nor* bit group 2-3 of the op code byte are 1's. If all four of bits 0-3 are 0's, addressing is direct, and the instruction is six bytes long. If any *one* of bits 0-3 is 1, one of the addresses is direct; the other address is base-displacement, and the instruction is five bytes long. If one bit from each of the bit groups is 1, all addressing is base-displacement 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 = 01, XR1 is used; if the bit group = 10, XR2 is used. Both addresses can use the same index register during one instruction.

Two-Address Instruction—Both Addresses Base Displacement

Op Code 0101 0110 1001 1010	Q-Byte	Operand 1 Displacement	Operand 2 Displacement

0 3
Bits

Two-Address Instruction—Operand 1 Address Direct

Op Code 0001 0010	Q-Byte	Operand 1 (High Order Address Byte)	Operand 1 (Low-Order Address Byte)	Operand 2 Displacement

0 3
Bits

Two-Address Instruction—Operand 2 Address Direct

Op Code 0100 1000	Q-Byte	Operand 1 Displacement	Operand 2 (High-Order Address Byte)	Operand 2 (Low-Order Address Byte)

0 3
Bits

Two-Address Instruction—Both Address Direct

Op Code 0000	Q-Byte	Operand 1 (High-Order Address Byte)	Operand 1 (Low-Order Address Byte)	Operand 2 (High-Order Address Byte)	Operand 2 (Low-Order Address Byte)

0 3
Bits

Chapter 2. Processor and Operator Controls

The processor controls the flow of input into the system, performs the operations on the data, and controls the output from the system.

MODES OF SYSTEM OPERATION

The system operates in three modes: burst, interrupt, and process.

Burst Mode

The system operates in *burst mode* while it transfers data between main storage and the disk. In burst mode the system provides a dedicated data path and, once data transfer starts, data moves rapidly between the disk and main storage until all the specified data has been transferred.

Interrupt Mode

At the end of most input and output operations, the microprocessor is signaled that the operation has ended and that the program should branch to a special interrupt handler routine. While the system is processing data in the interrupt routine, it is said to be operating in the *interrupt mode.*

Process Mode

The system is free to handle normal I/O control and data processing operations when it is not operating in either the burst mode or interrupt mode. At this time the system operates in *process mode.*

INSTRUCTION REGISTERS

Instruction Address Register (IAR)

The instruction address register holds the address of the first byte of the next sequential instruction in the stored program.

Address Recall Register (ARR)

← Also the J-register of Knuth's MIX

The system places the next sequential address (that is, the address of the instruction that follows the test-and-branch or branch-on-condition instruction in the program) in the address recall register whenever the program branches. At the end of the branch routine, the program can load the contents of the address recall register into the instruction address register; this returns the program to the point at which the branch occurred.

The address recall register is also affected by load-register, add-to-register, decimal-add-and-subtract, and insert-and-test-characters instructions. (All instructions are described in Chapter 3.)

Index Registers 1 and 2 (XR1 and XR2)

Each of these index registers holds a base address for base-displacement addressing.

INTERRUPT OPERATIONS

Certain I/O functions require special routines to handle data within a limited period of time. To provide for these special routines, the system uses interrupts. Generally, an interrupt implies that the processor must interrupt a current instruction sequence; perform an intervening instruction sequence requested by the interrupting keyboard, disk drive, printer, or BSCA; then return to the interrupted program. The interrupts for the disk drive, printer, and BSCA are handled entirely by the hardware; the program must provide interrupt handler routines for the keyboard.

An interrupt can occur for the keyboard unless the program disables the interrupt function. If the program disables an interrupt, it remains disabled until the program again enables it or until an initial program load (IPL) routine is performed.

Interrupts are discussed in *General I/O Operations* in this chapter.

INPUT/OUTPUT BLOCKS (IOBs)

Each input and output function has certain parameters that the program must define before the operation is performed. For some functions, the parameters are loaded into registers; for others, the parameters are moved into IOBs (that is, input/output blocks, which are consecutive main storage positions into which parameters are placed in defined fields).

Whenever an IOB is used to define the operation, the program must present the address of the leftmost byte of the IOB to the system (in XR1) at the start of the operation.

If an IOB is required for a function, this manual describes that IOB in the chapter that discusses the function.

GENERAL I/O OPERATIONS

The following paragraphs briefly discuss how each I/O unit functions with the user program.

Disk

The program sets up a disk IOB and branches to location 0008 (via a start-disk IOB instruction). The system performs its own queueing and dequeueing functions. The program need not issue any load, sense, or test instructions.

Printer

The program sets up a printer IOB; loads the character set size and image (line printer only) and the printer data field address; and places the IOB on the system queue.

The program issues a start-printer IOB instruction with XR1 pointing to the IOB. (At the end of the operation, the system automatically removes the IOB from the queue.)

Keyboard/Display Screen

The program sets up a keyboard/display screen IOB and loads the address of the IOB. For keyboard only, the program queues the keyboard IOB with XR1 pointing to the IOB address.

The program issues various instructions which enable and disable the keyboard and control the image on the display screen. The program also supplies an interrupt handler, which dequeues the IOB and disables the keyboard.

Diskette

The program loads the diskette control code from disk. (There is no IOB interface for the diskette.) The program loads the diskette data field address and diskette control field address and issues a start-diskette instruction to perform the function. The program also does its own sensing and testing.

BSCA

The program loads the BSCA control code for EBCDIC or ASCII mode and places a pseudo-IOB (three unexamined bytes – there is no true IOB interface) on the system queue with XR1 pointing to the IOB. The program also loads the current address register, transition address register, stop address register, and interrupt address register as required; loads the unit definition table; and enables and disables BSCA interrupts.

The program issues the BSCA control instruction to enable/disable the BSCA and/or the 2-second timeout, and issues the appropriate start instruction. The program senses the BSCA status bytes and responds to the status bit settings, and also senses the current address register. When the BSCA is disabled, the program removes the BSCA IOB from the system queue.

SYSTEM CONTROLS

The System/32 operator controls are situated on the operator's console (Figure 2-1); the CE controls are situated on the CE panel (Figure 2-2).

The operator's console consists of the keyboard/display screen and the operator processor controls. The CE panel serves as a diagnostic aid for locating machine and program malfunctions.

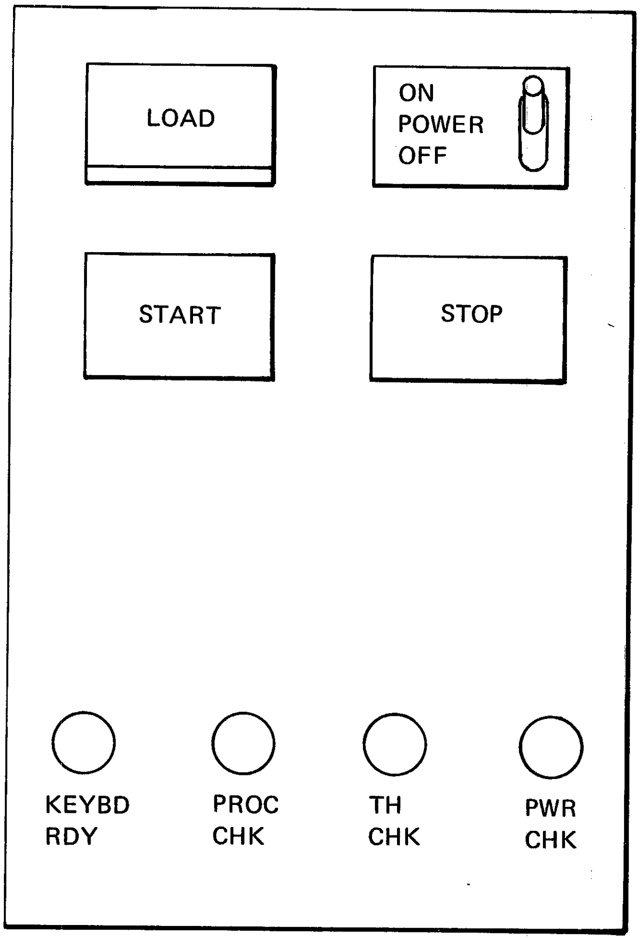


Figure 2-1. Operator's Console

Operator's Console

LOAD Key/Light

This is a combination key and light which resets the system, turns the LOAD light on, and turns on the nine event indicators on the CE panel. Releasing the key initiates the IPL routine, during which the system reads the initial program into storage from either the diskette or the disk, depending on the setting of the IPL switch on the CE panel. During an IPL routine, the LOAD light turns off.

Note: The LOAD key is always effective, and resets the system any time it is pressed. Before performing an IPL, set the MODE SELECTOR switch (on the CE panel) to PROC RUN and the IPL switch (on the CE panel) to the appropriate setting.

POWER ON/OFF Switch

Setting this switch to on turns on system power and resets the system. After about 10 seconds, the STOP light comes on, indicating that the initial program can be loaded. Setting the power switch to off turns off system power.

CAUTION
Removing power from the system destroys data stored in main storage.

START Key/Light

This is a combination key and light. Pressing START starts the system if it is stopped because:

- The STOP key was pressed.
- The MODE SELECTOR switch is in the SYS INSN STEP position.
- A main storage address compare stop has occurred.

The START light comes on during IPL procedures and stays on until:

- A disable-start-light instruction is issued by the program. The START light can be turned back on by an enable-start-light instruction followed by an I/O operation.
- The STOP key is pressed. The START light can be turned on again by pressing the START key.
- A main storage address compare stop condition occurs. The START light can be turned on again by pressing the START key.
- The MODE SELECTOR switch is in the SYS INSN STEP position. The START light can be turned on again by setting the MODE SELECTOR switch to PROC RUN and pressing the CE START key (on CE panel).

STOP Key/Light

This is a combination key and light that stops the system at the end of the operation in progress. The STOP light comes on at this time. The STOP light also comes on at the end of each power-up sequence (when the system is ready for the IPL operation) and during instruction-step and address-step operations. Pressing the LOAD key or the START key turns the STOP light off.

KEYBD RDY Light

This light indicates that the keyboard is enabled and is ready for operation. (The INQ, LINE, PAGE, PRINT, RESET, and ERROR RESET keys on the keyboard are always enabled.)

PROC CHK Light

This light indicates that an unrecoverable error (one requiring operator or CE attention) has occurred. Restart from an unrecoverable error requires that the initial program be loaded again.

TH CHK Light

If one of the system thermal sensors detects an over-temperature condition, the system turns off system power and turns on the thermal check light. The light remains on until the over-temperature condition is corrected and the POWER switch is turned off. Power can then be restored to the system by turning the POWER switch on.

PWR CHK Light

This light indicates that the system has shut off power because there is a problem in the power circuits. If this light is on, notify the IBM customer engineer. The CE panel display indicates which power level failed.

CE Panel

Although this is called a CE panel, some of the switches on the panel are used by the operator and the programmer. These switches are described in this section. CE panel switches not described in this section are used only by the IBM customer engineer.

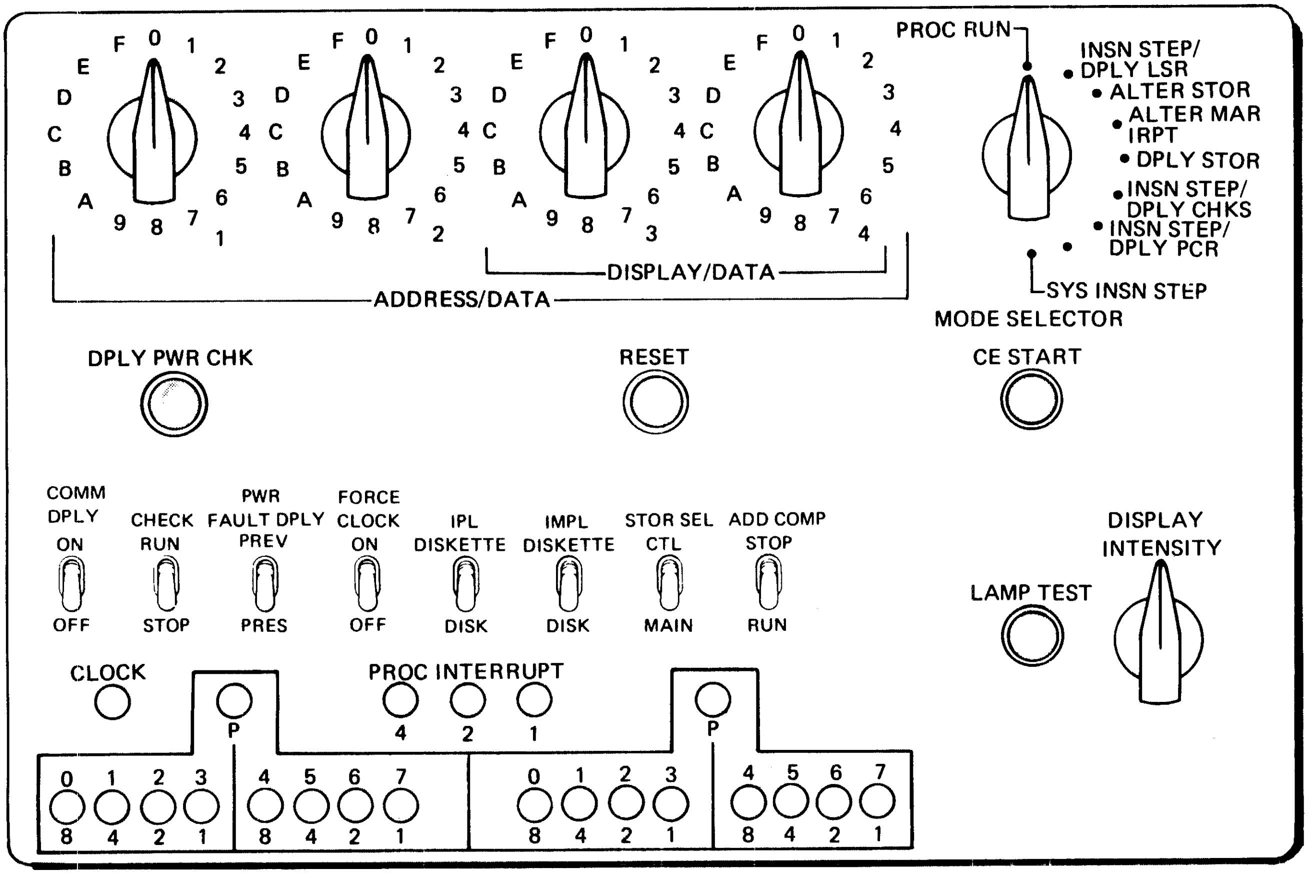


Figure 2-2. CE Panel

ADDRESS/DATA Switches

These four 16-position rotary switches are used to enter, alter, or display data stored in main storage or local storage registers when they are used in conjunction with other switches on the CE panel. (See *Manual Operating Procedures* in Chapter 7.)

Console Display Indicators

This group of lights displays the contents of certain system registers and presents system status information. The MODE SELECTOR switch selects the type of information to be displayed.

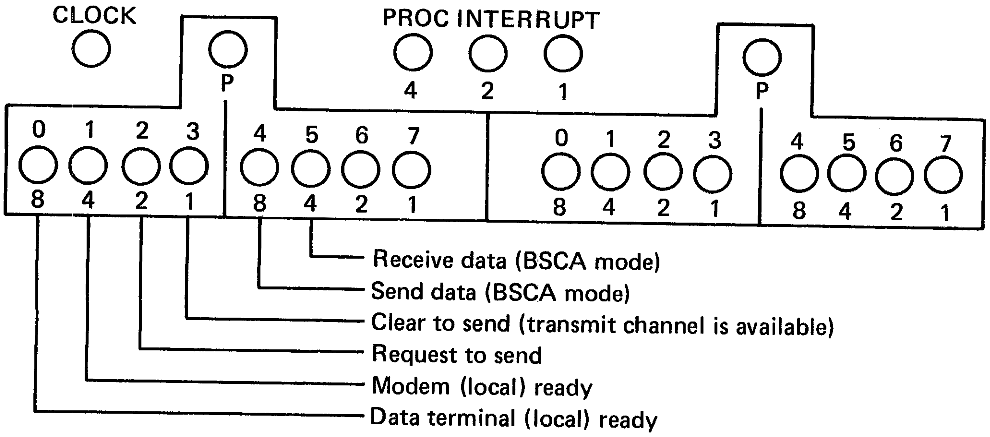
MODE SELECTOR Switch

During regular processing operations, the MODE SELECTOR switch must be set to the PROC RUN position. All other positions are associated with diagnostic procedures, and are specified in those procedures. (See *Manual Operating Procedures* in Chapter 7.)

Toggle Switches

Except for the COMM DPLY (communications display) switch, which can be set on when running BSCA programs, the toggle switches should all be set to the lower settings during normal processing operations.

The COMM DPLY switch is on the panel if the BSCA is installed on the system. When set at the ON position, the switch activates the leftmost six lights at the bottom of the CE panel. The lights indicate the following about the BSCA when the lights are on:



LAMP TEST Switch

Pressing the LAMP TEST switch tests all the electric bulbs on the CE panel and operator's console. If one of them does not light, notify the IBM customer engineer.

DISPLAY INTENSITY Control

This control regulates the intensity of the image on the display screen.

Chapter 3. Instructions

Each System/32 instruction is described here in detail. The instructions are organized into four groups:

- Arithmetic
- Data handling
- Logical
- Input/output handling

Arithmetic Instructions

ZERO AND ADD ZONED

Op Code (hex)	Q-Byte[1] (hex)	Operand Addresses[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
04	L1-L2 | L2-1	Operand 1 address		Operand 2 address	
14	L1-L2 | L2-1	Operand 1 address		Op 2 disp from XR1	
24	L1-L2 | L2-1	Operand 1 address		Op 2 disp from XR2	
44	L1-L2 | L2-1	Op 1 disp from XR1	Operand 2 address		
54	L1-L2 | L2-1	Op 1 disp from XR1	Op 2 disp from XR1		
64	L1-L2 | L2-1	Op 1 disp from XR1	Op 2 disp from XR2		
84	L1-L2 | L2-1	Op 1 disp from XR2	Operand 2 address		
94	L1-L2 | L2-1	Op 1 disp from XR2	Op 2 disp from XR1		
A4	L1-L2 | L2-1	Op 1 disp from XR2	Op 2 disp from XR2		

[1] The Q-byte designates operand length:
L1-L2 (4 bits) = the number of bytes in operand 1 minus the number of bytes in operand 2.
L2-1 (4 bits) = the number of bytes in operand 2 minus 1.
Maximum length of operand 1 is 31 bytes; maximum length of operand 2 is 16 bytes.

[2] The operands may overlap. Address operands by their rightmost bytes.

Operation

This instruction moves data from the second operand, byte by byte starting with the rightmost byte, into the first operand. If the first operand is longer than the second operand, the processing unit fills the extra positions with high-order decimal zeros (hex F0).

The processing unit sets the zone bits of all bytes except the rightmost byte in the first operand to hex F (binary 1111). It sets the zone bits of the rightmost byte in the first operand to (1) hex F if the value transferred is either zero or positive, or (2) hex D (binary 1101) if the value transferred is negative.

Program Note

The second operand remains unchanged unless the fields overlap.

CAUTION

Overlapping the operands with the rightmost byte of the first operand to the left of the rightmost byte of the second destroys part of the second operand before it is used in the operation.

Resulting Program Status Byte Settings

Bit	Name	Condition Indicated
7	Equal	Zero result
6	Low	Negative result
5	High	Positive result
4	Decimal overflow	Bit not affected
3	Test false	Bit not affected
2	Binary overflow	Bit not affected

Example

Instruction

04	22	00	10	00	20

Operand 1 Before Operation

	F7	F6	F3	F6	F9	
000C	000D	000E	000F	0010	← Storage Positions	

Operand 2 Before and After Operation

			F4	F2	F5	
			001E	001F	0020	← Storage Positions

Operand 1 After Operation

	F0	F0	F4	F2	F5	
000C	000D	000E	000F	0010	← Storage Positions	

Program Status Register Before Operation

00000000

0 7 ← Bits

Program Status Register After Operation

00000100

0 7 ← Bits

ADD ZONED DECIMAL

Op Code (hex)	Q-Byte[1] (hex)	Operand Addresses[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
06	L1-L2 L2-1	Operand 1 address		Operand 2 address	
16	L1-L2 L2-1	Operand 1 address		Op 2 disp from XR1	
26	L1-L2 L2-1	Operand 1 address		Op 2 disp from XR2	
46	L1-L2 L2-1	Op 1 disp from XR1	Operand 2 address		
56	L1-L2 L2-1	Op 1 disp from XR1	Op 2 disp from XR1		
66	L1-L2 L2-1	Op 1 disp from XR1	Op 2 disp from XR2		
86	L1-L2 L2-1	Op 1 disp	Operand 2 address		
96	L1-L2 L2-1	Op 1 disp from XR2	Op 2 disp from XR1		
A6	L1-L2 L2-1	Op 1 disp from XR2	Op 2 disp from XR2		

[1] The Q-byte designates the operand length:
L1-L2 (4 bits) = the number of bytes in operand 1 minus the number of bytes in operand 2.
L2-1 (4 bits) = the number of bytes in operand 2 minus 1.
Maximum length of operand 1 is 31 bytes; maximum length of operand 2 is 16 bytes.

[2] The operands may overlap. Address operands by their rightmost bytes.

Operation

This instruction algebraically adds the second operand to the first operand and stores the result in the first operand.

The processing unit sets the zone bits of all bytes except the rightmost byte in the first operand to hex F (binary 1111). It sets the zone bits of the rightmost byte in the first operand to (1) hex F if the result of the operation is either positive or zero, or (2) hex D if the result is negative.

Program Notes

- The second operand remains unchanged unless the fields overlap.

  CAUTION
  Overlapping the operands with the rightmost byte of the first operand to the left of the rightmost byte of the second operand destroys part of the second operand before it is used in the operation.

- The system does not check for valid decimal digits in either operand.

- The decimal overflow condition indicator, which may be set during this operation, is reset by:

  – A system reset
  – Testing decimal overflow with a branch-on-condition or jump-on-condition instruction
  – Loading a 0 in bit 4 of the program status register using the load-register instruction

- The system saves the starting address of operand 1 in the address recall register.

Resulting Program Status Byte Settings

Bit	Name	Condition Indicated
7	Equal	Zero result
6	Low	Negative result
5	High	Positive result
4	Decimal overflow	Carry occurred from the leftmost position of operand 1
3	Test false	Bit not affected
2	Binary overflow	Bit not affected

Example

Instruction

06	22	00	10	00	20

Operand 1 Before Operation

	F7	F6	F3	F6	F9	
	000C	000D	000E	000F	0010	← Storage Positions

Operand 2 Before and After Operation

			F4	F2	F5	
			001E	001F	0020	← Storage Positions

Operand 1 After Operation

	F7	F6	F7	F9	F4	
	000C	000D	000E	000F	0010	← Storage Positions

Program Status Register Before Operation

00000001		
0	7	← Bits

Program Status Register After Operation

00000100		
0	7	← Bits

SUBTRACT ZONED DECIMAL

Op Code (hex)	Q-Byte[1] (hex)		Operand Addresses[2] (hex)			
Byte 1	Byte 2		Byte 3	Byte 4	Byte 5	Byte 6
07	L1-L2	L2-1	Operand 1 address		Operand 2 address	
17	L1-L2	L2-1	Operand 1 address		Op 2 disp from XR1	
27	L1-L2	L2-1	Operand 1 address		Op 2 disp from XR2	
47	L1-L2	L2-1	Op 1 disp from XR1	Operand 2 address		
57	L1-L2	L2-1	Op 1 disp from XR1	Op 2 disp from XR1		
67	L1-L2	L2-1	Op 1 disp from XR1	Op 2 disp from XR2		
87	L1-L2	L2-1	Op 1 disp from XR2	Operand 2 address		
97	L1-L2	L2-1	Op 1 disp from XR2	Op 2 disp from XR1		
A7	L1-L2	L2-1	Op 1 disp from XR2	Op 2 disp from XR2		

[1] The Q-byte designates the operand length:
L1-L2 (4 bits) = the number of bytes in operand 1 minus the number of bytes in operand 2.
L2-1 (4 bits) = the number of bytes in operand 2 minus 1.
Maximum length of operand 1 is 31 bytes; maximum length of operand 2 is 16 bytes.

[2] The operands may overlap. Address operands by their rightmost bytes.

Operation

This instruction algebraically subtracts operand 2 from operand 1, byte by byte, and stores the result in operand 1. The processing unit sets the zone bits of all operand 1 bytes except the rightmost byte to hex F (binary 1111). It sets the zone bits of the rightmost byte in operand 1 to (1) hex F if the result of the operation is either positive or 0, or (2) hex D (binary 1101) if the result is negative.

Program Notes

- The second operand remains unchanged unless the fields overlap.

CAUTION
Overlapping the operands with the rightmost byte of the first operand to the left of the rightmost byte of the second operand destroys part of the second operand before it is used in the operation.

- The system does not check for valid decimal digits in either operand.
- The decimal-overflow-condition indication, which may be set during this operation, can be reset by:
  - A system reset
  - Testing decimal overflow with a branch-on-condition or jump-on-condition instruction
  - Loading a 0 in bit 4 of the program status register using the load-register instruction
- The system saves the starting address of operand 1 in the address recall register.

Resulting Program Status Byte Settings

Bit	Name	Condition Indicated
7	Equal	Zero result
6	Low	Negative result
5	High	Positive result
4	Decimal overflow	Carry occurred from the left-most position of operand 1
3	Test false	Bit not affected
2	Binary overflow	Bit not affected

Example

Instruction

07	22	00	10	00	20

Operand 1 Before Operation

	F7	F6	F3	F6	F9	
	000C	000D	000E	000F	0010	← Storage Positions

Operand 2 Before and After Operation

			F4	F2	F5	
			001E	001F	0020	← Storage Positions

Operand 1 After Operation

	F7	F5	F9	F4	F4	
	000C	000D	000E	000F	0010	← Storage Positions

Program Status Register Before Operation

00000001

0 7← Bits

Program Status Register After Operation

00000100

0 7← Bits

ADD LOGICAL CHARACTERS

Op Code (hex)	Q-Byte[1] (hex)	Operand Addresses[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
0E	L-1	Operand 1 address		Operand 2 address	
1E	L-1	Operand 1 address		Op 2 disp from XR1	
2E	L-1	Operand 1 address		Op 2 disp from XR2	
4E	L-1	Op 1 disp from XR1	Operand 2 address		
5E	L-1	Op 1 disp from XR1	Op 2 disp from XR1		
6E	L-1	Op 1 disp from XR1	Op 2 disp from XR2		
8E	L-1	Op 1 disp	Operand 2 address		
9E	L-1	Op 1 disp from XR2	Op 2 disp from XR1		
AE	L-1	Op 1 disp from XR2	Op 2 disp from XR2		

[1] The Q-byte designates the operand length:
L-1 = the number of bytes in either operand minus 1.
Maximum length of each operand is 256 bytes; both operands must be the same length.

[2] The operands may overlap. Address operands by their rightmost bytes.

Operation

This instruction adds the binary number in operand 2 to the binary number in operand 1 and stores the result in operand 1.

Program Notes

- **CAUTION**
  Overlapping the operands with the rightmost byte of the first operand to the left of the rightmost byte of the second operand destroys part of the second operand before it is used in the operation.

- The system resets the binary-overflow bit during this operation.

Resulting Program Status Byte Settings

Bit	Name	Condition Indicated
7	Equal	Zero result
6	Low	No carry occurred from the high-order byte and result not zero
5	High	Carry occurred from the high-order byte and result not zero
4	Decimal overflow	Bit not affected
3	Test false	Bit not affected
2	Binary overflow	Carry occurred from the high-order byte

Example

Instruction

5E	03	00	10

Note: Index register 1 = 0CC0

Operand 1 Before Operation

		00110101	11001011	11101101	01100100	
		0CBD	0CBE	0CBF	0CC0	← Storage Positions

Operand 2 Before and After Operation

		01011011	01010101	01111000	11001101	
		0CCD	0CCE	0CCF	0CD0	← Storage Positions

Operand 1 After Operation

		10010001	00100001	01100110	00110001	
		0CBD	0CBE	0CBF	0CC0	← Storage Positions

Program Status Register Before Operation

00000001

0 7 ← Bits

Program Status Register After Operation

00000010

0 7 ← Bits

SUBTRACT LOGICAL CHARACTERS

Op Code (hex)	Q-Byte[1] (hex)	Operand Addresses[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
0F	L-1	Operand 1 address		Operand 2 address	
1F	L-1	Operand 1 address		Op 2 disp from XR1	
2F	L-1	Operand 1 address		Op 2 disp from XR2	
4F	L-1	Op 1 disp from XR1	Operand 2 address		
5F	L-1	Op 1 disp from XR1	Op 2 disp from XR1		
6F	L-1	Op 1 disp from XR1	Op 2 disp from XR2		
8F	L-1	Op 1 disp from XR2	Operand 2 address		
9F	L-1	Op 1 disp from XR2	Op 2 disp from XR1		
AF	L-1	Op 1 disp from XR2	Op 2 disp from XR2		

[1] The Q-byte designates the operand length:
L-1 = the number of bytes in either operand minus 1.
Maximum length of an operand is 256 bytes; both operands must be the same length.

[2] The operands may overlap. Address operands by their rightmost bytes.

Operation

This instruction subtracts the binary number in operand 2 from the binary number in operand 1 and stores the result in operand 1. If the number stored in the second operand is larger than the number stored in the first operand, the answer develops as though the first operand has 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

Program Note

CAUTION
Overlapping the operands with the rightmost byte of the first operand to the left of the rightmost byte of the second operand destroys part of the second operand before it is used in the operation.

Resulting Program Status Byte Settings

Bit	Name	Condition Indicated
7	Equal	Zero result
6	Low	First operand smaller than second operand
5	High	First operand greater than second operand
4	Decimal overflow	Bit not affected
3	Test false	Bit not affected
2	Binary overflow	Bit not affected

Example

Instruction

AF	03	00	10

Note: Index register 2 = 0CC0

Operand 1 Before Operation

		10010110	01011010	01110111	10111111	
		0CBD	0CBE	0CBF	0CC0	← Storage Positions

Operand 2 Before and After Operation

		01110100	10000110	01100010	10100100	
		0CCD	0CCE	0CCF	0CD0	← Storage Positions

Operand 1 After Operation

		00100001	11010100	00010101	00011011	
		0CBD	0CBE	0CBF	0CC0	← Storage Positions

Program Status Register Before Operation

00000001

0 7← Bits

Program Status Register After Operation

00000100

0 7← Bits

ADD TO REGISTER

Op Code (hex)	Q-Byte[1] (binary)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
36	Rx	Operand 1 address	
76	Rx	Op 1 disp from XR1	
B6	Rx	Op 1 disp from XR2	

[1] Rx specifies the register whose contents are modified by the instruction.
[2] Operand 1 is a 2-byte field addressed by its rightmost byte; operand 2 is not used.

Operation

This instruction adds the binary number in operand 1 to the contents of the 2-byte register selected by the Q-byte and stores the result in the register. The Q-bytes used to specify various registers are:

Q-Byte Binary	Hex	Register Specified
0000 0000	00	None. The system ignores (no-ops) the instruction.
0000 0001	01	XR1.
0000 0010	02	XR2.
0000 0100	04	Program status register.
0000 1000	08	Address recall register.
0001 0000	10	Instruction address register.
0010 0000	20	Instruction address register.
0100 0000	40	None. The system ignores (no-ops) the instruction.
1000 0000	80	None. The system ignores (no-ops) the instruction.

Program Notes

- This instruction is used to modify the contents of only one register at a time.
- This instruction does not alter the operand.
- Subtract from the register by placing the 2s complement of the number to be subtracted in the operand.

Resulting Program Status Byte Settings

Bit	Name	Condition Indicated
7	Equal	Zero result
6	Low	No carry occurred from the leftmost byte and result not zero
5	High	Carry occurred from the leftmost byte and result not zero
4	Decimal overflow	Bit not used
3	Test false	Bit not used
2	Binary overflow	Carry occurred from the leftmost byte

Example

Instruction

36	00000010	00	04

Operand 1

01001000	00100000	
0003	0004	← Storage Positions

Index Register 2 Before Operation

00110101	01101010

Index Register 2 After Operation

01111101	10001010

Program Status Byte After Operation

00000010

0 7 ← Bits

MOVE HEXADECIMAL CHARACTER

Op Code (hex)	Q-Byte[1] (binary)	Operand Addresses[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
08	I	Operand 1 address		Operand 2 address	
18	I	Operand 1 address		Op 2 disp from XR1	
28	I	Operand 1 address		Op 2 disp from XR2	
48	I	Op 1 disp from XR1	Operand 2 address		
58	I	Op 1 disp from XR1	Op 2 disp from XR1		
68	I	Op 1 disp from XR1	Op 2 disp from XR2		
88	I	Op 1 disp	Operand 2 address		
98	I	Op 1 disp from XR2	Op 2 disp from XR1		
A8	I	Op 1 disp from XR2	Op 2 disp from XR2		

[1] I = one byte of immediate data that specifies which portion of each single-byte operand is used in the operation.

[2] Both operands are single-byte fields.

Operation

This instruction moves the numeric portion (bits 4-7) or the zone portion (bits 0-3) of the second operand to the numeric or zone portion of the first operand, as specified by the Q-byte. Q-byte coding is:

Hex	Binary	Meaning
00	0000 0000	Move data from operand 2 zone portion to operand 1 zone portion
01	0000 0001	Move data from operand 2 numeric portion to operand 1 zone portion
02	0000 0010	Move data from operand 2 zone portion to operand 1 numeric portion
03	0000 0011	Move data from operand 2 numeric portion to operand 1 numeric portion

Program Notes

- The six leftmost binary bits in the Q-byte immediate data should be 0's.
- The second operand is not changed unless the same byte is used for both operands.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

Example

Instruction

98	01	A0	65

Index register 1 = 2B15
Index register 2 = 1F20

Operand 1 Before Operation

2F

1FC0 ← Storage Position

Operand 2 Before and After Operation

4C

2B7A ← Storage Position

Operand 1 After Operation

CF

1FC0 ← Storage Position

MOVE CHARACTERS

Op Code (hex)	Q-Byte[1] (hex)	Operand Addresses[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
0C	L-1	Operand 1 address		Operand 2 address	
1C	L-1	Operand 1 address		Op 2 disp from XR1	
2C	L-1	Operand 1 address		Op 2 disp from XR2	
4C	L-1	Op 1 disp from XR1	Operand 2 address		
5C	L-1	Op 1 disp from XR1	Op 2 disp from XR1		
6C	L-1	Op 1 disp from XR1	Op 2 disp from XR2		
8C	L-1	Op 1 disp from XR2	Operand 2 address		
9C	L-1	Op 1 disp from XR2	Op 2 disp from XR1		
AC	L-1	Op 1 disp from XR2	Op 2 disp from XR2		

[1] The Q-byte designates the operand length:
L-1 = the number of bytes in either operand minus 1.
Maximum length of each operand is 256 bytes; both operands must be the same length.

[2] The operands may overlap. Address operands by their rightmost byte.

Operation

This instruction places the contents of operand 2, byte by byte, into operand 1. It is possible to propagate one character through an entire field by setting the operand 2 address one byte to the right of the operand 1 address.

Program Note

The second operand remains unchanged unless the fields overlap.

CAUTION

Overlapping the operands with the rightmost byte of the first operand to the left of the rightmost byte of the second operand destroys part of the second operand before it is used in the operation.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

Example:

Instruction

0C	05	1A	06	2B	5A

Operand 1 Before Operation

D1	C1	D4	C5	E2		
1A01	1A02	1A03	1A04	1A05	1A06	← Storage Positions

Operand 2 Before Operation

D9	D6	C2	C5	D9	E3	
2B55	2B56	2B57	2B58	2B59	2B5A	← Storage Positions

Operand 1 After Operation

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

MOVE INVERSE (NOT USED IN U.S.A.)

Op Code (hex)	Q-Byte[1] (hex)	Operand Addresses[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
00	L-1	Operand 1 address		Operand 2 address	
10	L-1	Operand 1 address		Op 2 disp from XR1	
20	L-1	Operand 1 address		Op 2 disp from XR2	
40	L-1	Op 1 disp from XR1	Operand 2 address		
50	L-1	Op 1 disp from XR1	Op 2 disp from XR1		
60	L-1	Op 1 disp from XR1	Op 2 disp from XR2		
80	L-1	Op 1 disp	Operand 2 address		
90	L-1	Op 1 disp from XR2	Op 2 disp from XR1		
A0	L-1	Op 1 disp from XR2	Op 2 disp from XR2		

[1] The Q-byte designates the operand length:
L-1 = the number of bytes in either operand minus 1.
Maximum length of each operand is 256 bytes; both operands must be the same length.

[2] The operands may overlap. Address operand 1 by its leftmost byte; operand 2 by its rightmost byte.

Operation

This instruction places the contents of operand 2, reversed byte by byte, into operand 1. Operand 2 remains unchanged unless the fields overlap. Note that operand 1 is addressed by its leftmost byte, while operand 2 is addressed by its rightmost byte.

Program Note

The move-inverse instruction is not used in IBM programming support in the United States.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

Example

Instruction

00	04	1A	05	2B	59

Operand 1 Before Operation

D1	D9	E6	C5	C2	
1A05	1A06	1A07	1A08	1A09	← Storage Positions

Operand 2 Before and After Operation

F1	F2	C1	C2	D1	
2B55	2B56	2B57	2B58	2B59	← Storage Positions

Operand 1 After Operation

D1	C2	C1	F2	F1	
1A05	1A06	1A07	1A08	1A09	← Storage Positions

EDIT

Op Code (hex)	Q-Byte[1] (hex)	Operand Addresses[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
0A	L1-1	Operand 1 address		Operand 2 address	
1A	L1-1	Operand 1 address		Op 2 disp from XR1	
2A	L1-1	Operand 1 address		Op 2 disp from XR2	
4A	L1-1	Op 1 disp from XR1	Operand 2 address		
5A	L1-1	Op 1 disp from XR1	Op 2 disp from XR1		
6A	L1-1	Op 1 disp from XR1	Op 2 disp from XR2		
8A	L1-1	Op 1 disp from XR2	Operand 2 address		
9A	L1-1	Op 1 disp from XR2	Op 2 disp from XR1		
AA	L1-1	Op 1 disp from XR2	Op 2 disp from XR2		

[1] The Q-byte designates the operand length:
L1-1 = the number of bytes in operand 1 minus 1.
Operand 2 contains the number of bytes in which there are hex 20s in operand 1.

[2] The operands may overlap. Address operands by their rightmost bytes.

Operation

This instruction replaces bytes containing hex 20 in operand 1 with characters from operand 2. Starting at the rightmost position in both operands, the processor examines operand 1 for hex 20s. When the system finds the first hex 20, it moves the first byte from operand 2 into that hex 20 location, then examines the following bytes in operand 1 for the next sequential hex 20. Locating the second hex 20, the system moves the second byte from operand 2 into that operand 1 position. The operation continues until the last byte in operand 1 has been examined for hex 20. During the operation, the system sets the zone bits of all replaced operand-1 bytes to hex F (binary 1111).

Program Note

Operand 2 remains unchanged during this instruction.

Resulting Program Status Byte Settings

Bit	Name	Condition Indicated
7	Equal	Second operand zero
6	Low	Second operand negative
5	High	Second operand positive
4	Decimal overflow	Bit not affected
3	Test false	Bit not affected
2	Binary overflow	Bit not affected

Example

Instruction

0A	0A	00	BF	00	07

Operand 1 Before Operation

$	20	,	20	20	20	.	20	20	ƀ	*	
00B5	00B6	00B7	00B8	00B9	00BA	00BB	00BC	00BD	00BE	00BF	←Storage Positions

Operand 2 Before and After Operation

0	1	0	8	0	R	
0002	0003	0004	0005	0006	0007	←Storage Positions

Note: R represents hex D9 (-9)

Operand 1 After Operation

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

Note: Storage position 00BD contains a 9 because the zone bits of all replaced characters in the edit pattern are set to hex F (binary 1111).

Program Status Byte After Operation

00000010

0 7←Bits

INSERT AND TEST CHARACTERS

Op Code (hex)	Q-Byte[1] (hex)	Operand Addresses[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
0B	L1-1	Operand 1 address		Operand 2 address	
1B	L1-1	Operand 1 address		Op 2 disp from XR1	
2B	L1-1	Operand 1 address		Op 2 disp from XR2	
4B	L1-1	Op 1 disp from XR1	Operand 2 address		
5B	L1-1	Op 1 disp from XR1	Op 2 disp from XR1		
6B	L1-1	Op 1 disp from XR1	Op 2 disp from XR2		
8B	L1-1	Op 1 disp	Operand 2 address		
9B	L1-1	Op 1 disp from XR2	Op 2 disp from XR1		
AB	L1-1	Op 1 disp from XR2	Op 2 disp from XR2		

[1] The Q-byte designates the operand length:
L1-1 = the number of bytes in operand 1 minus 1.
Operand 2 is a single-byte field.

[2] Address operand 1 by its leftmost position.

Operation

The single character at the operand 2 address replaces all the characters to the left of the first significant digit in operand 1. Only the decimal digits 1 through 9 are significant.

For example, if the leftmost byte of a field to be printed contains a dollar sign, the first operand address should be 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.

Program Notes

- Operand 2 remains unchanged.
- At the end of this operation, the address recall register contains the address of the first significant digit; if no significant digit is found, it contains the address of the byte to the right of the first operand. This new information remains in the register until the system executes the next decimal-add, decimal-subtract, branch, test-I/O-and-branch, or insert-and-test-character instruction.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

Example

Instruction

0B	09	00	B6	00	10

Operand 1 Before Operation

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

Operand 2 Before and After Operation

*	
0010	←Storage Position

Operand 1 After Operation

$	*	*	1	0	8	.	0	9	b̸	*	
00B5	00B6	00B7	00B8	00B9	00BA	00BB	00BC	00BD	00BE	00BF	←Storage Positions

Note: The first operand does not include address 00B5.

Address Recall Register After Operation

00	B8

MOVE LOGICAL IMMEDIATE

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
3C	I	Operand 1 address	
7C	I	Op 1 disp from XR1	
BC	I	Op 1 disp from XR2	

[1] I = one byte of immediate data (for example, one byte of actual data or a single-byte mask).
[2] Operand 1 is a single-byte field; operand 2 is not used.

Operation

This instruction moves the Q-byte into operand 1.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

Example

Instruction

3C	AF	2F	CB

Operand 1 Before Operation

00

2FCB ←— Storage Position

Operand 1 After Operation

AF

2FCB ←— Storage Position

SET BITS ON MASKED

Op Code (hex)	Q-Byte[1] (binary)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
3A	xxxx xxxx	Operand 1 address	
7A	xxxx xxxx	Op 1 disp from XR1	
BA	xxxx xxxx	Op 1 disp from XR2	

[1] The Q-byte contains a single-byte binary mask specifying operand bits to be turned on.
[2] Operand 1 is a single-byte field; operand 2 is not used.

Operation

The system examines the Q-byte, bit by bit. Whenever it encounters a binary 1 in the Q-byte, it sets the corresponding bit in the operand byte to 1; whenever the system encounters a binary 0 in the Q-byte, it leaves the corresponding bit in the operand unchanged.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

Example

Instruction

3A	01011010	00	20

Operand 1 Before Operation

00001100

0020 ←— Storage Position

Operand 1 After Operation

01011110

0020 ←— Storage Position

SET BITS OFF MASKED

Op Code (hex)	Q-Byte[1] (binary)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
3B	xxxx xxxx	Operand 1 address	
7B	xxxx xxxx	Op 1 disp from XR1	
BB	xxxx xxxx	Op 1 disp from XR2	

[1] The Q-byte contains a single-byte binary mask specifying operand bits to be turned on.

[2] Operand 1 is a single-byte field; operand 2 is not used.

Operation

The system examines the Q-byte, bit by bit. Whenever it encounters a binary 1 in the Q-byte, the system sets the corresponding bit in the operand byte to 0; whenever it encounters a binary 0 in the Q-byte, it leaves the corresponding bit in the operand unchanged.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

Example

Instruction

3B	10000001	00	30

Operand 1 Before Operation

01111001

0030 ← Storage Position

Operand 1 After Operation

01111000

0030 ← Storage Position

STORE REGISTER

Op Code (hex)	Q-Byte[1] (binary)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
34	Rx	Operand 1 address	
74	Rx	Op 1 disp from XR1	
B4	Rx	Op 1 disp from XR2	

[1] Rx specifies the register whose contents are to be stored.

[2] Operand 1 is a 2-byte field addressed by its rightmost byte; operand 2 is not used.

Operation

This instruction places the contents of the register specified by the Q-byte in the two-byte field specified by the operand address. The Q-bytes used to specify various registers are:

Q-Byte Binary	Hex	Register Specified
0000 0000	00	None. The system ignores (no-ops) the instruction.
0000 0001	01	XR1.
0000 0010	02	XR2.
0000 0100	04	Program status register.
0000 1000	08	Addressrecall register.
0001 0000	10	Instruction address register.
0010 0000	20	Instruction address register.
0100 0000	40	None (see *Enable START Light* in this chapter).
1000 0000	80	None. The system ignores (no-ops) the instruction.

Program Note

This instruction is used to store only one register at a time.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

Example

Instruction

34	00001000	32	BB

Address Recall Register

0A	CD

Operand Before Operation

2F	C2	
32BA	32BB	← Storage Positions

Operand After Operation

0A	CD	
32BA	32BB	← Storage Positions

LOAD REGISTER

Op Code (hex)	Q-Byte[1] (binary)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
35	Rx	Operand 1 address	
75	Rx	Op 1 disp from XR1	
B5	Rx	Op 1 disp from XR2	

[1] Rx specifies the register into which data is to be loaded.
[2] Operand 1 is a 2-byte field addressed by its rightmost byte; operand 2 is not used.

Operation

This instruction moves data from the 2-byte field specified by the operand address into the register specified by the Q-byte. The Q-bytes used to specify various registers are:

Q-Byte Binary	Hex	Register Specified
0000 0000	00	None. The system ignores (no-ops) the instruction.
0000 0001	01	XR1.
0000 0010	02	XR2.
0000 0100	04	Program status register.
0000 1000	08	Address recall register.
0001 0000	10	Instruction address register.
0010 0000	20	Instruction address register.
0100 0000	40	None (see *Disable START Light* in this chapter).
1000 0000	80	None. The system ignores (no-ops) the instruction.

Program Notes

- This instruction is used to load only one register at a time.
- The six rightmost bits (bits 10-15) of the program status register serve as condition indicators. These are commonly referred to as the program status byte throughout this manual. The other program status register bits are not used.
- You can use this instruction to perform an unconditional branch without disturbing the address recall register; simply load the branch-to address into the instruction address register. At the end of this instruction, the program advances to the instruction at the address specified by the contents of the instruction address register.

Resulting Program Status Byte Settings

This instruction does not affect the program status register unless that is the register specified by the instruction.

Example

Instruction

35	00000100	00	11

Operand

00000000	00000010

0010 0011 ←Storage Positions

Program Status Register Before Operation

00000000	00110001

0 7 8 15 ←Bits

Byte 0 Byte 1

Program Status Register After Operation

00000000	00000010

0 7 8 15 ←Bits

Byte 0 Byte 1

LOAD INDEX REGISTER

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
C2	Rx	Direct address	
D2	Rx	Op 1 disp from XR1	
E2	Rx	Op 1 disp from XR2	

[1] Rx specifies the index register to be loaded:
XR1 = hex 01 or 03
XR2 = hex 02 or 00

[2] A direct address is loaded when the instruction has a C2 op code. When the op code is D2, the system adds the instruction byte 3 value to the contents of XR1 and stores the result in the index register specified by the Q-byte. When the op code is E2 the system adds the instruction byte 3 value to the contents of XR2 and stores the result in the index register specified by the Q-byte.

Operation

This instruction loads the value specified by instruction byte 3 or instruction bytes 3 and 4 into the index register specified by the Q-byte.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

Example

Instruction

D2	02	05

Index Register 1

2A	15

Index Register 2 After Operation

2A	1A

HALT PROGRAM LEVEL

Op Code (hex)	Q-Byte[1] (hex)	R-Byte[2] (hex)
Byte 1	Byte 2	Byte 3
F0	00	00

[1] Q-byte should be hex 00, although it is not examined for this instruction.

[2] R-byte should be hex 00, although it is not examined for this instruction.

Operation

The system advances to the next sequential instruction without performing any operation.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

SUPERVISOR CALL

Op Code (hex)	Q-Byte (hex)	R-Byte (hex)
Byte 1	Byte 2	Byte 3
F4	xx	00

Operation

This instruction passes control to the system. If you use this instruction in your program, a program check will occur.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

Logical Instructions

COMPARE LOGICAL CHARACTERS

Op Code (hex)	Q-Byte[1] (hex)	Operand Addresses[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
0D	L-1	Operand 1 address		Operand 2 address	
1D	L-1	Operand 1 address		Op 2 disp from XR1	
2D	L-1	Operand 1 address		Op 2 disp from XR2	
4D	L-1	Op 1 disp from XR1	Operand 2 address		
5D	L-1	Op 1 disp from XR1	Op 2 disp from XR1		
6D	L-1	Op 1 disp from XR1	Op 2 disp from XR2		
8D	L-1	Op 1 disp from XR2	Operand 2 address		
9D	L-1	Op 1 disp from XR2	Op 2 disp from XR1		
AD	L-1	Op 1 disp from XR2	Op 2 disp from XR2		

[1] The Q-byte designates the operand length:
L-1 = number of bytes in operand 1 minus 1.
Maximum length of each operand is 256 bytes; both operands must be the same length.

[2] The operands may overlap. Address operands by their rightmost byte.

Operation

This instruction compares operand 1 with operand 2, byte by byte, and sets the condition register according to the result of the comparison. The comparison treats each operand as a binary quantity; that is, corresponding bytes from the two operands are compared, bit for bit.

Program Note

Neither operand is altered by the instruction.

Resulting Program Status Byte Settings

Bit	Name	Condition Indicated
7	Equal	Operand values are equal
6	Low	First operand value smaller than second operand value
5	High	First operand value greater than second operand value
4	Decimal overflow	Bit not affected
3	Test false	Bit not affected
2	Binary overflow	Bit not affected

Example

Instruction

0D	02	00	12	00	02

Operand 1 Before and After Operation

27	FA	26	
0010	0011	0012	← Storage Positions

Operand 2 Before and After Operation

23	FA	26	
0000	0001	0002	← Storage Positions

Program Status Byte Before Operation

00100001	
0 7	← Bits

Program Status Byte After Operation

00100100	
0 7	← Bits

COMPARE LOGICAL IMMEDIATE

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
3D	I	Operand 1 address	
7D	I	Op 1 disp from XR1	
BD	I	Op 1 disp from XR2	

[1] I = one byte of immediate data (that is, one byte of actual data that is to be used in binary form).

[2] Operand 1 is a single-byte field; operand 2 is not used.

Example

Instruction

3D	7F	00	21

Operand 1 Before and After Operation

75

0021 ← Storage Position

Program Status Byte After Operation

00000010

0 7 ← Bits

Operation

This instruction compares all the bits in the Q-byte with all the bits in operand 1 and stores the result in the program status byte.

Program Note

Neither the Q-byte nor operand 1 is changed by this operation.

Resulting Program Status Byte Settings

Bit	Name	Condition Indicated
7	Equal	Operand 1 value equal to Q-byte value
6	Low	Operand 1 value less than Q-byte value
5	High	Operand 1 value greater than Q-byte value
4	Decimal overflow	Bit not affected
3	Test false	Bit not affected
2	Binary overflow	Bit not affected

TEST BITS ON MASKED

Op Code (hex)	Q-Byte[1] (binary)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
38	xxxx xxxx	Operand 1 address	
78	xxxx xxxx	Op 1 disp from XR1	
B8	xxxx xxxx	Op 1 disp from XR2	

[1] The Q-byte contains a single-byte binary mask specifying operand bits for testing.

[2] Operand 1 is a single-byte field; operand 2 is not used.

Example

Instruction

38	00010110	00	21

Operand 1 Before and After Operation

10010101

0021 ←Storage Position

Program Status Byte After Operation

00010000

0 7 ←Bits

Operation

This instruction tests specified bits in the operand byte for an on state. For each mask bit (Q-byte bit) on, the system tests the corresponding bit in the operand. If any tested bit is off, the system turns the test false indicator (in the program status register) on.

Program Notes

- The operand and Q-byte remain unchanged.
- Test false condition is turned off by system reset, using test false as a condition in a branch-on-condition or a jump-on-condition instruction, or by loading a binary 0 into program status register bit 11 (bit 3 of the rightmost program status register byte).

Resulting Program Status Byte Settings

Bit	Name	Condition Indicated
7	Equal	Bit not affected
6	Low	Bit not affected
5	High	Bit not affected
4	Decimal overflow	Bit not affected
3	Test false	One of the tested bits not on
2	Binary overflow	Bit not affected

TEST BITS OFF MASKED

Op Code (hex)	Q-Byte[1] (binary)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
39	xxxx xxxx	Operand 1 address	
79	xxxx xxxx	Op 1 disp from XR1	
B9	xxxx xxxx	Op 1 disp from XR2	

[1] The Q-byte contains a single-byte binary mask specifying operand bits for testing.

[2] Operand 1 is a single-byte field; operand 2 is not used.

Operation

This instruction tests specified bits in the operand byte for a binary 1. For each mask bit (Q-byte bit) that is a 1, the system tests the corresponding bit in the operand. If any tested bit is a 1, the system turns the test false indicator (in the program status register) on.

Program Notes

- The operand and Q-byte remain unchanged.
- Test false condition is turned off by system reset, using test false as a condition in a branch-on-condition or jump-on-condition instruction, or by loading a binary 0 into program status register bit 11 (bit 3 of the rightmost program status register byte).

Resulting Program Status Byte Settings

Bit	Name	Condition Indicated
7	Equal	Bit not affected
6	Low	Bit not affected
5	High	Bit not affected
4	Decimal overflow	Bit not affected
3	Test false	One of tested bits on
2	Binary overflow	Bit not affected

Example

Instruction

39	01101100	00	25

Operand 1 Before and After Operation

10010100

0025 ← Storage Position

Program Status Byte After Operation

00010000

0 7 ← Bits

BRANCH ON CONDITION

Op Code (hex)	Q-Byte[1] (binary)	Branch-To Address (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
C0	xxxx xxxx	Direct address	
D0	xxxx xxxx	Disp from XR1	
E0	xxxx xxxx	Disp from XR2	

[1] The Q-byte contains a binary mask specifying which program status register positions are tested by the instruction.

Operation

This instruction tests the program status register (rightmost byte) under control of the Q-byte. If the register satisfies the condition established by the Q-byte, the system places the address of the next sequential instruction in the address recall register, places the branch-to address in the instruction address register, and branches to the branch-to address. If the register does not satisfy at least one condition established by the Q-byte, the system places the address of the next sequential instruction in the instruction address register, and the program advances to the next sequential instruction.

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

Bits 2 through 7 of the Q-byte define the program status register rightmost-byte bits to be tested. These bits, and the conditions they represent, are:

Bit	Condition Tested
1	None (bit should be set to 0)
2	Binary overflow
3	Test false
4	Decimal overflow
5	High
6	Low
7	Equal

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

Program Notes

- The address placed in the address recall register remains there until a decimal-add, decimal-subtract, test-I/O-and-branch, insert-and-test-characters, load-register, add-to-register, or another branch-on-condition instruction is executed.
- When the program status byte is not equal to hex 00:
  - Q-byte of hex 80, x7, or xF (where x is 0, 1, 2, 3, 4, 5, 6, or 7) causes the system to ignore (no-op) the instruction.
  - Q-byte of hex 00, x7, or xF (where x is 8, 9, A, B, C, D, E, or F) causes an unconditional branch.
- When the program status byte is hex 00, or is loaded with hex 00:
  - Q-byte of hex x7 or xF (where x is 0, 1, 2, 3, 4, 5, 6, or 7) causes an unconditional branch. (Hex 80 still causes a no-op to occur.)
  - Q-byte of hex x7 or xF (where x is 8, 9, A, B, C, D, E, or F) causes the system to ignore (no-op) the instruction. (Hex 00 still causes an unconditional branch.)

Resulting Program Status Byte Settings

Bit	Name	Condition
7	Equal	Bit not affected
6	Low	Bit not affected
5	High	Bit not affected
4	Decimal overflow	Turned off if tested; otherwise not affected
3	Test false	Turned off if tested; otherwise not affected
2	Binary overflow	Bit not affected

Example

Instruction

C0	10001000	02	BF	
0BCC	0BCD	0BCE	0BCF	←——Storage Positions

Condition Register Before Operation

00011001

Instruction Address Register After Operation

02	BF

Address Recall Register After Operation

0B	D0

Program Status Byte After Operation

00010001		
0	7	←——Bits

JUMP ON CONDITION

Op Code (hex)	Q-Byte[1] (binary)	R-Byte[2] (hex)
Byte 1	Byte 2	Byte 3
F2	xxxx xxxx	IAR disp

[1] The Q-byte contains a binary mask that indicates which status register bits (the bits in the rightmost byte of the program status register) are tested by the instruction.

[2] The R-byte is a displacement which, when added to the address in the instruction address register, provides a jump-to address.

Operation

This instruction tests the rightmost byte of the program status register under control of the Q-byte. If the register satisfies the conditions established by the Q-byte, the system adds the value stored in the instruction R-byte (byte 3) to the contents of the instruction address register and stores the result in the instruction address register. The program jumps to the new address stored in the instruction address register at the end of the jump-on-condition operation. If the register does not satisfy the condition(s) established by the Q-byte, the system advances to the next sequential instruction in the program. The Q-byte defines what conditions are tested and whether the jump is to occur on condition true (program status register bit is 1) or condition false (program status register bit is 0). When bit 0 of the Q-byte is 1, the jump occurs on condition true; when bit 0 of the Q-byte is 0, the jump occurs on condition false.

Bits 2 through 7 of the Q-byte define the program status byte to be tested. These bits, and the conditions they represent, are:

Bit	Condition Tested
1	None (bit should be set to 0)
2	Binary overflow
3	Test false
4	Decimal overflow
5	High
6	Low
7	Equal

When bit 0 is 1 (condition true), the jump occurs if any of the indicators tested is on (associated bit is 1). When bit 0 is 0 (condition false), the jump occurs if all of the indicators tested are off (associated bits all are 0).

Program Notes

- When the program status byte is not equal to hex 00:
  - Q-byte of hex 80, x7, or xF (where x is 0, 1, 2, 3, 4, 5, 6, or 7) causes the system to ignore (no-op) the instruction.
  - Q-byte of hex 00, x7, or xF (where x is 8, 9, A, B, C, D, E, or F) causes an unconditional jump.
- When the program status byte is hex 00, or is loaded with a hex 00:
  - Q-byte of hex x7 or xF (where x is 0, 1, 2, 3, 4, 5, 6, or 7) causes an unconditional jump. (Hex 80 still causes a no-op to occur.)
  - Q-byte of hex x7 or xF (where x is 8, 9, A, B, C, D, E, or F) causes the system to ignore (no-op) the instruction. (Hex 00 still causes an unconditional jump.)

Resulting Program Status Byte Settings

Bit	Name	Condition Indicated
7	Equal	Bit not affected
6	Low	Bit not affected
5	High	Bit not affected
4	Decimal overflow	Turned off if tested; otherwise not affected
3	Test false	Turned off if tested; otherwise not affected
2	Binary overflow	Bit not affected

Example

Instruction

F2	00110000	0F	
0BBD	0BBE	0BBF	← Storage Positions

Instruction Address Register After Operation

0B	CF

Program Status Byte Before Operation

00001001		
0	7	← Bits

Program Status Byte After Operation

00001001		
0	7	← Bits

LOAD PRINT BELT IMAGE REGISTER

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
31	E4	Operand 1 address	
71	E4	Op 1 disp from XR1	
B1	E4	Op 1 disp from XR2	

[1] With an op code of 31, 71, or B1, a Q-byte of E1, E3, E5, or E7 through EF is invalid and causes a program check.
[2] Operand 1 is a 2-byte field addressed by its rightmost byte; operand 2 is not used.

Operation

This instruction transfers the print belt image from operand 1 in main storage to the print belt image register in system logic.

Program Notes

- The operand field can be used by the program for any desired function after the print belt image register is loaded for the job being run.
- Systems equipped with the serial printer ignore (no-op) this instruction.
- The system loops on the instruction if the print buffer is busy, executing the instruction when the print buffer becomes not busy.
- The system loops on the instruction if a PRINT function key is being processed.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

LOAD CHARACTER SET SIZE REGISTER

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
31	E2	Operand 1 address	
71	E2	Op 1 disp from XR1	
B1	E2	Op 1 disp from XR2	

[1] With an op code of 31, 71, or B1, a Q-byte of E1, E3, E5, or E7 through EF is invalid and causes a program check.
[2] Operand 1 is a 2-byte field addressed by its rightmost byte. The leftmost byte of the operand is not used; the rightmost byte of the operand must contain two hex digits specifying the number of characters on the print belt being used. Operand 2 is not used.

Operation

This instruction transfers the line printer character-set size from the operand field specified by the operand address to the printer character-set size register.

Program Notes

- Data stored in operand 1 is not altered by this instruction.
- If the print buffer is busy when the system issues this instruction, the program loops on the instruction until the buffer is no longer busy, then the system executes instruction.
- If a PRINT function key is processed when this instruction is issued, the program loops on the instruction until function key processing is complete.
- Systems equipped with the serial printer ignore (no-op) this instruction.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

LOAD PRINTER DATA ADDRESS REGISTER

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
31	E6	Operand 1 address	
71	E6	Op 1 disp from XR1	
B1	E6	Op 1 disp from XR2	

[1] With an op code of 31, 71,or B1, a Q-byte of El, E3, E5, or E7 through EF is invalid and causes a program check.

[2] Operand 1 is a 2-byte field addressed by its leftmost byte. The operand contains the address of the field containing data to be printed by the system. Operand 2 is not used.

Operation

This instruction transfers the address of the printer data field in main storage to the printer data address register.

Program Notes

- Data stored in the operand is not altered by this instruction.
- If the print buffer is busy when the system issues this instruction, the program loops on the instruction until the print buffer is no longer busy, then the system executes the instruction.
- The system loops on this instruction if a PRINT function key is being processed.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

LOAD FORMS LENGTH AND CURRENT LINE NUMBER

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
31	E0	Operand 1 address	
71	E0	Op 1 disp from XR1	
B1	E0	Op 1 disp from XR2	

[1] With an op code of 31, 71, or B1, a Q-byte of E1, E3, E5, or E7 through EF is invalid and causes a program check.

[2] Operand 1 is a 2-byte field addressed by its rightmost byte. The rightmost byte holds the current line number; the leftmost byte holds the forms length. Operand 2 is not used.

Operation

This instruction transfers data from the operand to the forms length and line number register associated with the printer.

Program Notes

- Data stored in the operand is not altered by this instruction.
- The system loops on this instruction if the carriage is busy, and executes the instruction when the carriage is not busy.
- The system loops on this instruction if a PRINT function key is being processed.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

START PRINTER IOB

Op Code (hex)	Q-Byte[1] (hex)	R-Byte[2] (hex)
Byte 1	Byte 2	Byte 3
F3	xx	00

[1] E0 specifies continuous forms mode. 90 specifies single form/ledger cards mode. With an op code of F3, a Q-byte of E1 through EF or 91 through 9F is invalid and causes a program check.

[2] R-byte should be hex 00 although it is not examined for this instruction.

Operation

This instruction initiates the print, space, skip, or check reset operation specified by the Q-byte in the printer IOB. (See *Printer IOB and NCPODSW* in Chapter 5.)

Program Notes

- This instruction is always accepted.

- The program must build the IOB and put it on the queue before issuing the start-printer instruction.

- If a unit-check condition exists that prevents execution of the instruction and the IOB does not reset the check indication, or if the printer is not ready, the system sets the no-op bit in the status byte and the program advances to the next sequential instruction.

- If the printer is busy, the system waits until the printer is no longer busy, then executes the IOB.

- Never issue a start-printer-IOB instruction with a Q-byte of E0 when the next IOB is for single form/ledger cards; results are unpredictable.

- No operation is performed on the instruction if a start-printer-IOB instruction is issued with Q-byte of E0 while the hardware switch is set at the single form/ledger cards setting. No operation is performed with a Q-byte of 90 while the hardware switch is set at the continuous-form setting. If the Q-byte is 90 and the control store transient area is not loaded, a processor check occurs.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

SENSE FORMS LENGTH AND CURRENT LINE NUMBER

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
30	E0	Operand 1 address	
70	E0	Op 1 disp from XR1	
B0	E0	Op 1 disp from XR2	

[1] With an op code of 30, 70, or B0, a Q-byte of EI, E2, or E5 through EF is invalid and causes a program check.

[2] Operand 1 is a 2-byte field addressed by its rightmost byte. The leftmost byte holds the forms length; the rightmost byte holds the current line number. Operand 2 is not used.

Operation

This instruction transfers data from the printer forms length register and the printer current line number register to operand 1.

Program Notes

- Data previously stored in operand 1 is replaced by new data from the registers.
- If the carriage is moving, the destination line is stored.
- Data in the registers is not changed by this instruction.
- A sense instruction is accepted at any time.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

SENSE PRINTER STATUS

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
30	xx	Operand 1 address	
70	xx	Op 1 disp from XR1	
B0	xx	Op 1 disp from XR2	

[1] E3 specifies printer status bytes 0 and 1.
E4 specifies printer status bytes 2 and 3.
With an op code of 30, 70, or B0, a Q-byte of E1, E2, or E5 through EF is invalid and causes a program check.

[2] Operand 1 is a 2-byte field addressed by its rightmost byte. Status byte 0 or 2 is stored in the leftmost byte of the operand; status byte 1 or 3 is stored in the rightmost operand byte. Operand 2 is not used.

Operation

This instruction transfers the two bytes of printer status data specified by the Q-byte to the 2-byte main storage field specified by the operand address.

Program Note

The printer accepts this instruction immediately.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

START KEYBOARD/DISPLAY SCREEN IOB

Op Code (hex)	Q-Byte[1] (hex)	R-Byte[2] (hex)
Byte 1	Byte 2	Byte 3
F3	10	00

[1] With an op code of F3, a Q-byte of 11, 12, 13, 15, or 17 through 1F is invalid and causes a program check.

[2] R-byte should be hex 00, although it is not examined in this instruction.

Operation

This instruction initiates the operation specified by the keyboard/display screen IOB in the appropriate priority. (See *IOB Definition and Usage* in Chapter 7.)

Program Notes

- The program must build the keyboard/display screen IOB before issuing the start-keyboard/display-screen-IOB instruction.
- The program must load the address of the IOB with a keyboard/display-screen-IOB instruction before issuing the start-keyboard/display-screen-IOB instruction.
- If this instruction causes execution of an IOB that selects one of the three keyboard operating modes, the system enables the keyboard.
- If bit 3 or bit 7 of the IOB flag byte is set on, flag byte bits 0 and 1 must both be off. Otherwise, results are unpredictable.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

DISABLE KEYBOARD

Op Code (hex)	Q-Byte[1] (hex)	R-Byte[2] (hex)
Byte 1	Byte 2	Byte 3
F3	14	00

[1] With an op code of F3, a Q-byte of 11, 12, 13, 15, or 17 through 1F is invalid and causes a program check.

[2] R-byte should be hex 00, although it is not examined for this instruction.

Operation

This instruction disables the keyboard.

Program Note

Issuing a start-keyboard/display-screen-IOB instruction for an IOB that places the keyboard in one of the three operating modes (basic data entry, sequential data entry, or controlled sequential data entry) enables the keyboard.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

DISABLE KEYBOARD, START IOB, AND CALL OPERATOR

Op Code (hex)	Q-Byte[1] (hex)	R-Byte[2] (hex)
Byte 1	Byte 2	Byte 3
F3	16	00

[1] With an op code of F3, a Q-byte of 11, 12, 13, 15, or 17 through 1F is invalid and causes a program check.

[2] R-byte should be hex 00, although it is not examined for this instruction.

Operation

This instruction performs the following functions:

1. Disables the keyboard.
2. Performs all operations specified by the keyboard/display screen IOB in the priorities described in the keyboard/display screen IOB format in Chapter 7.
3. Flashes the display screen.

Subsequent depression of the ERROR RESET key stops the flashing, initiates a system interrupt, and reenables the keyboard.

Program Note

The IOB started by this instruction should not select one of the three operating modes, because the system enables the keyboard automatically if an operating mode is selected.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

LOAD KEYBOARD/DISPLAY SCREEN IOB ADDRESS

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
31	10	Operand 1 address	
71	10	Op 1 disp from XR1	
B1	10	Op 1 disp from XR2	

[1] With an op code of 31, 71, or B1, a Q-byte of 11,12, or 14 through 1F is invalid and causes a program check.

[2] Operand 1 is a 2-byte field addressed by its rightmost byte; operand 2 is not used.

Operation

This instruction transfers the contents of the operand to the keyboard/display screen IOB address register.

Program Notes

- Data stored in the operand is not altered by this instruction.
- Do not issue this instruction while the keyboard is enabled because interrupts are unpredictable when they are enabled. If the instruction is issued while the keyboard is enabled in the basic data entry (BDE) mode, the system loops on the instruction until the keyboard is no longer enabled, then executes the instruction.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

LOAD KEYBOARD/DISPLAY SCREEN INTERRUPT HANDLER ADDRESS

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
31	13	Operand 1 address	
71	13	Op 1 disp from XR1	
B1	13	Op 1 disp from XR2	

[1] With an op code of 31, 71, or B1, a Q-byte of 11, 12, or 14 through 1F is invalid and causes a program check.

[2] Operand 1 is a 2-byte field addressed by its rightmost byte; operand 2 is not used.

Operation

This instruction transfers the contents of the operand to the keyboard/display screen interrupt-handler address register.

Program Notes

- Data stored in the operand is not altered by this instruction.
- Do not issue this instruction while the keyboard is enabled because interrupts are unpredictable when they are enabled. If the instruction is issued and the keyboard is enabled in the basic data entry (BDE) mode, the system loops on the instruction until the keyboard is no longer enabled, then executes the instruction.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

SENSE ADDRESS/DATA SWITCHES

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
30	00	Operand 1 address	
70	00	Op 1 disp from XR1	
B0	00	Op 1 disp from XR2	

[1] With an op code of 30, 70, or B0, a Q-byte of 01 through 0F is invalid and causes a program check.

[2] Operand 1 is a 2-byte field addressed by its rightmost byte. The leftmost operand byte receives data from ADDRESS/DATA switches 1 and 2; the rightmost operand byte receives data from address/data switches 3 and 4. Operand 2 is not used.

Operation

This instruction transfers hex digits set up in the ADDRESS/DATA switches (on the CE panel) to the 2-byte operand specified by the operand address.

Program Notes

- Data previously stored in the operand is replaced by data set up in the ADDRESS/DATA switches.
- The system accepts a sense instruction at any time.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

LOAD DISKETTE CONTROL FIELD ADDRESS REGISTER

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
31	D0	Operand 1 address	
71	D0	Op 1 disp from XR1	
B1	D0	Op 1 disp from XR2	

[1] With an op code of 31, 71, or B1, a Q-byte of D2 through DF is invalid and causes a program check.
[2] Operand 1 is a 2-byte field addressed by its rightmost byte; operand 2 is not used.

Operation

This instruction loads the address of the diskette control field from main storage into the diskette control field address register.

Program Note

Operand 1 is not changed by this instruction.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

LOAD DISKETTE DATA FIELD ADDRESS REGISTER

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
31	D1	Operand 1 address	
71	D1	Op 1 disp from XR1	
B1	D1	Op 1 disp from XR2	

[1] With an op code of 31, 71, or B1, a Q-byte of D2 through DF is invalid and causes a program check.
[2] Operand 1 is a 2-byte field addressed by its rightmost byte; operand 2 is not used.

Operation

This instruction loads the address of the diskette data field from main storage into the diskette data address register.

Program Note

Operand 1 is not changed by this instruction.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

SEEK DISKETTE TRACK OR RECALIBRATE DISKETTE

Op Code (hex)	Q-Byte[1] (hex)	R-Byte (hex)
Byte 1	Byte 2	Byte 3
F3	D0	00

[1] With an op code of F3, a Q-byte of D3 or D8 through DF is invalid and causes a program check.

Operation

This instruction moves the diskette read/write head to the track specified by the diskette control field. If hex FF is specified as the track address, the system seeks to track 00 and recalibrates the access mechanism to 00.

Program Notes

- A recalibrate operation must be performed to clear a diskette drive not-ready indication.

- If the diskette drive has an associated unit check that prevents execution of the instruction, or if the drive is not ready for any instruction except a recalibrate-diskette instruction, the system sets the diskette no-op status bit and advances to the next sequential instruction without performing the seek operation.

- If the diskette has an associated unit-check condition that does not prevent instruction execution, the system executes the instruction and resets the unit-check status bit.

- An invalid address or seek failure is detected when the system compares the control field to the ID field from the diskette sector during the next read or write operation.

- A single-track seek has some unique functional characteristics that improve save/restore performance. A single-track-seek instruction requires a maximum of 170.83 milliseconds to execute and considerably decreases the time required to handle a following diskette read or write instruction. Seek for other than a single track requires a maximum of 106 + 53 (SD) ms (SD = seek displacement in tracks). Recalibrate time is 4.346 seconds, maximum.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

READ DISKETTE DATA AND CONTROL RECORD

Op Code (hex)	Q-Byte[1] (hex)	R-Byte (hex)
Byte 1	Byte 2	Byte 3
F3	D4	00

[1] With an op code of F3, a Q-byte of D3 or D8 through DF is invalid and causes a program check.

Operation

This instruction is similar to the read-diskette-data instruction. The diskette read/write head moves to the track address specified by the control field, then the system reads the number of records specified by the control field into contiguous positions of the diskette data field addressed by the diskette data address register. Reading starts at the record specified by the record number in the diskette control field, with the system adding 1 to the record number and subtracting 1 from the number of records to be read until the operation is complete. The operation ends when the number of records in the control field is hex FF.

If the system detects the end of cylinder during the read operation, it seeks the next sequential cylinder, adds 1 to the cylinder number in the control field, and resets the record number to 01. Whenever the system encounters a track that is flagged as deleted or defective (an ID field of hex FFFFFFFF), it always ignores that track and seeks the next sequential track.

There is one essential difference between this instruction and the read-diskette-data instruction. In the read-diskette-data and control-record instruction, when the system detects a control field in the specified sector, it reads the data from the control field into the diskette data field regardless of the character occupying the first position of the diskette control field. In such cases, the system turns on the control-address-mark status bit (byte 1, bit 3).

Program Notes

- A program cannot determine if the field moved is a data field or a control field by issuing a test-diskette instruction. Instead, the program must test the control-address-mark status bit (byte 1, bit 3). If this bit is on, one of the fields transferred is a control field, and the program must examine the records, one by one.

- If the diskette drive has a unit-check condition that prevents execution of the instruction when it is issued, or if the drive is not ready, the system sets the diskette no-op status bit and advances to the next sequential instruction without performing the read operation.

- If the diskette drive has a unit-check condition that does not prevent execution of the instruction, the system executes the instruction and resets the unit-check status bit.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

WRITE AND VERIFY DISKETTE DATA

Op Code (hex)	Q-Byte[1] (hex)	R-Byte (hex)
Byte 1	Byte 2	Byte 3
F3	D5	00

[1] With an op code of F3, a Q-byte of D3 or D8 through DF is invalid and causes a program check.

Operation

This instruction moves the read/write head to the track address specified in the diskette control field, then writes the number of records specified by the control field into contiguous positions of the data field addressed by the diskette data address register. Records are written, starting at the record number in the control field. Whenever the system encounters a track flagged as defective, it automatically seeks to the next sequential track.

To write each record, the system first reads the record ID field, then writes hex FB into the AM2 field to indicate that the following field is a data field, then finally writes 128 or 512 bytes of data from 128 or 512 sequential positions of the diskette data field in main storage. The system reads the same data back during the next revolution of the diskette to verify that it is written onto the diskette correctly.

If the system encounters the end of a cylinder during the operation, it automatically seeks to record 1 of the next sequential track to write the next record.

During the write-diskette-and-verify operation, the system adds 1 to the record number and subtracts 1 from the number of records to be accessed as each record is written. If a cylinder boundary is crossed, the system increases the cylinder number by 1 and sets the record number to 01. The other portions of the control field remain unchanged.

Program Notes

- If the diskette drive has a unit-check condition that prevents execution of the instruction, or if the drive is not ready when the instruction is issued, the system sets the diskette-no-op status bit and advances to the next sequential instruction without performing the write-and-verify-diskette-data instruction.

- If the diskette drive has a unit check that does not prevent execution of the instruction, the system resets the unit-check status and executes the instruction.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

WRITE AND VERIFY DISKETTE CONTROL ADDRESS MARKER

Op Code (hex) Byte 1	Q-Byte[1] (hex) Byte 2	R-Byte (hex) Byte 3
F3	D6	00

[1] With an op code of F3, a Q-byte of D3 or D8 through DF is invalid and causes a program check.

Operation

This instruction moves the read/write head to the track address specified by the diskette control field, then writes the number of 128-byte or 512-byte records specified by the control field from the diskette data field addressed by the diskette data address register. Writing starts at the record specified by the record number and continues into sequential sectors of the diskette until the operation is complete.

The difference between this instruction and the write-and-verify-diskette-data instruction is that in this instruction, the system writes hex F8 in the AM2 field before writing the data for each sector. As with the write-and-verify-data instruction, the system verifies the accuracy of data written by reading it during the next revolution of the diskette.

During this operation, if the system encounters a track flagged as defective, the system automatically seeks to the next sequential track. If the system encounters the end of a cylinder, it automatically seeks to record 1 of the next sequential track to write the next record.

During the write-and-verify-diskette-control-address-marker operation, the system adds 1 to the record number and subtracts 1 from the number of records to be accessed as each record is written. If the system crosses a cylinder boundary during the operation, it increases the cylinder number in the control field by 1 and sets the record number to 01. The other portions of the control field remain unchanged.

Program Notes

- If the diskette drive has a unit check that prevents execution of the instruction, or if the drive is not ready when the instruction is issued, the system sets the diskette no-op status bit and advances to the next sequential instruction without performing the write-and-verify-diskette-control-address-marker operation.

- If the diskette drive has a unit-check condition that does not prevent execution of the instruction, the system resets the unit check status and executes the instruction.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

READ DISKETTE ID

Op Code (hex)	Q-Byte[1] (hex)	R-Byte (hex)
Byte 1	Byte 2	Byte 3
F3	D2	00

[1] With an op code of F3, a Q-byte of D3 or D8 through DF is invalid and causes a program check.

Operation

The system immediately starts to read data from the track under the read/write head, transferring the four bytes from the first successfully read ID field into the leftmost four positions of the main storage diskette read data field specified by the diskette data address register.

Program Notes

- If the system cannot read at least one ID field on the track successfully, it turns on the no-orient bit in the diskette status byte (byte 2, bit 3).

- This instruction does not cause an automatic seek and the system does not change any portion of the control field.

- If the diskette drive has a unit-check condition that prevents execution of the instruction when it is issued, or if the drive is not ready, the system sets the diskette no-op status bit and advances to the next sequential instruction without performing the read-diskette-ID instruction.

- If the diskette has a unit-check condition that does not prevent execution of the instruction, the system executes the instruction and resets the unit-check status bit.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

READ DISKETTE DATA

Op Code (hex)	Q-Byte[1] (hex)	R-Byte (hex)
Byte 1	Byte 2	Byte 3
F3	D1	00

[1] With an op code of F3, a Q-byte of D3 or D8 through DF is invalid and causes a program check.

Operation

This instruction initiates an automatic seek to the track address specified in the control field and then, starting at the record number specified in the control field, reads the number of records specified by the control field into contiguous positions of the diskette data field addressed by the diskette data address register.

If the system encounters a track that is flagged defective, it automatically moves to the next track and examines that track number.

If the system encounters the end of cylinder before all the sectors specified for reading are read, the system automatically seeks to record 1 of the next track that is not flagged defective, then continues to read sectors until all specified sectors are read.

As the system performs this operation, it modifies the control field to reflect the current status of the read operation as each record is being read. After the system stops reading each sector, it increases the record number in the control field by 1 and decreases the number of records to be read by 1 if no unit-check condition is detected. If the system crosses a cylinder boundary, it increases the cylinder number in the control field by 1 and sets the record number to 01. The other portions of the control field remain unchanged.

Program Notes

- If the system detects a control field in the specified sector, and the first character in the data field is not a hex C4 or C6 (alphabetic D or F), the system sets the invalid-control-record-status bit (byte 1, bit 1) on, and sets the control-address-mark bit (byte 1, bit 3) on, terminating the read data operation after the field is read. If the system detects a control address mark with a hex C4 or C6 in the first position of the data field, the system ignores that record except to add 1 to the record number in the diskette control field; then the system reads the next record.
- If the diskette drive has a unit check that prevents execution of the instruction when it is issued, or if the drive is not ready, the system sets the diskette no-op status bit and advances to the next sequential instruction without performing the read operation.
- If the diskette drive has a unit-check condition that does not prevent execution of the instruction, the system executes the instruction and resets the unit-check status bit.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

WRITE AND VERIFY DISKETTE RECORD ID

Op Code (hex)	Q-Byte[1] (hex)	R-Byte (hex)
Byte 1	Byte 2	Byte 3
F3	D7	00

[1] With an op code of F3, a Q-byte of D3 or D8 through DF is invalid and causes a program check.

Operation

This instruction is used primarily for track initialization and defective track identification.

The instruction writes an entire track of data from the diskette data field in main storage into the track under the read/write head. As the system writes the data, it also numbers the sectors sequentially from 01 through 26 for standard interchange diskettes or 01 through 08 for diskettes in the extended format. After the system writes the ID field and data, it reads the data on the next revolution of the diskette to ensure that the sectors can be read. During this read operation, the system does not transfer the data anywhere.

During a write-and-verify-diskette-record-ID operation, the system increments the record number and decrements the X-byte in the diskette control field in main storage. At the end of the operation, these fields contain decimal 26 or 08 and hex FF, respectively.

If the cylinder portion of the control field contains hex FF at the start of the operation, the record number in the control field is not changed during the write-and-verify-diskette-ID operation; and the system writes hex FFFFFFFF, instead of valid ID data, into the ID field of each sector on the track. (During subsequent diskette operations, the system ignores tracks containing this ID field, and seeks the next sequential cylinder.) During automatic verification of this operation, the system always turns on the length-mismatch bit and the no-orient bit in the diskette status bytes. To verify these ID fields, use the read-ID instruction.

Program Notes

- If the diskette drive has a unit-check condition that prevents execution of the instruction, or if the drive is not ready when the instruction is issued, the system sets the diskette no-op status bit and advances to the next sequential instruction without performing the write-and-verify-diskette-record-ID instruction.
- If the diskette drive has a unit-check condition that does not prevent execution of the instruction, the system resets the unit-check status and executes the instruction.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

TEST DISKETTE DRIVE AND BRANCH

Op Code (hex)	Q-Byte[1] (hex)	Operand Address (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
C1	D0	Direct Address	
D1	D0	Disp from XR1	
E1	D0	Disp from XR2	

[1] With an op code of C1, D1, or E1, a Q-byte of D1 through DF is invalid and causes a program check.

Operation

This instruction tests the diskette drive for a not-ready or unit-check condition. If either condition exists, the program branches to the address specified in the branch-to address part of the instruction. (To determine the cause of the not-ready/unit-check condition, sense status bytes 0, 1, and 2 and test their bits for on conditions.) If neither condition exists, the program does not branch, and the system performs the next sequential instruction.

Program Notes

- A test instruction can be issued anytime, and is always accepted.
- This instruction does not test for a control-address-mark-record-found check.
- If a branch occurs, the system first stores the address of the next sequential instruction in the address recall register and the branch-to address in the instruction address register, then executes the branch.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

SENSE DISKETTE STATUS

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
30	xx	Operand 1 address	
70	xx	Op 1 disp from XR1	
B0	xx	Op 1 disp from XR2	

[1] D2 specifies the diskette status bytes 0 and 1.
D3 specifies the diskette status bytes 2 and 3.
With an op code of 30, 70, or B0, a Q-byte of D4 through DF is invalid and causes a program check..

[2] Operand 1 is a 2-byte field addressed by its rightmost byte; operand 2 is not used.

Operation

This instruction transfers the two bytes of status data specified by the Q-byte from the diskette status register to the main storage 2-byte field specified by the operand address.

Program Notes

- The diskette drive accepts this instruction immediately.
- All status bits except drive-not-ready are reset by the next diskette seek, read, or write instruction.
- To reset the not-ready bit, make the diskette drive ready, then recalibrate the diskette drive by issuing a diskette-seek-to-cylinder-FF instruction.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

SENSE DISKETTE CONTROL FIELD ADDRESS REGISTER

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
30	D0	Operand 1 address	
70	D0	Op 1 disp from XR1	
B0	D0	Op 1 disp from XR2	

[1] With an op code of 30, 70, or B0, a Q-byte of D4 through DF is invalid and causes a program check.

[2] Operand 1 is a 2-byte field addressed by its rightmost byte; operand 2 is not used.

Operation

This instruction transfers the 2-byte diskette control field address from the diskette control field address register to the 2-byte main storage field specified by the operand address.

Program Note

The diskette drive accepts this instruction immediately.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

SENSE DISKETTE DATA ADDRESS REGISTER

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
30	D1	Operand 1 address	
70	D1	Op 1 disp from XR1	
B0	D1	Op 1 disp from XR2	

[1] With an op code of 30, 70, or B0, a Q-byte of D4 through DF is invalid and causes a program check.

[2] Operand 1 is a 2-byte field addressed by its rightmost byte; operand 2 is not used.

Operation

This instruction transfers the address of the diskette data field in main storage from the diskette data address register to the 2-byte main storage field specified by the operand address.

Program Note

The diskette drive accepts this instruction immediately.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

RESET INTERRUPT

Op Code[1] (hex)	Q-Byte[1] (hex)	Operand 1 Address[1] (hex)		Control Code[2] (hex)
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
C0	87	00	04	01

[1] The op code, Q-byte, and operand address specify a branch to a general entry location in control storage.

[2] The control code is also called a request indicator byte (RIB); it specifies the system operation.

Operation

This instruction resets the interrupt that caused the branch to the current interrupt handler routine. When the reset interrupt operation is complete, the system branches back to the next sequential instruction of the program being processed.

[illegible]

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

WAIT FOR IOB

Op Code[1] (hex)	Q-Byte[1] (hex)	Operand 1 Address[1] (hex)		Control Code[2] (hex)
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
C0	87	00	04	02

[1] The op code, Q-byte, and operand address specify a branch to a general entry location in control storage.

[2] The control code is also called a request indicator byte (RIB); it specifies the system operation.

Operation

This instruction causes the system to wait for completion of the I/O operation specified in the indicated IOB.

Program Note

When the wait-for-IOB instruction is issued, XR1 must contain the address of the IOB for which the system is to wait.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

DISABLE BSCA, KEYBOARD, AND INQUIRY INTERRUPTS

Op Code[1] (hex)	Q-Byte[1] (hex)	Operand 1 Address[1] (hex)		Control Code[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
C0	87	00	04	09	[3]	[3]	[3]

[1] The op code, Q-byte, and operand address specify a branch to a general entry location in control storage. When the disable operation is complete, the system branches back to the next sequential instruction of the program being processed.

[2] Byte 5 of the control code is also called a request indicator byte (RIB); it specifies the system operation.

[3] Bytes 6, 7, and 8 (in control code) specify which interrupts are disabled:

40 00 00	specifies BSCA interrupt.
20 00 00	specifies keyboard interrupt.
00 00 01	specifies inquiry interrupt. (This code is valid only with the System/32 SCP provided by IBM.)

Combinations of the above coding are valid. For example, 20 00 01 specifies keyboard and inquiry interrupts.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

START DISK IOB

Op Code[1] (hex)	Q-Byte[1] (hex)	Operand 1 Address[1] (hex)		Control Code[2] (hex)
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
C0	87	00	04	03

[1] The op code, Q-byte, and operand address specify a branch to a general entry location in control storage.

[2] The control code is also called a request indicator byte (RIB); it specifies the system operation.

Operation

This instruction starts the operation specified by the disk IOB. When the start operation is complete, the system branches back to the next sequential instruction of the program being processed.

Program Note

XR1 must contain the address of the disk IOB when the program issues the start-disk-IOB instruction.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

ENABLE BSCA, KEYBOARD, AND INQUIRY INTERRUPTS

Op Code[1] (hex)	Q-Byte[1] (hex)	Operand 1 Address[1] (hex)		Control Code[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
C0	87	00	04	0A	[3]	[3]	[3]

[1] The op code, Q-byte, and operand address specify a branch to a general entry location in control storage. When the enable operation is complete, the system branches back to the next sequential instruction of the program being processed.

[2] Byte 5 of the control code is also called a request indicator byte (RIB); it specifies the system operation.

[3] Bytes 6, 7, and 8 (in control code) specify which interrupts are enabled:

40 00 00 specifies BSCA interrupt.
20 00 00 specifies keyboard interrupt.
00 00 01 specifies inquiry interrupt. (This code is valid only with the IBM System/32 SCP provided by IBM.)
Combinations of the above coding are valid. For example, 20 00 01 specifies keyboard and inquiry interrupts.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

INITIALIZE DISKETTE DRIVE

Op Code[1] (hex)	Q-Byte[1] (hex)	Operand 1 Address[1] (hex)		Control Code[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
C0	87	00	04	0F	00	01	00

[1] The op code, Q-byte, and operand address specify a branch to a general entry location in control storage.

[2] The control code is also called a request indicator byte (RIB); it specifies the system operation.

Operation

This instruction establishes the system control codes needed for diskette unit operations. When the initialize operation is complete, the system branches back to the next sequential instruction of the program being processed.

Program Note

The program must issue this instruction once per job.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

QUEUE/DEQUEUE PRINTER IOB

Op Code[1] (hex)	Q-Byte[1] (hex)	Operand 1 Address[1] (hex)		Control Code[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
C0	87	00	04	0E	[3]	06	00

[1] The op code, Q-byte, and operand address specify a branch to a general entry location in control storage.

[2] The control code is also called a request indicator byte (RIB); it specifies the system operation.

[3] Byte 6 (in control code) specifies what the system does with the printer IOB:
00 loads the IOB in the last position on the system IOB queue.
10 loads the IOB in the first position on the system IOB queue.
01 removes the IOB from the system IOB queue.

Operation

This instruction loads the printer IOB into the system IOB queue, or remôves the IOB from the queue, as specified by byte 6. When the queue/dequeue operation is complete, the system branches back to the next sequential instruction of the program being processed.

Program Note

XR1 must contain the address of the printer IOB when the program issues the queue/dequeue instruction.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

QUEUE/DEQUEUE KEYBOARD/DISPLAY SCREEN IOB

Op Code[1] (hex)	Q-Byte[1] (hex)	Operand 1 Address[1] (hex)		Control Code[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
C0	87	00	04	0E	[3]	04	00

[1] The op code, Q-byte, and operand address specify a branch to a general entry location in control storage.
[2] The control code is also called a request indicator byte (RIB); it specifies the system operation.
[3] Byte 6 (in the control code) specifies what the system does with the keyboard/display screen IOB:
00 loads the IOB in the last position on the system IOB queue.
10 loads the IOB in the first position on the system IOB queue.
01 removes the IOB from the system IOB queue.

Operation

This instruction loads the keyboard/display screen IOB into the system IOB queue, or removes the IOB from the queue, as specified by byte 6. When the queue/dequeue operation is complete, the system branches back to the next sequential instruction of the program being processed.

Program Note

XR1 must contain the address of the keyboard/display screen IOB when the program issues the queue/dequeue instruction.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

CONTROL BSCA

Op Code (hex)	Q-Byte[1] (hex)	R-Byte[2] (hex)
Byte 1	Byte 2	Byte 3
F3	80	xx

[1] With an op code of F3, a Q-byte of 84 through 8F is invalid and causes a program check.

[2] The R-byte serves as a control code that specifies the following:

Control Code (in hex)	Function Specified
00	Cancel 2-second timeout
04	Start 2-second timeout
80	Disable BSCA and cancel 2-second timeout
84	Disable BSCA and start 2-second timeout
C0	Enable BSCA and cancel 2-second timeout
C4	Enable BSCA and start 2-second timeout

Bits 2, 3, 4, 6, and 7 of the control code are not used by the system, but these should always be 0. If bit 0 is 0, the system does not examine bit 1.

Operation

The system performs the function specified by the R-byte (control code) of this instruction:

- *Two-Second Timeout:* This control instruction is provided to obtain a 2-second delay before the transmission of TTD or WACK. The start 2-second timeout must be given only with the control instruction. When the timeout is completed, an interrupt is generated. The BSCA is not busy when doing a 2-second timeout. It can be terminated by issuing any start BSCA instruction. A previously issued start 2-second timeout will be terminated if a noncontrol instruction is issued. The start 2-second timeout operation must not be issued while the adapter is busy.

  The BSCA need not be enabled to complete the 2-second timeout operation with an op-end interrupt.

- *Enable/Disable BSCA Control:* The enable BSCA function causes the communications adapter to become operable and allows it to connect to the modem and perform data handling functions.

Program Notes

- This is the only instruction that can start the 2-second timeout function. As good programming practice, this instruction should be used to start or cancel the 2-second timeout, and enable or disable the BSCA.

- If the BSCA is busy when a start 2-second timeout instruction is issued, the program loops on the instruction until the BSCA is not busy.

- Issuing a start BSCA receive initial, receive only, or transmit and receive instruction cancels the 2-second timeout.

Resulting Program Status Byte Setting

This instruction does not affect the program status register.

START BSCA RECEIVE INITIAL

Op Code (hex)	Q-Byte[1] (hex)	R-Byte[2] (hex)
Byte 1	Byte 2	Byte 3
F3	83	C0

[1] With an op code of F3, a Q-byte of 84 through 8F is invalid and causes a program check.
[2] The R-byte for a start BSCA receive initial instruction should be C0, as shown.

Operation

This operation allows the remote station to establish contact so it can transmit a message. The receive initial function is the only one that can be used by a tributary station for establishing contact in a multipoint network. In this operation the local communications adapter monitors the line until it receives an initialization sequence. Upon receiving the initialization sequence, the communications adapter stores the characters received in locations specified by the current address register. The BSCA adds +1 to the address register each time a character is stored. The operation ends and the BSCA generates interrupt request when: (1) the BSCA recognizes a change-of-direction character, (2) the current address register equals the stop address register, or (3) no synchronizing characters are received for three seconds after an initialization sequence is begun. Any of the control functions except start 2-second timeout can be combined with this operation.

Program Notes

- The program must enable the BSCA with a control only instruction and place a BSCA IOB on the system queue before issuing the start BSCA receive initial instruction.

- If the BSCA is busy when the program issues the start BSCA receive initial instruction, the system waits until the BSCA becomes not-busy before it executes the instruction.

- If a BSCA not-ready condition exists, the system advances to the next sequential instruction in the program without initiating the BSCA receive initial operation.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

START BSCA RECEIVE ONLY

Op Code (hex)	Q-Byte[1] (hex)	R-Byte[2] (hex)
Byte 1	Byte 2	Byte 3
F3	81	C0

[1] With an op code of F3, a Q-byte of 84 through 8F is invalid and causes a program check.
[2] The R-byte for a receive BSCA message instruction should be C0, as shown.

Operation

This operation accepts characters from the line and places them in storage at the location designated by the current address register. The BSCA adds +1 to the current address register each time a character is stored. The receive only operation ends: (1) when a change-of-direction character is received from the line, (2) when the current address register equals the stop address register, or (3) when no synchronizing characters are received from the line for three seconds.

Any of the control functions except start 2-second timeout can be initiated by this instruction.

Program Notes

- The program must enable the BSCA with a control only instruction and place a BSCA IOB on the system queue before issuing the start BSCA receive only instruction.

- If the BSCA is busy when the program issues the start BSCA receive only instruction, the system waits until the BSCA becomes not-busy before it executes the instruction.

- If a BSCA not-ready condition exists, the system advances to the next sequential instruction in the program without initiating the BSCA receive only operation.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

START BSCA TRANSMIT AND RECEIVE

Op Code (hex)	Q-Byte[1] (hex)	R-Byte[2] (hex)
Byte 1	Byte 2	Byte 3
F3	82	C0

[1] With an op code of F3, a Q-byte of 84 through 8F is invalid and causes a program check.

[2] The R-byte for a start BSCA transmit and receive instruction should be C0, as shown.

Operation

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 BSCA adds +1 to the current address register each time a character is transmitted. The last character to be transmitted must be a change-of-direction character and must be stored at an address one less than the address contained in the transition address register.

When the current address register is updated to equal the transition address register, the communications adapter stops transmitting and begins receiving characters from the line, storing the characters received into main storage at locations specified by the current address register. The BSCA adds +1 to the current address register each time a character is stored.

The operation ends and the BSCA generates an interrupt request when: (1) a change-of-direction character is received, (2) the current address register equals the stop address register, or (3) no synchronizing characters are received for three seconds. Any of the control functions except start 2-second timeout can be initiated by this instruction.

The transmit-and-receive operation can be used as a transmit only operation by loading the same address into both the transition address register and the stop address register. A transmit-and-receive operation with a zero length transmit field (initial value of the current address register and transition address register the same) is not allowed.

The transmit-and-receive function is provided to reduce line-turnaround time. The transmit-and-receive operation should be used in all transmit sequences that require a response.

Program Notes

- The program must enable the BSCA with a control only instruction and place a BSCA IOB on the system queue before issuing the start BSCA transmit and receive instruction.
- If the BSCA is busy when the program issues the start BSCA transmit and receive instruction, the system waits until the BSCA becomes not-busy before it executes the instruction.
- If a BSCA not-ready condition exists, the system advances to the next sequential instruction in the program without initiating the BSCA transmit and receive operation.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

LOAD BSCA UNIT DEFINITION TABLE REGISTER

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
31	85	Operand 1 address	
71	85	Op 1 disp from XR1	
B1	85	Op 1 disp from XR2	

[1] With an op code of 31, 71, or B1, a Q-byte of 80 or 86 through 8F is invalid and causes a program check.

[2] Operand 1 is a 2-byte field that is addressed by its rightmost byte. Operand 2 is not used by this instruction.

Operation

This instruction transfers the contents of the 2-byte operand to the BSCA unit definition table register.

Program Notes

- Data stored in the operand is not altered by this instruction.
- If the BSCA is busy when this instruction is issued, the program loops on the load instruction until the BSCA becomes not-busy, then loads the register.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

LOAD BSCA CURRENT ADDRESS REGISTER

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
31	84	Operand 1 address	
71	84	Op 1 disp from XR1	
B1	84	Op 1 disp from XR2	

[1] With an op code of 31, 71 or B1, a Q-byte of D2 through DF is invalid and causes a program check.

[2] Operand 1 is a 2-byte field that is addressed by its rightmost byte. Operand 2 is not used in this instruction.

Operation

This instruction transfers the contents of the 2-byte operand to the BSCA current address register.

Program Notes

- Data stored in the operand is not altered by this instruction.
- If the BSCA is busy when this instruction is issued, the program loops on the load instruction until the BSCA becomes not-busy, then loads the register.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

LOAD BSCA INTERRUPT ADDRESS REGISTER

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
31	83	Operand 1 address	
71	83	Op 1 disp from XR1	
B1	83	Op 1 disp from XR2	

[1] With an op code of 31, 71, or B1, a Q-byte of 80 or 86 through 8F is invalid and causes a program check.

[2] Operand 1 is a 2-byte field that is addressed by its rightmost byte. Operand 2 is not used in this instruction.

Operation

This instruction transfers the contents of the 2-byte operand (the address of the BSCA interrupt routine) to the BSCA interrupt address register.

Program Notes

- Data stored in the operand is not altered by this instruction.
- If the BSCA is busy when this instruction is issued, the program loops on the instruction until the BSCA becomes not-busy, then loads the register.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

LOAD BSCA STOP ADDRESS REGISTER

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
31	81	Operand 1 address	
71	81	Op 1 disp from XR1	
B1	81	Op 1 disp from XR2	

[1] With an op code of 31, 71, or B1, a Q-byte of 80 or 86 through 8F is invalid and causes a program check.

[2] The operand is a 2-byte field that is addressed by its rightmost byte. Operand 2 is not used in this instruction.

Operation

This instruction transfers the contents of the 2-byte operand to the stop address register.

Program Notes

- Data stored in the operand is not altered by this instruction.
- If the BSCA is busy when this instruction is issued, the program loops on the instruction until the BSCA becomes not busy, then loads the register.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

LOAD BSCA TRANSITION ADDRESS REGISTER

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
31	82	Operand 1 address	
71	82	Op 1 disp from XR1	
B1	82	Op 1 disp from XR2	

[1] When the op code is 31, 71, or B1, a Q-byte of 80 or 86 through 8F is invalid and causes a program check.

[2] The operand is a 2-byte field that is addressed by its rightmost byte. Operand 2 is not used in this instruction.

Operation

This instruction transfers the contents of the 2-byte operand to the BSCA transition address register.

Program Notes

- Data stored in the operand is not altered by this instruction.
- If the BSCA is busy when this instruction is issued, the program loops on the instruction until the BSCA becomes not-busy, then loads the register.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

SENSE BSCA CURRENT ADDRESS

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
30	84	Operand 1 address	
70	84	Op 1 disp from XR1	
B0	84	Op 1 disp from XR2	

[1] With an op code of 30, 70, or B0, a Q-byte of 80, 81, 82, or 85 through 8F is invalid and causes a program check.

[2] Operand 1 is a 2-byte field that is addressed by its rightmost byte. Operand 2 is not used in this instruction.

Operation

This instruction transfers the contents of the current address register to the 2-byte field specified by the operand address.

Program Note

The current address register identifies the position of main storage from which the BSCA will next move data onto the data transmission line (during a data transmit operation) or into which the BSCA will next store a character received from the data transmission line (during a receive operation).

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

SENSE BSCA STATUS

Op Code (hex)	Q-Byte[1] (hex)	Operand Address[2] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
30	83	Operand 1 address	
70	83	Op 1 disp from XR1	
B0	83	Op 1 disp from XR2	

[1] With an op code of 30, 70, or B0, a Q-byte of 80, 81, 82, or 85 through 8F is invalid and causes a program check.

[2] Operand 1 is a 2-byte field that is addressed by its rightmost byte. Operand 2 is not used in this instruction.

Operation

This instruction transfers the contents of the 2-byte BSCA status register into the field specified by the operand address.

Program Note

BSCA status byte 1 is stored in the rightmost byte of the 2-byte operand; status byte 2 is stored in the leftmost byte.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

INITIALIZE BSCA FOR EBCDIC MODE

Op Code[1] (hex)	Q-Byte[1] (hex)	Operand 1 Address[1] (hex)		Control Code[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
C0	87	00	04	0F	00	05	00

[1] The op code, Q-byte, and operand address specify a branch to a general entry location in system control storage.
[2] Byte 5 in the control code is also called a request indicator byte (RIB); it specifies the system operation.

Operation

This instruction establishes the system control codes needed for BSCA operation in EBCDIC mode.

Program Note

The program must specify either EBCDIC mode or ASCII mode once for each job that uses BSCA programming.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

INITIALIZE BSCA FOR ASCII MODE

Op Code[1] (hex)	Q-Byte[1] (hex)	Operand 1 Address[1] (hex)		Control Code[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
C0	87	00	04	0F	00	04	00

[1] The op code, Q-byte, and operand address specify a branch to a general entry location in system control storage.
[2] Byte 5 in the control code is also called a request indicator byte (RIB); it specifies the system operation.

Operation

This instruction establishes the system control codes needed for BSCA operation in the ASCII mode.

Program Note

The program must specify either ASCII mode or EBCDIC mode once for each job that uses BSCA programming.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

QUEUE/DEQUEUE BSCA IOB

Op Code[1] (hex)	Q-Byte[1] (hex)	Operand 1 Address[1] (hex)		Control Code[2] (hex)			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
C0	87	00	04	0E	[3]	02	00

[1] The op code, Q-byte, and operand address specify a branch to a general entry location in system control storage.
[2] Byte 5 in the control code is also called a request indicator byte (RIB); it specifies the system operation.
[3] Byte 6 in the control code specifies what the system is to do with the BSCA IOB:

00 Loads BSCA IOB in last position on the system IOB queue.
10 Loads BSCA IOB in first position on the system IOB queue.
01 Removes BSCA IOB from the system IOB queue.

Operation

This instruction loads the BSCA IOB onto the system IOB queue or removes the BSCA IOB from the queue, as specified by byte 6.

Program Notes

- Index register 1 must contain the address of any 2-byte field in main storage (other than 0000) when the program issues the queue/dequeue instruction.

- Although this instruction does not actually load a BSCA IOB onto the queue, it must be issued to allow the system to provide an op-end interrupt at the appropriate time.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

ADVANCE PROGRAM LEVEL

Op Code (hex)	Q-Byte[1] (hex)	R-Byte[2] (hex)
Byte 1	Byte 2	Byte 3
F1	00	00

[1] Q-byte should be hex 00, although it is not examined for this instruction.

[2] R-byte should be hex 00, although it is not examined for this instruction.

Operation

The system advances to the next sequential instruction without performing any operation.

Resulting Program Status Byte Settings

This instruction does not affect the program status register.

ENABLE START LIGHT

Op Code[1] (hex)	Q-Byte (hex)	Operand 1 Address[1] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
34	40	00	01

[1] The operand address must be valid, but no operand is addressed by the instruction.

Operation

This instruction sets the START light latch on, causing the START light to come on whenever:

- The system interrupts the program routine to service an I/O device other than the diskette.
- The operator presses the STOP key, then the START key.

DISABLE START LIGHT

Op Code (hex)	Q-Byte (hex)	Operand 1 Address[1] (hex)	
Byte 1	Byte 2	Byte 3	Byte 4
35	40	00	01

[1] The operand address must be valid, but no operand is addressed by the instruction.

Operation

This instruction resets the START light latch to off and immediately turns the START light off. Pressing the START key while the latch is off does not turn the START light on.

Chapter 4. Programming Considerations

INSTRUCTION TIMINGS

The following chart shows the instruction fetch and execution times for each type of instruction.

Instruction Name	Time in Microseconds
Zero and add zoned	49 + 9.2Z + 7(L2 - Z) + 5.6(L1 - L2) ± 2
Add zoned decimal (true add)	62 + 14.8(L2) + 14.8(L1 - L2 - 1) ± 3
Add zoned decimal (complement)	73 + 14.4(L2 - 1) + 14.6(L1 - L2 + 1) ± 3
Subtract zoned decimal (true)	62 + 14.8(L2) + 14.8(L1 - L2 - 1) ± 3
Subtract zoned decimal (complement)	73 + 14.4(L2 - 1) + 14.6(L1 - L2 + 1) ± 3
Recomplement	58 + 14.8(L2 -1) + 15(L1 - L2 + 1) ± 8
Move hex character[1]	28.4 to 33.8
Edit	48 + 5.8N + 14.2(Z - 1) + 12.6C ± 2
Insert and test characters	23.8 + 10.8L
Move character	20.6 + 4.4L
Move inverse	20.6 + 4.4L
Compare logical characters	28.2 + 5.6L
Add logical characters	32.6 + 8.2L
Subtract logical character	33.4 + 7.6L
Sense	69[2]
Load I/O registers	61[2]
Store registers[1]	19.8 to 25.2
Load registers[1]	22.6 to 24.4
Add to register[2]	26.2 to 38.2
Test bits on masked	23.2
Test bits off masked	23.0
Set bits on masked	21.8
Set bits off masked	21.8
Move logical immediate	17.8
Compare logical immediate	27.4
Branch on condition	24.6 (not taken), 28.2 (taken)
Test I/O unit and branch	69[2]
Load address	19.2
Halt program level	14.4
Advance program level	14.4
Jump on condition	26.2 (not taken), 28.8 (taken)
Start I/O function	128[2]
Supervisor call:	
Without I/O operation	Less than 100
With I/O operation	1500 to 2000

Note:

L = length
N = number of nonreplaced characters in operand 1 to the right of the first hex 20
C = number of characters replaced other than low-order zeros (operand 1)
Z = number of low-order zeros inserted before a significant digit is encountered in operand 2
L2 = length of operand 2
L1 = length of operand 1

[1] Depends on Q-byte

[2] Average nominal execution times for all I/O devices and I/O functions

CONDITIONING THE PROGRAM STATUS REGISTER

Instruction	Condition	Binary Overflow	Test False	Decimal Overflow	High	Low	Equal
Zero-Add Zoned Decimal	Set				Operand 2 positive	Operand 2 negative	Operand 2 zero
	Reset				Operand 2 negative	Operand 2 positive	Operand 2 not zero
Add and Subtract Zoned Decimal	Set			Result overflow	Result positive	Result negative	Result zero zero
	Reset				Result negative or zero	Result positive or zero	Result not zero
Edit	Set				Operand 2 positive	Operand 2 negative	Operand 2 zero
	Reset				Operand 2 not positive	Operand 2 not negative	Operand 2 not zero
Compare Logical Characters	Set				Operand 1 greater than operand 2	Operand 1 less than operand 2	Operand 1 equal to operand 2
	Reset				Operand 1 not greater than operand 2	Operand 1 not less than operand 2	Operands not equal
Compare Logical Immediate	Set				Operand 1 greater than immediate data	Operand 1 less than immediate data	Operand 1 equal to immediate data
	Reset				Operand 1 not greater than immediate data	Operand 1 not less than immediate data	Operand 1 not equal to immediate data
Add Logical Characters	Set	Carry out			Carry out and result not zero	No carry and result not zero	Result zero
	Reset	Reset at start of instruction			No carry or result zero	Carry out or result zero	Result not zero

CONDITIONING THE PROGRAM STATUS REGISTER (Continued)

Instruction	Condition	Binary Overflow	Test False	Decimal Overflow	High	Low	Equal
Subtract Logical Characters	Set				Operand 1 greater than operand 2	Operand 1 less than operand 2	Result zero
	Reset				Operand 1 not greater than operand 2	Operand 1 not less than operand 2	Result not zero
Add to Register	Set	Carry out			Carry out and result not zero	No carry and result not zero	Result zero
	Reset	At start of instruction			No carry or result zero	Carry out or result zero	Result not zero
Test Bits On			Tested bits not all ones				
Test Bits Off			Tested bits not all zeros				
Branch or Jump on Condition	Set						
	Reset		Reset if tested	Reset if tested			
Load Register (PSR)	Set	Set if loaded bit 10 on	Set if loaded bit 11 on	Set if loaded bit 12 on	Set if loaded bit 13 on	Set if loaded bit 14 on	Set if loaded bit 15 on
	Reset	Reset if loaded bit 10 off	Reset if loaded bit 11 off	Reset if loaded bit 12 off	Reset if loaded bit 13 off	Reset if loaded bit 14 off	Reset if loaded bit 15 off
System Reset	Set						Equal set on
	Reset	Binary overflow reset	Test reset	Decimal overflow reset	High reset	Low reset	

Chapter 5. Printer Functions

The system is equipped with either a line printer or a serial printer. Printer specifications are listed below.

Specification	Serial Printer	Line Printer
Length of print line	132 positions	132 positions
Pitch (characters per inch)	10	10
Lines per inch (vertical spacing)	6	6
Number of characters in character set	63 plus blank	64 plus blank or 48 plus blank
Maximum print rate (depends on system model)	40 characters/ second 80 characters/ second	50 lines/minute 100 lines/minute 155 lines/minute
Vertical forms control	Standard	Standard
Single form/ledger cards processing	Standard	None

Note: Because the left tractor on the serial printer can be moved only slightly, print position 1 on the serial printer is relatively fixed. This fact should be considered when designing or selecting continuous forms to be printed on the serial printer.

PRINTER CHARACTER SETS

The 48-character set available for the line printer consists of:

Alphabetic A through Z
Numeric 0 through 9
Special characters $, . + - * / % @ # '

The 63-character set used by the serial printer and the 64-character set used by the line printer consists of the above 48 characters plus the following special graphics:

(
)
=
_ (underscore)
:
;
!
"

?
¢
> (greater than)
< (less than)
| (concatenation)
¬ (logical not)
\ (reverse slash)
' (grave accent—not on 63-character set)

Both the line printer and the serial printer substitute blanks for unprintable characters if the unprintable character error is disabled. If the unprintable character error is enabled, each printer stops and a message is issued when the first unprintable character is detected.

OPERATIONAL CHARACTERISTICS

The program controls printing by building printer IOBs, then issuing start-printer-IOB instructions to initiate the actions. Programmed testing establishes printer status for branch decisions, using printer-sense instructions to perform these tests. The system provides an op-end interrupt request at the next operation on the IOB chain.

PRINT DATA AREA

The program must assign a 132-position area of main storage as a print data area (print data field); before chaining the printer operations the program must fully prepare this field.

Data loaded into this field corresponds, character for character, to the maximum-length print line. The leftmost character in the data field is assigned to print position 1; the next storage position is assigned to print position 2, etc. The program must load blanks into all positions of the print data field associated with unassigned print positions. For example, if data is printed from print positions 16-66 on a form, the program must load (1) blanks into print data field locations 1-15, (2) data to be printed (including blanks for spaces) into positions 16-66, and (3) blanks into positions 67-132.

The print data field must be loaded for each line printed on the form.

LINE PRINTER CHARACTER-SET IMAGE AND SIZE

For the line printer only, the following two parameters must be stored before the system can perform printing operations: The character-set size (the number of characters in the character set, excluding the blank), and the character-set image (a list of the sequence of characters on the print belt). The character-set size and image specified in the system configuration record are stored automatically by system control programming after each powering up. If the print belt is changed, the SET procedure can be used to change the character-set size and/or image in the system configuration record. (The system configuration record is described in *IBM System/32 System Data Areas and Diagnostic Aids,* SY21-0532. The SET procedure is described in *IBM System/32 System Control Programming Reference Manual,* GC21-7593.)

These parameters can be loaded by using the load-print-belt-image-register instruction and the load-character-set-size-register instruction. (For the serial printer only, these parameters are not required.) For information about changing print belts, see the *IBM System/32 Operator's Guide,* GC21-7591.

FORMS CONTROL, CONTINUOUS-FORMS MODE

The IOB controls forms movement. One load instruction (load-forms-length-and-current-line-number) defines both the forms length and current line number.

The maximum form length is 84 lines or 14 inches (355.6 mm). The sense-forms-length-and-current-print-line-number instruction indicates the length of the form and the current print-line location (the location of the next print line, if the printer is sensed while forms skipping or spacing is in progress).

The check-reset command, which is used to clear check conditions in the printer, also disables the end-of-forms condition. This allows the printer to process about six more inches (152.4 mm) of forms after the end of forms is indicated. The end-of-forms condition is reenabled when a skip or space operation is performed on the next form. Furthermore, if the forms length is greater than 51 lines, the end-of-forms condition is enabled if the destination line of a skip or space operation is within 15 lines of the specified forms length.

FORMS CONTROL, SINGLE FORM/LEDGER CARDS

Forms control for the single form/ledger cards mode of operation is similar to that for continuous forms mode operation. The IOB controls forms movement. The load-forms-length-and-current-line-number instruction (1) defines the length of the form being used (in print lines, where print lines equal the length of the form times 6) and (2) indicates the print line that is under the print mechanism when the operator completed the initial setup for the job to be run. Forms length and the current print line (the print line under the print mechanism when the instruction is issued) can be determined by issuing a sense instruction. The maximum form length is 84 line spaces or 14 inches (355.6 mm).

Whenever the system is operating in single form/ledger cards mode, space, skip, and print reset functions have expanded meanings:

- An attempt by a program to space or skip into the last one-half inch (three line spaces) on the form causes the System to eject the form and reset the current line number to the value residing in the forms-length register. (An eject is necessary because the friction feed rolls cannot control the last one-half inch of the form.)
- When the system operates in single form/ledger cards mode, the end-of-forms sensor is inoperative. (The operator turns off the end-of-forms sensor by pushing the paper release lever back.)

END-OF-OPERATION INTERRUPTS, PRINTER

The system initiates an end interrupt at the end of each printer operation, processes the interrupt, and updates the IOB as an indication to the main program that the operation is complete. (The program does not need an interrupt handler for the printer.) However, to prevent loss of check status and diagnostic information, the program must immediately handle any checks that occur during execution of an instruction.

PRINT BUFFER

The system has a print buffer that permits overlapping of printing and forms movement with other system I/O operations and instruction execution. The print operation does not alter data stored in the print data field in main storage; therefore, the complete print line is available for error recovery procedures.

PRINTER FUNCTION KEYS

The operator can exercise some control over the printer operation by using printer function keys on the keyboard. These keys are described in Figure 5-1.

Using one of the printer function keys resets all pending check conditions before the keyed function occurs and, on the serial printer, restores the print head if a printer check is pending. At the end of the function, the system generates an interrupt and executes any printer IOBs that were queued while the printer was busy executing the key-initiated function.

PRINTER STATUS BYTES

The system provides error and status indications which the program must interrogate at the end of each printer operation. If desired, the program can then display check or not-ready conditions and indications of appropriate recovery procedures on the display screen. Operator response is always through the keyboard. When the printer-function keys on the keyboard are used, the system resets all pending check conditions before executing the function, and issues an interrupt after the keyboard specified function is performed. This system action initiates any IOBs queued while the printer is busy performing the keyboard-specified function.

The system continuously monitors the printer and the printer-function keys on the keyboard for hardware checks that prevent proper printer IOB processing. Such checks indicate that the printer is not ready to perform the next printer IOB function. All checks should be processed immediately upon receipt of a printer op-end request.

Figures 5-2 and 5-3 show the line printer and the serial printer status bytes, respectively.

Key	Continuous-Forms Mode	Single Form/Ledger Cards Mode
LINE	Moves form up one line.	Moves form up one line to a maximum of forms length minus 3.
PAGE	Skips current form out of printer and moves next form into position to print on line 1.	Inoperative.
PRINT	Prints the six lines displayed on the display screen in the format displayed.	System attempts to print the data displayed on the display screen in the format displayed. **CAUTION** Pressing PRINT when the current line is nine or more lines from the bottom of the form causes the displayed data to print on the form. If there are fewer than nine print lines on the form, do not attempt to print the data; the system does not print all of the data displayed.
RESET	Resets the line counter to 1 to indicate that the print line under the print mechanism (the current print line) is line 1. (The form can be adjusted vertically to align print-line 1 under the print mechanism.)	Resets the line counter to 1, then initiates any operations on the queue. (The form can be adjusted vertically to align print-line 1 under the print mechanism.)

Figure 5-1. Key-Initiated Printer Functions

Priority	Status Byte[1]	Bit	Name of Indication	Indicates	Suggested Action[2]
None	0	0	CE diagnostic		None
None	0	1	CE diagnostic		None
None	0	2	CE diagnostic		None
4	0	3	Emitter check	Current print line may be in error.	1, 5, 13
6	0	4	Buffer data check	Data printed on current line is in error.	1, 5, 13
6	0	5	Hammer parity check	Incorrect printing.	1, 5, 13
None	0	6	No-op	An operation is abnormally terminated.	1, 8
7	0	7	Unprintable character	Data on last line printed includes one or more unprintable characters.	13, 9
8	1	0	Forms misfeed	Forms fail to move; multiple space instructions may be in error; overprinting may have occurred.	9
3	1	1	Belt speed check	Belt fails to start (no data lost).	1, 6, 13, 5
2	1	2	Carriage sync check	Improper carriage motion with possible incomplete print line.	2, 3, 4, 13, 5
9	1	3	End of forms	Last form passes end-of-forms switch (no data lost)	10, 13, 8
11	1	4	Throat interlock	Feed throat is not ready.	12, 13, 8
1	1	5	Coil current check	Hammer stayed on too long, current line has error.	1, 13, 5
5	1	6	Belt sync check	Printer is out of sync; last two lines may be in error.	3, 7
10	1	7	Cover interlock	Cover is not latched.	11, 13, 8
None None	2	0	Printer version	0 = line printer installed, 1 = serial printer installed.	None
None	2	1	Not used	These bits are not used.	None
	2	2			
	2	3			
	2	4			
	2	5			
	2	6			
	2	7			

[1] Status byte 3 has no meaning to the program.

[2] Actions are described in Part 2 of this figure.

Figure 5-2 (Part 1 of 2). Line Printer Status and Suggested Action

Suggested Action	
Number	Description
1	Reset check.
2	Manually position form to line 1.
3	Press RESET TO LINE 1 key.
4	Using the NEW LINE key on keyboard, space down to the last line printed.
5	Reissue the last start-printer-IOB instruction (overprinting occurs).
6	Ensure that print belt is correctly mounted on pulleys and that movable transducer is positioned correctly.
7	Reset check and continue. Data is lost; only last line can be retried.
8	Continue processing.
9	Inspect last line for overprinting. If overprinted, all but current line data is lost and only current line can be recovered. If data is lost, perform all steps; otherwise, do steps 4 and 5. 1. Sense the current print line before initiating recovery. 2. Manually position the form to line 1 (of the current form, if it is not damaged; if current form is damaged, to line 1 of the next form). Printing continues on the correct line of the form used. 3. Press the RESET TO LINE 1 key. 4. Correct the misfeed condition. 5. Skip to the line saved by step 1. The operation continues from the current line.
10	Thread new continuous forms through the forms tractor. Condition must be cleared before any further printer operations can be performed.
11	Close cover. Condition must be cleared before any further printer operations can be performed.
12	Close throat. Condition must be cleared before any further printer operations can be performed.
13	Program must provide operator instructions for the recovery needed for this type of error and for the job being run; the operator must perform the procedure specified by the program.

Figure 5-2 (Part 2 of 2). Line Printer Status and Suggested Action

Priority	Status Byte[1]	Bit	Name of Indication	Indicates	Suggested Action[2]
6	0	0	Forms hung check	Printer does not properly advance form; current line may be in error.	9, 2, 3, 4, 11, 10
7	0	1	Horizontal check	Error occurs during horizontal head movement.	1, 6, 2, 7
6	0	2	Forms runaway	Form is moved more than the number of line spaces specified; the system cuts carriage drive motor power. (Sense the current line number.)	9, 2, 3, 4, 11, 10
2	0	3	End of forms	Last form passed the END-OF-FORMS switch; more forms must be loaded into printer before processing the next form. (Printer not-ready bit is also on with this bit.)	2, 5, 1, 11
None	0 0 0	4 5 6	Not used	These bits are not used.	None
None	0	7	No-op	An operation is abnormally terminated.	1, 11
3	1	0	Printer not ready	End-of-forms condition or printer problem prevents printer operation.	1, 6, 2, 7
1	1	1	Wire check	At least one print hammer fires for excessive interval.	1, 6, 2, 7
None	1	2	Not used	This bit is not used.	None
None	1	3	Not used	This bit is not used.	None
4	1	4	Storage parity check	There is invalid parity in printer output data. (Buffered data is still good.)	2, 6, 1, 7
None	1	5	Not used	This bit is not used.	None
5	1	6	Unprintable character check	The printer attempts to print a character not defined in the character set.	2, 1, 7, 8
None	1	7	Not used	This bit is not used.	None
None	2	0	Printer version	0 = line printer installed. 1 = serial printer installed.	None
None	2 2 2 2	1 2 3 4	Not used	These bits are not used.	None
8	2	5	Forms mode	0 = continuous forms mode. 1 = cut forms mode.	12, 1, 2
None	2 2	6 7	Not used	These bits are not used.	None

[1] Status byte 3 has no meaning to the program.

[2] Actions are described in Part 2 of this figure.

Figure 5-3 (Part 1 of 2). Serial Printer Status and Suggested Action

Suggested Action	
Number	**Description**
1	Reset check.
2	Tell the operator what action to take for this type of error for the type of job being run.
3	Manually position form to line 1 of the current form or to line 1 of the next form. Printing continues on the correct line of whichever form is selected.
4	Press RESET TO LINE 1 key.
5	Insert new forms in the printer.
6	Current line may be in error. Operator may elect to overprint the line in which the error occurs or reprint that data on a new line.
7	Re-execute the last start-printer-IOB instruction.
8	If desired, suppress the unprintable character check by turning off bit 0 of the printer IOB Q-byte.
9	Sense the current line counter.
10	Skip to the saved current line and continue processing.
11	Continue processing.
12	Instruct the operator to select the proper mode (cut-forms continuous forms) for the program being run.

Figure 5-3 (Part 2 of 2). Serial Printer Status and Suggested Action

PRINTER IOB AND NCPODSW

Two areas of main storage that are used for programming the serial or line printer are the IOB field and the NCPODSW field. The IOB field (Figure 5-4) contains printer parameters and instructions that control the printer. The NCPODSW field is a single-byte area that can be used as a programmable and testable switch (Figure 5-5) and is situated at address 001A in main storage.

Displacement of Leftmost Byte of Hex	IBM Program Label	Length in Bytes	Field Description
0	PODCHAIN	2	This is the storage address of the next IOB in the chain. IOBs are chained only if the file requires more than one IOB.
2	PODDCMP (overlaid by PODDQ)	1	This is the printer completion code (bit significant): X'40' = operation complete X'20' = IOB active
2	PODDQ	1	This is the printer Q-byte: *Bit On* *Meaning* 0 Check for unprintable characters and set status bit if detected. 1 IOB operation is complete. 2 IOB operation (indicated by 6 and 7) is in progress; it is used by control storage to maintain active status of IOB. 3-5 Reserved. 6-7 Set to indicate desired operation: 00 = check reset 01 = space (R-byte must be 01, 02, or 03) 10 = skip 11 = print
3	PODDR	1	The printer R-byte specifies (in binary) the amount of carriage movement for any space or skip operation.
4	PODDNEXT	2	This is the address of the next available system IOB. This field is optional.

Figure 5-4. Printer IOB Format

Bit	Meaning	Remarks
0	Error during last operation	The system sets this bit to 1 to indicate the error. The program must reset this bit to 0.
1	Print buffer not busy	The program must set this bit to 0 to indicate a buffer-loaded condition. The system sets this bit to 1 when data is successfully printed.
2	Print operation complete	The system sets this bit to 1 upon completion of any print operation (whether successful or not). The program can reset the bit as desired.
3-7	Not used	The system should set these bits to 0.

Note: This byte is located at address 001A in main storage.

Figure 5-5. NCPODSW Byte (Testable Switch) Bit Meanings and Address

The disk provides high performance magnetic storage media for the IBM System/32. Disk data tracks reside in two concentric regions on a single surface accessed by two heads mounted on a common access arm. A third head generates track servo timing and index and sector timing (Figure 6-1).

PHYSICAL CHARACTERISTICS

The disk unit provides 5.0 or 9.1 million bytes of accessible storage. The specifications of the drive are:

Item	Magnitude	Unit
Rotational speed	2964 ± 3.0%	RPM
Average rotational delay or latency	10.1	ms
Average seek time (excluding latency) over one-third of the disk:		
5.0 megabytes	70.0	ms
9.1 megabytes	72.5	ms
Capacity		
Sectors per track	60	sectors
Bytes per sector	256	bytes
Bytes per track	15,360	bytes
Tracks per cylinder	2	tracks
Bytes per cylinder	30,720	bytes
Cylinders (5.0 megabytes)	169	cylinders
Capacity (5.0 megabytes)	5,053,440	bytes
Cylinders (9.1 megabytes)	303	cylinders
Capacity (9.1 megabytes)	9,169,920	bytes
Data rate (1.13 µs/byte or 141 ns/bit)	889,000	bytes/sec (nominal)

FILE ORGANIZATION

All data tracks are on the top surface of the disk (Figure 6-2). Cylinder 0 is on the innermost part of the surface; cylinder 168 (5.0 megabyte disk) or 302 (9.1 megabyte disk) is on the outermost area of the disk. The organization of data and use of cylinders is shown in Figures 6-1 and 6-2.

Cylinder Number	Head 0	Head 1
000	IMPL and diagnostics	System control storage
001	System control transients	VTOC
002	Alternate sectors	Alternate sectors
003 to 166 or 003 to 300	Customer data files, program products, and application programs	
167 or 301[1]	Diskette diagnostic cylinders	
168 or 302[1]	CE cylinder	CE cylinder

[1] 168 is the CE track for a 5.0-megabyte disk; 167 is a diagnostic track. 302 is the CE track for a 9.1-megabyte disk; 301 is a diagnostic track.

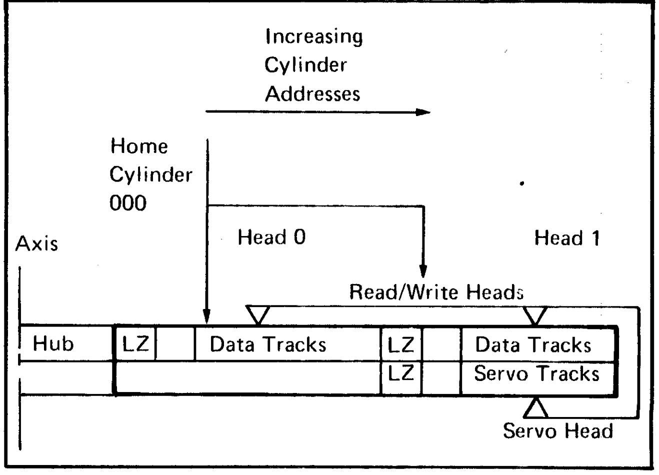
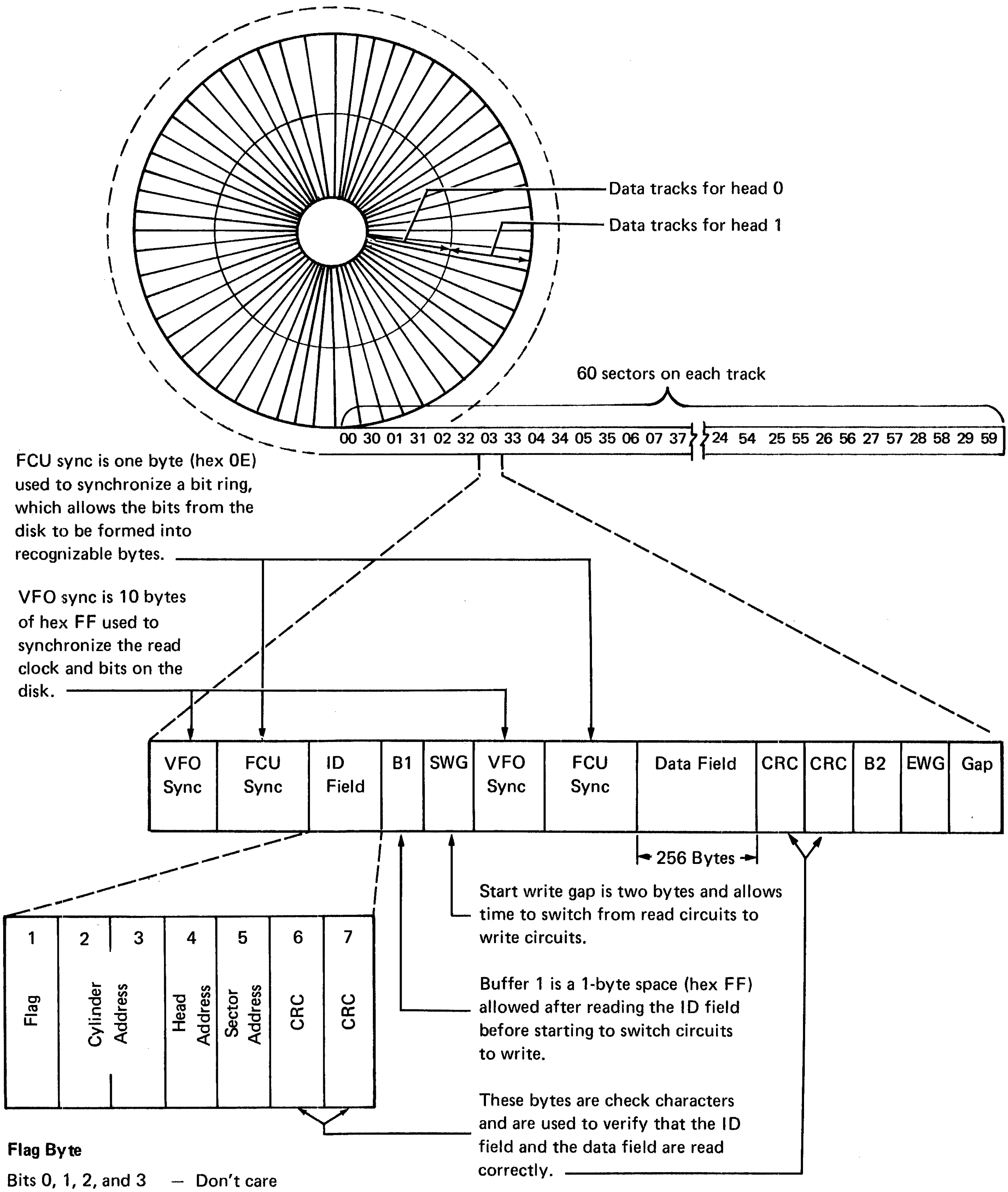


Figure 6-1. Disk Surface Partitioning Diagram



FCU sync is one byte (hex 0E) used to synchronize a bit ring, which allows the bits from the disk to be formed into recognizable bytes.

VFO sync is 10 bytes of hex FF used to synchronize the read clock and bits on the disk.

Start write gap is two bytes and allows time to switch from read circuits to write circuits.

Buffer 1 is a 1-byte space (hex FF) allowed after reading the ID field before starting to switch circuits to write.

These bytes are check characters and are used to verify that the ID field and the data field are read correctly.

Flag Byte

Bits 0, 1, 2, and 3	– Don't care		
Bit 4	– Unassigned		
Bit 5	– Data field may contain bad data		
Bits 6 and 7	– 00	– Good primary sector	
	10	– Defective primary sector	
	01	– Good alternate sector	
	11	– Defective alternate sector	

Figure 6-2. Disk Sector Detailed Description

OPERATIONAL CHARACTERISTICS

Initiating a Disk Operation

The control program schedules and controls the transfer of data between main storage and disk storage. The program branches to main storage location hex 0008 with XR1 pointing to a previously assembled 24-byte IOB located somewhere above hex 0800 in main storage. The system accesses the IOB and performs the specified operation.

The most commonly used operations are:

- *Seek* – positions the disk access mechanism at a specified location.
- *Read* – seeks (if necessary) and reads data from the disk data field in main storage.
- *Write* – seeks (if necessary) and records data from the disk data field in main storage onto the disk, with or without verification.
- *Scan* – seeks (if necessary) and sequentially examines a disk file for a specified item of data.

All of the usual load, test, and sense operations are performed under control of the system.

Disk Addressing, Relative Sector Addresses

One type of disk address used to identify disk data areas is a relative sector address (SS). The SS is a binary number, starting at 0001, which increases by 1 for each sector on the track (cylinder). This disk address starts at cylinder 1, head 1, sector 0. The relative sector addressing extends through the last physical sector on the disk. Tracks 0 through 2 cannot be addressed using SS, so the data on these tracks cannot be accessed or accidentally destroyed. (Cylinder 0, head 0 and 1, and cylinder 1, head 0, cannot be addressed from main storage.)

The formula for determining SS is:

$$SS = 120C + 60H + S-179$$

where
C = cylinder number
H = head designation
S = sector number

Disk Addressing, Actual Sector Addresses

The second type of disk addressing that can be used is optional, and requires that the user provide the actual 6-byte control field that the IBM disk IOS (input/output system) normally provides from the relative sector address. This 6-byte control field, referred to as NFCCHS, is found in bytes 19-24 of the IOB. The control field is defined as:

Byte	Name	Description
1	N	The N-byte specifies the number of sectors to be accessed. (N is one less than the number of sectors to be accessed.) All multiple sector data operations can cross track and cylinder boundaries; therefore, there are no invalid N-bytes. When one sector is to be transferred, the proper N-byte is hex 00.
2	F	The flag byte identifies (flags) the usability of a sector. Only the four low-order bits (4-7) are used for F-byte orientation. The four high-order bits (0-3) are not compared and can be used at the discretion of the programmer. The significance of bits in the flag byte is:

Bits	Meaning
0-3	Not used.
4	Unassigned; can be used for unique flagging purposes.
5	Data field may contain bad data. The alternate sector assignment routine uses this bit to flag data that could not be recovered error-free.
6-7	Condition of the sector: 00 = good primary sector 01 = good alternate sector 10 = defective primary sector 11 = defective alternate sector

Byte Name Description

3-4 CC The cylinder address is a 2-byte logical binary address. A valid data cylinder address is a function of the installed disk capacity:

Capacity	Cylinder	Hex Address
5.0 megabytes	003 to 166	0003 to 00A6
9.1 megabytes	003 to 300	0003 to 012C

An invalid CC address sets the invalid-seek-address status (byte 1, bit 2):

During any seek operation

During a data operation, if the system detects an end-of-file condition

During a data operation, if the system also posts the no-record-found indication

5 H The head address is a single-byte binary address used by all disk commands to address one of two heads per cylinder. The valid head addresses are hex 00 and 01. Any attempt to execute a data operation with an invalid head address sets the no-record-found status.

6 S The sector address is a single-byte binary address used by all disk commands to address one of the 60 sectors per track. The disk unit uses an interleaving technique for arranging sectors on a track. Sector interleaving is used because of the limitations on read and write recovery time. Logical addresses and physical addresses are arranged on the track as follows:

Physical Address	Logical Address
00	00
01	30
02	01
03	31
04	02
05	32
06	03
07	33
08	04
09	34
10	05
11	35
12	06
13	36
14	07
15	37
16	08
17	38
18	09
19	39
20	10
21	40
22	11
23	41
24	12
25	42
26	13
27	43
28	14
29	44
30	15
31	45
32	16
33	46
34	17
35	47
36	18
37	48
38	19
39	49
40	20
41	50
42	21
43	51
44	22
45	52
46	23
47	53
48	24
49	54
50	25
51	55
52	26
53	56
54	27
55	57
56	28
57	58
58	29
59	59

If the system cannot find a logical sector address, it sets the no-record-found status bit.

The disk address in the control field specifies one of the following:

- The destination track address for a seek command
- The starting track and sector address, and the number of sectors accessed (N + 1) in a data operation.

All data operations begin when the control field sector is located by a series of comparisons with either a physical or logical disk sector address (orientation procedure). The comparisons are limited to track boundaries. The NFCCHS control field in the IOB (Figure 6-3) is not changed during the operation. It contains the same sector address that is in the IOB when the operation is issued.

00	01	02	03	04	05	06	07
IOS queue pointer (need not be initialized)		Completion code (need not be initialized)	Q-byte of start I/O	R-byte of start I/O	SS[1]		Number of sectors minus 1 (N)[1]

08	09	0A	0B	0C	0D	0E	0F
Pointer to calling routine's data area start address		Device status sense information (need not be initialized) (Sense bytes 0-1)		Device status sense information (need not be initialized) (Sense bytes 2-3)		Disk I/O error counter (need not be initialized)	Indicator to request special I/O operations

10	11	12	13	14	15	16	17
Enqueueing priority	Unassigned	N	F	C	C	H	S
		Six-byte disk control field (if used)					

End of Basic IOB

[1] Bytes 05, 06 and 07 are present, but not used.

Figure 6-3 (Part 1 of 4). Disk IOB Format

Displacement of Leftmost Byte in Hex	IBM Program Label	Length in Bytes	Field Description
0	IOBCHN	2	This is the chain pointer field used by the system. It contains either the address of the next IOB on the queue, or hex FFFF if this is the last IOB on the queue. This field need not be initialized and is set up by the system. When the I/O operation is complete, this field contains the SS of the last sector accessed. For requests using NFCCHS, this field is unpredictable.
2	IOBCMP	1	This is the completion-code byte set by the system (when the I/O operation is complete) to inform the calling routine of the requested operation's status. It is the responsibility of the calling routine to check this byte before assuming the data is transferred without error. The following codes are used: *For All Operations except Scan* X'40' – successful completion X'41' – permanent I/O error *For Scan Operations* X'40' – scan hit X'41' – permanent error X'42' – scan not found X'44' – scan equal found
3	IOBQB	1	The calling routine sets this byte to indicate the type of operation desired. (This is equivalent to the Q-byte in the I/O instructions.) Figure 6-4 shows the possible values and their meanings for the Q-byte.
4	IOBRB	1	The calling routine sets this byte to further define the type of operation requested. This is equivalent to the R-byte in the I/O instructions. Figure 6-4 shows the possible values and their meanings for the R-byte.
5	IOBSS	2	This is the SS from the beginning of the disk, starting at sector 1. Sector addressing is independent of the physical organization of the disk and sectors are numbered consecutively from 1 to the end of the disk. The system modifies this field only at the end of a scan operation. After a scan operation, SS points to the sector where the hit occurred, or to the next sector if not hit occurred. The caller must initialize this field before requesting the I/O operation if the program is using the relative sector addressing method. Otherwise, see bit 3 of the flag byte.
7	IOBNB	1	The calling routine must set this byte to the hex value of the number of sectors, minus one, involved in the data transfer. For example, if five sectors are to be processed, this byte contains hex 04. This byte is not changed by the system.
8	IOBDAT	2	The calling routine must set this 2-byte pointer to the address of the leftmost byte of the calling routine data area. The data area is given in multiples of 256 bytes, with a minimum of 256 required.

Figure 6-3 (Part 2 of 4). Disk IOB Format

Displacement of Leftmost Byte in Hex	IBM Program Label	Length in Bytes	Field Description
A	IOBSNS1	2	These two bytes are used by the system to keep bytes 1 and 2 of the device status sense information. This field need not be initialized. These bytes are not filled unless an error occurs during processing.
C	IOBSNS2	2	These two bytes are used by the system to keep bytes 3 and 4 of the device status sense information. This field need not be initialized. These bytes are not filled unless an error occurs during processing.
E	IOBERR	1	The system uses this byte to count the retries required to complete an I/O operation. This field need not be initialized.
F	IOBFLG	1	This byte contains bit indicators to request special handling of I/O operations. The bit settings are: *Bit* *Meaning when Set to 1* 0 No error recovery to be attempted. 1 No verification is to be done on write operations by the system control program. 2 This bit is set if the routine calling the disk drive resides in the transient area. This bit indicates that error logging is not evoked during the execution of this disk operation. It is deferred until the transient area is not busy. 3 The calling program is supplying the NFCCHS within the IOB. The SS is not used. NFCCHS immediately follows the unassigned field (byte 18) in the IOB. If the CC field of NFCCHS is equal to 0, or if CC is equal to 1 and H equals 0 and the request is from main storage, a program check results. 4 The request does not use disk data management (no DTF). It is ignored by the system. 5 Do not return on an unrecoverable disk error. When a hard disk error occurs, the system issues an abnormal termination message. If this bit is off when a hard disk error occurs, the control returns to the user with a hex 41 completion code. 6 This bit is reserved and must be set to 0. 7 Prevent the drive from performing an automatic seek. *Note:* Bits 0 through 5 and bit 7 are set by the calling routine and are never modified by the system. Bit 6 is used only by the system and must not be modified by the calling routine. If the calling routine does not desire any special handling, this byte must be initialized to hex 00.

Figure 6-3 (Part 3 of 4). Disk IOB Format

Displacement of Leftmost Byte in Hex	IBM Program Label	Length in Bytes	Field Description
10	IOBPRTY	1	This byte is reserved for priority queueing. The caller should set it to hex 00.
11	IOBPAD	1	This byte is presently unassigned and should be initialized to hex 00.
12	IOBN	1	This is the 6-byte control field (NFCCHS) of the calling routine. (This field is described in this chapter under *Disk Addressing, Actual Sector Addresses.)* To access this field, turn on bit 3 in the flag byte (displacement hex F into the IOB). This field does not change.
13	IOBF	1	
14	IOBCC	2	
16	IOBH	1	
17	IOBS	1	

Figure 6-3 (Part 4 of 4). Disk IOB Format

Operation		Q-Byte	R-Byte[1]
		0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
Control	Seek	1 0 1 0 0 0 0 0	— — — — — — — 0
	Recalibrate	1 0 1 0 0 0 0 0	— — — — — — — 1
Read	Data	1 0 1 0 0 0 0 1	N N — — 0 N 0 0
	ID	1 0 1 0 0 0 0 1	N N — — N N 0 1
	Diagnostic	1 0 1 0 0 0 0 1	N N — — 0 N 1 0
	Verify	1 0 1 0 0 0 0 1	N N — — 0 N 1 1
Write	Data	1 0 1 0 0 0 1 0	N N — — 0 N — 0
	ID	1 0 1 0 0 0 1 0	N N — — N N — 1
Scan	Equal	1 0 1 0 0 0 1 1	N N — — 0 N 0 0
	Low or equal	1 0 1 0 0 0 1 1	N N — — 0 N 0 1
	High or equal	1 0 1 0 0 0 1 1	N N — — 0 N 1 0

[1] The R-byte control field specifies the function modifiers. The significance of each bit is shown in Figure 6-4, part 2.

— = Don't care

N = The bit can have a 0 or 1 value

Figure 6-4 (Part 1 of 2). Disk Start I/O Q-Byte and R-Byte Codes

Bit	Description
0	Data field wrap control causes the same 256-byte data area to be used for each sector accessed. Therefore, N+1 sectors use the same data area.
1	Control storage low only causes only the low-order byte positions of control storage to be accessed. Bit 5 must be set concurrently.
2	Don't care.
3	Don't care.
4	Fast sync extend control is used only with the read-ID and write-ID operation to read or write a skewed ID field.
5	Control storage address select allows a control storage address to be accessed instead of a main storage address.
6-7	These two bits modify the N portion of the Q-byte and further define the disk I/O operation.

Figure 6-4 (Part 2 of 2). Disk Start I/O Q-Byte and R-Byte Codes

DISK OPERATIONS

A start-disk instruction is always initially accepted, but may be ignored (no-oped) if an outstanding check condition is considered to be nonrecoverable. If an outstanding check condition is recoverable, the system resets the check status bits prior to instruction execution.

Control Seek

The disk access mechanism moves the read/write head to the logical cylinder and head address specified by SS or to the actual control field address (NFCCHS). Any attempt to seek beyond the disk capacity sets the invalid-seek-address status bit. The disk is busy during execution of the seek. The system does not indicate invalid head addresses and does not perform seek verification. Normally, a seek failure results in a no-record-found status when the program attempts a subsequent data operation. The program must provide the SS or the CCHS portion of the control field. The system automatically performs a seek at the end of the cylinder during multiple-sector logical-data operations that overflow a cylinder (that is, include sectors on more than one cylinder). The logical-data operations include read-data, read-verify, write-data, and any scan operation. The system automatically seeks to the next logical track by increasing the cylinder number by 1 and setting the head selected to 0. If the program attempts to seek beyond end of file, the system turns on the invalid-seek-address status bit.

The seek time for the 5.0-megabyte drive is:

Cylinder to cylinder	13.0 ms (maximum)
Average seek (56 cylinders)	70.0 ms (maximum)
Maximum seek (168 cylinders)	179.8 ms (maximum)

The seek time for the 9.1-megabyte drive is:

Cylinder to cylinder	14.2 ms (maximum)
Average seek (101 cylinders)	72.5 ms (maximum)
Maximum seek (302 cylinders)	166.0 ms (maximum)

The general seek equation for determining seek time for the 5.0-megabyte disk is:

$T = 13$ if the seek crosses one cylinder boundary
$T = 16$ if the seek crosses two cylinder boundaries
$T = 16.5 + 1.9(N-3)$ if the seek crosses three to seven cylinder boundaries
$T = 23.0 + 0.98(N-8)$ if the seek crosses 8 to 168 cylinder boundaries

where:
N = number of cylinder boundaries crossed
T = maximum seek time, excluding latency, in milliseconds

The general seek equation for determining seek time for the 9.1-megabyte disk is:

$T = 14.2 + 3.25(N-1)$ if the seek crosses 1 to 5 cylinder boundaries
$T = 22.5 + 0.78(N-6)$ if the seek crosses 6 to 15 cylinder boundaries
$T = 32.5 + 0.47(N-16)$ if the seek crosses 16 to 302 cylinder boundaries

where:
N = number of cylinder boundaries crossed
T = maximum seek time, excluding latency, in milliseconds

Control Recalibrate

The recalibrate operation initializes the disk actuator mechanism to the fixed reference point, cylinder 0. The disk is busy during execution of the seek.

Read Data

This operation reads data, beginning at the logical sector specified by the SS or the NFCCHS. The system reads N+1 sectors into contiguous positions of storage, beginning at the data area start address specified by the IOB. Head switching and cylinder overflow are performed automatically by the system during this operation.

Read ID (Identifier)

The read-ID operation recovers a 5-byte identifier field (FCCHS), from the logical sector address specified by the SS or the NFCCHS. The system translates the logical sector address into a physical sector address, placing the recovered ID in the data area specified by the data area start address in the IOB. The system starts the read-ID operation at the index marker and counts the sequential sector markers to locate the specified sector. An invalid sector number sets the no-record-found bit in the status byte. The physical sectors are numbered sequentially from 00 through 59.

The read-ID operation can be modified by bit 4 of the R-byte to recover a skewed or displaced ID field. The system provides no specific indication that the ID field is displaced or skewed. However, if a standard read-ID instruction fails to recover the ID field, the system automatically attempts to recover the displaced ID using the read-ID-with-skew command.

Read Data Diagnostic

This operation recovers a single sector data field when a defect in the identifier area prevents sector orientation. Data recovery relies on the fact that the physical sector position is known and that the data field sync byte can be recognized. The system starts the read-data-diagnostic operation at the index marker and counts sequential sector marks until the logical sector specified in the SS or the NFCCHS reaches the read/write head. Then the system reads the data field into contiguous positions of the data area specified by the data area start address in the IOB.

Read Verify

The read-verify operation is similar to the read-data operation, except that the system does not transfer data to main storage. Read-verify simply verifies that the specified sectors are readable, not that the data has been written correctly.

Write Data

This operation transfers data from storage to the specified disk sector. The system starts the write-data operation by seeking the logical sector address specified by the SS or the NFCCHS. When the system detects a matching address, it transfers the 256 bytes of data from the data area specified by the data area start address in the IOB to the sector data field on disk. This operation continues until data from contiguous positions in storage has been written into N+1 disk sector. Head switching and cylinder overflow are automatic system functions.

System-generated CRC bytes are appended to each sector written.

Write ID (Identifier)

The write-ID operation initiates the writing of a single sector format containing only the identifier field. The operation does not overwrite or destroy any part of the data field. The program must assemble the sector format containing the ID field and store it in the main storage data area addressed by the data area start address in the IOB. This sector format must be set up before the program issues the write-ID request. Writing begins at the logical sector specified by the SS or the NFCCHS. The sector format in the data area must appear as follows:

Value	Number of Bytes (Decimal)
X'FF'	10 (This number must be 64 for write-ID displacement.)
X'0E'	6
F	
C	
C	
H	
S	
X'FF'	2

The system generates a CRC field and places it at the end of the ID field. Alternate sector or track assignment procedures use the write-ID operation. If a sector defect occurs within the ID region, the system displaces the beginning of the ID field 64 bytes down the track. ID displacement increases the first VFO sync region of the sector format in the data area in storage by 64 bytes hex FF. The data field is partially destroyed.

Scan Equal

The scan-equal operation begins at the logical sector specified by the SS or the NFCCHS and compares N+1 sectors read from disk to a single 256-byte main storage data field addressed by the data area start address in the IOB. Hex FF bytes can be used to mask any bytes within the main storage data field that are not to be examined. The scan operation ends when the system detects the first matching sector, and the system resets the scan-not-hit sense bit, and sets the scan-equal sense bit. Head switching and cylinder overflow are automatic system functions.

Scan Low or Equal

The scan-low-or-equal operation is similar to the scan-equal operation. It starts at the logical sector specified by the SS or the NFCCHS and compares N+1 sector from disk to a single 256-byte data field in storage.

A low-or-equal condition means that the disk data field contains either a high-order byte with a lower binary value than the corresponding byte in the data field in storage, or that the fields are equal. The first successful sector scan terminates the operation and resets the scan-not-high sense bit. Head switching and cylinder overflow are automatic system functions.

Scan High or Equal

This operation is similar to the scan-low-or-equal operation except that the system performs a scan test for high or equal. A high-or-equal condition means that the disk data field contains either a high-order byte with a greater binary value than the corresponding byte in the data field in storage, or that the fields are equal.

DISK CHECK CONDITIONS AND STATUS

These bits show the conditions that result after execution or attempted execution of an operation requested by the IOB. The system automatically performs disk error recovery routines; this disk function need not be programmed. Refer to Figure 6-5 for a list of operations that set the following bits:

Byte 0

Bit **Meaning**

0 *Disk not ready* indicates either that power-up is delayed or that the drive (1) is not up to rotational speed, (2) failed to recalibrate after initial power-up; (3), is unsafe, or (4) has a motor brake failure. Turning power off, then on, resets this bit.

1 *Alternate sector process* indicates the error recovery failed during an attempt to process an unassigned alternate sector. This bit is set on the initial seek from the primary track to the alternate track and is reset when seek back to the primary track is complete. A disk read, write, or seek operation resets this bit.

2 *Sector sync check* indicates the sync byte compare failed on either an ID or data field sync byte. A disk read, write, or seek operation resets this bit.

3 *Off track check* indicates an off-servo-track condition. A disk read, write, or seek operation resets this bit.

4 *CRC check* indicates a cyclic redundancy check miscompare after initial ID orientation on (1) a data field during a read or scan operation, or (2) a subsequent ID field. The no-record-found status is set concurrently. A disk read, write, or seek operation resets this bit.

5 *Parallel parity check* indicates a parity error on the data bus-out during a disk I/O operation. (The equipment-check status bit is set concurrently.) A disk read, write, or seek operation resets this bit.

6 *Write echo check* indicates a miscompare between serial write data and write data echo from the file. (The attachment-equipment-check status bit is set concurrently.) A disk read, write, or seek operation resets this bit.

7 *Channel overrun* indicates that the system did not move data between the disk and storage quickly enough, so at least one byte was lost. (The attachment-equipment-check status bit is set concurrently.) A disk read, write, or seek operation resets this bit.

Byte 1

Bit Meaning

0 *No operation (no-op)* is set when a disk read, write, or seek instruction is issued while the drive is not ready. The next disk read, write, or seek operation resets this bit.

1 *Data unsafe* indicates that a select-unsafe, write-unsafe, or servo-unsafe condition is detected. This implies that the hardware has failed so that errors may go undetected in processing or that data may be lost. (The not-ready status bit is set concurrently on a second retry only.) The next read, write, or seek operation resets this bit.

2 *Invalid seek address* indicates that the program specified a seek beyond the cylinder capacity of the drive. A disk read, write, or seek operation resets this bit.

3 *Attachment equipment check* indicates a hardware check. A list of the conditions that set equipment check are found in Figure 6-5. A disk read, write, or seek operation resets this bit.

4 *No record found* indicates that the sector specified in the SS or the NFCCHS was not found within one complete revolution. This generally results from a seek failure, an encountered alternate sector, an incorrect control field specification, or surface defects in the ID field. A disk read, write, or seek operation resets this bit.

5 *Scan equal hit* indicates a scan-equal condition during a scan operation. A disk read, write, or seek operation resets this bit.

6 *Scan not hit* indicates that a scan-hit condition was not detected during a scan operation. The status is saved during nonscan operations until this error is corrected. A disk read, write, or seek operation resets this bit.

Note: This bit is on during non-scan operations.

7 *Seek check* indicates that a hardware failure detected during a seek operation was caused by (1) a failing actuator, (2) an actuator reaching the restricted behind-home region, or (3) the phase locked oscillator being out of sync during a data operation. A read, write, or seek operation resets this bit.

Byte 2

Bit Meaning

0 *Serializer-deserializer check* indicates that a mismatch was detected between parallel and serial hardware checks. (The attachment equipment check status bit is set concurrently.) A disk read, write, or seek operation resets this bit.

1 *Write check trigger* indicates that write current is being applied to data head when it should not be. A disk read, write, or seek operation resets this bit.

2 *Channel transfer error* indicates that a hardware failure is detected during data transfer in the I/O channel.

3 This bit is not used.

4 This bit is not used.

5 This bit is not used.

6 This bit is not used.

7 *Head select bit* on indicates read/write head 01 was selected. If this bit is off, read/write head 00 was selected.

Byte 3

Bit Meaning

0 This bit is not used.

1 *Select unsafe* indicates incorrect head selection during a write operation. (The data-unsafe and not-ready status bits are set concurrently.) Correcting the error condition resets the select unsafe bit.

2 *Write unsafe* indicates that one of the following conditions were detected during a write operation: (1) write was selected and not write transitions occurred or (2) write was not selected and write current was on. (The data-unsafe and not-ready bits are set concurrently.) Correcting the error condition resets the write unsafe bit.

3 *Brake failure* indicates that a failure in the disk drive motor brake is detected. (The not-ready status bit is set concurrently.) The system removes power from the disk. Correcting the error condition resets the brake failure bit.

4 *Servo unsafe* indicates that one of the following conditions was detected during a write operation (1) write was selected and the head was off the track, or (2) write was selected and phase lock oscillator was out of sync. (The data-unsafe and not-ready bits are set concurrently.) Correcting the error condition resets the servo-unsafe bit.

5 This bit is not used.

6 *9.1 megabyte drive* indicates that a 9.1 megabyte drive is attached.

7 This bit is not used.

Status Byte 0

Bit	Name and Indicates	Seek	Recalibrate	Read Data	Read ID	Read Diagnostic	Read Verify	Write Data	Write ID	Scan High or Equal	Scan Equal	Scan Low or Equal
0	*Disk not ready* is part of unit-check condition.	x	x	x	x	x	x	x	x	x	x	x
1	*Alternate sector process* is part of unit-check condition.			x			x	x		x	x	x
2	*Sector sync check* is part of unit-check condition. Equipment check is set concurrently.			x		x	x	x		x	x	x
3	*Off track check* is part of unit-check condition. Equipment check is set concurrently.	x	x	x	x	x	x	x	x	x	x	x
4	*CRC* is part of unit-check condition. Equipment check is set concurrently.			x	x	x	x			x	x	x
5	*Parallel parity check* is part of unit-check condition. Equipment check is set concurrently.	x		x	x	x	x	x	x	x	x	x
6	*Write echo check* is part of unit-check condition. Equipment check is set concurrently.							x	x			
7	*Channel overrun* is part of unit-check condition. Equipment check is set concurrently.			x	x	x	x	x	x	x	x	x

Status Byte 1

Bit	Name and Indicates	Seek	Recalibrate	Read Data	Read ID	Read Diagnostic	Read Verify	Write Data	Write ID	Scan High or Equal	Scan Equal	Scan Low or Equal
0	*No operation* is part of unit-check condition.	x	x	x	x	x	x	x	x	x	x	x
1	*Data unsafe* is part of unit-check condition.			x	x	x	x	x	x	x	x	x
2	*Invalid seek address* is part of unit-check condition.	x										
3	*Attachment equipment check* is part of unit-check condition.	x		x	x	x	x	x	x	x	x	x
4	*No record found* is part of unit-check condition.			x	x	x	x	x		x	x	x
5	*Scan equal hit.*									x	x	x
6	*Scan not hit.*									x	x	x
7	*Seek check* is part of unit-check condition. Equipment check is set concurrently.	x	x	x	x	x	x	x	x	x	x	x

Figure 6-5 (Part 1 of 2). Disk Operation Ending Conditions and Status

Status Byte 2

Bit	Name and Indicates	Seek	Recalibrate	Read Data	Read ID	Read Diagnostic	Read Verify	Write Data	Write ID	Scan High or Equal	Scan Equal	Scan Low or Equal
0	*Serializer-deserializer check* is part of unit-check condition. Equipment check is set concurrently.			x	x	x	x					
1	*Write check trigger* and equipment check are set concurrently.											
2	*Channel transfer error* is part of unit-check condition. Equipment check is set concurrently.	x	x	x	x	x	x	x	x	x	x	x
3	Not used.											
4	Not used.											
5	Not used.											
6	Not used.											
7	Not used.											

Status Byte 3

Bit	Name and Indicates	Seek	Recalibrate	Read Data	Read ID	Read Diagnostic	Read Verify	Write Data	Write ID	Scan High or Equal	Scan Equal	Scan Low or Equal
0	This bit is not used but is part of unit-check condition.											
1	*Select unsafe* is part of unit-check condition.	x	x	x	x	x	x	x	x	x	x	x
2	*Write unsafe* is part of unit-check condition.	x	x	x	x	x	x	x	x	x	x	x
3	*Brake failure* is part of unit-check condition.	x	x	x	x	x	x	x	x	x	x	x
4	*Servo unsafe* is part of unit-check condition.	x	x	x	x	x	x	x	x	x	x	x
5	Not used.											
6	*9.1-megabyte* drive.	x	x	x	x	x	x	x	x	x	x	x
7	Not used.	x	x	x	x	x	x	x	x	x	x	x

Figure 6-5 (Part 2 of 2). Disk Operation Ending Conditions and Status

DISK OPERATING PROCEDURES

Disk Program Load Procedure

Pressing the LOAD key on the operator's panel initiates program load functions (see *System Controls* in Chapter 2). IMPL and IPL toggle switch settings determine which device reads data during the IMPL and IPL procedures. (If you are using IBM programming support, both switches must be set to DISK.)

A disk IMPL procedure initiates an immediate recalibrate to cylinder 000, track 0 and loads sectors 00 through 15 (2048 words) into contiguous control storage locations from address hex 0000 through 2047. The LOAD light turns off when the system successfully completes an IMPL procedure. The system initiates IPL from the disk immediately after a successful IMPL provided the device select switch for IPL is set to DISK. Prompting on the display screen signals the end of the IPL procedure if IBM system control programming is used.

Alternate Sector Assignment Procedure

The system uses an alternate sector assignment procedure for handling defective disk sectors. If a sector is defective, the system automatically assigns a good alternate sector to replace the defective sector. The system does not perform the procedure for cylinders 000 and 001, which are guaranteed at time of purchase from IBM.

In the alternate assignment procedure, the system:

1. Recovers the data portion of the defective sector.

2. Seeks the alternate sector cylinder (002) and locates the next available good alternate sector, using a read-disk-ID instruction.

3. Writes the ID field of the alternate sector, writing hex 01 in the F-byte to indicate a good alternate sector and writing the address of the defective sector in the CCHS bytes.

4. Writes the data recovered from the defective sector into the data field of the alternate sector, and verifies the ID and data portion of the alternate sector by reading it.

5. Seeks to the original primary track and changes the ID field of the defective sector, writing hex 02 in the F-byte and the address of the alternate sector in the CCHS bytes.

6. Verifies the ID field and, if the ID field cannot be recovered successfully, rewrites the ID field using a write-disk-ID-with-skew instruction. Then the system verifies the ID field using a read-disk-ID-with-skew instruction.

Alternate Sector Processing

Whenever an ID mismatch occurs during read-data, read-verify, write-data, and scan operations, the system automatically seeks to the track containing the alternate sector, locates the alternate sector, and performs the operation specified. If an ID mismatch occurs on the alternate sector, the system sets the no-record-found and alternate-track status bits in the disk status byte. Otherwise (if the alternate sector is located and the operation performed), the system seeks to the track containing the defective sector at the end of the operation, and processing continues.

Sector Initialization Procedure

The initialization procedures described in this section are for a single sector. A complete track or total surface initialization procedure can be done by single-sector-initialization operations. The initialization objectives are:

1. Previously flagged defective sectors are not reinitialized.

2. Defective sectors are reassigned alternate sectors.

Note: If the sector has been a previously assigned alternate sector, initialize the alternate sector according to the alternate sector assignment procedure.

However, if the primary sector has no assigned alternate, attempt to initialize the ID of this sector. The initialization procedure is as follows:

1. Initialize the ID portion of the sector using the write-ID operation.

2. Read ID to verify a successful recovery. If unsuccessful, go to the alternate sector assignment procedure.

3. Verify the sector integrity of the data portion by completing two successful executions of a scan-equal operation for a hex DEB6 pattern that is repeated 128 times in the data field. If this is unsuccessful, use the alternate sector assignment procedure to assign an alternate sector.

Error Recovery Procedure

The system performs disk error recovery without special programming. However, if bit 0 of byte 15 in the IOB. is on, the system does not do error recovery; in this case, error recovery is a program function.

All Checks Except Data Unsafe and Seek. For a data or seek operation, the system resets the check and executes the IOB.

Seek Check or No Record Found. For a data operation, the system resets the check, recalibrates, and seeks to the current cylinder, then executes the IOB. For a seek, the program must reset the check, recalibrate and initiate a seek.

Invalid Seek. Any data operation is bypassed (no-op). For a seek, the check is reset and then the disk drive is recalibrated if the operation is a recalibrate. Otherwise, the operation must be retried.

Data Unsafe and Ready. The first time this check occurs for a data or seek operation, the unsafe condition is reset, a recalibrate and seek to the current cylinder is performed, and the instruction is reexecuted. The second time no operation is performed on the instruction and the system sets the no-op status bit. The system must be reloaded to recover from no-op status due to a data-unsafe condition.

Not Data Unsafe and Not Ready. Whenever a data or seek instruction is issued, the system ignores (no-ops) the instruction.

Error Recovery by IBM IOS

If a unit check persists after 16 instruction retries, the specific error recovery depends on the type of check status. If the system determines that the error cannot be recovered, the system error routine assembles an operator message. If the alternate-sector bit is on when the system enters the error routine, the system also enters an alternate sector assignment routine. The system calls this routine during execution of the start-disk instruction if no-record-found occurs during logical address operations. For all other unit check status conditions, the system error routine logs the error data.

Chapter 7. Keyboard/Display Screen Functions

Every IBM System/32 has a keyboard and a display screen on which information can be displayed.

The operator uses the keyboard to control the system and enter data into the system. Data is entered from the keyboard through any of three groups of keys: alphameric, special character, and numeric. Command and function keys control system operations (Figure 7-1).

Alphameric and Special Character Keys

The alphameric and special character keys, which are arranged in a typewriter format, form the main body of the keyboard. Letters, decimal digits and special characters are entered with these keys.

Ten-Key Numeric Keyboard

A 10-key numeric keyboard to the right of the alphabetic keyboard resembles an adding machine keyboard. The 10-key keyboard is used to enter numeric data and the decimal point; two associated keys (ENTER- and ENTER+) control whether a number keyed into the keyboard is entered as a positive or a negative value.

Function Keys

Except for FIELD BKSP, REC ADV, ENTER+ and ENTER-, all function keys are to the left of the alphabetic keyboard.

Command Keys

Command keys redefine the top row of the keyboard so that special tasks can be performed by using this row of keys.

PHYSICAL CHARACTERISTICS

The display screen displays alphameric characters on a screen similar to that of a television set. Up to 240 characters can be displayed at the same time in six 40-character lines. A marker (called a cursor) usually indicates which position the next character entered via the keyboard occupies on the display. However, the cursor can be displayed in any combination of the 240 character positions.

The keyboard supports a 63-character set; the display screen supports the same set and also supports display of the backward slash (\). The display screen has a display-intensity control on the CE panel which controls readability of the information displayed.

The keyboard character set is:

A through Z
0 through 9
$, . + - * / % @ & ' # (48-character commercial set)
( )
= > <
! " ? ¢
- | Γ : ;
Space

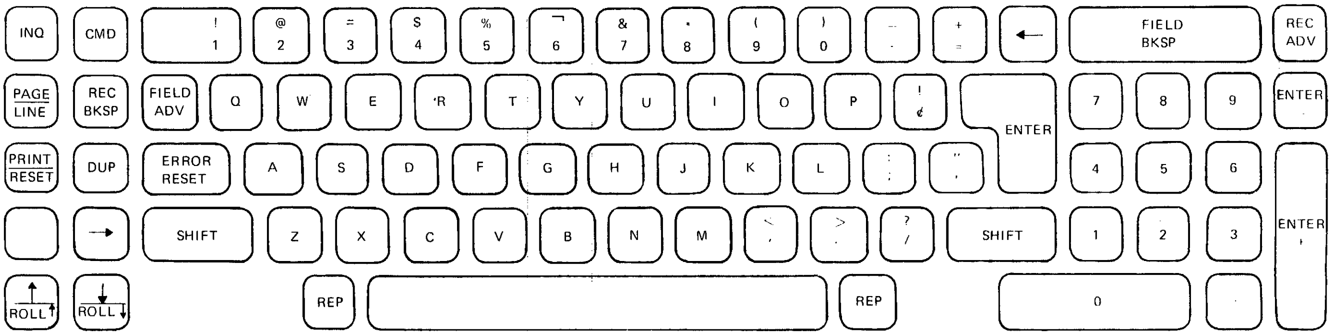


Figure 7-1. EBCDIC Keyboard Used in U.S.A.

OPERATIONAL CHARACTERISTICS -- KEYBOARD KEY FUNCTIONS

The INQ (inquiry), PAGE/LINE, PRINT/RESET, and ERROR RESET keys can be used any time the system is working, even when the KEYBD RDY (keyboard ready) indicator on the operator's console is off. All other keys are available for use only when KEYBD RDY is on.

The PAGE/LINE and PRINT/RESET keys are printer control keys, and are not used by the program. If the printer is not ready when you press a function key, the function does not occur.

If the printer is ready and not printing when you press the key, the system resets all pending printer checks and performs the keyed functions.

LINE

Pressing LINE moves the form up one line in the printer.

PAGE

Pressing PAGE in continuous-forms mode skips the current form out of the printer and moves the next form into position to print on line 1.

Pressing PAGE in single form/ledger cards mode ejects the current form from the printer. Concurrently, the system loads the forms length from the forms length register into the line counter.

PRINT

Pressing PRINT in continuous-forms mode prints the six lines of data displayed on the display screen in the format displayed.

Pressing PRINT in single form/ledger cards mode causes the system to print lines of data from the display screen, in the format displayed, until the forms length -2 is reached. (That is, printing occurs until the next line to be printed is on the last 1/3 inch of the form.) Then the system ends the print operation.

CAUTIONS

- Altering the data being displayed on the display screen while printing the display screen image may result in incorrect printed data.
- In single form/ledger cards mode, do not press PRINT unless the current line number is 9 or more lines from the bottom of the form.

RESET

Pressing RESET in continuous-forms mode resets the line counter to 1 to indicate that the current print line is line 1. (The form should be adjusted vertically to align print line 1 under the print mechanism. This can be done by adjusting the carriage control knobs.)

Pressing RESET in single form/ledger cards mode resets the line counter to F6 (line 1 minus 10), then initiates any operations on the queue.

ERROR RESET

This key enables the keyboard and turns the KEYBD RDY indicator on after:

- A keyboard overrun occurs.
- An operator error occurs.
- A programmed disable-keyboard, start-IOB, and call-operator operation is performed.

An overrun error occurs if you press a key on an enabled keyboard before the system handles the last key you pressed. An operator error occurs:

- If you try to key data outside the limits of the defined field.
- If you try to key alphabetic data into a numeric field.

INQ (Inquiry)

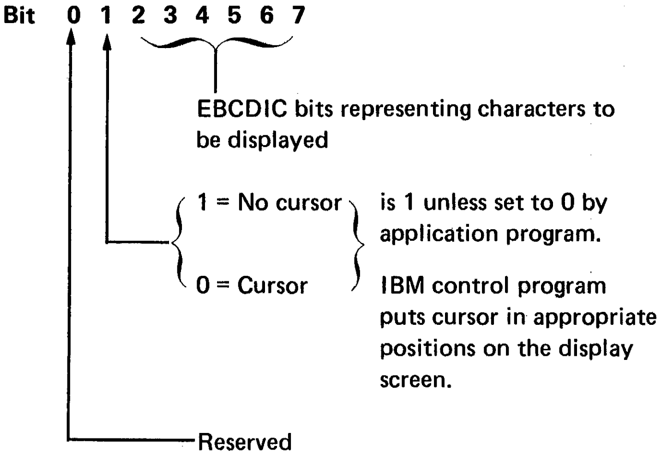
If your program is running under the system control program provided by IBM, the system displays one of the following inquiry options on the display screen whenever you press INQ:

- Ignore inquiry request and return to the program.
- Indicate that the INQ key has been pressed. (The system communication area address is at location hex 0010 and 0011, at a displacement of hex 12 (NCSCH) into the system communication area. The hex 01 bit is turned on (NCAMIEXC). Your program must check this bit to detect whether or not the INQ key has been pressed.
- Cancel the job. (The end-of-job transit is called, and a controlled cancel indication is passed to the system control program.)

To have complete control over the INQ key:

- Determine the address of the program level communication area.
- At a displacement of hex 33 (NPSCH8) into the program level communication area, turn on bit 02 (NPAMIREQ).
- When the INQ key is pressed, bit 01 (NPAMIHI7) in the same byte is turned on by the system.

CHARACTER DISPLAY FORMAT


Bit 0 1 2 3 4 5 6 7
EBCDIC bits representing characters to be displayed
1 = No cursor
0 = Cursor
is 1 unless set to 0 by application program.
IBM control program puts cursor in appropriate positions on the display screen.
Reserved

Bit 1 of each byte of data in the 240-byte display field determines if the cursor is displayed with that byte. For each EBCDIC character that has bit 1 = 0, a cursor is displayed directly below that character. All valid EBCDIC characters have a 1 in the bit-1 position unless set to 0 by a program. (See Appendix B for the EBCDIC displayable subset.)

Although bits 2-7 in Appendix B show the \ (backward slash) as 101010, the system displays hex 6A as \ in EBCDIC.

Display Screen

The keyboard/display screen uses an IOB to interface with the system. The IOB location and the information in the IOB must be provided and updated by the application program or by the IOB system control program.

IOB Definition and Usage

The keyboard/display screen IOB is an 11-byte field in main storage (Figure 7-2). The first column in the figure is the number of bytes from the IOB starting address; the balance of the table describes the IOB fields.

SHIFT

The SHIFT key is used in conjunction with keys having two meanings. (One meaning is defined by the upper half of the keyface and the other meaning defined by the lower half of the keyface.)

Pressing SHIFT places the keyboard in upper shift mode. In upper shift mode, all the meanings on the upper half of the keyface of two-meaning keys apply.

Releasing SHIFT places the keyboard in lower shift mode. In lower shift mode, all the meanings on the lower half of the keyface of two-meaning keys apply.

The SHIFT key does not affect single-meaning keys, though it always sets the shift bit in the status byte (sense byte 0 – see Figures 7-2 and 7-3).

REP (Repeat)

When you press both the REP key and a character or function key concurrently, the keyboard repeats the keyed character or function about seven times per second until you release one of the keys. Pressing REP alone causes no action.

CMD (Command)

The CMD key operates in conjunction with the top row of keys on the alphabetic keyboard. (The program can assign special meanings and function control for both upper shift and lower shift mode for each of these keys.) Pressing the CMD key as the job is being run causes the next key pressed to specify the special function assigned to it by the program.

Keys 1 through = (lower shift mode) are defined as command keys 1 through 12; keys 1 through = (upper shift mode) are defined as command keys 13 through 24.

Templates (GX21-7638) are available from IBM. The assigned meaning of each key for one job can be printed on the template. At the start of each job, the operator can insert the template above the keyboard; the command keys are then properly identified, by function, for the job to be processed.

Displacement of Leftmost Byte in Hex	IBM Program Label	Length in Bytes	Field Description
0	KBCHN	2	This is the address of the next IOB in the chain (chain pointer).
2	KBCMP	1	This byte contains the completion code: Bit 0 – Not used. Bit 1 – Operation complete. Bits 2-7 – Scan code of key. (This is the code presented to the attachment before the system converts it to EBCDIC.)
3	KBSNS1	1	The system places sense byte 0 of current keystroke into this field when the system is operating in BDE (only function keys are sensed), SDE, or CSDE mode (Figure 7-3).
4	KBSNS2	1	The system places sense byte 1 of current keystroke into the field when the system is operating in BDE (only function keys are sensed), SDE, or CSDE mode (Figure 7-3).
5	KBNCP	1	This is the position of the cursor within display screen buffer. The program must place the hex value of the display screen character position that is to contain the cursor in this field. If no cursor is to be displayed, the field must contain hex 00.
6	KBRSP	1	The record start position contains displacement into the display screen buffer of leftmost byte of record to be moved to or from the buffer.
7	KBLEN	1	The record length contains the maximum length of the current record (1 to 240 decimal characters).
8	KBFLAG	1	The flag byte defines the keyboard/display screen operation to be performed, as follows: Bits 0 and 1 = 00 – Specifies a data-move operation during which data is moved as specified by flag byte bits 3 and 7. 01 – BDE mode. (The keyboard/display screen processes data one record at a time.) 10 – SDE mode. (Keyboard/display screen does no processing, but passes characters to the processor and provides sense bytes.) 11 – CSDE mode. (Keyboard/display screen processes keystrokes, but also passes characters to processor and provides sense bytes.) Bit 2 = 1 – Numeric mode. (Check for numeric key entry (0-9). Bit 3 = 1 – Data is moved from display screen buffer to main storage. Bit 4 = 1 – Display screen buffer contents are rolled up one line. Bit 5 = 1 – Display contents of display screen buffer on display screen. Bit 6 = 1 – Erase display screen and reset. Bit 7 = 1 – Move data from main storage to display screen buffer. *Note:* The system executes these operations in the following sequence when the flag byte bits specify more than one operation: 1. Enable the keyboard, if specified. 2. Erase the display screen. 3. Roll the display screen buffer up one limit. 4. Move data either to or from the display screen buffer. 5. Display contents of display screen buffer.
9	KBMS@	2	Direct address of data in main storage to be moved to or from display screen buffer.

Figure 7-2 . Keyboard/Display Screen IOB Format

Key Pressed	Status Byte—Sense Byte 0 (in hex)	Data Byte—Sense Byte 1 (in hex)
ERROR RESET (u/c)	28	06
ERROR RESET (l/c)	20	06
↑ (u/c)	28	0A
↓ (u/c)	28	0B
← (l/c)	20	0C
→ (l/c)	20	0D
ENTER (u/c)	28	10
ENTER (l/c)	20	10
ENTER + (u/c)	28	11
ENTER + (l/c)	20	11
ENTER - (u/c)	28	12
ENTER - (l/c)	20	12
FIELD ADV (u/c)	28	13
FIELD ADV (l/c)	20	13
REC ADV (u/c)	28	14
REC ADV (l/c)	20	14
FIELD BKSP (u/c)	28	15
FIELD BKSP (l/c)	20	15
REC BKSP (u/c)	28	16
REC BKSP (l/c)	20	16
DUP (u/c)	28	17
DUP (l/c)	20	18
ROLL↑ (l/c)	20	19
ROLL↓ (l/c)	20	1A
←(u/c)	28	1B
→ (u/c)	28	1C
CMD (u/c)	28	30
CMD (l/c)	20	30
Data keys (u/c)	48	EBCDIC characters
Data keys (l/c)	40	EBCDIC characters

Notes:

1. The status byte format is as follows:
   Bit 0 = not used; always set to 0
   Bit 1 = data key (A-Z, 0-9, special characters)
   Bit 2 = function key (nondata key)
   Bit 3 = not used; always set to 0
   Bit 4 = shift key (shift occurs on current keystroke)
   Bit 5 = not used; always set to 0
   Bit 6 = not used; always set to 0
   Bit 7 = not used; always set to 0
2. u/c means uppercase and denotes a SHIFT key operation.
3. l/c means lowercase and denotes no SHIFT key operation.

Figure 7-3. Contents of Keyboard/Display Screen Sense Bytes for Keys that Cause an Interrupt

KEYBOARD OPERATION

Modes

Regardless of the mode specified in the IOB, the INQ, PAGE/LINE, PRINT/RESET, and ERROR RESET keys are always enabled.

In BDE mode, the system processes:

- The data keys—A-Z, 0-9, and special characters
- The cursor control keys—↑, ↓, → (unshifted), and ← (unshifted)

Sense byte information is set up for function keys only, and processing of the user's program continues.

In CSDE mode, all keys generate sense bytes in the IOB; control passes to the program after the system processes the data or cursor keys.

In SDE mode, all keys generate sense bytes in the IOB; control passes to the system control program. The system control program does not alter or affect the display screen buffer with the characters and functions.

In BDE mode, all keys except ERROR RESET, data keys, cursor keys, and printer function keys generate sense bytes in the IOB; control passes to the system control program. The functions are performed in the display screen buffer and the results are displayed.

While the system operates in BDE mode, the display screen logic:

- Accepts the keystroke as it is keyed.
- Passes the keystroke and control to the system if the key is not a data key, a cursor control key, the INQ key, the PAGE/LINE key, the PRINT/RESET key, or the ERROR RESET key.
- Checks the keystroke against the numeric mode bit (bit 2 = 1) of the IOB flag byte (numeric = 0-9).
- Checks addition of the keystroke to the current record to ensure that the result does not cause the record length to exceed the record length in the IOB.
- Updates the display screen buffer and displays the character (or moves the cursor) at the current cursor position.
- Rolls the display screen image up one line if the keyed character is the last (fortieth) character on the last line of the display screen and the record length specified in the IOB is not exceeded. By rolling up the display screen image, the top line of the display is erased from the display screen, blanks are moved into the bottom (sixth) line of the display screen, and the cursor is positioned at the far left position of the bottom line.
- Disallows any keyboard/display screen interrupt requests after the first nonfunction key is pressed until a function key is pressed.

The system disables the keyboard and turns off the KEYBD RDY indicator light whenever the mode of the keyed character is incorrect (not numeric when in numeric mode) or the maximum record length is exceeded. Press the ERROR RESET key to reactivate the keyboard.

Programming Considerations for Keyboard

The system does not execute any keyboard instructions while the keyboard is enabled in BDE mode (that is, from the time the first keystroke is made in a data field while the keyboard is enabled in BDE mode until an interrupting key, such as ENTER, is pressed). The system loops on any keyboard instructions while in BDE enabled mode, and executes them as soon as the enabled condition no longer exists.

When the keyboard is enabled in CSDE or SDE mode, do not issue keyboard instructions. For example, do not issue a start-keyboard/display-screen-IOB instruction to alter the display in the middle of a field; also, do not change the IOB address or interrupt address in the middle of a field.

KEYBOARD HARDWARE CHARACTERISTICS

Keystroke Encoding

The keyboard generates a unique keyboard code on the keyboard input bus for a given key depression. The code, called the scan code, is an 8-bit code; the first bit indicates a shift and the other seven bits identify each key position. All keys except the SHIFT and REPEAT keys encode data. A specific key generates the same basic 7-bit code regardless of keyboard shift.

Data Handling and Interrupts

The program may set the keyboard ready (enabling keystrokes to enter data into the keyboard attachment) or not-ready. Depressing a key sends the 8-bit code generated by the keyboard to system logic.

Keyboard interrupt request 1 occurs:

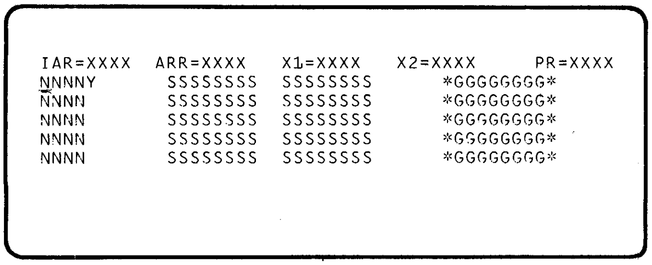
- Whenever a key is pressed (other than INQ, PRINTER CONTROL, or ERROR RESET) and the keyboard is in SDE or CSDE mode.
- When a move operation is completed (after execution of a start-keyboard/display-screen-IOB instruction with flag byte bits 0 and 1 = 00 and either bit 3 = 1 or bit 7 = 1).
- When the keyboard is in BDE mode and a key other than one of the following keys is pressed:
  Data key
  Cursor control key
  Printer function key
  ERROR RESET key

MANUAL OPERATING PROCEDURES

Data in registers or in main storage can be displayed on the display screen. This data can be altered, or new data can be entered into registers or main storage, by using the keyboard.

Data is displayed on the display screen as shown in the following example,

where:
X = register value (Hex digits)
N = main storage address
Y = M (code character identifying the preceding four characters as a main storage address)
= E (code character identifying the end of an alter or display operation; not required)
S = actual data stored at the address shown on that line (hex digit)
G = EBCDIC graphic of the stored value



```
IAR=XXXX  ARR=XXXX   X1=XXXX   X2=XXXX     PR=XXXX
NNNNY       SSSSSSSS  SSSSSSSS       *GGGGGGGG*
NNNN        SSSSSSSS  SSSSSSSS       *GGGGGGGG*
NNNN        SSSSSSSS  SSSSSSSS       *GGGGGGGG*
NNNN        SSSSSSSS  SSSSSSSS       *GGGGGGGG*
NNNN        SSSSSSSS  SSSSSSSS       *GGGGGGGG*
```

Displaying Data Stored in Registers and Main Storage

To display data that is stored in an instruction register or in main storage:

1. Set the IPL switch on the CE panel to DISKETTE.
2. Stop the system by pressing the STOP key on the operator's panel or by reaching a main storage address previously set into the address switches and placing the address compare switch at the STOP position.
3. Press the INQ key on the keyboard.

The hex digits representing the values presently stored in the instruction address register, the address recall register, index register 1, index register 2, and the program status register appear on the top line of the display in place of the Xs. The display screen also shows five more lines of data, which is data from main storage, starting with the byte stored at the address set into the four address switches on the CE panel. To display data stored in other storage locations:

1. Key in the address of the leftmost byte of the field to be displayed. The new address replaces the old address at the left end of line 2 on the display screen (shown on the example as NNNNY). As you key the hex digits, the cursor (shown as a straight line under the first N) moves right and identifies the location of the next character to be keyed.
2. Key an M as the fifth character on line 2. This identifies a main storage display.
3. Press ENTER on the keyboard.

The system now displays data stored at the new main storage address. Repeat the preceding three steps to continue displaying the contents of main storage. To end the display of stored data:

1. Press START on the operator's console to restart the system
2. Set the IPL switch on the CE panel to DISK (down).

Altering Contents of Main Storage

To alter data stored at a specified storage location:

1. Display the contents of the specified storage position. (See *Displaying Data Stored in Registers and Main Storage* in this chapter.)

2. Use the function keys on the keyboard to move the cursor until it is at the first character (or blank) to be changed.

3. Enter the new data. Each new character entered replaces the character previously stored at the storage location. The new character appears on the display screen and the cursor automatically moves to the next higher storage position.

   Note: If you press any key except 0 through 9 or A through F, the KEYBD RDY light turns off, indicating that an invalid key was pressed. Press the ERROR RESET key to make the keyboard ready and continue.

4. Press ENTER when all the desired changes on the data displayed are made.

5. Press START on the operator's console to end the alter-display operation and restart the system.

Altering Contents of Instruction Registers

To alter the contents of instruction registers:

1. Display the contents of the instruction registers. (See *Displaying Data Stored in Registers and Main Storage* in this chapter.)

2. Use the function keys on the keyboard to move the cursor to the first character to be altered in the register. (The register contents are displayed on the top line of the display screen, and the stored data follows the register identification and equal sign.)

   Note: If you press any key except 0 through 9 or A through F, the KEYBD RDY light will turn off, indicating that an invalid key has been pressed. Press the ERROR RESET key to ready the keyboard and continue.

3. Enter the new data. As you key each new character, the system replaces the character previously stored in the register. The new character appears on the display screen; then the cursor moves to the next register data position.

4. When you have entered all the required new data, press ENTER.

5. Press START on the operator's console to end the alter instruction register procedure and restart the system.

Error Recovery Procedures

Keyboard Recovery Procedures

Pressing the ERROR RESET key initiates recovery from a keyboard overrun or a keying error. The ERROR RESET key restores the keyboard and turns the KEYBD RDY indicator on.

Keyboard overrun occurs when you press two keys on the keyboard and the system does not accept or process either key.

A keying error occurs whenever keyed data causes the cursor to move outside the limits of the record, or when you key alphabetic data in a numeric-only field.

Display Screen Recovery Procedures

If the operator or the system recognizes any display screen irregularities, retry the operation at least once. If the irregularities persist, call the IBM customer engineer.

Chapter 8. Diskette Drive Functions

The IBM System/32 diskette drive has a single head for reading and writing. The diskette supports two primary system functions:

- Data interchange, using the IBM diskette for standard data interchange.
- Storing data from the disk to save it for future use, then loading it back into the system at a later time when needed for a job.

PHYSICAL CHARACTERISTICS

Only one side of the diskette is used. The diskette surface is divided into tracks. Each diskette surface contains 77 tracks; track 00 is the outside track and track 76 is the inside track.

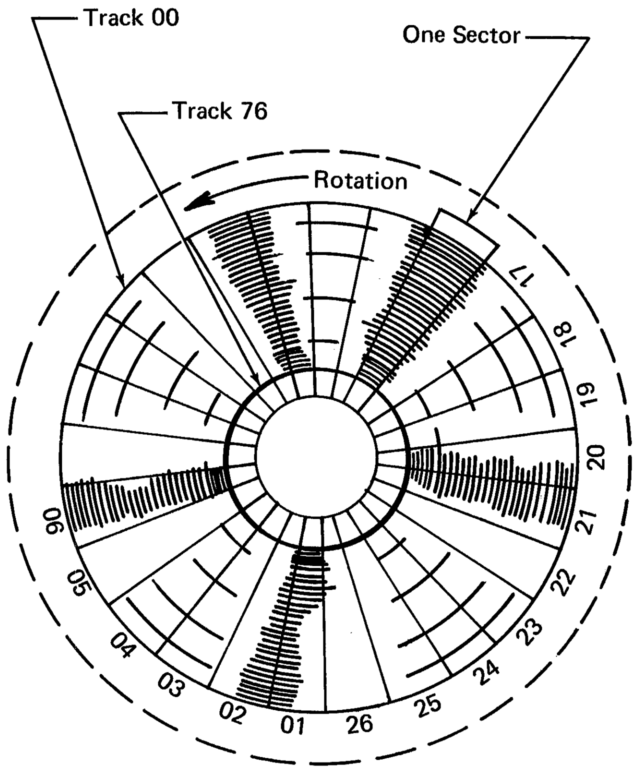


Figure 8-1. Standard Interchange Diskette Surface Recording Arrangement

Of the 77 tracks, only 75 are normally used. Track 00 contains the volume label; tracks 1-74 are primary tracks used to contain data records. Tracks 75 and 76 are available for data storage in the event that one or two of the primary data tracks become defective.

For standard data interchange, each track is divided into 26 sectors (Figure 8-1). Each sector is 128 bytes long, so it is possible to store 242,944 bytes of information on tracks 1-73. Data can be stored on track 74 of a diskette in the standard data interchange format, allowing for the storage of 246,272 bytes of data on the diskette, if the data on track 74 is to be read only by a System/32.

On diskettes in the extended format, track 00 is divided into 26 sectors, but each remaining track is divided into eight 512-byte sectors, making it possible to store 303,104 bytes of information on tracks 1-74.

The data stored in each sector is called a record. Therefore, because the diskette is formatted into tracks and sectors, each record on the diskette has a definite address consisting of a track number and sector number. This address is recorded at the record's physical location on the diskette. Diskettes that contain prerecorded record addresses are known as initialized diskettes; each record consists of an ID field and a data field.

Rotational speed of the diskette drive is 360 ± 2.5% RPM. The nominal data transfer rate of the diskette drive is 31,250 bytes per second. The diskette drive reads 128-byte records from a diskette and writes 128-byte records to a diskette at the following rates:

	Using Diskettes in the Standard Interchange Format	**Using Diskettes in the Extended Format**
Reads	Up to 3,400 records per minute	Up to 4,100 records per minute
Writes	Up to 1,800 records per minute	Up to 2,200 records per minute

RECORD FORMAT

GAP1	SYNC	AM1	ID	CRC	GAP 2	SYNC	AM2 or AM3	DATA or CONTROL	CRC	GAP 1

Field	Description

GAP 1 The gap between the data field and the next record. It is variable length and contains all 1's.

SYNC A 6-byte field of 0's which synchronizes the hardware circuits prior to reading the information from the diskette.

AM1 A single-byte identifier field address mark which identifies the ID data and contains hex FE.

ID A 4-byte associate sector address in the format CHRN, where:

C = Cylinder address. Valid one-byte binary addresses are:

Decimal	=	00-76
Hex	=	00-4C

H = One-byte binary head address. The valid address is X'00'.

R = One-byte binary record address. Valid addresses are:

	Standard Interchange Format	*Extended Format*
Decimal	01-26	01-08
Hex	X'01'-X'0A'	X'01'-X'08'

N = One-byte record length indicator used in the record length algorithm: 128 x 2^N. N is hex 00 for standard 128-byte records, or hex 02 for 512-byte records.

CRC A 2-byte cyclic redundancy check field which verifies that the ID field and data field were read correctly. The system generates these bytes during a write operation and automatically performs a read-back check to verify their accuracy.

GAP 2 Interrecord gap between the ID field and the data field. It contains hex FF and is generated by the system during write operations.

Field	Description

AM2 or AM3 A 1-byte field containing either X'FB' or X'F8'. X'FB' is a data field address marker (AM2) that identifies subsequent information as a data field. X'F8' is a control field address marker (AM3) that indicates that the following field contains control data.

DATA or CONTROL The length of this field is specified by the record length indicator (N) contained in the ID field. The System/32 record length of 128 bytes (N=0) conforms to the specifications of the IBM diskette for standard data interchange.

Displacement of Leftmost Byte in Hex	IBM Program Label	Length in Bytes	Field Description
0	IIOBCAN	2	This is the sector address of the last sector accessed.
2	IIOBCMP	1	This is the diskette completion code (in hex). 40 = Successful completion. 41 = Permanent I/O error. 42 = End of volume. 43 = Drawer is opened, but diskette has same volume-ID. 45 = Bad sector not fixed by user. 47 = Stand-alone version of I1IOS or drawer is opened but volume label is invalid or volume ID not the same. 49 = Unsupported control record.

3	IIOBQB	1	This is the Q-byte (in hex).
			D0 = Seek.
			D1 = Read data.
			D2 = Read ID.
			D4 = Read data/CAM. (D and F control records are not squeezed out as in I1PRD.)
			D5 = Write data.
			D6 = Write control address marks.
			D7 = Write ID.

4	IIOBRB	1	This is the R-byte (in hex). 80 = Seek after. 00 = Null.
5	IIOBSS	2	This is the sector address.
7	IIOBNB	1	This is the number of sectors, minus 1, involved in data transfer.
8	IIOBDAT	2	This is the data buffer address.
A	IIOBSNS1	2	Sense bytes 0 and 1. *Byte 0 (in hex)* 80 = Missing data address mark (two consecutive AMs found). 40 = CRC found in ID field. 20 = CRC found in data field. 10 = Cylinder byte in ID miscompare. 08 = Head byte in ID miscompare. 04 = Record byte in ID miscompare. 02 = Length byte in ID miscompare. 01 = First ID is found (no error).

Figure 8-2 (Part 1 of 2). Diskette IOB Format

Displacement of Leftmost Byte in Hex	IBM Program Label	Length in Bytes	Field Description
			Byte 1 (in hex) 80 = Due to prior condition, no action attempted. 40 = Invalid control record found (not D or F). 20 = Lines to I1 were not set/read correctly. 10 = Control record(s) squeezed out. 08 = Physical cylinder number greater than 76. 04 = Reserved. 02 = Reserved. 01 = Reserved.
C	IIOBSNS2	2	Sense bytes 2 and 3. *Byte 2 (in hex)* 80 = I1 index pulses too close together. 40 = I1 index pulses too far apart. 20 = End-of-cylinder found (not tested). 10 = No IDs on track. 08 = Read overrun. 02 = Write overrun. 01 = Write parity check *Byte 3 (in hex)* 80 = Head loaded. 40 = Low gate current set. 20 = Write gate on. 10 = Erase gate on. 08 = Seek to track 3 or 0 on. 04 = Seek to track 0 or 1 on. 02 = Seek to track 1 or 2 on. 01 = Seek to track 2 or 3 on.
E	IIOBERR	1	This is the error retry count.
F	IIOBFLG	1	This is the flag byte. **Bit On Meaning** 0 No ERPs attempted. 1 Automatic error display and correction allowed. 2 No error logging. 3 Allow seek past logical cylinder 74 (to 75 or 76). 4 Do not return to user program if ERP completion code is I1COMPER, I1NOTFIX, or I1INVCAM.
10	Reserved	1	Reserved.
11	IIOBEXP	1	This is the sector size.
12	IIOBXR2	2	This is a save area for XR2.
14	IIOBARR	2	This is a save area for the ARR.
16	IIOBDTF	2	This is the DTF address.

Figure 8-2 (Part 2 of 2). Diskette IOB Format

DISKETTE DRIVE OPERATING CHARACTERISTICS

The program must build a diskette IOB (Figure 8-2) to control diskette operations. Diskettes perform most efficiently if programs use sequential data organization, which is normally used for data interchange and data file save/restore functions. The system performs seek, read, and write which are typical direct access storage device functions. Diskette input and output operations cannot be overlapped with any other system function.

Diskette Control Fields and Data Fields

Before issuing a seek, read, or write instruction, the program usually must initialize the diskette control fields and data fields. These fields, which can be located anywhere in main storage, are addressed by the control address register (CAR) and data address register (DAR), respectively.

Control Field

The 5-byte control field (CHRNX) is used to specify:

- For a seek operation, the destination track address (00-76 or hex FF); the cylinder address of hex FF specifies a recalibrate operation. H, R, N, and X are not used for a seek operation.
- For a seek, read-data-and-control-record, write-data, or write-control-address-mark-operation:
  C is the destination cylinder for the automatic seek.
  R is the starting record number.
  N is the number of bytes per record (128 x 2^N).
  X+1 is the number of records to be accessed.

Before processing any data field, the system compares the CHRN portion of the control field and the corresponding portion of the identifier field of the record read. This process is called orientation. If orientation cannot be done, the system turns on the no-orient status bit (status byte 2, bit 3).

As instruction execution proceeds, the system modifies the R- and X-bytes of the control field after each record is successfully processed; this maintains a log of instruction execution for error recovery procedures.

The program must load the address of the leftmost byte of the control field into the control address register prior to instruction execution.

The control field (CHRNX) has the following meaning:

- *C (cylinder number)* is a 1-byte address; valid decimal addresses are 0 through 76. Cylinder addresses 77-254 set the invalid cylinder address (byte 1, bit 4); cylinder address 255 specifies recalibrate. This byte is not changed during execution unless a cylinder boundary is crossed during a read-data, write-data, or write-control-address-mark operation.

  Note: For this device, the term cylinder and track are synonymous.

- *H (head number)* is a 1-byte address; the only valid decimal address is 0. This byte remains unchanged during command execution.
- *R (record number)* is a 1-byte binary address that specifies the first record to be processed in a single-record or multiple-record data operation. Decimal record numbers 1 through 26 are valid for standard interchange diskettes, 1 through 8 for extended format diskettes; invalid record numbers set the record-mismatch status bit (byte 0, bit 5). The system increments the record number by 1 after each record is processed if a not-ready/unit-check condition is not detected.
- *N (record length indicator)* is a 1-byte binary number that indicates the physical record length. N must be 0 for 128-byte records, 2 for 512-byte records. This byte remains unchanged during command execution.
- *X (number of records)* is a 1-byte binary number plus 1 that specifies the number of records to be processed; the system decrements this byte by 1 after each record is processed if a not-ready/unit-check condition is not detected. If the system does not find a control record during a read-data operation, this byte is not decremented.

Data Field

The system transfers data to the data field during read operations, and from the data field during write operations. The data field is addressed by the leftmost byte and can be located arbitrarily in main storage. Prior to instruction execution, the program must load the data field address into the data address register. For a read-data, read-data-and-control-record, write-data, or write-control-address-mark operation, the record length indicator and the number of records to be processed determine the data field length as follows:

Data field length = (X + 1) 128 X 2^N
where X and N come from the control field CHRNX.

Control Address Register

This 2-byte field in control storage contains the address of the leftmost byte of the control field. The control address register must be initialized by load-diskette-control-field-address-register instruction, and is not modified during command execution.

Data Address Register

This 2-byte field in control storage contains the address of the data byte following the last data field byte accessed. It is initalized to the address of the leftmost byte of the data field before the data field is used to execute an instruction. The data address register changes after a record is transferred and no errors are found.

DISKETTE OPERATIONS

Diskette Seek

During a control seek, the access mechanism is moved so that the read/write head is at the track address specified in the control field. The R-byte of the seek instruction has no significance. A recalibration to cylinder 00 is performed by specifying FF in the cylinder address bytes (CC) of the control field.

The seek operation does not check for an invalid address prior to seek verification. Instead, the system detects an invalid address or a seek failure when it compares the control field and record ID field during the next read or write operation.

A single-track seek has some unique functional characteristics that improve save/restore performance. A single-track seek command requires a maximum of 170.83 ms to execute and considerably decreases the time required to handle the following diskette read or write instruction.

A seek beyond a single track seek requires a maximum of 106 + 53 ms SD. Where SD equals seek displacement in tracks. Recalibrate time is 4.346 seconds maximum.

Read Data

This operation initiates an automatic seek to the logical track address specified in the control field and then, beginning at the record number specified in the control field, reads X+1 records into contiguous positions of the data field addressed by the data address register. When a flagged track is encountered, an automatic seek to the next record is executed. A maximum of three automatic seeks are attempted before the operation is terminated.

If the end of a cylinder is encountered before X+1 records have been accessed, an automatic seek to record 1 of the next track occurs and execution continues.

Read Data and Control Record

This operation is very similar to the read-data operation. It initiates an automatic seek to the logical track address specified in the control field, and then reads X+1 records into contiguous positions of the data field addressed by the data address register. The system reads the records beginning at the record number specified within the control field. Whenever the system encounters a flagged track, it seeks the next record. The system tries a maximum of three automatic seeks before termination of the operation.

If the system encounters the end of cylinder before X+1 records have been accessed, it executes an automatic seek to record 1 of the next track and continues with the operation.

There is one difference between this operation and the read-data operation. In this command, if a control address mark is detected, the system sets control-address-mark status (byte 1, bit 3) (this bit cannot be tested by a TIO) and reads the control record data field into the data field in main storage regardless of the control character in the first position of the record data field.

For multiple record operations, the system alters the control field when each record is read. The record number (R) is incremented by 1 and the number of records to be accessed (X) is decremented by 1. If a cylinder boundary is crossed, the cylinder number is increased by one and the record number is reset to 1.

The record number (R) is incremented by 1 and number of records to be accessed (X) is decremented by 1 as each record is read. If a cylinder boundary is crossed, the cylinder number (C) also increases by 1 and the record number is reset to 1. The rest of the control field does not change.

Note: If a control address mark (hex F8), is detected and the first character in the data field is not hex C4 or C6 (alpha D or F), the system posts invalid control record status (byte 1, bit 1) and control address mark (byte 1, bit 3) and ends the read operation. If an alphabetic D or F is in the first position of the data field, the system ignores that record and reads subsequent records.

Read ID (Identifier)

This operation initiates the recovery of a single 4-byte identifier field, CHRN, from the current selected track. The system places the first identifier read without error in the data field at the address specified in the data address register. If no ID on the current track can be read successfully, the system posts the no-orient status (byte 2, bit 3).

No automatic seek occurs, and the control field remains unchanged.

Note: If a defect appears in an ID field, your program can delete the entire track by writing a defective track identifier in all records on that track. This ID is CHRN = hex FFFFFFFF. All track information, including ID fields, must be put on the next-higher-numbered track.

Write/Verify Data

This operation initiates an automatic seek to the logical track address specified in the control field and writes X+1 records obtained from contiguous positions of the data field addressed by the data address register. Records are written beginning at the record number specified within the control field. Whenever the system encounters a flagged track, it automatically seeks the next track. A maximum of three automatic seeks are attempted before the operation is terminated. Each record is 128 or 512 bytes long and is written in accordance with the record format described in *Diskette Control Fields and Data Fields* in this chapter. At the same time, the system writes a data address mark (hex FB) in the byte immediately preceding the data field in the record.

Data verification automatically occurs during the next diskette revolution.

If the system encounters the end of cylinder before X+1 records have been accessed, an automatic seek to record 1 of the next track occurs and execution continues.

The system increments by 1 and decrements the number of records to be accessed by 1 as each record is written. If a cylinder boundary is crossed, the system also increments the cylinder number by 1 and sets the record number to 01. The system does not change the rest of the control field.

Write/Verify Control Address Mark

This operation is very similar to a write-data operation. It initiates an automatic seek to the logical track address specified in the control field and then writes X+1 records obtained from contiguous positions of the data field addressed by the data address register. Records are written, beginning at the record number specified within the control field. The only difference between this operation and the write-data operation is that a control address mark (hex F8) is written in the byte immediately preceding the data field in the record. The data field of each record written is 128 or 512 bytes long and is written in accordance with the record format reference. When the system encounters a flagged track, an automatic seek to the next track occurs. The system attempts a maximum of three automatic seeks before ending the operation.

Data verification occurs during the next diskette revolution.

If the end of a cylinder is encountered before X+1 records have been accessed, an automatic seek to record 1 of the next track occurs and execution continues.

The control field remains unchanged except for the record number and number of records to be accessed during multiple record operations; these values are incremented by 1 and decremented by 1, respectively, as each record is written. If a cylinder boundary is crossed, the system adds 1 to both the cylinder number and sets the record number to 01.

Write/Verify ID (Identifier)

The write-ID/verify operation initiates the writing of a full track of data in accordance with the record format described in *Diskette Control Fields and Data Fields* in this chapter.

The system forces the record number in the control field to 1 so that the first record written after the index point is record 1. The system generates subsequent record numbers by incrementing the record number byte by 1 as each record is written. Data for each record data field is obtained from the same 128-byte or 512-byte data field in main storage. The system writes an entire 128-byte or 512-byte data field from storage for each disk record written.

Data verification automatically occurs during the next diskette revolution.

The record number is the only byte in the control field that changes; it is decimal 26 or 8 at the end of the operation.

Note: Write ID is intended to be used for track initialization procedures. If the cylinder portion of the CHRN field is specified as hex FF, the record number portion is not incremented during write-ID execution. This writes a defective track identifier field equal to hex FFFFFFFF. During automatic data verification, length-mismatch, and no-orient status are set. ID verification should be performed by the read-ID instruction.

CHECK CONDITION AND STATUS INFORMATION

Refer to Figure 8-3 for a summary of diskette operations, diskette indicators set, and suggested restart procedures.

Status Byte 0

Bit **Description**

0 *Missing data address mark* indicates that a data address marker is not detected after an ID field. The next diskette operation or system reset operation resets this bit.

1 *ID CRC* indicates a cyclic redundancy check noncompare in an ID field:

- When the searching for an ID field and orientation can not be achieved

- After initial orientation, when a subsequent ID CRC is detected.

The next diskette operation or system reset operation resets this bit.

2 *Data CRC* indicates a cyclic redundancy check noncompare in the data field after initial record orientation. The next diskette operation or system reset operation resets this bit.

3 *Cylinder mismatch* indicates a mismatch between the cylinder address portion of the ID field and the control field during an ID search. This bit is reset prior to diskette I/O instruction execution or by a system reset operation.

Status Byte 0 (Continued)

Bit **Description**

4 *Head mismatch* indicates a mismatch between the head address portion of the ID field and the control field during an ID search. This bit is reset prior to the next diskette operation or by the next system reset operation.

5 *Record mismatch* indicates that no match between the record address portion of any ID field and the control field occurred during an ID search. This bit is reset prior to the next diskette operation or by the next system reset operation.

6 *Length mismatch* indicates a mismatch between the record length portion of the ID field and the control field during an ID search. This bit is reset prior to the next diskette operation or by the next system reset operation.

7 This bit is reserved.

Status Byte 1

Bit **Description**

0 *No operation* indicates that a diskette I/O instruction cannot be executed because of an outstanding not-ready status. This bit is reset by the next diskette I/O instruction or by a system reset operation.

1 *Invalid control record* indicates that the leftmost byte of a control record data field contains other than an F or D control graphic. This bit is reset by a diskette I/O instruction or by system reset.

2 *Control error* indicates that a low write current existed during a diskette write operation or write or erase gate selection was made during any diskette I/O operation except write. This bit is reset prior to a diskette I/O operation or by a system reset operation.

3 *Control address mark record found* indicates that a control address mark was found. This record was skipped and the next one was processed in its place if the operation was read-data and the leftmost byte of the data field contained hex D or F. This condition is not tested by a test instruction. The next diskette read, write, or seek operation resets the control-address-mark-record-found bit.

Status Byte 1 (Continued)

Bit **Description**

4 *Cylinder address invalid* indicates that the logical cylinder number in the control field exceeds 76. This bit is reset prior to a start diskette command or by a system reset operation.

5 *Write error* indicates that either a write overrun (status byte 2, bit 6) or a write parity check (byte 2, bit 7) was detected during a write operation. This bit is reset by the next diskette I/O instruction or by a system reset operation.

6 This bit is used by the system.

7 This bit is used by the system.

Status Byte 2

Bit **Description**

0 *Diskette fast* indicates that the diskette rotates faster than the maximum specified rate of 369 RPM, or 162.50 ms per revolution. This bit is reset prior to the next diskette operation or by a system reset operation.

1 *Not ready* indicates that:

The diskette is not inserted, or
The door is not closed, or
The diskette is inserted backwards, or
The diskette unit is malfunctioning.

Ready is conditioned by a recalibrate operation if the diskette is up to proper speed. This is the only way this bit can be reset.

2 This bit is unassigned.

3 *No orient* indicates that a record specified by the record address in the control field of a data operation could not be found. This bit is reset prior to the next diskette operation or by a system reset operation.

4 *Read overrun* indicates that the minimum data transfer rate was not maintained during a data transfer operation from the diskette to main storage. This bit is reset prior to the next diskette operation or by a system reset operation.

Status Byte 2 (Continued)

Bit **Description**

5 This bit is unassigned.

6 *Write overrun* is similar to a read overrun except that the minimum data transfer rate is not maintained during data transfer from main storage to the diskette. The next diskette read, write, or seek operation or system reset operation resets this bit.

Note: This status bit can be sensed by the program for a write-ID operation only. Write-error status (byte 1, bit 5) is set simultaneously; the program can sense this bit.

7 *Write parity check* indicates that a mismatch between data-bus-out parity and system generared serial-write-data parity was detected during a write operation. The next diskette read, write, or seek operation or system reset operation resets this bit.

Note: This status bit can be sensed by the program for a write-ID operation only. Write-error status (byte 1, bit 5) is set simultaneously; the program can sense this bit.

Status Byte 3

The status bits in this byte cannot be sensed by your program. Bits 0 and 4-7 may be on if a 1-track seek preceded the I/O operation bits 1, 2, and 3 will be off.

Bit **Description**

0 The head is loaded.

1 The low write current to diskette is set.

2 The write gate to diskette is on.

3 The erase gate to diskette is on.

4 Seek to track 3 or 0 is on.

5 Seek to track 0 or 1 is on.

6 Seek to track 1 or 2 is on.

7 Seek to track 2 or 3 is on.

IMPL (Initial Microprogram Load) and IPL (Initial Program Load)

IMPL and IPL are selected by toggle switches located on the CE panel and are initiated by pressing the LOAD key located on the operator's panel.

A diskette IMPL initiates the transfer of the 4096-byte IMPL record located on track 00, record 01 into storage. This 4096-byte record is a special record used by the system; it is never used by the programmer.

The IPL from the diskette is not the normal mode and is used primarily for installing a new supervisor. IPL from diskette is initiated by the system immediately following a successful IMPL if the DEVICE SELECT switch is set to select IPL from the diskette.

Suggested Diskette Error Recovery Procedures

The recovery procedures presented in Figure 8-3 can be used to restore system operation. The type of action necessary is determined by performing a sense-diskette operation, then testing to determine which diskette status bits are on.

Status Byte 0

Bit	Name	Condition Set By: · Seek Q-Byte D0	· Read Data Q-Byte D1	· Read ID Q-Byte D2	· Read Data and Control Data Q-Byte D4	· Write Data (Write Cycle) Q-Byte D5	· Write Data (Verify Cycle) Q-Byte D5	· Write Control (Write Cycle) Q-Byte D6	· Write Control (Verify Cycle) Q-Byte D6	· Write ID (Write Cycle) Q-Byte D7	· Write ID (Verify Cycle) Q-Byte D7	Suggested Action[1]
0	Missing data address mark	X	X		X		X		X		X	3
1	ID CRC		X	X	X	X	X	X	X		X	3
2	Data CRC		X		X		X		X		X	3
3	Cylinder mismatch		X		X	X	X	X	X		X	3
4	Head mismatch		X		X	X	X	X	X		X	3
5	Record mismatch		X		X	X	X	X	X		X	3
6	Length mismatch		X		X	X	X	X	X		X	3
7	Not used											3

Status Byte 1

Bit	Name	Seek Q-Byte D0	Read Data Q-Byte D1	Read ID Q-Byte D2	Read Data and Control Data Q-Byte D4	Write Data (Write Cycle) Q-Byte D5	Write Data (Verify Cycle) Q-Byte D5	Write Control (Write Cycle) Q-Byte D6	Write Control (Verify Cycle) Q-Byte D6	Write ID (Write Cycle) Q-Byte D7	Write ID (Verify Cycle) Q-Byte D7	Suggested Action[1]
0	No op	[2]	X	X	X	X	X	X	X	X	X	3
1	Invalid control record		X		X							4
2	Control error	X	X	X	X	X	X	X	X	X	X	3
3	Control address mark record found	[3]	[3]	[3]	[3]	[3]	[3]	[3]	[3]	[3]	[3]	
4	Cylinder address invalid	X	X		X	X	X	X	X			2
5	Write error					[3]	[3]	[3]	[3]	[3]	[3]	
6	Not used											
7	Not used											

[1] Actions are described in Part 3 of this figure.
[2] No-op is set by all operations except recalibrate.
[3] This condition is not tested for error recovery procedure.

Figure 8-3 (Part 1 of 3). Diskette Operation Ending Conditions, Status, and Suggested Restart

Status Byte 2

Bit	Name	Condition Set By:										Suggested Action[1]
		Seek Q-Byte D0	Read Data Q-Byte D1	Read ID Q-Byte D2	Read Data and Control Data Q-Byte D4	Write Data (Write Cycle) Q-Byte D5	Write Data (Verify Cycle) Q-Byte D5	Write Control (Write Cycle) Q-Byte D6	Write Control (Verify Cycle) Q-Byte D6	Write ID (Write Cycle) Q-Byte D7	Write ID (Verify Cycle) Q-Byte D7	
0	Diskette fast	X	X	X	X	X	X	X	X	X	X	3
1	Not ready	X	X	X	X	X	X	X	X	X	X	1
2	Not used											
3	No orient		X	X	X		X		X		X	3
4	Read overrun		X	X	X		X		X		X	3
5	Not used											
6	Write overrun					X		X		X		3
7	Write parity check					X		X		X		3

[1] Actions are described in Part 3 of this figure.
[2] No-op is set by all operations except recalibrate.
[3] This condition is not tested for error recovery procedure.

Figure 8-3 (Part 2 of 3). Diskette Operation Ending Conditions, Status, and Suggested Restart

Suggested Action	
Number	**Description**
1	1. Recalibrate. 2. Read volume label. 3. Check volume 1 standard and volume identification for validity. (If not valid issue message to insert correct diskette.) 4. Repeat until valid diskette is inserted or until job is canceled. 5. Post possible wrong-diskette-inserted return code. 6. Return.
2	1. Post end-of-volume return-code. 2. Return.
3	1. Retry three times or until successful. a. If not successful, seek forward and backward one cylinder three times or until successful. b. If step 1a is not successful, and if the system is executing a read data operation, evoke an automatic sector rebuild. If the system is not executing a read data operation, post the permanent-error return. 2. Log the diskette error. 3. Return.
4	1. Post invalid-control-record-read return code. 2. Return.

Figure 8-3 (Part 3 of 3). Diskette Operation Ending Conditions, Status, and Suggested Restart

Initialization Procedures

Diskette initialization procedures described below guarantee a serviceable diskette. Basic initialization objectives are that:

- Track 00 must be free of defects.
- Diskettes are initialized in accordance with the basic interchange record format defined in this chapter under *Diskette Control Fields and Data Fields.*
- Defective tracks are flagged defective and the ERMAP is updated.

To initialize the diskette:

1. Reinitialize track 00.
   a. Write ID/verify with 128 bytes of hex E5 in the data field. If unsuccessful, retry a maximum of two times, and if still unsuccessful, go to step 3.
   b. Write data/verify with 128 bytes of hex 40 in the data field. If unsuccessful, retry a maximum of two times, and if still unsuccessful, go to step 3.

2. Continue to sequentially reinitialize the remainder of the diskette by repeating step 1. (Be sure to reinitialize sectors of the appropriate length – 128 bytes each or 512 bytes each.)

3. If track 00 cannot be successfully initialized, discard the diskette. If track 00 is successfully initialized and some other track is found defective, use the surface defect procedures, flag the defective track, and update the ERMAP.

Surface Defect Procedures

The track must be flagged as defective and the ERMAP label must be updated during a reformatting operation if less than two tracks have been previously flagged.

The diskette must be discarded if track 0 is defective, or if two tracks have been flagged as defective and an additional track is found defective.

Chapter 9. Telecommunications Functions

The binary synchronous communications adapter is a special feature for System/32. It provides the system with the ability to function as a point-to-point or multipoint processor terminal. Operation is half-duplex, synchronous, and serially by bit, serially by character over either switched voice grade 2-wire facilities, or nonswitched 2- or 4-wire condition lines.

Operation of the BSCA is fully controlled by a combination of System/32 stored program instructions and BSCA logical responses to line control characters. With the feature installed, the system can both transmit and receive during a single communication, although half-duplex operation prevents simultaneous transmission and reception of data.

Point-to-Point Communications Networks

The BSCA functions in either a switched or nonswitched point-to-point network. Normally, contention cannot occur because the called station must be made ready to receive before a call can be completed. However, a 2-second timeout can be programmed to resolve any contention situations that may occur.

In a point-to-point network, System/32 can be designated, by programming, as either the calling (transmitting) or called (receiving) station.

Multipoint Communications Networks

System/32 can be used on a multipoint network as a tributary station. Support required to operate System/32 as a control station is not provided.

System/32 polling and addressing characters are shown in Appendix F.

Data Rates

The communications adapter can operate at various data rates between 600 and 7200 BPS (bits per second). The customer selects the data rate to be used based on his modem type, and his communications adapter is equipped with an appropriate line interface. Interconnected units must operate at the same data rate.

Modems

The modem receives the data serially by bit and serially by character from the communications line during receive operations and presents the bits to the communications adapter. During transmit operations the communications adapter receives characters from storage serially by character, then makes them available serially by bit, serially by character to the modem. The modem places each bit on the communications line as soon as it receives the bit from the communications adapter.

Transmission Rate Control

A timing device, called a clock, controls the rate at which data is transmitted and received. Clocking is furnished either by a special feature for the communications adapter or by the modem, depending on which type of modem is selected. Connected units must use the same clocking source (modem or business machine).

Transmission Codes

Data can be transferred in either of two codes, EBCDIC (extended binary coded decimal interchange code) or ASCII (the IBM version of the American National Standard Code for Information Interchange). In each job that uses the BSCA, the customer must specify once which code is being used in the job. Only units using the same code can communicate with each other.

EBCDIC is the standard, 8-bit plus parity, internal binary code of System/32 (this code is illustrated in Appendix B). The parity bit, used for internal checking, is not transmitted over the communications network.

ASCII is a 7-bit plus parity code. It is illustrated in Appendix F. Unlike EBCDIC, which numbers its bits 0 through 7 starting at the high-order bit, ASCII numbers its bits 1 through 7 starting at the low-order bit (Figure 9-1).

All characters are transmitted over the line low-order bit first. For ASCII, the high-order bit must be a zero bit from main storage on transmit. If the adapter does not receive a high-order zero from main storage, it will generate and send out a wrong parity (P) bit. In addition, the invalid ASCII character status bit will be set on causing a unit check condition.

On receive, the first bit received is transferred into low-order main storage position and so on. For ASCII, the adapter fills a zero into the high-order bit position in main storage except when the character has a VRC error.

EBCDIC and ASCII have different coding structures to represent characters. When ASCII is used with the communications adapter, the program must translate data from EBCDIC before transmission and to EBCDIC after reception. This translation is not performed by the communications adapter.

	First Hex	Second Hex
	High	Low
Transmission	8 7 6 5	4 3 2 1
EBCDIC	0 1 2 3	4 5 6 7
ASCII	P 7 6 5	4 3 2 1

Note: The complete EBCDIC coded character set is shown in Appendix B. The ASCII coded character set is shown in Appendix F.

Figure 9-1. Bit Positions and Significance

STANDARD BSCA CAPABILITIES

The following capabilities are standard with each BSCA for System/32.

Transmission Code Selection. Each BSCA can transmit and receive both EBCDIC and ASCII data (only units using the same transmission code can communicate with each other). The transmission code used can be varied from job to job according to the particular requirements of each job.

Autoanswer. The autoanswer capability (switched network only) enables the communications adapter to respond to a telephone request for data communications automatically without operator intervention (manual answer) if the modem has unattended answer capability.

Intermediate Block Checking. The intermediate block checking capability allows the reception of checking (ITB) characters for checking the accuracy of communication without interrupting the steady flow of information from the transmitting station to the receiving station.

Full Transparent Text Mode. The full transparent text mode capability (EBCDIC only) allows all the 256 possible bit combinations available in EBCDIC to be transmitted as data. Thus, with this capability the EBCDIC character combinations designated as line control characters can, as required, be transmitted as data.

Rate Select. The rate select capability allows programs to select a transmission rate of half the normal speed, provided that the system is equipped with a modem that can operate at either rate.

SPECIAL FEATURES

The following special communications adapter features are available on System/32.

EIA Interface

The EIA (Electronics Industry Association) Interface feature provides a cable and interface for attaching the communications adapter to a stand-alone IBM modem or a non-IBM modem. The EIA Interface may require the Internal Clock feature (described in the following paragraph) if the external modem does not provide its own clocking. The EIA Interface cannot be installed with the IBM 1200 BPS Integrated Modem or the IBM 2400 BPS Integrated Modem, both of which are special features described in following paragraphs.

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

The Internal Clock feature requires either the EIA Interface or the IBM 1200 BPS Integrated Modem. It cannot be installed with the IBM 2400 BPS Integrated Modem.

IBM 1200 BPS Integrated Modem

This feature permits the communications adapter to operate at a data transmission rate of 1200 bits per second over nonswitched facilities or in a switched network. The device communicating with System/32 must also be equipped with an IBM 1200 BPS Integrated Modem.

This feature comes in two versions:

- *The nonswitched version* attaches to 2- or 4-wire facilities by means of an IBM-provided cable directly to the line (facility D5, type 3002).
- *The switched with autoanswer version* attaches to a switched network by means of an IBM-provided cable to a common carrier arrangement (type CBS or equivalent).

The IBM 1200 BPS Integrated Modem requires the Internal Clock. It cannot be installed with the EIA Interface or the IBM 2400 BPS Integrated Modem.

IBM 2400 BPS Integrated Modem

This feature permits the communications adapter to operate at a data transmission rate of 2400 bits per second over nonswitched facilities or in a switched network. The device communicating with System/32 must also be equipped with an IBM 2400 BPS Integrated Modem or with an IBM 3872 Modem.

This feature comes in three versions:

- *The nonswitched point-to-point* and *nonswitched multipoint tributary versions* attach to 2- or 4-wire facilities by means of an IBM-provided cable directly to the line (facility D7, type 3002).
- *The switched with autoanswer version* attaches to a switched network by means of an IBM-provided cable to a common carrier arrangement (type CBS or equivalent).

The IBM 2400 BPS Integrated Modem requires the Processing Unit Expansion feature, upon which the modem is mounted. The IBM 2400 BPS Integrated Modem cannot be installed with the EIA Interface or the IBM 1200 BPS Integrated Modem.

Switched Network Backup

This feature provides for backup attachment of System/32 to the public switched network if the primary System/32 facility is the IBM 2400 BPS Integrated Modem on a nonswitched line. This feature permits communication with another IBM 2400 BPS Integrated Modem equipped with the switched network capability and permits communication with an IBM 3872 Modem equipped with the switched network capability. Attachment to the switched network is by means of a common carrier arrangement (type CDT or equivalent).

Selection at a given time of the primary facility or the Switched Network Backup is made by System/32 system control programming. Calls must be made and answered manually if the backup is selected.

This feature requires the IBM 2400 BPS Integrated Modem and the Processing Unit Expansion feature. It cannot be installed with the Switched Network Backup with Autoanswer feature, which is described in the following paragraph.

Switched Network Backup with Autoanswer

This feature is the same as the Switched Network Backup, described in the preceding paragraphs, except that it automatically answers incoming calls when attached to a common carrier arrangement (type CBS or equivalent).

This feature requires the IBM 2400 BPS Integrated Modem and the Processing Unit Expansion feature. It cannot be installed with the Switched Network Backup feature.

CONTROL STORAGE REGISTERS USED BY COMMUNICATIONS ADAPTER

Four control storage registers are provided for the communications adapter: the current address register, the transition address register, the stop address register, and the unit definition table register.

Current Address Register

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 address is incremented by +1 under control of the adapter each time a character is stored in or fetched from main storage.

Transition Address Register

The transition address register contains the address at which a reversal 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 and begins to store the characters received from the communications line.

If System/32 is a multipoint tributary, the transition address register is used during the receive initial operation to store the system's assigned address.

Stop Address Register

The stop address register contains the address at which the communications adapter I/O operation must stop. When the address in the current address register equals the address in the stop address register, the communications adapter ends its operation and generates an interruption request.

Unit Definition Table Register

The unit definition table register contains the communications adapter unit definition table. The unit definition table is a 2-byte field that describes the configuration of the communications adapter feature. The BSCA unit definition table contains:

Byte 1	Byte 2
Bit 0 on = Half rate	Not used
Bit 1 on = Internal clock	
Bit 2 on = IBM modem	
Bit 3 on = Answer tone	
Bit 4 on = Switched network backup	
Bit 5 on = Multipoint tributary	
Bit 6 on = Switched network[1]	
Bit 7 on = Point-to-point network[1]	

[1] Bits 6 and 7 on together = Switched CDSTL (connect data set to line). Switched CDSTL is available only from World Trade Corporation.

BSCA TERMINAL CONTROL

Adapter controls are called into action at each station by:

- Starting codes, to enter certain modes and to begin to accumulate BCC
- Modifiers, sync characters, and data link escape functions (ITB, SYN, DLE)
- Ending codes, to terminate blocks and activate checking functions

Control Characters and Sequences

When transmitting, the adapter prepares to receive when the current address register is equal to the transition address register. The program must ensure that the last character of the change-of-direction sequence is at a location one less than the transition address. When receiving, any change-of-direction character or sequence causes the adapter to terminate the receive operation and issue an op-end interrupt.

BSCA control characters and sequences are described in the following paragraphs and listed in Figure 9-2.

- *SOH or STX* resets control state mode and sets the adapter to data mode. The first SOH or STX after line turnaround resets the BCC buffer and BCC accumulation commences with the following character.
- *ETB or ETX* resets data mode in the adapter and is the last character included in the BCC accumulation. At the master station, the adapter transmits the BCC and the pad character. At the slave station, the adapter compares its BCC accumulation with the BCC(s) received following the ETB or ETX.
- *EOT* indicates the end of a transmission, which may contain more than one message, and resets all stations on the line to control mode. EOT is also transmitted as a negative response to a polling sequence. EOT cannot be immediately preceded by any character other than SYN. To be recognized as a control character, EOT must be followed by four contiguous 1's. They are stored in the four low-order bit positions of the main storage location following the EOT. The four high-order bit positions of this byte should be ignored. When transmitting, the adapter automatically generates the four 1's by sending the trailing PAD character.
- *ENQ* resets data mode in the adapter.

Name	Mnemonic	EBCDIC	ASCII
Start of heading	SOH	SOH	SOH
Start of text	STX	STX	STX
End of transmission block[1]	ETB	ETB	ETB
End of text[1]	ETX	ETX	ETX
End of transmission[1]	EOT	EOT	EOT
Enquiry[1]	ENQ	ENQ	ENQ
Negative acknowledge[1]	NAK	NAK	NAK
Synchronous idle	SYN	SYN	SYN
Data link escape	DLE	DLE	DLE
Intermediate block character	ITB	IUS	US
Even acknowledge[1]	ACK 0	DLE (70)	DLE 0
Odd acknowledge[1]	ACK 1	DLE/	DLE 1
Wait before transmit–pos. ack.[1]	WACK	DLE,	DLE;
Mandatory disconnect[1]	DISC	DLE EOT	DLE EOT
Reverse interrupt[1]	RVI	DLE@	DLE<
Temporary text delay[1]	TTD	STX ENQ	STX ENQ
Transparent start of text	XSTX	DLE STX	
Transparent intermediate block	XITB	DLE IUS	
Transparent end of text[1]	XETX	DLE ETX	
Transparent end of trans. block[1]	XETB	DLE ETB	
Transparent synchronous idle	XSYN	DLE SYN	
Transparent block cancel[1]	XENQ	DLE ENQ	
Transparent TTD[1]	XTTD	DLE STX DLE ENQ	
Data DLE in transparent mode	XDLE	DLE DLE	

[1] Change of direction character

Figure 9-2. Control Characters and Sequences

- *NAK* indicates that the previous transmission block was in error and the receiver is ready to accept a retransmission of the block. NAK is also the *not-ready* reply to multipoint station selection sequences and point-to-point initialization sequences. NAK must be followed by four contiguous 1's to be recognized as a control character. The 1's are stored in the four low-order bit positions of the main storage location following the NAK. The four high-order bit positions of this byte should be ignored. When transmitting, the adapter automatically generates the four 1's by sending the trailing PAD character.

- *SYN* is generated and transmitted automatically by the adapter to establish and maintain synchronism. SYN does not enter BCC or main storage. A SYN from main storage at the transmitting station is transmitted, but does not enter main storage at the receiving station nor BCC accumulation at either station.

- *SYN SYN* is the sync pattern in nontransparent mode. Two contiguous SYN characters are always transmitted immediately following an ITB or XITB, BCC sequence. SYN is also used as a time fill character for a transmit only instruction terminated by ITB or XITB until the next transmit and receive instruction is issued.

- *DLE* alerts the adapter to test the following character for a defined control sequence. In nontransparent data mode, DLE is treated as data.

- *ITB* is included in the BCC and causes the BCC (s) to be sent or compared. Both adapters continue in data mode with the new BCC accumulation starting with the first non-SYN character.

- *ACK 0 and ACK 1* are positive acknowledgements by the receiving station that the previous even-numbered (ACK 0) or odd-numbered (ACK 1) transmission block was received. In data mode, ACK indicates that the last block check character received matched the block check character generated by the adapter. In control mode, ACK indicates that the adapter is ready to receive. ACK always requires a response from the station that receives it. ACK causes the receiving adapter to issue an op-end interrupt request.

- *WACK* signals that the last data block was received correctly but the receiving station is not able to continue receiving. During line initialization, a received WACK means that the remote station is temporarily unable to receive but will be able to receive in a short time. Receiving WACK from the remote station causes the adapter to generate an op-end interrupt.

- *DISC* is transmitted (switched point-to-point networks only) to signal the remote station that the transmitting station is going to disconnect from the line. DISC causes the receiving adapter to generate an op-end interrupt.

- *RVI* is transmitted by a slave station to request that the master station end its transmission and allow the slave to transmit. RVI is transmitted in place of ACK. Successive RVIs can be transmitted only in response to ENQ. RVI causes the receiving adapter to generate an op-end interrupt.

- *TTD* is transmitted by a master to a slave station to inform the slave station that (1) there will be a delay exceeding two seconds in transmitting the next data block, or (2) the master station wishes to cancel the transmission in progress. The slave responds to TTD by transmitting NAK. TTD causes the receiving adapter to generate an op-end interrupt.

- *XSTX* resets control state and sets the adapter to data mode and transparent mode. Unless preceded by SOH ––, XSTX resets the BCC register and BCC accumulation commences with the following character. In transparent mode, the first DLE in each 2-character DLE sequence does not enter BCC or main storage; the second character does, if it is not SYN. Also, the transmitting adapter inserts a DLE for each DLE received from main storage.

- *XITB* causes the same adapter action as ITB and, in addition, resets transparent mode.

- *XETX or XETB* causes the same adapter action as ETX or ETB and, in addition, resets transparent mode.

- *XSYN* is the sync pattern for maintaining synchronism in transparent mode. It does not enter BCC or main storage.

- *XENQ* resets data mode and transparent mode in the adapter.

- *XTTD* serves the function of TTD in transparent mode.

- *XDLE* is interpreted in transparent mode as a valid data byte – hex 10.

Pad Characters

The BSCA generates and sends one PAD character for each change-of-direction character transmitted. If the change-of-direction sequence calls for a BCC character, the PAD character follows the BCC character; otherwise, the PAD character follows the change-of-direction character in the message being transmitted. This PAD character is hex FF.

The BSCA also generates and transmits a hex FF (PAD) character as the second character of the NAK and EOT control character sequences.

When transmission starts, the adapter automatically generates and inserts a PAD character (in this case, a hex 55) ahead of the initial synchronizing sequence. No leading or trailing PAD character (except a PAD character immediately following either EOT or NAK) is stored during receive operations.

BSCA Synchronization

The BSCA receives timing pulses externally from the modem which, in this case, establishes and maintains bit synchronism. The adapter starting to transmit automatically sends two SYNs required for establishing character synchronism at the receiving adapter. The receiving adapter establishes character synchronism by decoding two consecutive SYNs.

An adapter with Internal Clock feature establishes and maintains bit synchronism on its own. For this purpose, the BSCA automatically sends two additional hex 55s preceding the character synchronism pattern.

To maintain character synchronism, the transmitting adapter (master) inserts a synchronization pattern, SYN SYN, at every transmit timeout. The synchronization pattern does not enter BCC or main storage. In transparent mode, the transparent synchronous idle (DLE SYN) is used.

If a transmit only operation is terminated with ITB or XITB, the synchronization pattern, SYN SYN, is transmitted immediately following the BCC(s).

FRAMING THE MESSAGE

The program at the transmitting station must frame the data to be sent with appropriate line control characters. These characters are stored at the receiving station, so the program must allow space for them in storage. When transmitting, the BSCA automatically generates and transmits SYN, PAD, and CRC characters as required for establishing and maintaining synchronism with the remote station and for error checking. When receiving, the BSCA removes all SYN and CRC characters and some PAD characters received from the data being sent to storage. The PAD character following an NAK or EOT *is not* removed by the adapter.

Response characters (ACK 0, ACK 1, WACK, and NAK) are inserted by the stored program, not the transmitting BSCA. They are not stripped by the receiving BSCA. The program must store these characters in a known location so that the program can test them to determine what action to take next.

OP-END INTERRUPT

If enabled, an op-end interrupt occurs at the end of the following BSCA operations:

- Transmit and receive
- Receive initial
- Receive
- Two-second timeout (The BSCA need not be enabled to complete the 2-second timeout operation with an op-end interrupt.)

In a receive type operation, an op-end interrupt is generated when a change-of-direction character is decoded, when the current address equals the stop address, or when a receive timeout occurs.

In a transmit only operation, the interrupt is generated when the current address, transition address, and stop address are all equal.

On a start 2-second timeout operation, an op-end interrupt is generated at the end of the 2-second period.

BSCA OPERATIONS

All operations on the communications line are controlled through a combination of instructions in the system processor and the automatic controls initiated by line control characters and sequences. Figure 9-3 is a basic flowchart of a suggested generalized routine to place the BSCA in operation.

Enable/Disable BSCA

Enable BSCA sets on the data terminal ready line to the modem; disable BSCA sets off the data terminal ready line and resets the BSCA. Power on reset, system reset, or IPL also sets off the data terminal ready line and resets the BSCA.

Since data terminal ready controls switching the modem to the communications channel, enable BSCA is a prerequisite to establish a switched network connection. Disable BSCA is used to disconnect from a switched network. Sufficient time must be allowed for the modem to disconnect from the switched network before the program again enables BSCA. The 2-second timeout may be used to ensure this.

Initialization Sequences

Initialization sequences are defined in *General Information Binary Synchronous Communications,* GA27-3004, and are transmitted by the transmit and receive instructions. Receive initial instruction is defined for receiving initial sequences. The receive initial operation depends on the data link (point-to-point nonswitched, point-to-point switched, or multipoint) selected by the customer.

Receive Initial Operation (Point-to-Point Nonswitched)

On a nonswitched network, receive initial causes the BSCA to hunt for sync. When character sync is established, the adapter sets *busy; receive timeout* then becomes effective; and the following sequence (starting with the first non-SYN character) is stored in the main storage area specified by the current address register. The stop address register should be loaded with the initial current address plus the maximum number of characters received plus 1. The operation is terminated and an interrupt generated when a change-of-direction character is received, the current address and stop address become equal, or a receive timeout occurs.

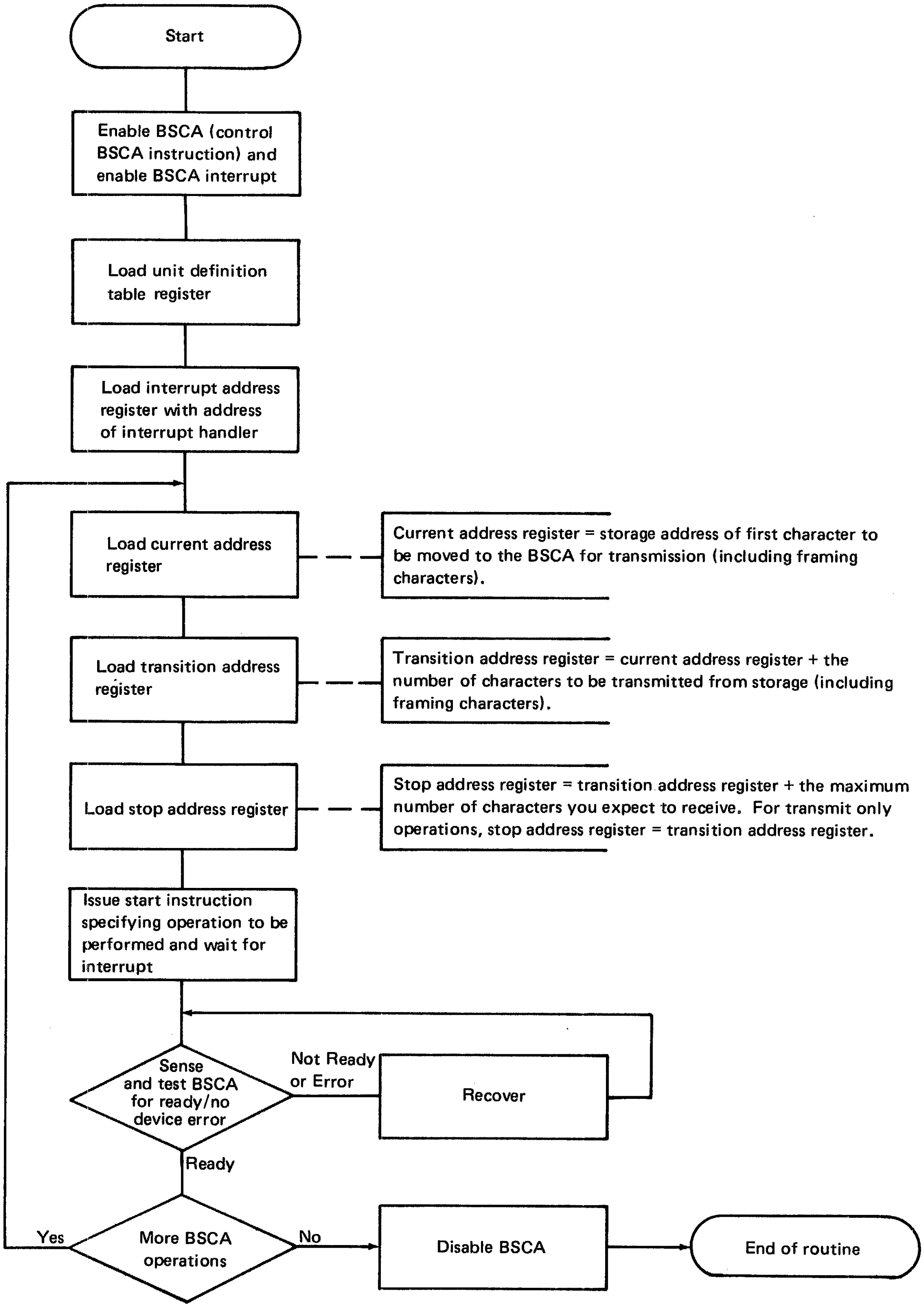

Start
Enable BSCA (control BSCA instruction) and enable BSCA interrupt
Load unit definition table register
Load interrupt address register with address of interrupt handler
Load current address register
Current address register = storage address of first character to be moved to the BSCA for transmission (including framing characters).
Load transition address register
Transition address register = current address register + the number of characters to be transmitted from storage (including framing characters).
Load stop address register
Stop address register = transition address register + the maximum number of characters you expect to receive. For transmit only operations, stop address register = transition address register.
Issue start instruction specifying operation to be performed and wait for interrupt
Sense and test BSCA for ready/no device error
Not Ready or Error
Recover
Ready
More BSCA operations
Yes
No
Disable BSCA
End of routine

Figure 9-3. Initiating BSCA Operations

Receive Initial Operation (Point-to-Point Switched)

On a switched network, receive initial conditions the BSCA to set *busy* as soon as *data set ready* comes up with the call. Receive timeout becomes effective and the BSCA attempts to establish sync.

When character sync is established, the following sequence of received characters (starting with the first non-SYN character) is stored in the main storage area specified by the current address register. The stop address register should be loaded with the initial current address plus one more than the maximum number of characters to be received. As above, the operation is terminated and an interrupt generated when a change-of-direction character is received, the current address and the stop address become equal, or a receive timeout occurs. In the case of a receive timeout, the recovery procedure is to issue the receive only instruction.

Receive Initial Operation (Multipoint Tributary)

Receive initial is used to receive polling and selection sequences on a multipoint network. The stop address register should be loaded with the initial current address plus one less than the maximum number of characters in the polling/selection sequence. A 2-character station address is used. For this operation, the low-order (rightmost) byte of the transition address register must be loaded with the station address. The EBCDIC 2-bit or the ASCII 6-bit of the first station address character received is disregarded; however, both characters of the address received must be identical.

For example, assuming EBCDIC code, if the transition address register is loaded with either XB or XS, the adapter recognizes either BB or SS as the station address. The high-order byte in the transition address register is not used.

The basic mode of BSCA is monitor mode for this operation. In this mode, the BSCA hunts for sync. With character sync established, it monitors the line. All line control characters are decoded and the respective functions are executed, but data is not stored. When a valid EOT sequence is received, control mode is set.

In control mode, the BSCA monitors for its station address. If it is not detected, the BSCA continues monitoring the line. A decoded SOH or STX drops control mode and puts the BSCA back into monitor mode. If the station address is decoded as the first non-SYN characters after establishing character sync in control mode, the BSCA immediately enters addressed mode and transfers the sequence, starting with the second station address character, into the main storage area specified by the current address register. The operation is terminated and an interrupt is generated when a change-of-direction character is received, current address and stop address are equal, or a receive timeout occurs.

Transmit and Receive Operation

The transmit and receive instruction is used for any type of transmission; that is, for control sequences or text data. It sets the BSCA to transmit mode, then takes characters from main storage and transmits them onto the line. BCC accumulation, data mode, and transparent mode are set, depending on the type of line control characters fetched from storage. Transmission proceeds until the current address register equals the transition address register, which turns the adapter around to receive mode under the same instruction.

In receive mode, the BSCA hunts for sync, then stores the characters received into main storage. As in transmit, the detail function on receive depends on the particular line control characters received.

The operation is terminated and an interrupt generated when a change-of-direction sequence is received, the current address register equals the stop address register, or a receive timeout occurs. At this time the status bits can be interrogated.

The reason for this combined transmit and receive instruction is the required fast response between the two operations. The effect of the current address, transition address, and stop address on the control sequences or text data is shown in Figure 9-4.

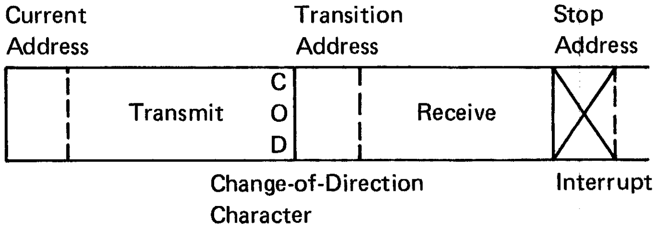


Figure 9-4. I/O Area and Address Register Contents at Start of Transmit and Receive Operation

The transmit and receive instruction is used by both the control and the tributary; that is, to send data and receive the reply, and to send the reply and receive data.

The current address specifies the beginning of the combined transmit-receive field. A +1 is added to the current address each time a character is fetched or stored. The transition address register specifies the beginning of the receive field and must be loaded with the initial current address plus the number of characters to be transmitted. The stop address register specifies the end of the transmit and receive field and should be loaded with the transition address plus one more than the number of characters to be received.

The current, transition, and stop addresses are unrestricted 2-byte addresses, except that a zero-length transmit field is not permitted. There is no maximum restriction in block length; that is, current, transition, and stop addresses. Each is a 16-bit address. If the stop address is equal to the transition address, the instruction becomes a transmit only operation.

At the start of the transmit and receive operation, the adapter sends hex 55 (two additional hex 55s if the Internal Clock feature is installed), and two SYN characters. During transmit, the BSCA inserts the sync pattern, SYN SYN, at every transmit timeout. SYN is not accumulated in the BCC and does not enter main storage. BCC compare takes place when an ITB, ETB, or ETX is received.

If the adapter entered data mode by receiving an STX or SOH, then only ETB, ETX, and ENQ are considered valid change-of-direction sequences. Outside of data mode, all turnaround sequences are considered valid change-of-direction sequences and will terminate the operation.

Busy stays on with the transmit and receive instruction throughout both sections of the operation until interrupt occurs. Interrupt occurs before the stop address is reached if a change-of-direction sequence is received.

ITB Operation

The IUS/US character is interpreted as the ITB control character to activate the ITB function. The master sends the BCC after the ITB, the slave receives and compares it; both stations continue transferring more data immediately thereafter with no line turnaround.

For nontransparent data, the master can (1) transmit all ITB blocks in a single transmit and receive instruction or (2) transmit each ITB block in a transmit only instruction as described for transparent ITBs in the next section.

When the slave receives an ITB character, the adapter remains busy and proceeds to receive the next ITB block. This continues until a change-of-direction character is recognized. When the ending sequence – ETB, ETX, or ENQ – is received, it is stored and an op-end interrupt occurs. At this time, the program checks the status bits to determine the appropriate reply.

Transparent Operation

In transmitting and receiving data, transparent mode is set by the contiguous sequence DLE STX. In transparency, the transmitting adapter automatically inserts a second DLE preceding each DLE from storage (except DLE STX), which is stripped by the receiving BSCA. The additional DLE does not enter BCC accumulation.

Either ETB, ETX, ITB, or ENQ ends transparent mode at the master if it is at a location one less then the transition address. Due to this coincidence, the master BSCA inserts a DLE so that the single DLE followed by ETB, ETX, ITB, or ENQ tells the slave to leave transparent mode. This DLE is stripped by the slave and is not included in the BCC at either station.

The use of the transition address to point to the control ETB, ETX, or ENQ allows replies to transparent data to consist of any number of characters. Limited conversational operation is possible in transparent, as well as nontransparent mode.

Each ITB block of transparent data must be transmitted with its own transmit instruction. No turnaround takes place after the ITB, and the adapter inserts at least two SYN characters (more, if necessary) until the next instruction is issued. During this period the adapter is not busy. Every ITB block must start out with DLE STX to again set transparent mode.

Disconnect Operation

The program can perform a disconnect operation on a switched network by giving a disable BSCA instruction, which drops the *data terminal ready* line to the modem. It should previously transmit a DLE EOT sequence with a transmit instruction to inform the other station that it is going *on-hook.* A received DLE EOT sequence should cause the slave station program to perform a disconnect operation.

Sufficient time must be allowed for the disconnect to occur before the program again enables BSCA. The 2-second timeout may be used to ensure this.

Receive Operation

The receive instruction is defined for use when it is necessary to perform a receive operation after termination of the previous instruction, such as when a receive timeout has occurred. The operation is the same as the receive part of the transmit and receive operation. The BSCA is busy for the entire operation.

This instruction must be used as a result of a receive timeout during a receive initial operation on a switched network.

Two-Second Timeout

This control code function is provided to obtain a 2-second delay before transmitting a TTD or WACK. The start 2-second timeout must be given only with the control instruction. When the timeout is completed, the BSCA generates an interrupt. The BSCA is not busy when doing a 2-second timeout. It can be terminated by issuing any BSCA SIO. Start 2-second timeout must not be issued if the adapter is busy.

The BSCA does not need to be enabled to perform the 2-second timeout operation.

Loading the Registers

Load instructions are used to load the unit definition table register, interrupt address register, current address register, transition address register, and stop address register.

Sensing

Sense instructions are used to store the current address register and BSCA status bytes. Figure 9-5 describes the BSCA status bytes. Byte 1 is stored in the location addressed by the operand address of the sense BSCA status instruction. Byte 2 is stored in the next lower storage location.

Data Checking

As the remote station transmits messages, it generates block check characters from the data bits transmitted. As these bits are received at the local communications adapter, the adapter generates a similar block check character from the data bits it receives. Each time the remote station transmits an ITB, ETB, or ETX character, it also transmits its block check characters. The local communications adapter compares these block check characters that it receives from the line with the block check characters that it generated from the data bits it received from the line. If the block check characters generated by the local communications adapter do not match the block check characters received from the line, the CRC/LRC/VRC status bit is set. While servicing the interrupt resulting from an ETB or ETX character, the program must sample the status bits and determine if a block check has occurred.

Byte	Bit	Meaning When On	Reset Off By
1	0-5	Not assigned.	
1	6	Data set ready. This indicates that the modem is ready to operate and that the BSCA has been enabled.	Modem losing its ready state or BSCA being disabled
1	7	Not assigned.	
2	0	Timeout status. A receive timeout (3.25 seconds) occurred during a receive operation.	Any noncontrol BSCA SIO
2	1	Data check during receive operation. a. A CRC compare check occurred (EBCDIC). b. A VRC check occurred (ASCII). *Note:* Characters having VRC checks are distinguished by a high-order bit in main storage. These characters are never recognized as control characters by the BSCA.	Any noncontrol BSCA SIO
2	2	Not assigned.	

Figure 9-5. (Part 1 of 2) BSCA Status Indications

Byte	Bit	Meaning When On	Reset Off By
2	3	Overrun. BSCA did not move a character to or from main storage before the next character had to be moved to accommodate the line. An overrun does not terminate the operation.	Any noncontrol BSCA SIO
2	4	Invalid ASCII. BSCA found leftmost bit in ASCII byte on during transmit operation.	Any noncontrol BSCA SIO
2	5	Abortive disconnect. Indicates BSCA on switched network was enabled, then the modem became ready, then not ready. This indicates the connection has been released and causes data terminal ready to turn off. The program must allow enough time for a forced disconnect (BSCA-controlled) to occur. The program can use the 2-second time-out to ensure this.	Disable BSCA
2	6	Adapter busy.	Op-end interrupt
2	7	Not assigned.	

Figure 9-5. (Part 2 of 2) BSCA Status Indications

If the interruption is the result of an ETB or ETX character, the result of the block check compare determines which response character should be sent. The positive acknowledgement characters alternate; ACK 0 is transmitted in response to even-numbered blocks and ACK 1 is 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 master station assumes that a block of data or heading was missed and initiates an error recovery procedure.

When block checking is initiated by ITB, the result of the block check compare is not transmitted immediately. Instead, if the block check compare is equal, the communications adapter continues to receive and store characters. If the block check is incorrect, the VRC/LRC/CRC status bit is set on to indicate that a block check noncompare occurred. When the next ETB or ETX character is received, it is stored and an interruption is generated. The status bits are sensed and tested to determine if all data was received correctly. An ENQ character also terminates the receive operation.

The lost data check is a program function. When the current address equals the stop address, a lost data error is indicated.

Suggested Error Recovery Procedures

At the end of every transmit and/or receive operation, the program should sense the BSCA status bytes. Test the status bits and perform the procedures for recovering from the error in the order given in Figure 9-6. The program must check for lost data and analyze the last two characters received to detect an abnormal response error.

Priority	Status		Error Condition	Error Recovery Procedure (Recommended Program Action)
	Byte	Bit		
1	2	4	Invalid ASCII character	All cases—Action 1
2	2	5	Abortive disconnect	All cases—Action 1
3	2	3	Overrun	Control mode—Action 5 Slave—Action 4 Master—Action 3
4	2	0	Receive timeout	Receive initial (switched)—Action 8 Control mode—Action 5 Slave—Action 4 Master—Action 3
5	2	1	CRC/LRC/VRC	Control mode—Action 5 Slave—Action 2 Master—Action 3
	Program detected error [1]		Lost data (CAR=SAR on receive)	
6	Program detected error [1]		Abnormal response	Control mode—Action 5 Slave: Absence of initial STX or terminal ETB/ETX—Action 4 Master: Improper ACK immediately preceded by timeout—Action 6 Master: Any response other than proper ACK or EOT—Action 7

[1] The program should provide lost data detection.

Action Table

1. Permanent error occurred — operator must restart.
2. NAK was transmitted and received — retransmit data.
3. ENQ was transmitted and received — retransmit last response N times.
4. Issue receive portion of previous operation N times.
5. Retry last operation M times.
6. T&R last text. This is an intermediate action within a recovery procedure; it is taken by the master each time it transmits text, times out on receive, transmits ENQ, and receives the improper ACK. A system hangup will not occur because of the limitation on action 3.
7. T&R ENQ once. If response is NAK, do action 6 N times. If invalid response reoccurs, do action 1.
8. Issue SIO receive instruction.

The value M should be equal to or greater than N.
The value N should be a minimum of 7.

When M or N is reached (permanent error), the program should cancel the job and tell the operator the nature of the error condition by some means (such as an error message). Operator intervention is then required and the procedure is either to completely restart the job or to continue with the next job.

Note: A processor check stop causes an immediate cancel.

Figure 9-6. BSCA Error Conditions and Recovery Procedures

System and Error Statistics

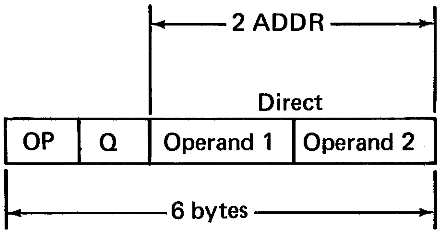
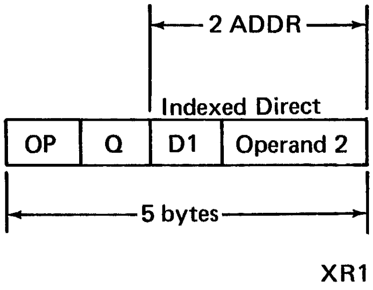
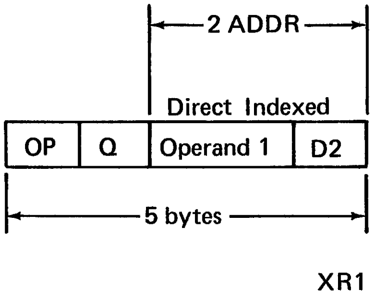
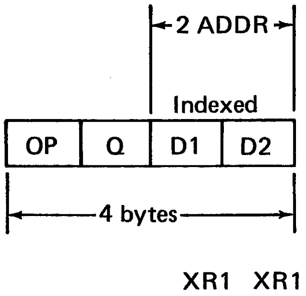
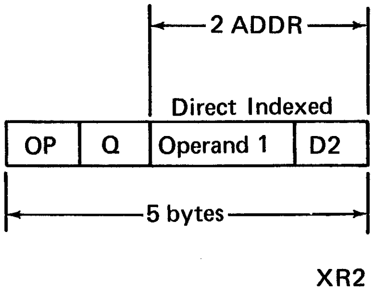
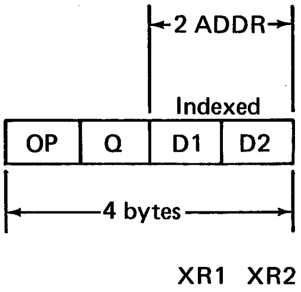
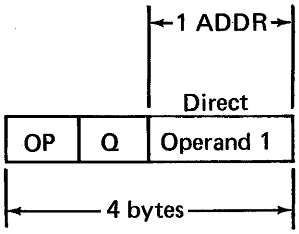
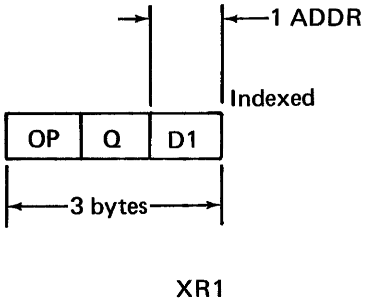
The user program should accumulate the following information for BSCA as a diagnostic aid. This data should be logged to disk storage at close time.

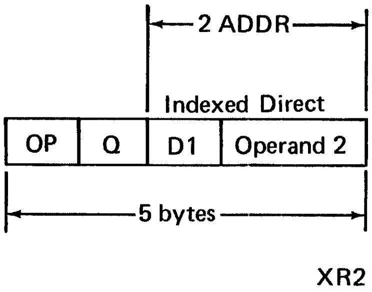
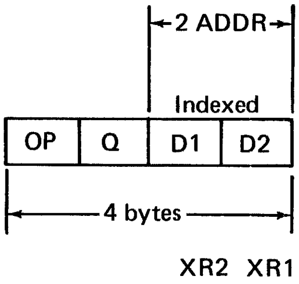
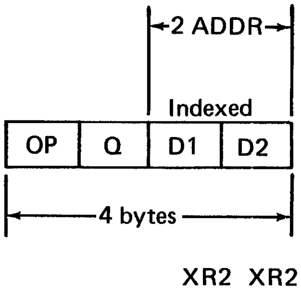
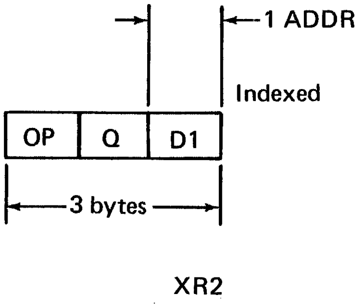
Transmission Statistics

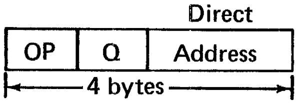
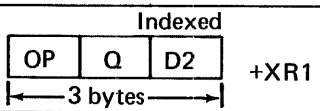
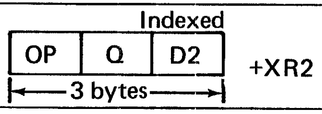
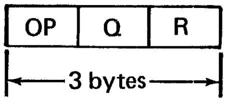
1. A count of data blocks transmitted successfully, as proven by the receipt of valid affirmative responses.
2. A count of data blocks that result in a negative response from the slave.
3. A count of invalid or no-response replies to transmitted data blocks and to following ENQ control characters.
4. A count of slave station terminations (EOT in lieu of normal response to text).
5. A count of overrun checks on transmit operations.

Reception Statistics

1. A count of data blocks received correctly.
2. A count of data blocks received with BCC (or VRC) errors.
3. A count of ENQ characters received in message transfer state as a request from the master station to transmit the last response. ENQ as reponse to a transmitted WACK should not be included.
4. A count of master station forward terminations (TTD/NAK/EOT sequences).
5. A count of overrun checks on receive operations.

Op Code	Type	Op Code	Type
00 04 06 07 08 0A 0B 0C 0D 0E 0F	 2 ADDR Direct OP Q Operand 1 Operand 2 6 bytes	40 44 46 47 48 4A 4B 4C 4D 4E 4F	 2 ADDR Indexed Direct OP Q D1 Operand 2 5 bytes XR1
10 14 16 17 18 1A 1B 1C 1D 1E 1F	 2 ADDR Direct Indexed OP Q Operand 1 D2 5 bytes XR1	50 54 56 57 58 5A 5B 5C 5D 5E 5F	 2 ADDR Indexed OP Q D1 D2 4 bytes XR1 XR1
20 24 26 27 28 2A 2B 2C 2D 2E 2F	 2 ADDR Direct Indexed OP Q Operand 1 D2 5 bytes XR2	60 64 66 67 68 6A 6B 6C 6D 6E 6F	 2 ADDR Indexed OP Q D1 D2 4 bytes XR1 XR2
30 31 34 35 36 38 39 3A 3B 3C 3D		70 71 74 75 76 78 79 7A 7B 7C 7D	

Op Code	Type
80 84 86 87 88 8A 8B 8C 8D 8E 8F	 2 ADDR Indexed Direct OP Q D1 Operand 2 5 bytes XR2
90 94 96 97 98 9A 9B 9C 9D 9E 9F	 2 ADDR Indexed OP Q D1 D2 4 bytes XR2 XR1
A0 A4 A6 A7 A8 AA AB AC AD AE AF	 2 ADDR Indexed OP Q D1 D2 4 bytes XR2 XR2
B0 B1 B4 B5 B6 B8 B9 BA BB BC BD	

Op Code	Type
C0 C1 C2	
D0 D1 D2	
E0 E1 E2	
F0 F1 F2 F3 F4	

Appendix B. EBCDIC Code Meanings

Below are definitions of the column headings used in the following table.

Hex Value. The internal EBCDIC code used by the system, expressed as a hexadecimal notation.

Binary Value. The internal EBCDIC code expressed as a binary notation.

Printer Graphic. The graphic printed by this system for the EBCDIC code shown. For example, graphics printed for the EBCDIC code stored in the print data field (the field specified by the printer IOB as data to be printed) correspond to the binary values shown in the chart. Hence, a main storage value of hex 6C is printed as %.

Related Keyboard Key. This column specifies the key that must be pressed to send the associated EBCDIC code to the keyboard/display screen IOB as a data byte. For example, pressing the ENTER key on the keyboard stores hex 10 in the keyboard/display screen IOB data byte.

Display Screen Graphic. This column shows the graphic that is displayed on the display screen for the associated main storage EBCDIC code shown in the binary value column. For example, if the program issues a start-keyboard/display-screen IOB instruction, and if the IOB specifies that a a field in main storage be transferred to the keyboard/display screen buffer and subsequently displayed, the characters displayed for the EBCDIC binary values in the bytes moved from main storage correspond to the associated graphics shown in the display screen graphic column. A main storage value of hex 50 is displayed as &.

Communications Character. The system may display data entered from a diskette prepared on a 3741 with a communications feature. In such cases, the communications characters shown in this column correspond to the EBCDIC binary value on the table. For example, SYN is displayed as graphic $\underline{2}$, but is not printed (hex 32 has no printable graphic).

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character
00	00000000				
01	00000001		INQ	A	SOH
02	00000010			B	STX
03	00000011			C	ETX
04	00000100			D	
05	00000101			E	
06	00000110		ERROR RESET	F	
07	00000111			G	
08	00001000			H	
09	00001001			I	
0A	00001010		↑	¢	
0B	00001011		↓	.	
0C	00001100		←	<	
0D	00001101		→	(	
0E	00001110			+	
0F	00001111			|	
10	00010000		ENTER	&	DLE
11	00010001		ENTER +	J	
12	00010010		ENTER -	K	
13	00010011		FIELD ADV	L	
14	00010100		REC ADV	M	
15	00010101		FIELD BKSP	N	
16	00010110		REC BKSP	O	
17	00010111		DUP (SHIFT)	P	
18	00011000		DUP	Q	
19	00011001		ROLL ↑	R	
1A	00011010		ROLL ↓	!	
1B	00011011		← (SHIFT)	$	
1C	00011100		→ (SHIFT)	*	
1D	00011101			)	
1E	00011110			;	
1F	00011111			¬	ITB
20	00100000			-	
21	00100001			/	
22	00100010			S	
23	00100011			T	
24	00100100			U	
25	00100101			V	
26	00100110			W	ETB
27	00100111			X	
28	00101000			Y	
29	00101001			Z	
2A	00101010			\	
2B	00101011			,	
2C	00101100			%	
2D	00101101			_	ENQ

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character
2E	00101110			$\underline{>}$	
2F	00101111			$\underline{?}$	
30	00110000		CMD	$\underline{0}$	
31	00110001			$\underline{1}$	
32	00110010			$\underline{2}$	SYN
33	00110011			$\underline{3}$	
34	00110100			$\underline{4}$	
35	00110101			$\underline{5}$	
36	00110110			$\underline{6}$	
37	00110111			$\underline{7}$	EOT
38	00111000			$\underline{8}$	
39	00111001			$\underline{9}$	
3A	00111010			$\underline{:}$	
3B	00111011			$\underline{\#}$	
3C	00111100			$\underline{@}$	
3D	00111101			$\underline{'}$	NAK
3E	00111110			$\underline{=}$	
3F	00111111			$\underline{''}$	
40	01000000	Blank	Space	Blank	Blank
41	01000001			A	
42	01000010			B	
43	01000011			C	
44	01000100			D	
45	01000101			E	
46	01000110			F	
47	01000111			G	
48	01001000			H	
49	01001001			I	
4A	01001010	¢	¢	¢	¢
4B	01001011	.	.	.	.
4C	01001100	<	<	<	<
4D	0100110	(	(	(	(
4E	01001110	+	+	+	+
4F	01001111	|	|	|	|
50	01010000	&	&	&	&
51	01010001			J	
52	01010010			K	
53	01010011			L	
54	01010100			M	
55	01010101			N	
56	01010110			O	
57	01010111			P	
58	01011000			Q	
59	01011001			R	
5A	01011010	!	!	!	!
5B	01011011	$	$	$	$

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character
5C	01011100	*	*	*	*
5D	01011101	)	)	)	)
5E	01011110	;	;	;	;
5F	01011111	¬	¬	¬	¬
60	01100000	—	—	—	—
61	01100001	/	/	/	/
62	01100010			S	
63	01100011			T	
64	01100100			U	
65	01100101			V	
66	01100110			W	
67	01100111			X	
68	01101000			Y	
69	01101001			Z	
6A	01101010			\	
6B	01101011	,	,	,	,
6C	01101100	%	%	%	%
6D	01101101	_	_	_	_
6E	01101110	>	>	>	>
6F	01101111	?	?	?	?
70	01110000			0	(70)
71	01110001			1	
72	01110010			2	
73	01110011			3	
74	01110100			4	
75	01110101			5	
76	01110110			6	
77	01110111			7	
78	01111000			8	
79	01111001			9	
7A	01111010	:	:	:	:
7B	01111011	#	#	#	#
7C	01111100	@	@	@	@
7D	01111101	'	'	'	'
7E	01111110	=	=	=	=
7F	01111111	"	"	"	"
80	10000000			_	
81	10000001			A̲	
82	10000010			B̲	
83	10000011			C̲	
84	10000100			D̲	
85	10000101			E̲	
86	10000110			F̲	
87	10000111			G̲	

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character
88	10001000			H	
89	10001001			I	
8A	10001010			¢	
8B	10001011			.	
8C	10001100			<	
8D	10001101			(	
8E	10001110			+	
8F	10001111			|	
90	10010000			&	
91	10010001			J	
92	10010010			K	
93	10010011			L	
94	10010100			M	
95	10010101			N	
96	10010110			O	
97	10010111			P	
98	10011000			Q	
99	10011001			R	
9A	10011010			!	
9B	10011011			$	
9C	10011100			*	
9D	10011101			)	
9E	10011110			;	
9F	10011111			¬	
A0	10100000			-	
A1	10100001			/	
A2	10100010			S	
A3	10100011			T	
A4	10100100			U	
A5	10100101			V	
A6	10100110			W	
A7	10100111			X	
A8	10101000			Y	
A9	10101001			Z	
AA	10101010			\	
AB	10101011			,	
AC	10101100			%	
AD	10101101			_	
AE	10101110			>	
AF	10101111			?	
B0	10110000			0	
B1	10110001			1	
B2	10110010			2	
B3	10110011			3	
B4	10110100			4	
B5	10110101			5	

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character
B6	10110110			$\underline{6}$	
B7	10110111			$\underline{7}$	
B8	10111000			$\underline{8}$	
B9	10111001			$\underline{9}$	
BA	10111010			$\underline{:}$	
BB	10111011			$\underline{\#}$	
BC	10111100			$\underline{@}$	
BD	10111101			$\underline{\text{'}}$	
BE	10111110			$\underline{=}$	
BF	10111111			$\underline{\text{"}}$	
C0	11000000			Blank	
C1	11000001	A	A	A	A
C2	11000010	B	B	B	B
C3	11000011	C	C	C	C
C4	11000100	D	D	D	D
C5	11000101	E	E	E	E
C6	11000110	F	F	F	F
C7	11000111	G	G	G	G
C8	11001000	H	H	H	H
C9	11001001	I	I	I	I
CA	11001010			¢	¢
CB	11001011			.	.
CC	11001100			<	<
CD	11001101			(	(
CE	11001110			+	+
CF	11001111			|	|
D0	11010000			&	
D1	11010001	J	J	J	J
D2	11010010	K	K	K	K
D3	11010011	L	L	L	L
D4	11010100	M	M	M	M
D5	11010101	N	N	N	N
D6	11010110	O	O	O	O
D7	11010111	P	P	P	P
D8	11011000	Q	Q	Q	Q
D9	11011001	R	R	R	R
DA	11011010			!	
DB	11011011			$	
DC	11011100			*	
DD	11011101			)	
DE	11011110			;	
DF	11011111			¬	
E0	11100000		\	–	
E1	11100001			/	
E2	11100010	S	S	S	S
E3	11100011	T	T	T	T

Hex Value	Binary Value	Printer Graphic	Related Keyboard Key	Display Screen Graphic	Communications Character
E4	11100100	U	U	U	U
E5	11100101	V	V	V	V
E6	11100110	W	W	W	W
E7	11100111	X	X	X	X
E8	11101000	Y	Y	Y	Y
E9	11101001	Z	Z	Z	Z
EA	11101010			\	
EB	11101011			,	
EC	11101100			%	
ED	11101101			—	
EE	11101110			>	
EF	11101111			?	
F0	11110000	0	0	0	0
F1	11110001	1	1	1	1
F2	11110010	2	2	2	2
F3	11110011	3	3	3	3
F4	11110100	4	4	4	4
F5	11110101	5	5	5	5
F6	11110110	6	6	6	6
F7	11110111	7	7	7	7
F8	11111000	8	8	8	8
F9	11111001	9	9	9	9
FA	11111010			:	
FB	11111011			#	
FC	11111100			@	
FD	11111101			'	
FE	11111110			=	
FF	11111111			"	

Appendix C. Powers of Two Table

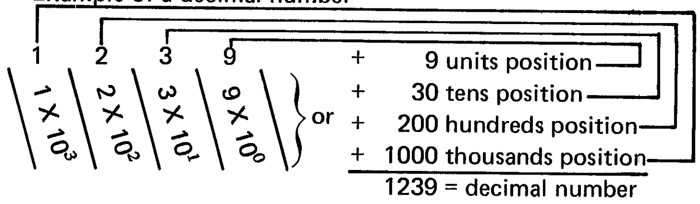
2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
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

BINARY NUMBER NOTATION

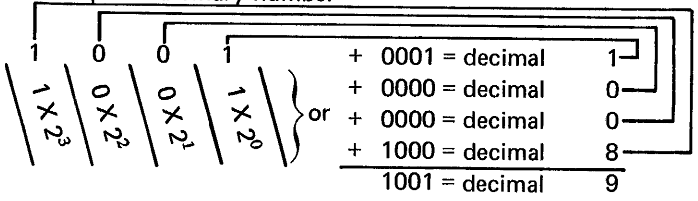
A binary number system uses a base of two. A base-of-two number system can be compared with the base-of-ten (decimal) number system.

Decimal Number	Binary Number Equivalent
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 sytem allows counting to 10 in each position, from units to tens to hundreds to thousands, etc. The binary system allows counting to two in each position. CE panel displays are in binary form: a bit light on is indicated by a 1; a bit light off is indicated by 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 hex 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 hex position. The following table shows the comparable values of the three number systems:

Decimal	Binary	Hex
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	Hex
16	0001 0000	10
17	0001 0001	11
18	0001 0010	12
19	0001 0011	13
20	0001 0100	14
21	0001 0101	15
etc	etc	etc

Remember that the computer deals only with binary. However, 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 hex value of 1E13. This is easier to state than the string of 1's and 0's.

Appendix E. Hexadecimal-Decimal Conversion Tables

The tables in this appendix provide direct conversion of decimal and hexadecimal numbers in these ranges:

Hex	Decimal
000 to FFF	0000 to 4095

For numbers outside the range of the tables, add the following values to the table figures:

Hex	Decimal
1000	4096
2000	8192
3000	12288
4000	16384
5000	20480
6000	24576
7000	28672
8000	32768

Three-position hex values composed of the numerals listed at the side and top of the tables convert to the decimal values listed inside the tables. Decimal values inside the tables convert to hex values composed of the coordinate numerals at the side and top of the tables.

	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

	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

	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

	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

	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	3819
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

Appendix F. Polling and Addressing Characters for System/32 Tributary Stations

Polling and addressing characters must be used together in certain pairs: that is, once a polling character is selected, the complementary addressing character is determined; once an addressing character is selected, the complementary polling character is determined.

The pairs of valid polling and addressing characters for both EBCDIC and ASCII code are as follows:

EBCDIC Code

Polling Character	Hexadecimal Representation	Addressing Character	Hexadecimal Representation
BB	C2C2	SS	E2E2
CC	C3C3	TT	E3E3
DD	C4C4	UU	E4E4
EE	C5C5	VV	E5E5
FF	C6C6	WW	E6E6
GG	C7C7	XX	E7E7
HH	C8C8	YY	E8E8
II	C9C9	ZZ	E9E9
JJ	D1D1	11	F1F1
KK	D2D2	22	F2F2
LL	D3D3	33	F3F3
MM	D4D4	44	F4F4
NN	D5D5	55	F5F5
OO	D6D6	66	F6F6
PP	D7D7	77	F7F7
QQ	D8D8	88	F8F8
RR	D9D9	99	F9F9

ASCII Code

Polling Character	Hexadecimal Representation	Addressing Character	Hexadecimal Representation
AA	4141	aa	6161
BB	4242	bb	6262
CC	4343	cc	6363
DD	4444	dd	6464
EE	4545	ee	6565
FF	4646	ff	6666
GG	4747	gg	6767
HH	4848	hh	6868
II	4949	ii	6969
JJ	4A4A	jj	6A6A
KK	4B4B	kk	6B6B
LL	4C4C	ll	6C6C
MM	4D4D	mm	6D6D
NN	4E4E	nn	6E6E
OO	4F4F	oo	6F6F
PP	5050	pp	7070
QQ	5151	qq	7171
RR	5252	rr	7272
SS	5353	ss	7373
TT	5454	tt	7474
UU	5555	uu	7575
VV	5656	vv	7676
WW	5757	ww	7777
XX	5858	xx	7878
YY	5959	yy	7979
ZZ	5A5A	zz	7A7A

To specify polling or addressing characters in the ADDR-nn parameter of the SETR utility control statement or the OVERRIDE command statement format, give the hex representation of one of the addressing characters. It will be duplicated by the system to provide two characters. At the same time, the corresponding polling characters will be determined.

For example, ADDR-E7 is given to specify the EBCDIC addressing characters XX and the corresponding polling characters GG. ADDR-70 is given to specify the ASCII addressing characters pp and the corresponding polling characters PP.

The ASCII coded character set is shown in the following chart (the EBCDIC coded character set is described in Appendix B). The data link control characters recognized by System/32 are listed in Chapter 9.

ASCII Codes

		Main Storage Bit Positions 0, 1, 2, 3															
Main Storage Bit Positions		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
4, 5, 6, 7	Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL	DLE	SP	0	@	P	`	p								
0001	1	SOH	DC1	!	1	A	Q	a	q								
0010	2	STX	DC2	"	2	B	R	b	r								
0011	3	ETX	DC3	#	3	C	S	c	s								
0100	4	EOT	DC4	$	4	D	T	d	t								
0101	5	ENQ	NAK	%	5	E	U	e	u								
0110	6	ACK	SYN	&	6	F	V	f	v								
0111	7	BEL	ETB	'	7	G	W	g	w								
1000	8	BS	CAN	(	8	H	X	h	x								
1001	9	HT	EM	)	9	I	Y	i	y								
1010	A	LF	SUB	*	:	J	Z	j	z								
1011	B	VT	ESC	+	;	K	[	k	{								
1100	C	FF	FS	,	<	L	\	l	¦								
1101	D	CR	GS	-	=	M	]	m	}								
1110	E	SO	RS	.	>	N	¬	n	~								
1111	F	SI	US	/	?	O	_	o	DEL								

Index

Index of Instructions by Machine Code Sequence

READER'S COMMENT FORM

IBM System/32
Functions
Reference Manual

GA21-9176-1

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*

Due to the current paper shortage, we will not send a reply to your comments unless you check the box below.

I would like a reply. □

Name ____

Address ____

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

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
5775D Glenridge Drive N.E.
Atlanta, Georgia 30301
(USA Only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

GA21-9176-1

IBM System/32 Functions Reference (File No. S32-01) Printed in U.S.A. GA21-9176-1

International Business Machines Corporation
General Systems Division
5775D Glenridge Drive N.E.
Atlanta, Georgia 30301
(USA Only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)