IBM SYSTEM/3

IBM System/3 Model 10 Components Reference Manual

GA21-9103-5
File No. S3-01

Preface

This reference manual is intended for persons interested in the operation and characteristics of IBM System/3 Model 10. It is assumed that the reader is familiar with programming and with data processing terms. For more information about binary communications concepts, data link operations, transmission codes, message formats, etc., consult *General Information – Binary Synchronous Communications Manual,* GA27-3004. For a comprehensive listing of IBM teleprocessing publications refer to *IBM Tele-Processing Bibliography,* GA24-3089. For a list of associated IBM System/3 publications, refer to *IBM System/3 Bibliography*, GC20-8080. For detailed information about features, operations, and procedures of the various System/3 Model 10 I/O features and RPQs, see the publications specified at the right.

Name of Publication	**Order Number**
IBM 1255 Magnetic Character Reader Components Description Manual	GA24-3542
IBM System/360 Component Description – IBM 1270 Optical Reader/Sorter Manual	GA19-0035
IBM 1403 Printer Component Description	GA24-3073
IBM 1442 Card Read Punch Manual	GA24-3119
IBM 3410/3411 Magnetic Tape Subsystem Component Summary	GA32-0015
IBM 3881 Optical Mark Reader Models 1 and 2 Reference Manual and Operator's Guide	GA21-9143
IBM 3881 Optical Mark Reader Forms Kit	GC20-1750
IBM 3881 Mark Reader Systems Design Guide	GC20-1751
Multiple Line Terminal Adapter RPQ Program Reference and Component Description Manual	GC21-7560
IBM 96-Column Card Reference Manual	GA21-9125

Sixth Edition (May 1974)

This is a reprint of GA21-9103-4, incorporating Technical Newsletter GN21-0169 dated February 28, 1974.

Changes are periodically made to the information herein; before using this publication, consult the latest IBM System/3 Bibliography, GC20-8080 for editions that are applicable and current.

Requests 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 has been removed, comments may be addressed to IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901. Comments become the property of IBM.

IBM System/3
Model 10
Components
Reference Manual

Contents

The IBM System/3 (Figure 1-1) extends the use of stored-program data processing to the smaller data processing users. With its high internal speed and new, extended-capacity card, it is possible to perform operations in a single pass of the cards that formerly required two or more passes on various machines. The minimum configuration card system (processing unit, line printer, and multi-function card unit) provides most of the functions a punched card accounting installation can perform.

SYSTEM CONFIGURATION

Figure 1-2 shows the units that make up System/3. Each 96-column card System/3 Model 10 must have the following:

1. An IBM 5410 Processing Unit. The basic storage size is 8,192 bytes. Additional storage is available as shown in Figure 1-2. Basic storage size for program supported disk system is 12,288 bytes. Included as special features are dual programming, which allows the operation of two independent programs at the same time, a serial input/output channel, and a binary synchronous communications adapter (BSCA).
2. An IBM 5424 Multi-Function Card Unit. This unit provides the combined functions of a card reader, a card punch, an interpreter, a sorter, and a collator.
3. An IBM 5203 or 1403 Printer. These printers are referred to as line printers to distinguish them from the printing function of the multi-function card unit.

A minimum configuration disk oriented system that uses the IBM system control program consists of an IBM 5410 Model A13, either an IBM 5203 Printer or an IBM 1403 Printer, an IBM 5444 Disk Drive, and either (1) an IBM 5424 MFCU or (2) an IBM 1442 Card Read Punch and an IBM 5422 Disk Enclosure (for the IBM 5444 Disk Drive).

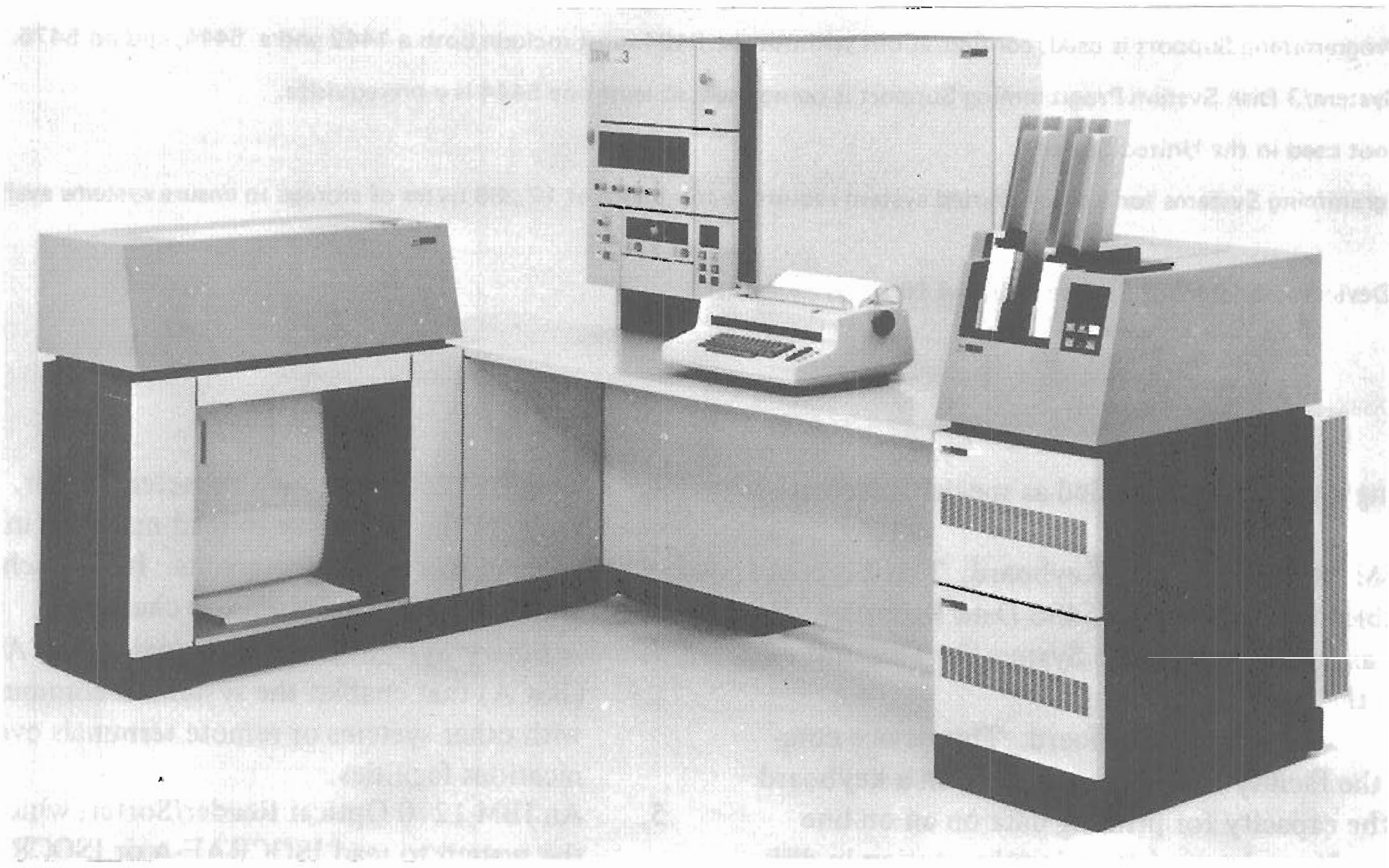
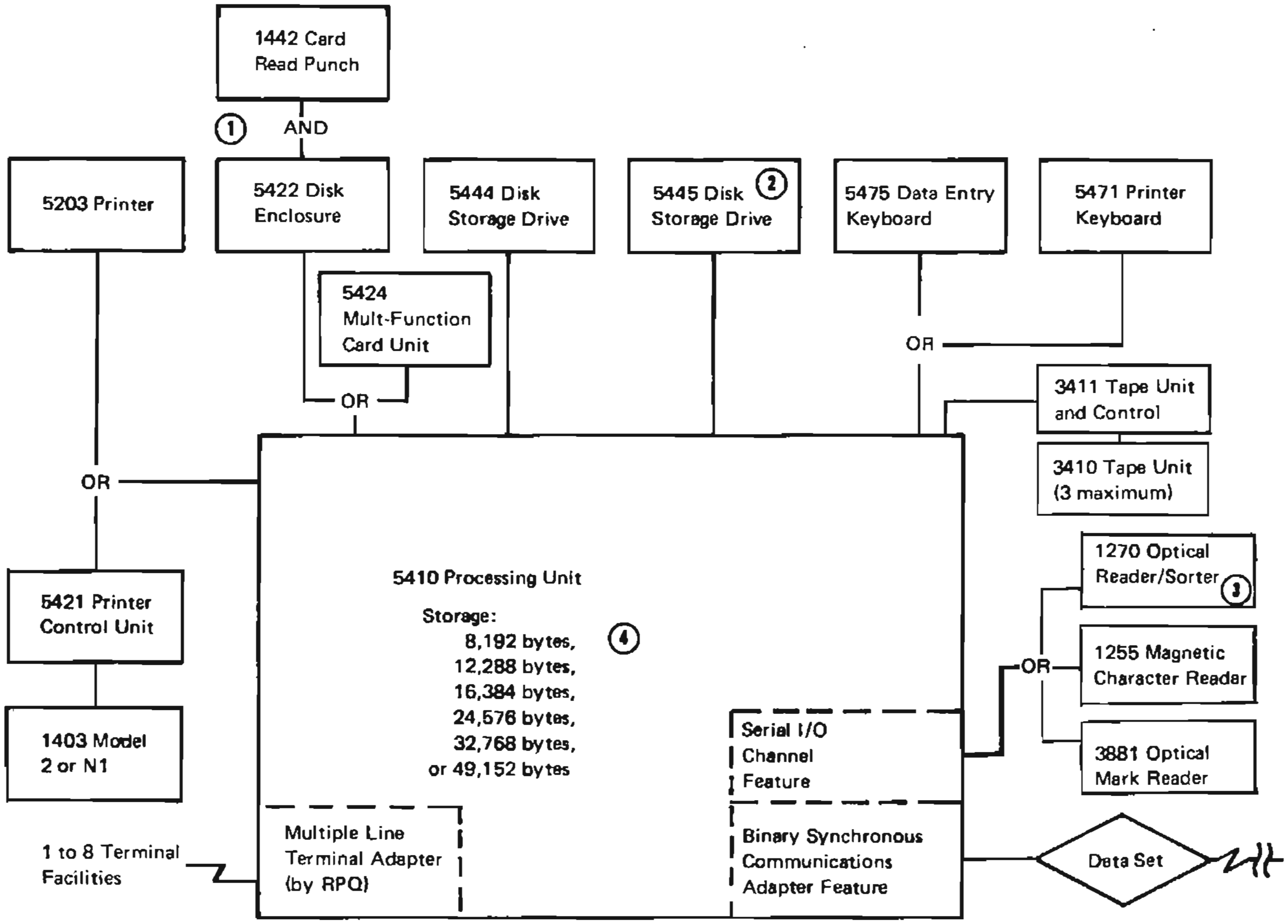


Figure 1-1. IBM System/3 Model 10 Typical Configuration



① If IBM Programming Support is used, configurations without the 5424 must include both a 1442 and a 5444, and *no* 5475.

② If IBM System/3 Disk System Programming Support is being used, at least one 5444 is a prerequisite.

③ Usually not used in the United States.

④ IBM Programming Systems for a disk-oriented system requires a minimum of 12,288 bytes of storage to ensure systems availability.

Figure 1-2. Devices Available for System/3 Model 10

The following units can be included as special features:

1. An IBM 5475 Data Entry Keyboard. This keyboard resembles that of the IBM 5496 Data Recorder. It serves as an input device to System/3. This unit excludes the printer-keyboard.
2. An IBM 5471 Printer-Keyboard. This device combines the facility for entering data from a keyboard with the capacity for printing data on an on-line printer. It can be used as an inquiry station in disk systems. This unit excludes the data entry keyboard.
3. An IBM 1255 Magnetic Character Reader. This unit provides the capability to read magnetic ink characters and sort paper documents. It is attached to the system through the serial I/O channel.
4. A Binary Synchronous Communications Adapter (BSCA) that enables the system to communicate with other systems or remote terminals over Communications facilities.
5. An IBM 1270 Optical Reader/Sorter, which enables the system to read ISOCRAF-A or ISOCRAF-B digits and four special characters from a single horizontal line on source documents, and to sort these documents into either six or 12 pockets, depending upon the 1270 model.

6. An IBM 5445 Disk Storage Drive. This unit reads data from and writes data onto the removable IBM 2316 Disk Pack. This disk pack also operates on the IBM 2314 as an I/O unit for the IBM System/360 and on the IBM 2319 as an I/O unit for the IBM System/360. If IBM System/3 programming conventions are followed on System/360 and System/370, data on a 2316 can be accessed by a System/3 Model 10, by a System/360, or by a System/370.
7. An IBM 3411 Tape Unit and Control, either alone or with one, two, or three IBM 3410 Tape Units. Tapes produced on the 3410 and 3411 Tape Units are interchangeable with tapes produced on other systems, such as System/360, if half-inch tape of the same density is used.
8. An IBM 1442 Card Read Punch. This unit allows System/3 Model 10 to read and punch 80-column cards. The 1442 is available as an RPQ for systems that use the 5424. For systems that use the 5422, the 1442 is available as a special feature.
9. An IBM 3881 Optical Mark Reader. This unit reads hand printed and machine printed marks from forms, converts the marks into EBCDIC or image format code, and sends the coded characters to the system.

For a more detailed configuration of the system, including the required and available special features, see *IBM System/3 Model 10 Configurator* (GA21-9135).

SYSTEM/3 DATA

96-column cards data can be entered into System/3 through the medium of punched cards. The card (Figure 1-3) provides 96 positions for recording data in three tiers of 32 columns each. Each of the four print lines provides 32 positions for printing. An IBM publication, *IBM 96-Column Card* (GA21-9125), provides general information about card fields, card layout, card terms, card storage, card handling, and special feature cards. It also provides information about specifications for card stock, card printing, punching registration, and how to use the 96-column card gauge.

Data Formats

Data is stored in System/3 in extended binary coded decimal interchange code (EBCDIC) using eight bits plus a parity bit for each byte.

Zoned Decimal Format

In zoned decimal format each byte of data is considered to be divided into two groups of four bits each. Bits 0-3 constitute the zone portion and bits 4-7 constitute the digit portion. Figure 1-4 shows the byte as interpreted for zoned decimal format. When data is handled in this format, the zone bits do not participate in any arithmetic operations. The zone bits of the low-order byte are used to indicate the sign of the field for arithmetic operations.

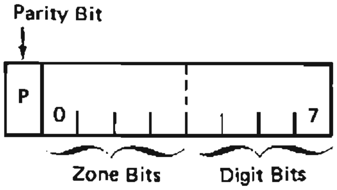


Figure 1-4. Zoned Decimal Format

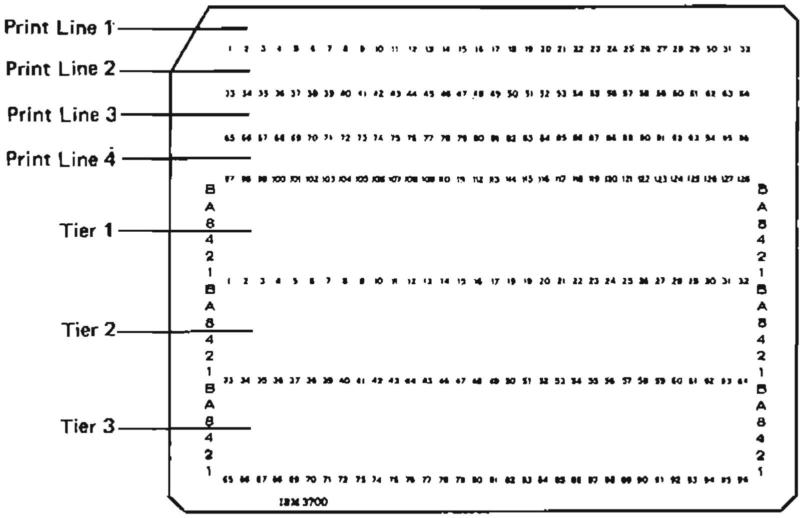


Figure 1-3. System/3 Card

Binary Format (Logical Data)

Data handled in binary format is treated as an eight-bit binary integer as shown in Figure 1-5. Note that all data in storage looks the same to the processing unit, eight binary bits. Instructions in the processing unit determine whether the data is treated as zoned decimal, graphic characters, or as binary integers.

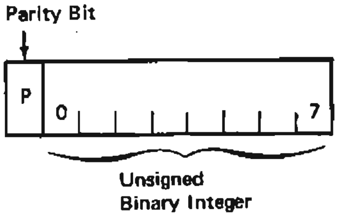


Figure 1-5. Binary Format

Parity

In addition to the eight data bits, each byte contains one other bit called a parity bit. This bit is used to maintain an odd number of bits in the byte. Any time a byte is used, it is checked to ensure that it contains an odd number of bits. If an even number of bits is detected, the processing unit stops with a process check.

Six-Bit Card Code

Data is stored in the System/3 card in six-bit form. Each of the ninety-six columns in the card can contain one six-bit character, which is converted during input operations to eight-bit extended binary coded decimal interchange code (EBCDIC) format. On output to the punch, EBCDIC is converted to card code. Figure 1-6 shows the card code representation of each character.

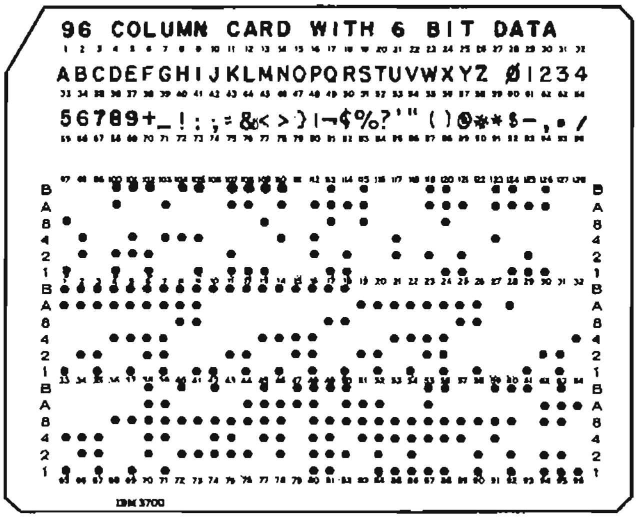

96 COLUMN CARD WITH 6 BIT DATA
ABCDEFGHIJKLMNOPQRSTUVWXYZ Ø1234
56789+_!:;=&<>)|¬¢%?'"()@**$-,./
B
A
8
4
2
1
IBM 5700

Figure 1-6. System/3 Card Code

Eight-Bit Program Card Code

Six-bit card code limits the number of characters (bit patterns) to sixty-four. The eight-bit bytes used internally in the processing unit allow a maximum of 256 different combinations. The instructions to the system require all of the 256 different combinations. The system provides a method of reading eight bits into storage while using six-bit card code.

Eight-bit code is read by using tier 3 of the card to provide two extra bits for each column in tiers 1 and 2. These bits are designated C and D. For tier 1 columns, the 4 bit of the corresponding tier 3 column serves as the C bit, and the 8 bit serves as the D bit. For tier 2 the 1 bit of the corresponding tier 3 column serves as the C bit, and the 2 bit serves as the D bit. For example, columns 1 and 33 use coluriin 65 for their C and D bits, columns 2 and 34 use column 66, etc. Figure 1-7 shows an example of program card code punching. The full card code including the characters that must be punched to obtain eight-bit code are shown in *Appendix B.*

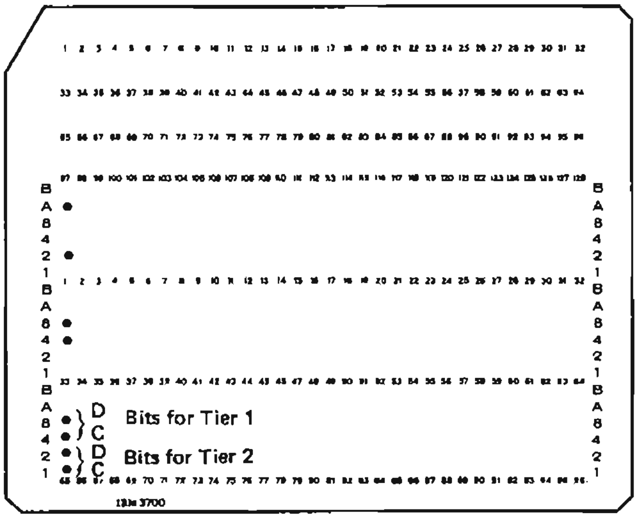


Figure 1-7. Program Card Code

ADDRESSING

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

An address used to refer to main storage can be specified by either of two methods: direct addressing or base-displacement addressing. The type of addressing to be used is specified by bits 0-3 of the first byte 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.

Direct Addressing

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

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

Base-Displacement Addressing

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 one-byte address in the instruction is added to the contents of a two-byte address in an index register. The index register to be used is determined by the bit of the bit group that is 1. If the low-order bit (bit 1 or bit 3) is 1, index register 1 is used. If the high-order bit of the bit group (bit 0 or bit 2) is 1, index register 2 is used. Both bit groups can 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

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

As Figure 1-8 shows, all instruction formats have two elements in common: the op code and the Q code. Each of these elements is one byte. The op code determines the type of addressing to be performed (and thereby the length of the instruction) and the operation to be performed. The function of the Q byte is determined by the instruction being performed and will be discussed with each individual instruction.

Command Instructions

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

1. Device address and function specification.
2. Jump condition.
3. Halt identifier (tens position).

The command instruction is distinguished by having bits 0-3 of the op code all ones.

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 ones. The two bits that are not both one (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 index register 1 (01) or index register 2 (10) is used. The Q byte of a one-address instruction can contain:

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

Command Instruction

Op Code 1111	Q Byte	Command

0 3
Bits

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 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 Addresses 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

Figure 1-8. Instruction Formats

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 both ones. If all four of bits 0-3 are zero, addressing is direct, and the instruction is six bytes long. If any *one* of bits 0-3 is one, 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 one, 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, index register 1 is used; if the bit group = 10, index register 2 is used. Both addresses can use the same index register during one instruction.

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

PROCESSING UNIT DATA FLOW

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

1. Op code register.
2. Q register.
3. Condition register.
4. One of the local storage registers (LSRs).
5. Out to an I/O unit.

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

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

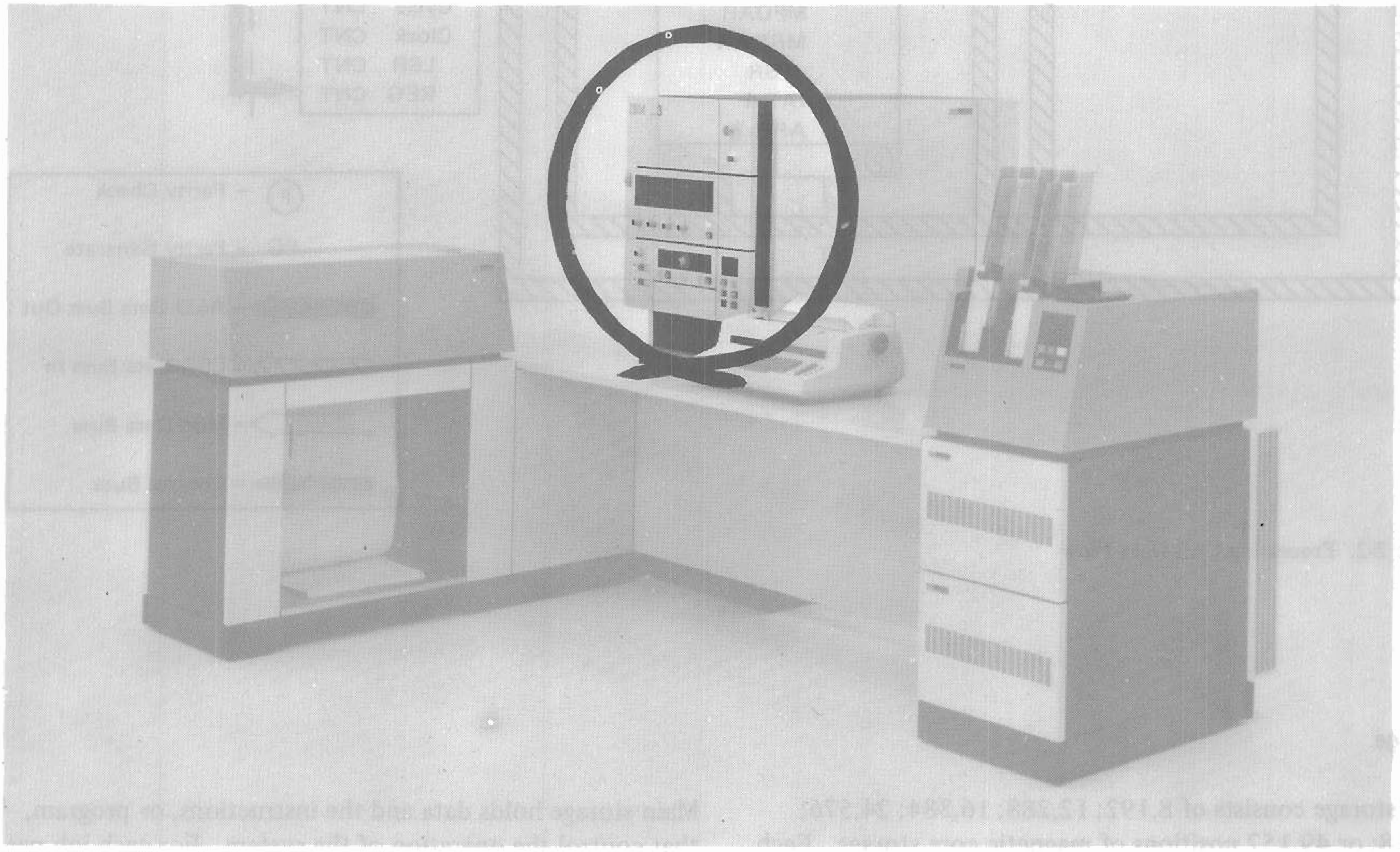


Figure 2-1. Processing Unit

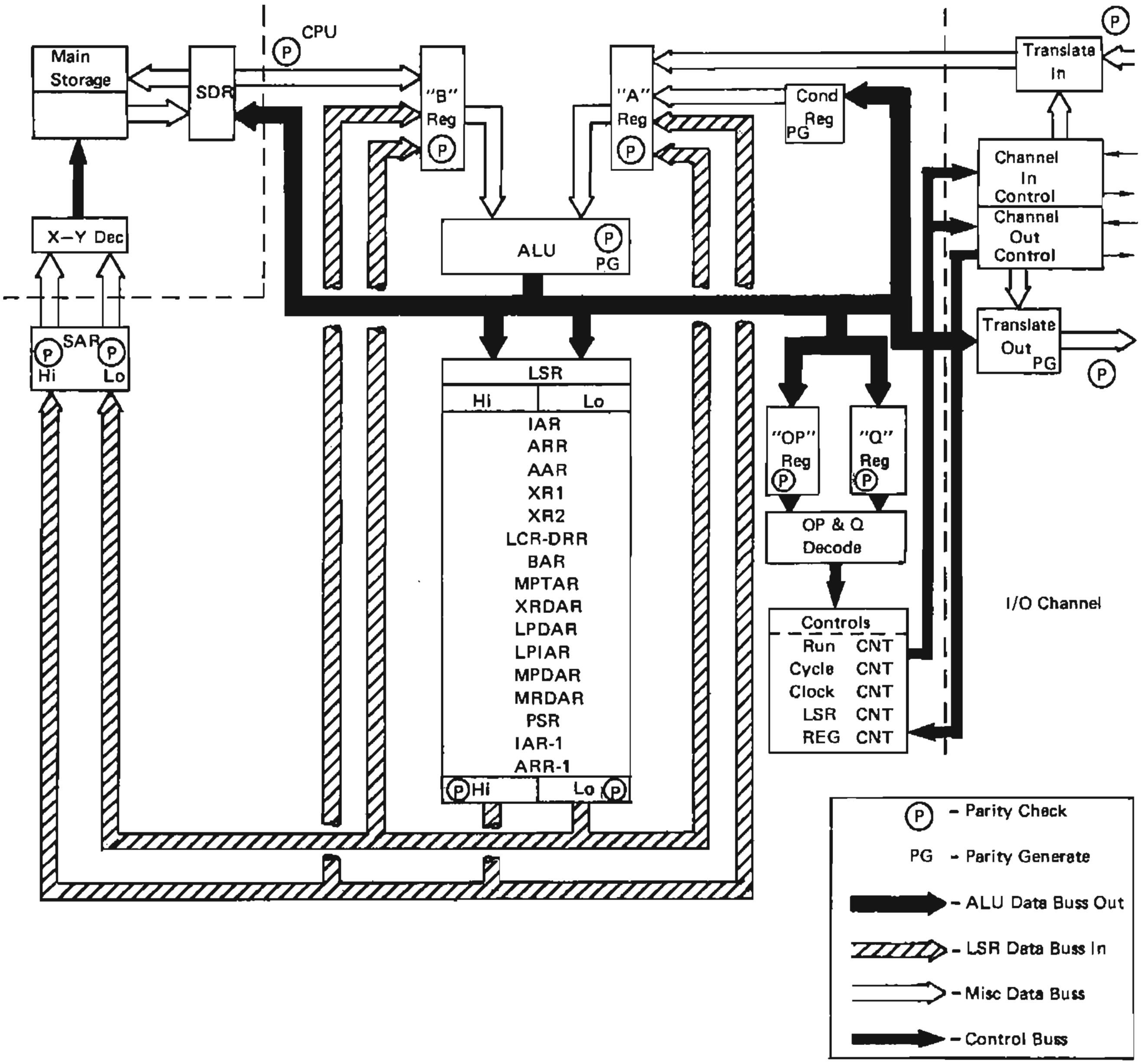


Figure 2-2. Processing Unit Data Flow

Storage

Main storage consists of 8,192; 12,288; 16,384; 24,576; 32,768; or 49,152 positions of magnetic core storage. Each position has its own distinct address (method of locating the position). Each position can store an eight-bit unit of information and a parity bit called a byte. Coded combinations of bits within a byte can represent alphabetic, numeric, or logical data.

Main storage holds data and the instructions, or program, that control the operation of the system. For each job performed by the system, certain portions of main storage are assigned to store instructions and certain portions to store data to be processed.

Storage Data Register

This register serves as a place to store data temporarily as it passes between the processing portions of the processing unit and main storage. Data can enter the storage data register from ALU or from main storage. Data can be sent from the storage data register to either the B register or to main storage. The storage data register in an 8,192 storage position CPU holds one byte of data. In every other CPU, the storage data register holds two bytes of data; however, only one of these bytes is addressed during a single CPU cycle.

A and B Registers

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

Arithmetic-and-Logical Unit

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

Storage Address Register

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

Condition Register

The condition register stores bits that indicate what happened as the result of processing various operations. For example, bits in the condition register can indicate a high, a low, or an equal condition after a compare operation; after an arithmetic operation, bits may indicate that a binary or decimal overflow has occurred. The program can test this register for these conditions.

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

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

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

Q Register

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

Op Code Register

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

Local Storage Registers

The local storage registers contain data and addresses required for the execution of instructions. Each of the local storage registers contains two bytes.

In the following text, these acronyms apply: P1 = program level 1, P2 = program level 2 (which applies only to systems with the dual programming feature), IAR = instruction address register, PSR = program status register, XR = index register, and DAR = data address register.

Pressing the system reset key resets P1 IAR, P1 PSR, and P2 PSR to zero. *Pressing the program load key* resets P1 IAR, P1 PSR, P2 PSR, and the I/O (MFCU or Disk) DAR that was used for the program load function, to zero. *All other IARs, XRs, and I/O LSRs* must be initialized by a load register instruction or a load I/O instruction prior to their use. Fetching the first instruction from storage sets the PSR to condition equal. After the execution of the first instruction, the P1 PSR resets to condition equal unless the instruction itself has caused some condition other than equal to exist, in which case the P1 PSR sets to the resulting condition.

Instruction Address Register (IAR)

This register contains the address of the instruction bytes as they are addressed. It is increased by one each time an instruction byte is taken from storage. The basic group of registers contains two instruction address registers: the instruction address register for program level 1 (the main program in the basic system, which does not have the dual programming feature installed) and the instruction address register for interrupt level 1 (the interrupt level for the data entry keyboard or the printer-keyboard). In feature 1 there are instruction address registers for: program level 2 (if dual programming feature is installed), interrupt level 0 (if dual programming feature is installed), interrupt level 2 (if BSCA is installed), and interrupt level 4 (if SIOC is installed). When the system reset or program load key is pressed, the IAR register is reset to zero.

Address Recall Register (ARR)

The address recall register is used by certain instructions to store a beginning address for execution of the instruction or the address of the next sequential instruction. As with the instruction address register there are address recall registers for program levels 1 and 2 and for interrupt levels 0, 1, and 4. The address recall register is affected only by branch, decimal, and insert-and-test characters instructions.

After a system reset or program load key is pressed, the values of the ARR and the IAR may have been exchanged, so the value in the ARR should not be used as though it remained constant throughout the system reset or program load operation.

Operand 2 Address Register (AAR)

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

Index Registers (XR1 and XR2)

Two index registers are provided in the base system. These 16-bit registers contain the base address for base-displacement addressing. A second pair of index registers is provided with the dual programming feature.

Program Status Register (PSR)

Separate program status registers are provided for the base system (program level 1 program status register) and for the second program level provided by the dual programming feature (program level 2 program status register). The high-order byte is used as a length count recall register during an interrupt. The low-order byte is used as a condition recall register during an interrupt. Loading this register automatically sets the condition register to the same value.

Operand 1 Address Register (BAR)

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

MFCU Print Data Address Register (MPTAR)

This register holds the address of the high-order (leftmost) byte of the MFCU print data field in CPU storage.

Line Printer Data Address Register (LPDAR)

This register holds the address of the high-order (leftmost) byte of the line printer (5203 or 1403) data field in CPU storage.

Line Printer Image Address Register (LPIAR)

This register holds the address of the high-order (leftmost) byte of the CPU storage field that stores an image of the character order for the line printer chain (or train).

MFCU Punch Data Address Register (MPDAR)

This register holds the address of the high-order (leftmost) byte of the MFCU punch data field in CPU storage.

Length Count/Data Recall Registers (LCR and DRR)

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

MFCU Read Address Register (MRDAR)

This register contains the address of the high-order (leftmost) byte of the MFCU read data field in CPU storage (the storage area into which data is entered by the MFCU reader).

Disk Control Address Register (DCAR)

This register is installed on systems equipped with an IBM 5444 Disk Storage Drive. It holds the address of the high-order (leftmost) byte of the 5444 disk control field in CPU storage.

Disk Drive Control (Address) Register (DDCR)

This register is installed on systems equipped with an IBM 5445 Disk Storage Drive. It holds the address of the high-order (leftmost) byte of the 5445 disk drive control field (DDCF) in CPU storage.

Disk Data Address Register (DDAR)

This register is installed on systems equipped with an IBM 5444 Disk Storage Drive. It holds the address of the high-order (leftmost) byte of the CPU storage area reserved for 5444 disk data; that is, this register indicates the CPU storage location assigned to receive data from or to supply data to 5444 disk storage.

Disk Drive Data (Address) Register (DDDR)

This register is installed on systems equipped with an IBM 5445 Disk Storage Drive. It holds the address of the high-order (leftmost) byte of the CPU storage area reserved for the 5445 disk data; that is, this register indicates which CPU storage area is to receive information from or to supply information to 5445 disk storage.

Serial I/O Channel Address Register (SIAR)

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

1442 Data Address Register

This register is installed on systems equipped with an IBM 1442 Card Read Punch. It holds the address of the high-order (leftmost) byte of the 1442 data area in storage.

CYCLES AND PHASES

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

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

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

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

Storage access cycles are designated by the phase in which they occur, and the type of operations performed in them is as follows:

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

TIME SHARING

System/3 operates in a mode known as time sharing. Time sharing is a mode of operation in which I/O operations are overlapped with processing operations so that processing operations can continue while I/O operations are in process. This is accomplished by allowing I/O units to steal processing unit cycles between processing cycles. The processing unit must check the I/O units to determine when an I/O operation is completed and the input data can be used or the output data can be replaced with the new data.

INTERRUPT

Certain I/O units require special subroutines to handle data entered by them within a limited period of time or for other similar reasons. To provide for these special subroutines, an interrupt system is installed. This interrupt system permits the processing unit to change state as a result of a condition external to the system. External conditions encountered in System/3 originate at an I/O device that has requested special attention by the processing unit. Generally, an interrupt implies that the processing unit must interrupt a current instruction sequence, perform an intervening instruction sequence requested by the interrupting I/O device, and return to the interrupted program.

It is apparent from the nature of an interrupt that a means of retaining the stopping point of an interrupted program and the starting point of an intervening program is an important characteristic. The system provides an instruction address register and an address recall register for each level of interrupt.

Interrupt Priorities

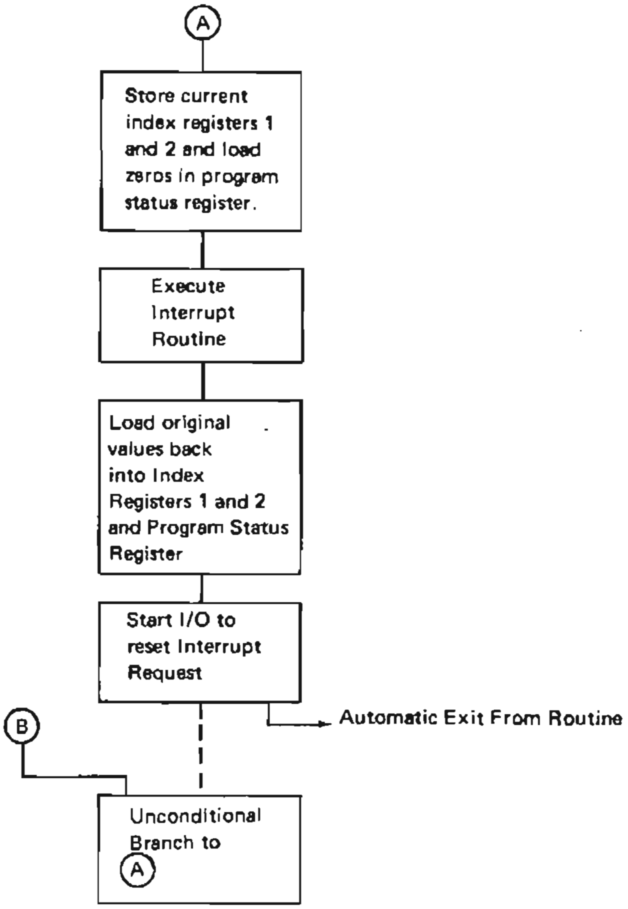
Five levels of interrupt have been implemented in System/3. I/O devices and their interrupt levels, in order of descending priorities, are:

Serial I/O Channel	Level 4
Unassigned	Level 3
BSCA	Level 2
Data Entry Keyboard or Printer Keyboard	Level 1
Dual Programming Control (Interrupt Key)	Level 0

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

Interrupt Operation

When an interrupt is acknowledged, at the completion of the instruction in process, the processing unit interrupts execution of further instructions based on the interrupted program's instruction address register and proceeds to execute those instructions designated by the interrupting level's instruction address register. The interrupted instruction address register and address recall register remain intact. The interrupting program is responsible for storing and restoring index registers 1 and 2 and the program status register for the interrupted program. The end of the interrupt routine is signaled by a start I/O instruction telling the interrupting device to reset its interrupt request. Figure 2-3 shows the recommended generalized interrupt routine.


A
Store current index registers 1 and 2 and load zeros in program status register.
Execute Interrupt Routine
Load original values back into Index Registers 1 and 2 and Program Status Register
Start I/O to reset Interrupt Request
Automatic Exit From Routine
B
Unconditional Branch to A

Note: The interrupt instruction address register must be set to the address of A or B before the first interrupt occurs. The normal operation of the processing unit will leave the interrupt instruction address register at the address of B at the end of the interrupt routine.

Figure 2-3. Interrupt Routine

INPUT/OUTPUT FACILITIES

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

The processing unit communicates with the I/O devices via an interface called the Input/Output Channel. The I/O channel consists of:

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

Channel Organization

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

Device Control

The following instructions control I/O devices:

1. Start I/O.
2. Load I/O.
3. Sense I/O.
4. Test I/O and Branch.
5. Advance Program Level.

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

The interrupt mode was discussed earlier in this chapter. The second mode of operation is the cycle steal mode. In this mode of operation the I/O device is started by a start I/O instruction, then is left to perform its operations until storage is required by those operations. When storage is needed, the I/O device is allowed to steal one or more cycles from other processing unit operations in order to store or retrieve from storage the necessary data. The processing unit then continues to perform other operations until the I/O unit requires storage cycles again.

Channel Limitations

In certain system configurations, overlapped I/O operations may cause I/O devices to experience data overrun conditions. Data overrun occurs when requests for I/O cycle steals are not granted in the time limit required for a device. The result of the overrun may cause loss of data and on the 5424 and 5203 this condition is not detected. Therefore, when programming I/O device operations, only those devices should be overlapped which do not cause overrun conditions. The following chart gives those allowable configurations:

Devices	Groups of Devices that Avoid Data Overrun – Read Down					
5444	X	X			X	
5445	X		X	X		
5203*		X	X			X
1255, 1270, 3881**	X	X	X	X	X	X
SIOC (50 KB)		X		X	X	X
3410/3411				X	X	X
BSCA	X	X	X	X	X	X
5424	X	X	X	X	X	X
MLTA	X	X	X	X	X	X
1442	X	X	X	X	X	X
1403*	X	X	X	X	X	X
5471, 5475***	X	X	X	X	X	X

*5203, 1403	Mutually exclusive devices
**1255, 1270, 3881	Are SIOC attached devices. They are mutually exclusive and each excludes all other devices on the SIOC.
***5471, 5475	Mutually exclusive devices

IBM Required Special Equipment Engineering can determine whether configurations involving high data rate devices, such as the RPQ items installed, will result in data overrun if IBM program products are not being used. Contact your IBM Sales Representative for this information.

DUAL PROGRAMMING FEATURE

The dual programming feature provides the system with the capability to execute two independent programs on a time-sharing basis. This feature allows the processing unit to transfer to a different program when the current program must wait for completion of an I/O operation. This uses the high performance capabilities of the processing unit rather than forcing it to wait for completion of the execution by active I/O devices.

The dual programming feature is program supported only on disk systems with 12,288 or more bytes of main storage. The feature allows two independent object programs to reside in storage simultaneously.

The transfer from one program level to the other is called program level advance. Program level advance can be either automatic or program controlled. Unlike interrupt, program level advance does not require that index registers 1 and 2 and the program status register be stored, because separate index registers, instruction registers, address recall registers, and program status registers are provided for each program level.

An automatic program level advance occurs when:

1. Operation on one program level is instructed to halt.
2. An I/O device is instructed to operate when the device requires operator attention.
3. An I/O device is instructed to operate when the device is already performing an operation.

A program controlled program level advance is accomplished by issuing an instruction.

Program Note: After a system reset, a program level advance from program level 1 to program level 2 will initialize the condition register to the high condition.

Because one program can finish operating before the other, and thus require a new program to be entered while one of the old programs is running, it is the responsibility of the supervising program to ensure that the two programs do not use the same I/O devices or overlapping storage areas.

INSTRUCTION AND PROGRAMMING CONSIDERATIONS

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

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

ARITHMETIC INSTRUCTIONS

Zero and Add Zoned

Mnemonic: ZAZ

Op Code	Q Byte	Operand Address (2 to 4 Bytes)			
X4	L1 ⁝ L2				

Operation: The second operand is placed byte by byte in the first operand. Extra high-order zeros are inserted into those positions by which the first operand exceeds the second in length. The zone bits in each byte of the result except the rightmost are set to all ones. The zone bits of the rightmost byte of the first operand are set to all ones if the result is positive or zero. If the result is negative, the zone bits of the rightmost byte are set to 1101. The operands are addressed by their rightmost bytes. Zero results are positive.

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

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

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

Resulting Condition Register Settings:

Equal	Result is zero.
Low	Result is negative.
High	Result is positive.
Decimal Overflow	Not affected.
Test False	Not affected.
Binary Overflow	Not affected.

Instruction Timing: Time in microseconds = 1.52 (N + L1 + L2) + 1.52 (R).

Example:

Instruction

04	22	00	10	00	20

Operand 1 Before Operation

F7	F6	F3	F6	F9	
000C	000D	000E	000F	0010	(Addresses)

Operand 2

F4	F2	F5	
001E	001F	0020	(Addresses)

Operand 1 After Operation

F0	F0	F4	F2	F5	
000C	000D	000E	000F	0010	(Addresses)

Condition Register Before Operation

00000000

0 Bits 7

Condition Register After Operation

00000100

0 Bits 7

Next Instruction Address: Next Sequential Instruction

Add Zoned Decimal

Mnemonic: AZ

Op Code	Q Byte		Operand Addresses	(2 to 4 Bytes)		
X6	L1	L2				

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

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

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

The second operand remains unchanged unless the fields overlap.

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

Resulting Condition Register Settings:

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

Program Note: The decimal overflow condition code is reset only by machine reset or by testing decimal overflow with a branch-on-condition or jump-on-condition instruction. Data stored in the address recall register is altered by this instruction.

Instruction Timing: Time in microseconds = 1.52 (N + L1 + L2) + 1.52 (R).

Example:

Instruction

06	22	00	10	00	20

Operand 1 Before Operation

F7	F6	F3	F6	F9	
000C	000D	000E	000F	0010	(Addresses)

Operand 2

F4	F2	F5	
001E	001F	0020	(Addresses)

Operand 1 After Operation

F7	F6	F7	F9	F4
000C	000D	000E	000F	0010

Condition Register Before Operation

00000000
0 7
Bits

Condition Register After Operation

00000100
0 7
Bits

Next Instruction Address: Next Sequential Instruction

Subtract Zoned Decimal

Mnemonic: SZ

Op Code	Q Byte		Operand Addresses (2 to 4 Bytes)			
X7	L1	L2				

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

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

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

The second operand remains unchanged unless the fields overlap.

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

Resulting Condition Register Settings:

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

Program Note: The decimal overflow condition code is reset only by machine reset or by testing decimal overflow with a branch-on-condition or jump-on-condition instruction. Data stored in the address recall register is altered by this instruction.

Instruction Timing: Time in microseconds = 1.52 (N + L1 + L2) + 1.52 (R).

Example:

Instruction

07	22	00	10	00	20

Operand 1 Before Operation

F7	F6	F3	F6	F9	
000C	000D	000E	000F	0010	(Addresses)

Operand 2

F4	F2	F5	
001E	001F	0020	(Addresses)

Operand 1 After Operation

F7	F5	F9	F4	F4
000C	000D	000E	000F	0010

Condition Register Before Operation

00000000
0 Bits 7

Condition Register After Operation

00000100
0 Bits 7

Next Instruction Address: Next Sequential Instruction

Add Logical Characters

Mnemonic: ALC

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

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

The Q byte specifies the length of the operands. L is 1 less than the length in bytes of either operand. Both operands must be the same length. The maximum length of the operands is 256 (1 + hex FF) bytes.

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

Operand 2 is not changed unless it overlaps operand 1.

Resulting Condition Register Settings:

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

Program Note: Binary overflow bit is reset during the instruction phase of this operation.

Instruction Timing: Time in microseconds = 1.52 (N + 2L).

Example:

Instruction　　　　Index Register 1 = 0CC0

5E	03	00	10	NSI	

Operand 1 Before Operation

00110101	11001011	11101101	01100100	
0CBD	0CBE	0CBF	0CC0	(Addresses)

Operand 2

01011011	01010101	01111000	11001101	
0CCD	0CCE	0CCF	0CD0	(Addresses)

Operand 1 After Operation

10010001	00100001	01100110	00110001
0CBD	0CBE	0CBF	0CC0

Condition Register Before Operation

00000000
Bits

Condition Register After Operation

00000010
Bits

Next Instruction Address: Next Sequential Instruction

Subtract Logical Characters

Mnemonic: SLC

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

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

First operand	0110 1101
Second operand	0111 1110
Result	1110 1111

The Q byte specifies the length in bytes of the operands. L is one less than the length of either operand. Both operands must be the same length. The maximum length of the operands is 256 bytes (1 + hex FF).

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

The second operand is not changed unless the operands overlap.

Resulting Condition Register Settings:

Equal	Result is zero.
Low	First operand is smaller than second operand.
High	First operand is greater than second operand.
Decimal Overflow	Not affected.
Test False	Not affected.
Binary Overflow	Not affected.

Instruction Timing: Time in microseconds = 1.52 (N + 2L).

Example:

Instruction Index Register 2 = 0CC0

AF	03	00	10		

Operand 1 Before Operation

10010110	01011010	01110111	10111111	
0CBD	0CBE	0CBF	0CC0	(Addresses)

Operand 2

011101 00	10000110	01100010	10100100	
0CCD	0CCE	0CCF	0CD0	(Addresses)

Operand 1 After Operation

00100001	11010100	00010101	00011011
0CBD	0CBE	0CBF	0CC0

Condition Register Before Operation

00000000

Bits

Condition Register After Operation

0000100

Bits

Next Instruction Address: Next Sequential Instruction

Add to Register

Mnemonic: A

Op Code	Q Byte	Operand Address
Y6	Q	

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

The Q code selects the register to be modified. The high-order bit (bit 0) of the Q code determines which of two groups of registers will be modified. The remaining bits of the Q code determine which specific register within the group will be modified.

If bit 0 of the Q code is 0, the remaining bits cause modification of the registers as follows:

Bit	*Register*
1	Program level 2 instruction address register.
2	Program level 1 instruction address register.
3	Instruction address register in use when the add-to-register instruction is executed.
4	Address recall register.
5	Program status register.
6	Index register 2.
7	Index register 1.

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

Bit	*Interrupt Level*
None	Interrupt level 0.
1	Interrupt level 1.
2	Interrupt level 2.
3	Interrupt level 3.
4	Interrupt level 4.

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

Resulting Condition Register Settings:

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

Program Note: Even though this instruction can modify the program status register, the contents of the condition register will be placed in the low-order byte of the program status register during I-phase of the next instruction.

The binary overflow bit in the condition register is turned off during I-phase of this instruction.

Instruction Timing: Time in microseconds = 1.52 (N + 2).

Example:

Instruction

36	00000010	00	04

Operand 1

01001000	00100000
0003	0004

Index Register 2

00110101	01101010

Before Operation

01111101	10001010

After Operation

Condition Register After Operation

00000100

DATA HANDLING INSTRUCTIONS

Move Hex Character

Mnemonic: MVX

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

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

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

Hex Value of Q Code	*Operand 2*		*Operand 1*
00	Zone	to	Zone
01	Numeric	to	Zone
02	Zone	to	Numeric
03	Numeric	to	Numeric

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

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

Instruction Timing: Time in microseconds = 1.52 (N + 2).

Example:

Instruction

98	01	A0	65

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

Operand 1 Before Operation

2F

1FC0

Operand 2

4C

2B7A

Operand 1 After Operation

CF

1FC0

Move Characters

Mnemonic: MVC

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

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

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

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

Instruction Timing: Time in microseconds = 1.52 (N + 2L).

Example:

Instruction

0C	05	1A	06	2B	5A

Operand 1 Before Operation

	D1	C1	D4	C5	E2	40
Addresses	1A01	1A02	1A03	1A04	1A05	1A06

Operand 2

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

Operand 1 After Operation

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

Edit

Mnemonic: ED

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

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

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

Resulting Condition Register Settings:

Equal	Second operand is zero.
Low	Second operand is negative.
High	Second operand is positive.
Decimal Overflow	Not affected.
Test False	Not affected.
Binary Overflow	Not affected.

Instruction Timing: Time in microseconds = 1.52 (N + L1 + L2).

Example:

Instruction

0A	0A	00	BF	00	07

Operand 1 Before Operation

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

Operand 2

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

Note: "R" represents "–9"

Operand 1 After Operation

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

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

Condition Code

00000010

0 7

Bits

Insert and Test Characters

Mnemonic: ITC

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

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

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

At the end of this operation, the address of the first significant digit is placed in the address recall register. If no significant digit is found, the address of the byte to the right of the first operand is placed in the address recall register. The address recall register will be changed again only by a decimal, branch, insert and test characters, or test I/O instruction.

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

Instruction Timing: Maximum time in microseconds = 1.52 (N + 1 + L1).

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

Operand 2

*
0010

Operand 1 After Operation

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

Note that address 00B6 was not included in the first operand.

Move Logical Immediate

Mnemonic: MVI

Op Code	Q Byte	Operand Address
YC	10	

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

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

Instruction Timing: Time in microseconds = 1.52 (N +1).

Example:

Instruction

3C	AF	2F	CB

Operand Before Operation

00

2FCB

Operand After Operation

AF

2FCB

Set Bits On Masked

Mnemonic: SBN

Op Code	Q Byte	Operand Address	
YA	M		

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

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

Instruction Timing: Time in microseconds = 1.52 (N + 1).

Example:

Instruction

3A	01011010	00	20

Operand Before Operation

00001100

Operand After Operation

01011110

Set Bits Off Masked

Mnemonic: SBF

Op Code	Q Byte	Operand Address	
YB	M		

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

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

Instruction Timing: Time in microseconds = 1.52 (N + 1).

Example:

Instruction

38	10000001	00	30

Operand Before Operation

01111001
0030

Operand After Operation

01111000
0030

Store Register

Mnemonic: ST

Op Code	Q Byte	Operand Address	
Y4	Q		

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

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

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

Bit	*Register*
1	Program level 2 instruction address register.
2	Program level 1 instruction address register.
3	Instruction address register in use when the store register instruction is executed.
4	Address recall register.
5	Program status register. The high-order byte of this register is the length count recall register and has no program significance. The low-order byte is the image of the condition register.
6	Index register 2.
7	Index register 1.

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

Bit	*Interrupt Level*
None	Interrupt level 0.
1	Interrupt level 1.
2	Interrupt level 2.
3	Interrupt level 3.
4	Interrupt level 4.

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

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

Instruction Timing: Time in microseconds = 1.52 (N + 2).

Example:

Instruction

34	00001000	B2	BB

Address Recall Register

0A	CD

Operand Before Operation

2F	C2
B2BA	B2BB

Operand After Operation

0A	CD
B2BA	B2BB

Load Register

Mnemonic: L

Op Code	Q Byte	Operand Address
Y5	Q	

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

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

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

Bit	*Register*
1	Program level 2 instruction address register.
2	Program level 1 instruction address register.
3	Instruction address register in use when the load register instruction is executed.
4	Address recall register.
5	Program status register. The high-order byte of this register is the length count recall register and has no program significance. The low-order byte of this register holds a condition code and is loaded under special conditions described in the programming notes for this instruction.
6	Index register 2.
7	Index register 1.

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

Bit	*Interrupt Level*
None	Interrupt level 0.
1	Interrupt level 1.
2	Interrupt level 2.
3	Interrupt level 3.
4	Interrupt level 4.

Program Notes:

1. This instruction must not be used to load more than one register at a time. The attempt to load more than one register can result in incorrect register contents.
2. When the program status register is selected, the contents of the low-order byte of the operand have the following significance:

Bit 7 = 1:	Set equal condition.
Bit 6 = 1:	Set low condition if bit 7 = 0.
Bit 6 = 0:	Set high condition if bit 7 = 0.
Bit 4 = 1:	Set decimal overflow condition.
Bit 3 = 1:	Set test false condition.
Bit 2 = 1:	Set binary overflow condition.

When bit 7 of the operand = 0, bit 5 of the low-order byte of the program status register is set to 1 when bit 6 of the operand = 0 and set to 0 when bit 6 of the operand = 1. Bits 0, 1, and 5 of the operand are ignored. The condition register is set at the same time as the program status register under these same controls.

3. If program level 1 has been halted and this instruction is used by an interrupt routine to load program level 1 instruction address register, program level 1 will be reset from the halt state and will proceed after all interrupts and I/O cycle steals have been serviced. The program level 1 halt indicators will be turned off. If the dual-programming feature is installed and this instruction is used in either program level or in an interrupt routine to load the instruction address register for a halted program level, that program level will be reset from the halt state and will proceed according to normal priority. The halt indicators for that program level will be turned off.

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

Instruction Timing: Time in microseconds = 1.52 (N + 2).

Example:

Instruction

35	00000100	00	11

Operand

00000000	00000000
0010	0011

Program Status Register Before Operation

00001100	00110001

Program Status Register After Operation

00000000	00000100

Condition Register After Operation

00000100

Load Address

Mnemonic: LA

Op Code	Q Byte	Operand	
Z2	Q		

Operation: If the instruction is in the 4-byte format (op code C2), the 2-byte operand is taken from the instruction stream and loaded into the register specified by the Q byte.

If the instruction is in the 3-byte format (op code D2 or E2), the 1-byte operand is taken from the instruction stream and added to the contents of the index register specified by the op code. The result of this addition is loaded into the register specified by the Q byte.

Only index registers can be loaded with this instruction. Bits 6 and 7 of the Q code specify which index register to load as follows:

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

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

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

Instruction Timing: Time in microseconds = 1.52 (N).

Example:

Instruction

D2	02	05

Index Register 1

BA	15

Index Register 2 After Operation

BA	1A

LOGICAL INSTRUCTIONS

Compare Logical Characters

Mnemonic: CLC

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

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

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

Resulting Condition Register Settings:

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

Instruction Timing: Time in microseconds = 1.52 (N + 2L).

Example:

Instruction

0D	02	00	12	00	02

Operand 1

27	FA	26

Operand 2

23	FA	26

Condition Register

00000100

Compare Logical Immediate

Mnemonic: CLI

Op Code	Q Byte	Operand Address
YD	10	

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

Resulting Condition Register Settings:

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

Instruction Timing: Time in microseconds = 1.52 (N + 1).

Example:

Instruction

3D	7F	00	21

Storage Operand

7F

0021

Condition Register After Operation

00000001

Test Bits On Masked

Mnemonic: TBN

Op Code	Q Byte	Operand Address
Y8	Mask	

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

Resulting Condition Register Settings:

Equal	Not affected.
Low	Not affected.
High	Not affected.
Decimal Overflow	Not affected.
Test False	Turned on if any of the designated bits in the storage operand is not = 1.
Binary Overflow	Not affected.

Program Notes:

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

Instruction Timing: Time in microseconds = 1.52 (N + 1).

Example:

Instruction

38	00010110	00	21

Storage Operand

10010101

0021

Condition Register After Operation

00010000

Test Bits Off Masked

Mnemonic: TBF

Op Code	Q Byte	Operand Address	
Y9	Mask		

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

Resulting Condition Register Settings:

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

Program Notes:

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

Instruction Timing: Time in microseconds = 1.52 (N + 1).

Example:

Instruction

39	01101100	00	25

Storage Operand

10010100

0025

Condition Register After Operation

00010000

Branch On Condition

Mnemonic: BC

Op Code	Q Byte	Branch Address
ZO	Q	

Operation: The condition register is tested for the condition or conditions specified by the Q code. Bit 0 of the Q code specifies whether the branch is to be performed on condition true (1) or condition false (0). Bit 1 is not used.

If bit 0 of the Q code is a one, and at least one of the conditions specified by bits 2 through 7 is present, the address of the next sequential instruction (IAR) is placed in the address recall register. The branch address is placed in the IAR and therefore becomes the address of the next instruction.

The address recall register will be changed by the next decimal, insert and test character, branch, or test I/O instruction

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

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

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

Resulting Condition Register Settings:

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

Program Notes:

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

Instruction Timing: Time in microseconds = 1.52 (N).

Example:

Instruction

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

Condition Code Before Operation

00011001

Instruction Address Register After Operation

02	BF

Address Recall Register After Operation

0B	D0

Condition Register After Operation

00010001

Jump On Condition

Mnemonic: JC

Op Code	Q Byte	Control Code
F2	Q	

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

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

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

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

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

Resulting Condition Register Settings:

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

Program Notes:

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

Instruction Timing: Time in microseconds = 4.56.

Example:

Instruction

F2	00110000	0F
0BBD	0BBE	0BBF

Condition Register Before Operation

00001001

Instruction Address Register After Operation

0B	CF

Condition Register After Operation

00001001

HALT INSTRUCTIONS

Halt Program Level

Mnemonic: HPL

Op Code	Halt Identifier	
F0	Tens Code	Units Code

Operation: This instruction prevents the execution of the next sequential instruction and displays a halt identifier on a message display unit on the system control panel. The message display unit consists of fourteen indicators arranged as shown in Figure 2-4. These indicators are individually controlled by the bits in the halt-identifier bytes. The bits control the indicators as follows:

Bit	*Indicator Lighted*
0	Reserved
1	1
2	2
3	3
4	4
5	5
6	6
7	7

The hex digits required in a byte to produce the common characters used as halt identifiers are shown in Figure 2-5. The display unit is turned off (reset to blank) when the halt operation is terminated.

In systems without the dual programming feature the halt operation performs as a continuous branch to the beginning of the halt operation until the system start key is operated. Pressing the start key allows execution of the next sequential instruction and turns off the display unit.

In systems with the dual programming feature this operation results in an automatic program level advance. The re-entry point for the program containing the halt instruction is the address of the halt instruction. The halted program can be continued by pressing the halt reset key for that program level. This will also reset the display unit for that program level.

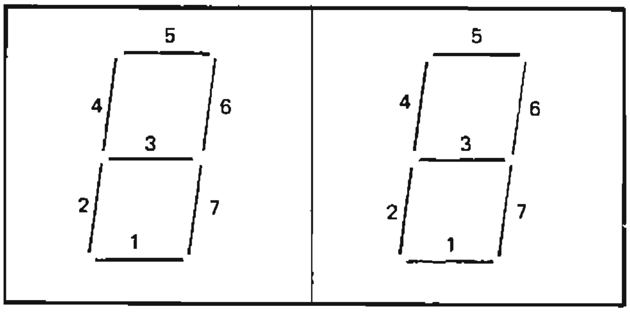


Figure 2-4. Message Indicator Light Arrangement

CHAR-ACTER	HEX CODE	DISPLAY SEEN	CHAR-ACTER	HEX CODE	DISPLAY SEEN
None	00		A	3F	A
1	03	1	b	79	b
2	76	2	C	6C	C
3	57	3	d	73	d
4	1B	4	E	7C	E
5	5D	5	F	3C	F
6	7D	6	H	3B	H
7	07	7	J	63	J
8	7F	8	L	68	L
9	5F	9	P	3E	P
0	6F	0	U	6B	U

Figure 2-5. Coding for Halt Identifier Characters

Example:

```
2 = Bits  - 0 1 2 3 4 5 6 7
    State - X 1 1 1 0 1 1 0

    Hex Codes   7       6
```

Program Notes:

1. The halt program level instruction performs as a no-op when it is used in an interrupt level program sequence.
2. Program level 1 or program level 2 can be stopped with a halt program level instruction to wait for an interrupt request. The interrupt routine can modify an appropriate program level instruction address register with a load register instruction to return to the halted program level at an instruction other than the halt instruction. The halted program level resumes operation and the display unit is turned off immediately after such a load register instruction is executed and the interrupt is reset. The program level resumes operation according to normal priority.

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

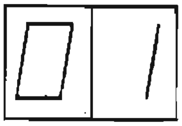
Instruction Timing: Time in microseconds = 4.56.

Example:

Instruction

F0	6F	03

Display Unit



INPUT/OUTPUT INSTRUCTIONS

Five specific instructions are provided for I/O operations. The op code for these instructions is the same for all I/O devices. Bits 0 through 4 of the Q byte of these instructions provide a device address (DA) and modifier (M) bits. Bits 5 through 7 of the Q byte provide an N code. These instructions and the specific bits that should occupy each of these Q byte bit positions are discussed under the specific I/O unit or operation that uses them.

Start I/O

Mnemonic: SIO

Op Code	Q Byte			Control Code
F3	DA	M	N	

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

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

Program Notes:

1. The start I/O instruction is no-oped if a unit check condition that prevents the execution of the start I/O instruction exists in the addressed device.
2. A start I/O instruction that specifies the reset of an interrupt condition is executed regardless of any unit check condition in the addressed device.
3. Any unit check condition that does not prevent the execution of a start I/O instruction is reset by the start I/O instruction, and the instruction is executed.
4. In systems with the dual programming feature, a start I/O instruction addressed to a device that is busy or not ready results in a program level advance. In systems without the dual programming feature a similar start I/O instruction results in a loop on that instruction until the device is ready or not busy.

Instruction Timing: Time in microseconds = 4.56.

Sense I/O

Mnemonic: SNS

Op Code	Q Byte			Operand Address
Y0	DA	M	N	

Operation: The contents of a data source specified by the N code portion of the Q byte are placed in the two-byte field specified by the operand address. A Q byte of 00 specifies that the data source is to be the address/data switches on the system control panel. Specifications for other data sources are discussed with the appropriate I/O device sense I/O instruction. The operand is addressed by its rightmost byte.

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

Instruction Timing: Time in microseconds = 1.52 (N + 2).

Load I/O

Mnemonic: LIO

Op Code	Q Byte			Operand Address
Y1	DA	M	N	

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

Program Note: In systems with the dual programming feature installed, a load I/O instruction to a busy device results in a program level advance. In systems without the dual programming feature, a load I/O instruction to a busy device causes the program to loop at the load I/O instruction until the device becomes not busy.

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

Instruction Timing: Time in microseconds = 1.52 (N + 2).

Test I/O and Branch

Mnemonic: TIO

Op Code	Q Byte	Branch-to-Address	
Z1	DA M N		

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

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

Instruction Timing: Time in microseconds = 1.52 (N).

Advance Program Level

Mnemonic: APL

Op Code	Q Byte	
F1	DA M N	Not Used

Operation: In systems with the dual programming feature installed the program level advances if the conditions specified by the N code of the Q byte exist at the addressed device. The re-entry point of the discontinued program level is the starting address of the advance program level instruction unless the program level advance is unconditional. The re-entry point for unconditional program level advance instructions is the next sequential instruction. If the specified condition does not exist, the operation is no-oped and no program level advance occurs. An unconditional program level advance occurs if the Q byte is 00.

If the dual programming feature is not installed, this operation causes the program to loop on the advance program level instruction until the specified condition no longer exists at the device. The program then proceeds with the next sequential instruction. An unconditional program level advance becomes a no-op in systems that do not have the dual programming feature installed.

Program Note: The use of an N field specifying advance on unit check will result in a discontinuation of the program level when a unit check exists. If the dual programming feature is not installed, the program goes into a one instruction loop.

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

Instruction Timing: Time in microseconds = 4.56.

DUAL PROGRAMMING INSTRUCTIONS

The following instructions must be incorporated in the loader/supervisor program for dual programming control.

Start I/O

Mnemonic: SIO

Op Code	Q Byte	Control Code
F3	00	

Operation: This instruction controls the dual programming mode of operation and the dual programming interrupt level. The control code specifies the operation to be performed as follows:

Bit	*Operation*
0	Reserved.
1	Reserved.
2	Reserved.
3	Reserved.
4	Reserved.
5	Enable dual programming mode when bit is 1; disable dual programming mode when bit is 0.
6	Enable interrupt level 0 (system control panel interrupt key) when bit is 1; disable interrupt level 0 when bit is 0.
7	Reset interrupt request 0.

The start I/O instruction to enable or disable dual programming mode provides programmed control over the system's ability to execute a program level advance. Enabling the dual programming mode allows both the automatic and the programmed advance of the program levels to occur. Disabling dual programming mode inhibits all program level advances and transforms them into wait operations. This instruction can be issued in either program level or in any interrupt level and will enable or disable all program level advances until another enable or disable instruction is given.

Program Notes:

1. Program level advances are not executable in an interrupt level. Unconditional program level advances result in no-op operations, and conditional program level advances result in wait operations.
2. To enable interrupt level 0, bits 5 and 6 of the control code must both be present. Interrupt level 0 cannot be enabled unless dual programming mode is enabled.

Instruction Timing: Time in microseconds = 4.56.

Test I/O and Branch

Mnemonic: TIO

Op Code	Q Byte	Branch-to Address
Z1	0000 M N	

Operation: This instruction tests the setting of the dual programming control switch on the system control panel. The N code specifies the condition to be tested for as follows:

Bits 6 7	*Condition*
0 0	Cancel program level.
0 1	Load program level from MFCU.
1 1	Reserved.
1 0	Load program level from printer keyboard.

Bit 5 defines the program level to be operated on:

Bit 5 = 0: Program level 1.
Bit 5 = 1: Program level 2.

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

Instruction Timing: Time in microseconds = 1.52 (N).

The system control panel (Figure 3-1) contains the switches and lights required to operate and control the system. System controls are divided into three sections: operator controls, customer engineering (CE) controls, and console display.

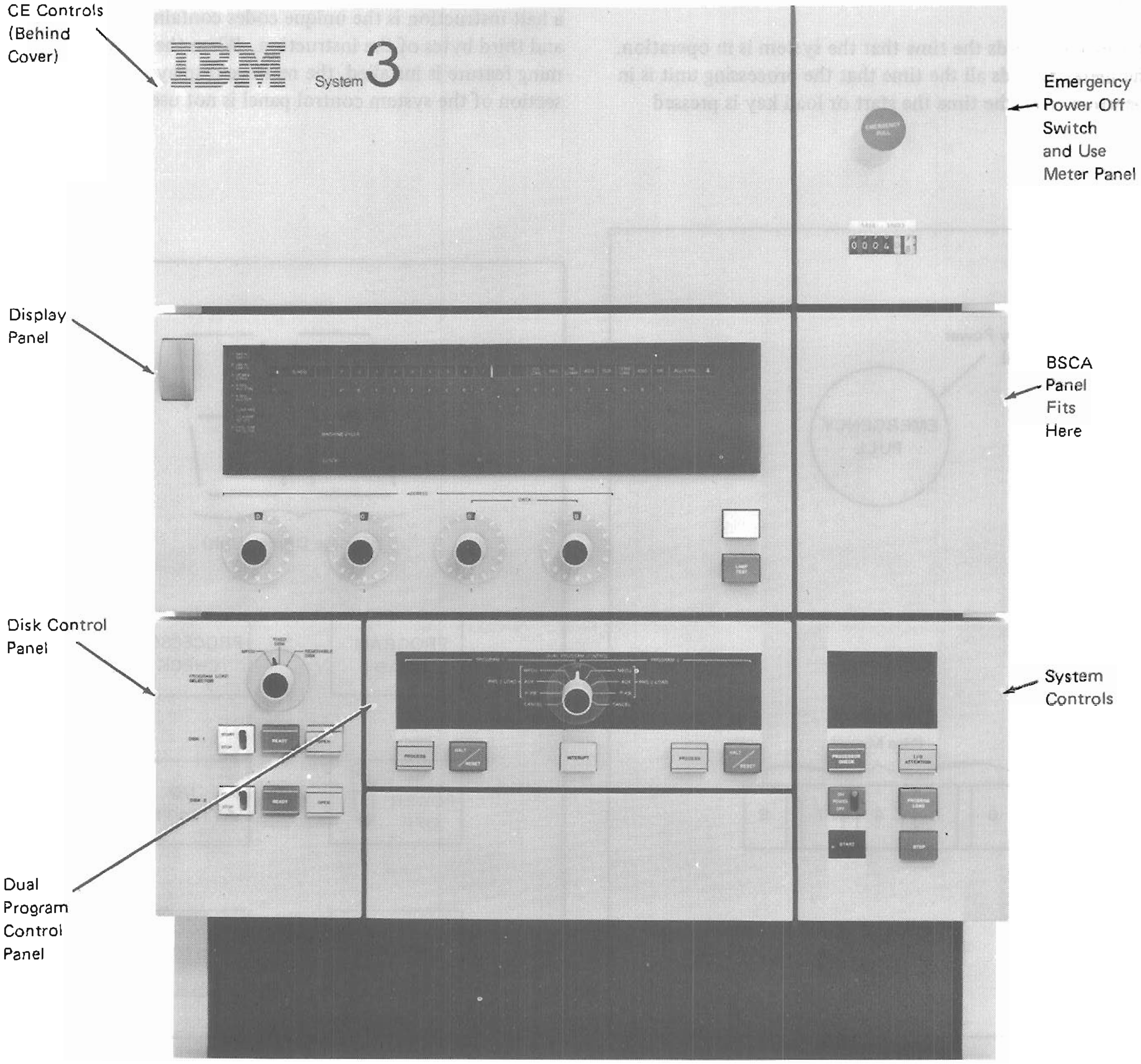

CE Controls (Behind Cover)
IBM System 3
Emergency Power Off Switch and Use Meter Panel
Display Panel
BSCA Panel Fits Here
Disk Control Panel
System Controls
Dual Program Control Panel

Figure 3-1. System Control Panel

OPERATOR CONTROLS

Emergency Power Off and Meter Panel (Figure 3-2)

Emergency Power Off

This switch controls all power to the system. The switch is operated by pulling out on the knob and locks in the out position. Power can be restored to the system only by intervention of maintenance personnel. The integrity of the data in storage is not guaranteed after operation of this switch.

Usage Meter

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

System Controls (Figure 3-3)

Message Display Unit

The two-position display unit lights from the halt identifier portion of a halt instruction. The halt identifier portion of a halt instruction is the unique codes contained in the second and third bytes of the instruction. When the dual programming feature is installed, the message display unit in this section of the system control panel is not used.

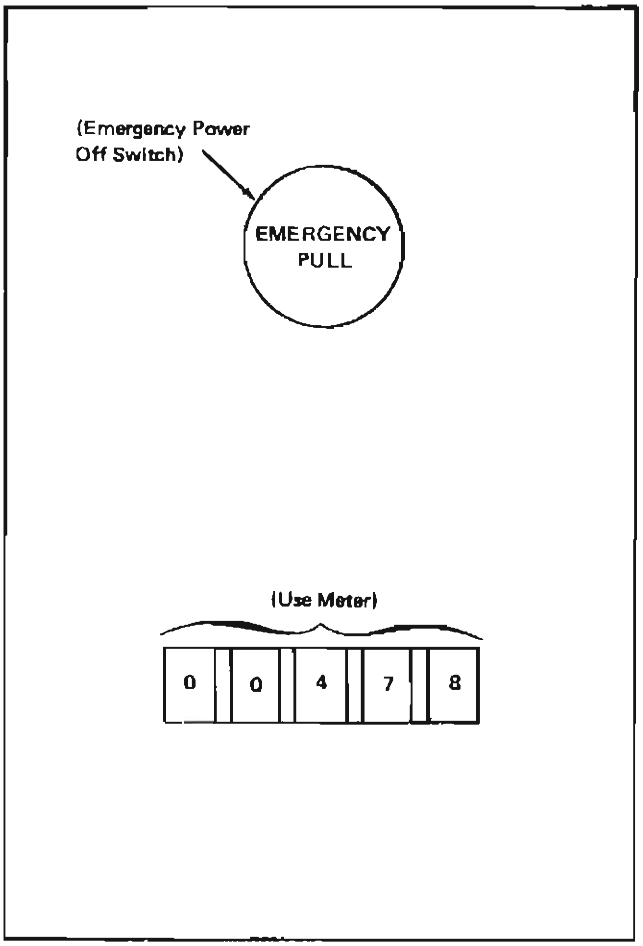


Figure 3-2. EPO and Meter Panel

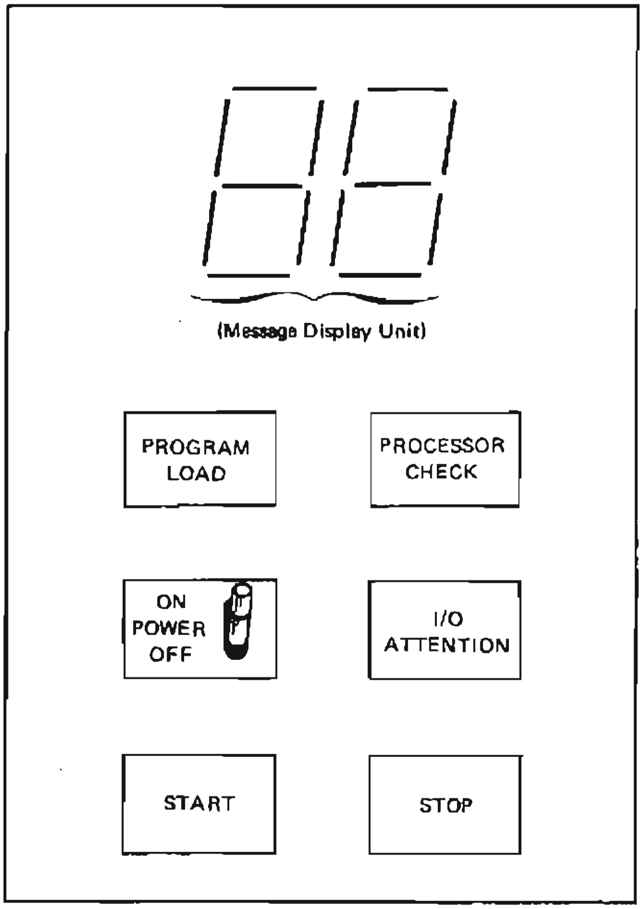


Figure 3-3. System Controls

Processor Check Light

The checks that turn on this light are:

1. Invalid op code.
2. Invalid address.
3. Parity check in the processing unit.
4. Invalid Q byte in an I/O instruction.
5. Parity error on an I/O data.
6. Incorrect selection of an I/O local storage register by an I/O device.

A system reset or operating the CE check-reset key turns this light ofi.

The checks that light this indicator cause the processing unit to stop immediately. When the processing unit stops, data from any I/O operation that is in progress is lost. The specific check that caused the processing unit to stop is indicated in the display panel section of the system control panel.

I/O Attention Light

This light turns on when an I/O unit is addressed and requires normal operator attention. The light turns off when the requirement for operator attention has been removed. Normal processing unit operation does not stop, but the I/O unit requiring attention will not accept a start I/O instruction until the condition is corrected. The I/O unit requiring attention lights an indicator to show what attention is required.

Typical I/O conditions that cause this indicator to light are:

1. Printer forms run-out.
2. Printer cover open.
3. MFCU hopper empty.
4. MFCU stacker full.
5. MFCU chip box full.
6. MFCU cover open.

Power On/Off Switch

This toggle switch controls the power to the system when the emergency power off switch has not been operated. When this switch is turned on, a system reset is performed in such a manner that no I/O operations are performed until explicitly directed. The integrity of data in storage is not guaranteed after this switch is operated.

Program Load Key

This key initiates loading the program into main storage. The following actions occur when this key is pressed:

1. All I/O and machine register, controls, and status indicators are reset.
2. The instruction address register in use is set to zero.
3. The MFCU read address register is set to zero.
4. In disk systems the disk data address register is reset to zero.
5. The first card in the primary feed of the MFCU or the first record on one of the disks in disk drive 1 is read into storage starting at location 0000. The unit that provides the first record is selected by a switch in disk system. In card systems only the MFCU primary feed can be used for program loading.

When the program load key is released, the processing unit executes the instructions read into storage by pressing the program load key, starting at location 0000.

If the selected I/O device is not ready, the I/O attention light will light when the program load key is pressed. It is necessary only to make the I/O device ready to complete the program load function.

Stop Key/Light

Pressing this key causes the processing unit to stop and the key to light. The processing unit stops at the end of the operation during which the key is pressed. I/O data transfers are completed without loss of information. Processing can be continued by pressing the start key.

Start Key

Pressing this key takes the processing unit out of the stopped condition and turns off the stop key light. The start key is also used in conjunction with other controls on the system control panel to perform certain manual operations. In systems without dual programming, the start key clears the message unit and allows the program to proceed after a programmed halt operation.

Dual Program Control Panel (Figure 3-4)

Message Display Units

A message display unit is provided for each program level. These units operate in the same manner as the message display unit in the system controls.

Process Lights

These lights indicate which program level is functioning at any time. If an interrupt is being serviced, this indicator shows which index registers and program status register are in use.

Halt Reset Keys

These keys are used to take a program level out of the programmed halt state. Pressing either of these keys clears the corresponding message display unit and allows the corresponding program to continue its normal operation.

Interrupt Key/Light

Pressing this key when it is illuminated causes the program in operation at that time to halt its normal operation and enter the interrupt-handling subroutine for interrupt level 0. Normal programmed operation will be resumed after the interrupt routine signals completion of interrupt servicing with a start I/O instruction to reset interrupt request 0.

The interrupt key is lighted only when the system is in dual programming mode and interrupt level 0 is enabled. Selection of whether the system is to be used in the dedicated or the dual programming mode is accomplished via the start I/O instruction. The start I/O instruction is also used to enable or disable the use of interrupt level 0.

Dual Program Control Switch

This rotary switch is normally used in conjunction with the console interrupt key. The status of this switch is sampled by the test-I/O-and-branch instruction.

File Control Panel (Figure 3-5)

Program Load Selector Switch

This switch is used to select the unit from which program loading is to be done. The fixed disk and removable disk positions refer to drive 1 only.

Start/Stop Switches

These switches (one per drive) turn the disk drive power on or off when system power is on. With the switch in the off position, the removable disk can be replaced.

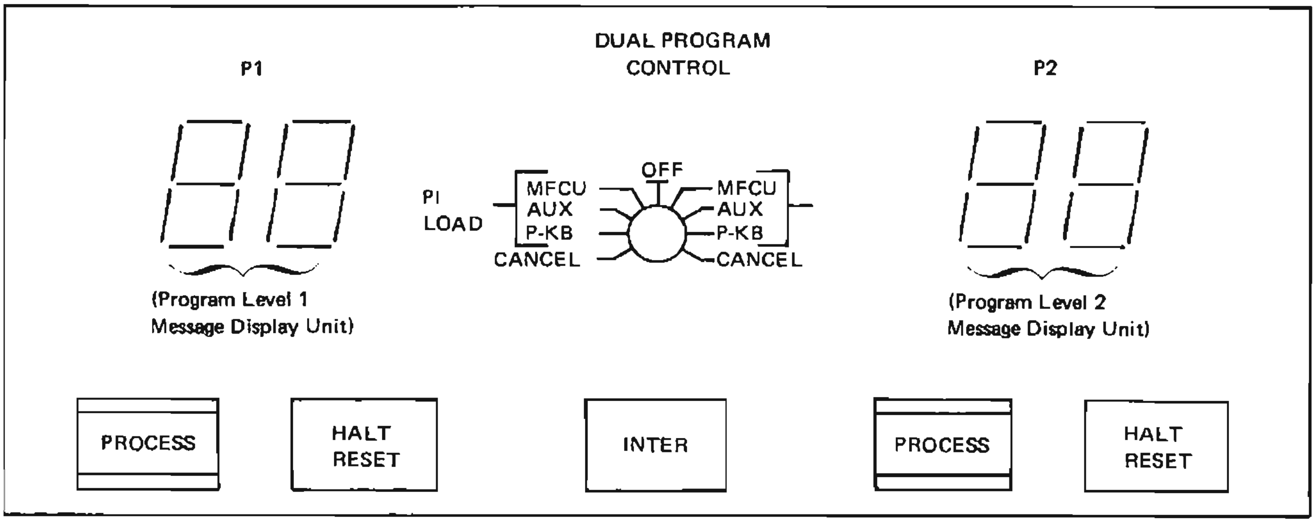


Figure 3-4. Dual Programming Control Panel

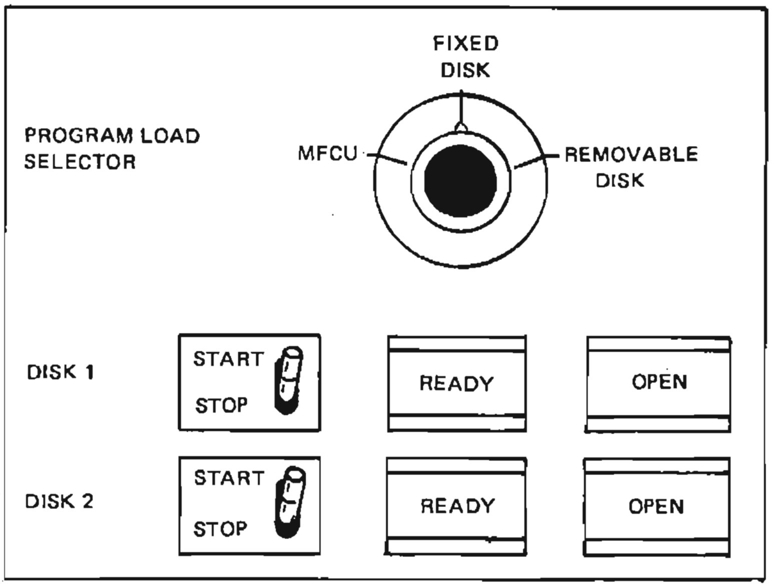


Figure 3-5. Disk Control Panel

Ready Lights

These lights (one for each drive) light when the disk drive is ready for use. If operation of the drive is attempted before this light turns on, the I/O attention light on the control panel will light.

Open Lights

These lights (one for each drive) indicate that the associated drive drawer can be opened for changing the removable disk. This light turns on when the start/stop switch is turned to the stop position, the read/write head has been retracted, and the disk has come to a stop.

CONSOLE DISPLAY

Display Panel (Figure 3-6)

Address/Data Switches

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

Register Display Unit

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

Strip Number	*Display*
1	SAR HI and SAR LO—contents of the two-byte storage address register.
2	LSR HI and LSR LO—contents of a local storage register selected by a switch on the CE panel.
3	OP REG—contents of the op register.
	Q REG—contents of the Q register.
4	B REG—contents of the B register.
	ALU CTL—state of the following ALU controls:

DIG CAR	Digital Carry
DEC	Decimal Instruction
RECOMP	Recomplement
ADD	Addition
SUB	Subtraction
TEMP CAR	Temporary Carry
AND	Logical And
OR	Logical Or

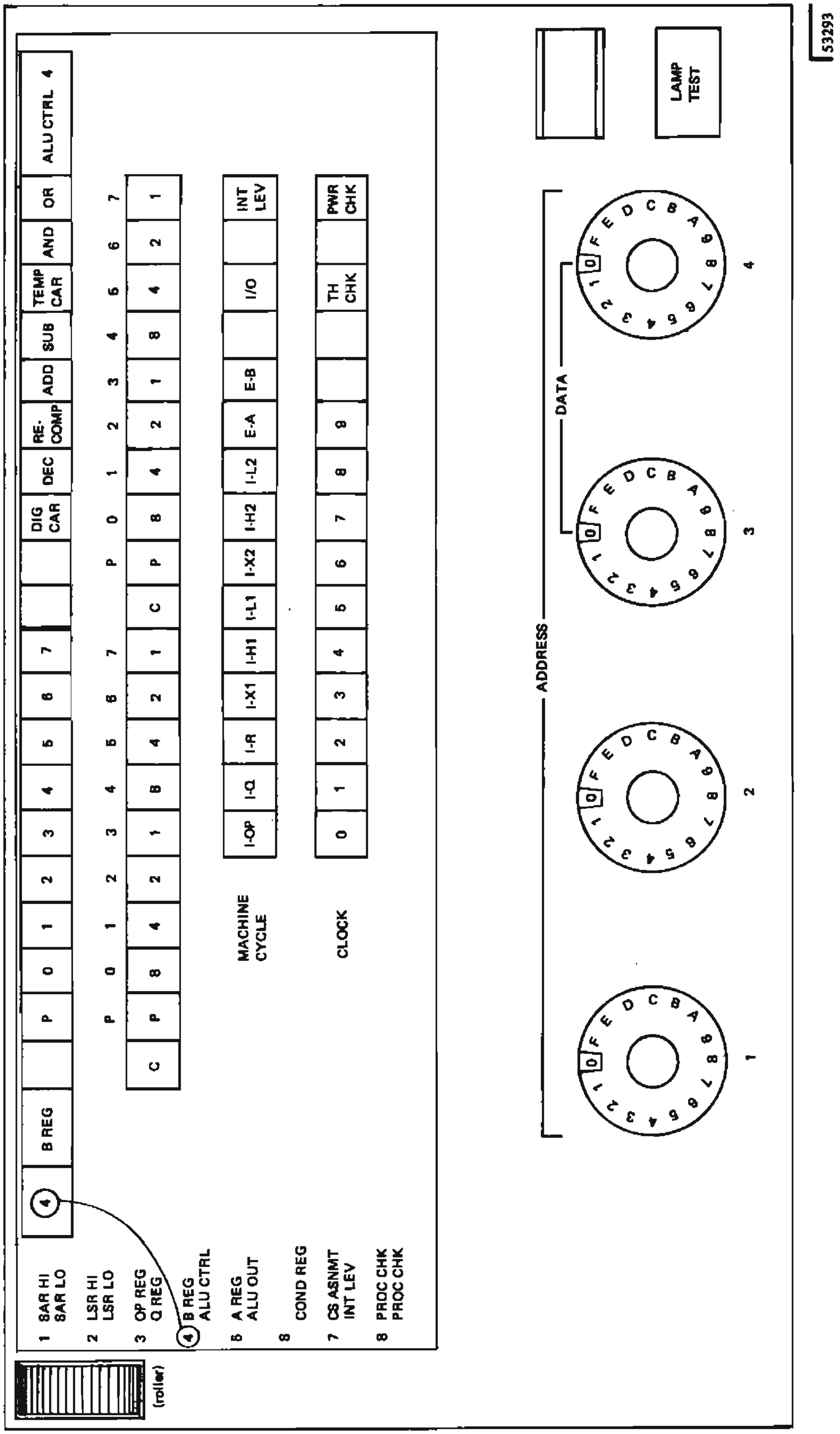


Figure 3-6. Display Panel

Strip Number *Display*

5 A REG—contents of the A register.

ALU OUT—output of the ALU.

6 COND REG—contents of the condition register as follows:

BIN OVF	Binary Overflow.
TF	Test False.
DEC OVF	Decimal Overflow.
HI	High.
LO	Low.
EQ	Equal.

7 CS ASNMT—cycle steal assignment as it is presented to the I/O devices.

INT LEV—Interrupt level indicating which I/O device is interrupting the program.

8 PROC CHK—the following causes of processor checks are displayed. Most of these indications are useful only to the customer engineer, but some of them are useful in analyzing programming errors.

I/O LSR—indicates that the selection of an LSR by an I/O device was not performed correctly. The CE LSR selector switch must be set to NORMAL to obtain an indication of an LSR parity check.

LSR F1—indicates that parity is incorrect on the output of the LSRs associated with disk storage and certain optional features.

LSR F2—indicates that parity is incorrect on the output of the LSRs associated with certain optional features.

LSR HI—indicates that parity is incorrect on the high-order byte output of the LSRs associated with the basic card system.

LSR LO—indicates that parity is incorrect on the low-order byte output of the LSRs associated with the basic card systèm.

Strip Number *Display*

8 PROC CHK— (Continued)

SAR HI—indicates that parity is incorrect in the high-order byte of the storage address register.

SAR LO—indicates that parity is incorrect in the low-order byte of the storage address register.

INV ADDR—indicates that the address contained in the storage address register is outside the address range of the system.

SDR—indicates that parity is incorrect in the storage data register.

CAR—indicates that the carry out of the ALU is incorrect.

CPU DBO—indicates that the processing unit attempted to send data with incorrect parity to an I/O device.

OP/Q—indicates that the op register or the Q register contains incorrect parity.

INV OP—indicates that the byte in the op register does not specify a valid operation.

CHAN DBO—indicates that the processing unit sent data with correct parity to an I/O device, but the I/O device received data with incorrect parity.

INV Q—indicates that the Q byte in the Q register is not valid.

DBI—parity is incorrect on data received from an I/O device.

A/B—parity is incorrect in the A register or the B register.

ALU—parity is incorrect at the output of the ALU.

Machine Cycles Display

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

Clock Cycles Display

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

Power Check Light

The power check light lights whenever the power switch is in the on position and power is not completely applied to the system, or whenever the power switch is in the off position and power is not completely removed from the system (except in those areas within the power control circuitry where power is never completely removed). The following statements apply to power check light operation:

1. When the power switch is turned on, the power check light will be on until power has sequenced all the way up and the system is ready to operate.
2. When the power switch is turned off, the power check light will be on until power has sequenced all the way down.
3. If system power is on and is then removed from the system because an over temperature has been detected (see Thermal Check Light), the power check light will be on until the power switch is turned off.
4. If system power is on and is then removed from the system because a power fault has been detected, the power check light will be on until the power switch is turned off.

After the power fault has been corrected, power is restored to the system by placing the power switch in the off position, pressing the check reset key, then turning the power switch to the on position.

Thermal Check Light

Whenever one of the system thermal sensors (located in the processing unit and in the line printer) detects an over-temperature condition, power is removed from the system and the thermal check light comes on. (The power check light also comes on, remaining on until the power switch is moved to the off position.) The thermal check light remains on until the over-temperature condition has been corrected and the power switch has been turned off. Power can then be restored to the system by turning the power switch on.

Figure 3-7 summarizes power check/thermal indications and the required action.

Fault	Power On/ Off Switch	Indicators		Action
		Power Check	Thermal	
Internal Power Supply Malfunction	On	On	Off	1. Call CE 2. Turn power switch to OFF 3. Correct problem 4. Depress Check Reset 5. Turn power ON
Thermal Condition	On	On	On	1. Turn power switch to OFF 2. Power check indicator goes off 3. Thermal light stays on until condition is removed
Customer Power Source Loss	On	On	On	1. Turn power switch to OFF 2. All indicators turn OFF 3. Turn power switch to ON and continue operation
Emergency Power Off (EPO) Activated	On	Off	Off	1. Call CE 2. Turn power switch to OFF 3. Correct problem 4. Restore EPO interlock 5. Turn power switch to ON

Figure 3-7. Power Check/Thermal Indications and Action

Lamp Test Key

Pressing this key turns on all the processing unit display lights.

BSCA Operator's Panel (Figure 3-8)

BSCA Attention Light

The following table shows the conditions indicated by this light.

Instruction	Condition Indicated
Any receive or transmit and receive or (on non-switched and multipoint networks only) receive initial.	Data set is not ready.
Auto call or receive initial on switched network.	Auto call unit power is off or data line is being used.
Any SIO except control SIO.	Either (1) the BSCA is disabled or (2) the external test switch is on and BSCA is not in test mode.
None.	Data set is not ready.

Unit Check Light

This light turns on when any bit in status byte 2 is on. Also, when an SNS transition or SNS stop register instruction is executed, it is possible for an LSR, S register, or DBI register parity check to occur, resulting in a unit check condition with the unit check light on. Under such a condition, the status byte 2 bits may all be zero.

The unit check indicator signifies that the BSCA program should enter an error recovery procedure.

Data Terminal Ready Light

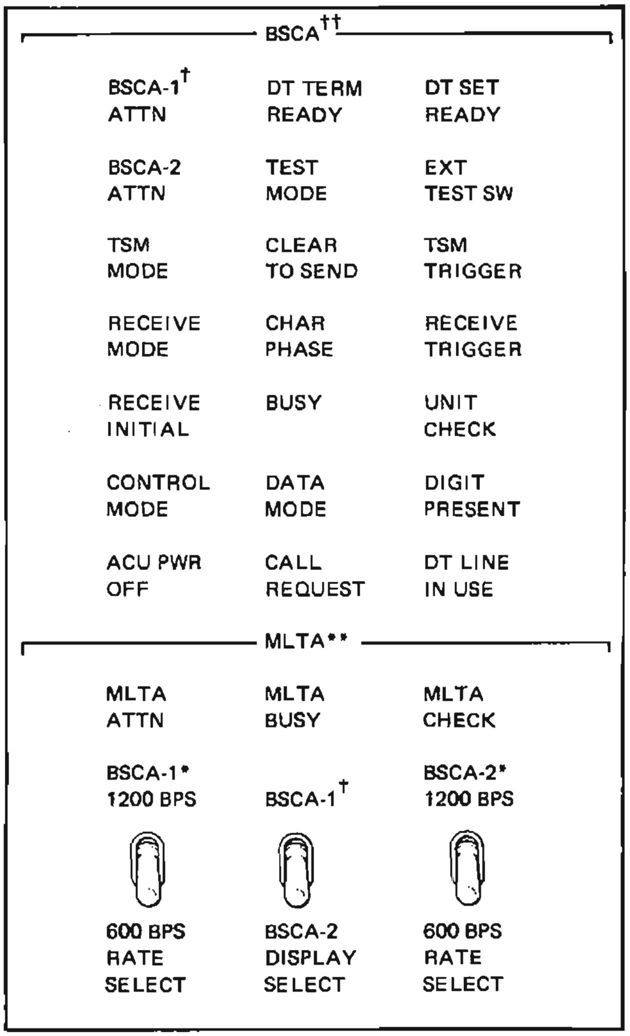
This light indicates that the BSCA is enabled and that the data terminal is ready for use.

Data Set Ready Light

The DT SET READY light indicates that the data set ready line from the data set is on and that the data set is ready for use.

Clear To Send Light

This light indicates that the clear to send line from the data set is on and that the adapter may now transmit.



BSCA††

BSCA-1† ATTN	DT TERM READY	DT SET READY
BSCA-2 ATTN	TEST MODE	EXT TEST SW
TSM MODE	CLEAR TO SEND	TSM TRIGGER
RECEIVE MODE	CHAR PHASE	RECEIVE TRIGGER
RECEIVE INITIAL	BUSY	UNIT CHECK
CONTROL MODE	DATA MODE	DIGIT PRESENT
ACU PWR OFF	CALL REQUEST	DT LINE IN USE

MLTA**

MLTA ATTN	MLTA BUSY	MLTA CHECK
BSCA-1* 1200 BPS	BSCA-1†	BSCA-2* 1200 BPS
600 BPS RATE SELECT	BSCA-2 DISPLAY SELECT	600 BPS RATE SELECT

* Rate select switch is for machines used outside the United States. If the rate selection feature is specified on either of the BSCAs, it will be made available to both.

** MLTA is available by RPQ only.

† This reads LCA on machines equipped with the local communications adapter (LCA) feature.

†† This reads LCA/BSCA on machines equipped with the LCA feature.

Figure 3-8. BSCA/LCA Control Panel

Receive Trigger Light

This light indicates the status of the receive trigger. The light is on when the trigger is at a binary 0 state.

Transmit Trigger Light

The TSM TRIGGER light indicates the status of the transmit trigger. The light is on when the trigger is at a binary 0 state.

Receive Mode Light

This light indicates that the adapter has been instructed to perform a receive operation.

Transmit Mode Light

The TSM MODE light indicates that the adapter has been instructed to perform a transmit operation.

Receive Initial Light

This light is turned on by an SIO receive initial instruction. It is turned off at the end of the receive initial operation.

Busy Light

This light indicates that the communication adapter is executing a receive initial, transmit and receive, auto call, receive or loop test instruction.

Character Phase Light

The CHAR PHASE light indicates that the adapter has established character synchronism with the transmitting station. The light is turned off at the end of receive operations and whenever character synchronism is lost.

Data Mode Light

This light is turned on by the decoding of an SOH or STX during a transmit or a receive operation. It is turned off at the end of the transmit or receive operation.

Control Mode Light

This indicator is used only on systems that have the station select feature installed. The light is turned on by an EOT sequence during a transmit, receive, or receive initial monitor operation when the station select feature is installed. It is turned off by the decoding of an SOH or STX.

Digit Present Light

This light indicates that a digit has been obtained from storage for the auto call unit when the auto call feature has been installed.

Auto Call Unit Power Off Light

The ACU PWR OFF light indicates that the auto call unit (special feature) power is off.

Call Request Light

On systems with the auto call feature installed, this light indicates that the communication adapter has received an SIO auto call instruction and is performing an auto call operation.

Data Line In Use Light

On systems with the auto call unit installed, the DT LINE IN USE light indicates that the data line occupied line from the auto call unit is on.

Test Mode Light

This light indicates that the program has placed the adapter in a test mode of operation.

External Test Switch Light

The EXT TEST SW light indicates that the switch at the data set end of the medium speed data set cable is in the test position. For high speed data sets, this indicator is active when the local test switch on the CE panel is in the on position.

Rate Select Switch

This switch, which is present only on systems installed outside the U.S.A. that have the rate selection feature as well, controls the rate of transmission and reception of data.

CE CONTROLS

CE control switches should be altered only when the system is stopped.

Cable Test Switch

This switch is part of the plug at the remote end of the BSCA data cable (that is, at the data set end of the cable). The switch should be set at the operate setting except during BSCA diagnostic operations. This switch is provided with data cables to medium speed data sets only.

CE Panel (Figure 3-9)

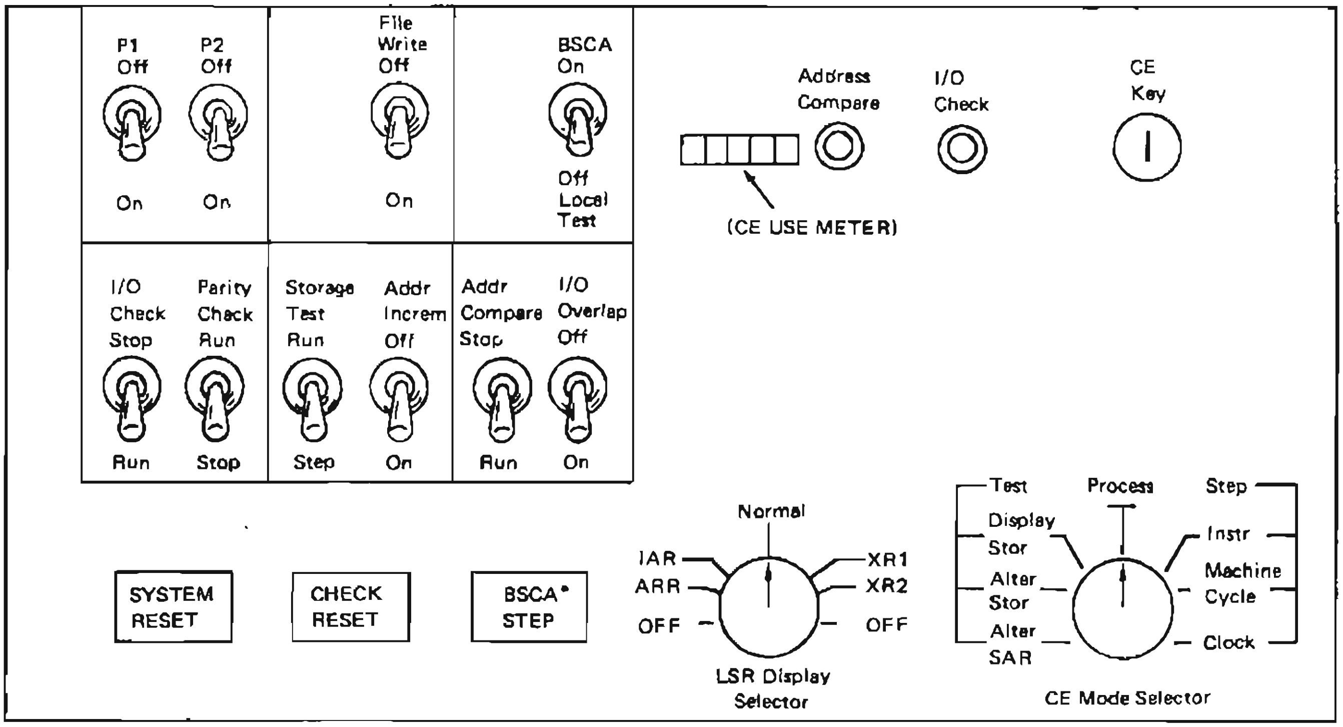
CE Key Switch

This switch is operated by the CE to prevent recording time when the system is being serviced.

CE Mode Selector

This rotary switch selects one of three processing unit operating modes: the normal PROCESS mode, the STEP mode, or the TEST mode. Process is the normal mode for normal programmed system operation.

In the step mode the rotary switch setting controls the manner in which the processing unit performs the stored program.



* If local communications adapter feature is installed, this switch Is labeled LCA/BSCA STEP.

Figure 3-9. CE Control Panel

1. Instruction Step—Each time the start key is pressed and released the processing unit peiforms one complete instruction. Any SIO instruction causes the next sequential instruction to be executed without start key operation.
2. Machine Cycle Step—Each time the start key is pressed and released the processing unit executes one machine cycle.
3. Clock Step—Each time the start key is pressed and released the processing unit executes two clock cycles. During an SIO instruction the clock runs automatically from the start of data transfer until data transfer is complete. The start key does not work while data transfer is taking place.

I/O operations operate in their normal manner during step mode operations.

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

1. Alter SAR—With the CE mode selector switch set to this position, pressing the start key transfers the setting in the address/data switches to the instruction address register in use at the time and into the storage address register.
2. Alter Storage—With the switch in this position, pressing and releasing the start key transfers the data specified by the two rightmost address/data switches into the A register and into storage at the address specified by the SAR.
3. Display Storage—Pressing and releasing the start key transfers the data at the address specified by the storage address register to the B register, and then to the Q register, where it can be displayed by the roller switch on the display panel. The data is not destroyed in storage.

The storage test switch must be in the step position to avoid a processor check when the CE mode selector switch is moved between the alter storage position and the display storage position.

Note: No test is made for invalid storage addresses when the CE mode selector switch is in one of the test positions.

System Reset Key

Pressing this key with the CE mode selector switch set at PROCESS mode causes all I/O and machine registers (not local storage registers) to be reset to zero. Program level instruction address register and both program status registers are reset to zero. All other local storage registers are unaffected. A complete program restart is normally required after a system reset.

Check Reset Key

Pressing this key resets the processor checks, input/output checks, and power check. Resetting these checks allows the processing unit to resume processing when the start key is pressed. The check reset key also immediately resets all 5445 functions and status indicators. Therefore, do not press it while the 5445 attachment is processing I/O instructions.

BSCA Step Key

The BSCA STEP key, which is effective only when the communication adapter is in step mode, causes the communication adapter to advance one bit-time for each key depression.

BSCA Local Test Switch

This toggle switch sets the high speed data set into local test mode and causes data to be wrapped around through the data set with a start I/O loop test instruction in test mode.

File (5444 Disk) Write Switch

When this switch is in the off position, write operations cannot be performed on 5444 disk storage.

File (Disk) Write Switch

When this switch is in the off position, write operations cannot be performed on disk storage.

Address Compare Switch

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

CE Servicing Switches

The following switches are used only by the customer engineer:

1. Storage Test.
2. Address Increment.
3. I/O Overlap.
4. I/O Check.

5. Parity.
6. P1.
7. P2.
8. BSCA Local Test.

Address Compare Light

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

I/O Check Light

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

LSR Display Selector Switch

This switch selects the local storage register to be displayed by the LSR position of the register display switch. The LSRs displayed are the LSRs in use (program level 1, program level 2, or an interrupt level). In the normal position, the register in use at any particular instant is the one displayed. The off positions are reserved for CE use. The switch must be in the normal position for LSR parity checks to be displayed.

MANUAL OPERATION PROCEDURES

Altering Storage Addresses

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

1. Press the stop key.
2. Turn the storage test switch to STEP.
3. Turn the CE mode selector switch to ALTER SAR.
4. Set the address/data switches to the desired address.
5. Press the start key.

If the CE mode selector switch is now turned to PROCESS and the start key is pressed, the processing unit will begin processing with the instruction located at the address just set in the SAR.

Altering Storage

1. Press the stop key.
2. Set the storage test switch to STEP.
3. Set the CE mode selector switch to ALTER SAR.
4. Set the address of the storage position you want to alter in the address/data switches.
5. Press the start key.
6. Turn the CE mode selector switch to ALTER STOR.
7. Set the two rightmost address/data switches to the hex value you want in storage.
8. Press the start key.

In order to resume normal operation it will be necessary to set the storage address register to the address of the instruction with which you wish to begin. This is accomplished by the procedure described in "Altering Storage Addresses" above.

3

Displaying Storage

1. Press the stop key.
2. Set the storage test switch to STEP.
3. Turn the CE mode selector switch to ALTER SAR.
4. Set the address of the storage location you want to display in the address/data switches.
5. Turn display roller to setting 5.
6. Press the start key.
7. Turn the CE mode selector to DISPLAY STOR.
8. Press the start key. The byte stored will be displayed in the console lights.

To resume normal operation it will be necessary to set the storage address register to the address of the instruction with which you wish to begin processing. This is accomplished by the procedure in "Altering Storage Addresses" above.

Displaying Local Storage Registers

1. Press the stop key.
2. Turn the register display roller switch to LSR HI/ LSR LO.
3. Turn the LSR display selector switch to the desired LSR.

Stopping at a Particular Address

1. Set up the desired address in the address/data switches.
2. Set the register display switch to SAR HI/SAR LO.
3. Set the address compare switch to STOP.

Press the start key if the system is stopped.

At the end of the cycle in which the desired address is used to access storage, the processing unit stops with the address compare light on.

CHECK CONDITIONS

Processor Checks

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

Invalid Address

This check indicates that the storage address register contains an address outside the address range of the processing unit.

Invalid Op Code

This check indicates that the op register contains a code that is not recognized as a valid op code.

Parity Check In The Processing Unit

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

Invalid Q Byte In An I/O Instruction

This check indicates that the device address contained in the Q code of an I/O instruction addressed a unit that is not available to that system or that the N code in the Q byte is not valid for that I/O device.

I/O Attention

This check indicates that the processing unit has addressed an I/O device that requires attention because of a condition that occurs during the normal course of operating the system. Such conditions are: empty hopper, full stacker, full chip box, or forms runout. This check does not stop the processing unit. Recovery from this condition is accomplished by returning the I/O device to ready status.

Unit Check

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

IBM 5424 Multi-Function Card Unit (MFCU)

The MFCU (Figure 4-1) is the primary input/output unit of the card system, providing the capability of performing unit record functions. The unit can feed cards from either of two hoppers, read the cards, punch the cards, print on the cards, and stack the cards in any of four stackers.

Figure 4-2 shows the path cards take through the MFCU. Two hoppers are provided: the primary and the secondary. Cards can enter the unit and be read from either hopper. After the reading station, cards from the primary go to an upper level wait station; cards from the secondary go to a lower level wait station. From these wait stations either the primary or the secondary card can be advanced through the punching and printing stations to the stackers.

The following operations can be performed.

1. Feed.
2. Feed and read.
3. Punch and feed.
4. Punch, feed, and read.
5. Print and feed.
6. Print, feed, and read.
7. Punch, print, and feed.
8. Punch, print, feed, and read.
9. Selection of the card leaving the wait station into any of four stackers.

These operations can be performed from either the primary or the secondary feed.

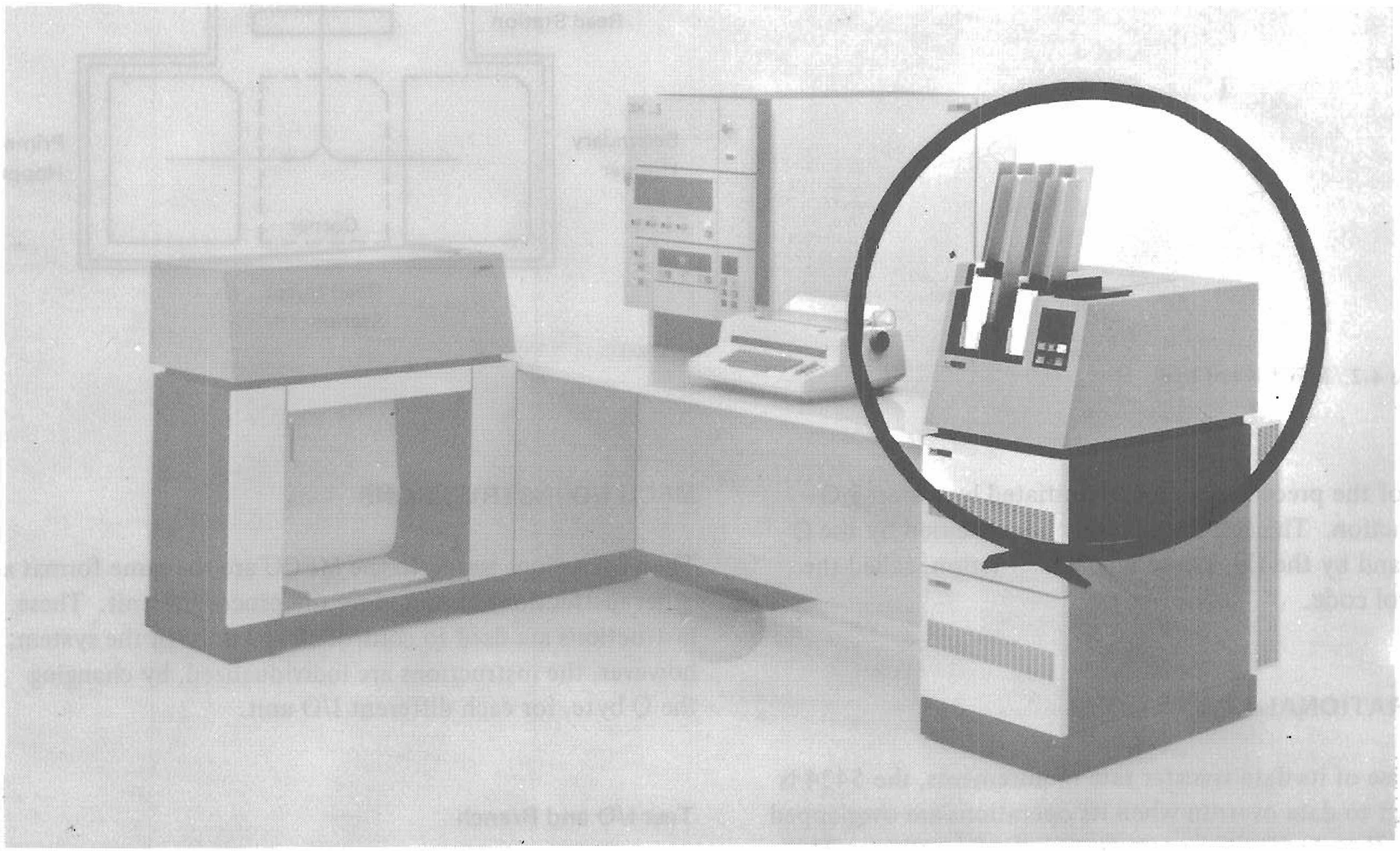


Figure 4-1. IBM 5424 Multi-Function Card Unit

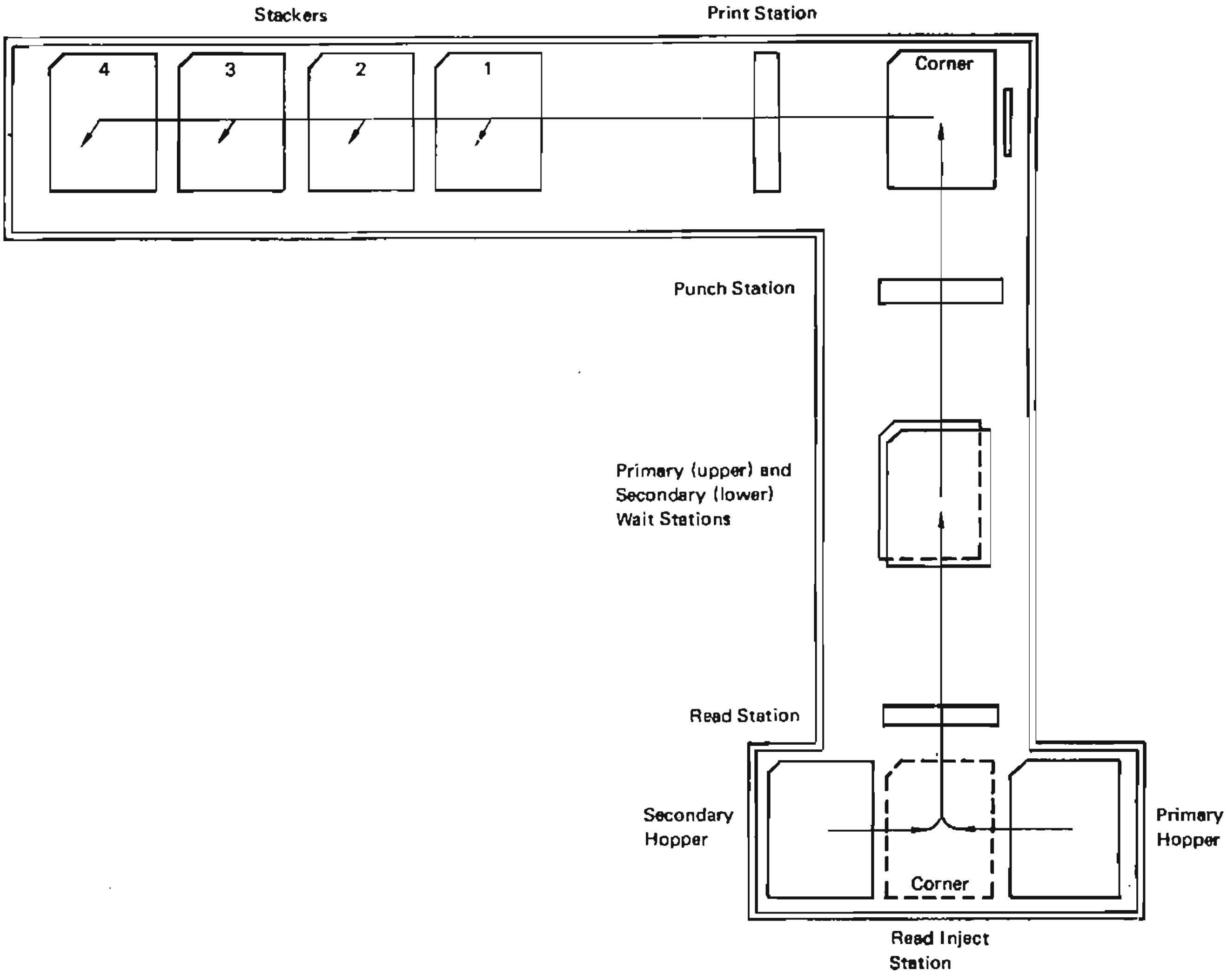


Figure 4-2. MFCU Card Path

Any of the preceding actions is initiated by a start I/O instruction. The action to be taken is specified by the Q byte and by the third byte of the instruction, called the control code.

OPERATIONAL LIMITATIONS

Because of its data transfer rate requirements, the 5424 is subject to data overrun when its operations are overlapped with other devices in certain system configurations. This condition is not detected by the 5424 and may result in loss of data. Refer to Chapter 2, "Channel Limitations" for allowable overlapped device configuration that will not cause overrun in the system.

MFCU I/O INSTRUCTIONS

The instructions issued to the MFCU are the same format as other instructions executed by the processing unit. These instructions are used to control all I/O units in the system; however, the instructions are individualized, by changing the Q byte, for each different I/O unit.

Test I/O and Branch

Mnemonic: TIO

Op Code	Q Byte	Branch-to Address
Z1	F M N	

Operation: The condition specified by the N portion of the Q byte is tested. If the condition exists, the next instruction is taken from the address specified in the address portion of the instruction. If the condition does not exist, the next sequential instruction is executed.

The Q byte contains the device address, the M bit, and the N code. Bits 0-3 of the Q byte contain the device address (always F for the MFCU). Bit 4 is the M bit. When this bit is 0, the primary feed is tested; when the bit is 1, the secondary feed is tested.

Bits 5, 6, and 7 of the Q byte constitute the N code. The N code specifies the conditions that are to be tested as follows:

N Code	*Condition*
000	Specified feed not ready/check condition exists.
001	Read/feed busy.
010	Punch data busy.
011	Either or both read/feed or punch data is busy.
100	Card printer busy.
101	Read/feed is busy, card printer is busy, or both read/feed and card printer are busy.
110	Punch data is busy, card printer is busy, or both punch data and card printer are busy.
111	Any or all of the following are busy: read/feed, punch data, or card printer.

Read/feed becomes busy as soon as a start I/O instruction for the MFCU is accepted by the MFCU. Punch data becomes busy when the MFCU accepts a start I/O instruction that specifies punching. Acceptance of an MFCU instruction that specifies printing causes a card printer busy indication. The card printer becoming not busy does not indicate that the print operation is complete, because this indication drops (to allow another print instruction to be issued) before the print operation is completed. The occurrence of a feed check while any one of the busy conditions is active turns off the busy condition immediately. Otherwise, the busy condition is turned off at the end of the I/O operation (except as noted for the card printer busy indication).

Program Note: The address not used for the next instruction (branch-to address for no-branch condition or next sequential instruction address for branch condition) is retained in the address recall register until the next decimal, insert-and-test characters or branch instruction is executed.

Example:

Instruction

C1	F7	02	C4

Resulting Operation:

If the MFCU is performing any operation on the primary feed, the next instruction is taken from location 02C4; otherwise the next sequential instruction is executed.

Advance Program Level

Mnemonic: APL

Op Code	Q Byte			
F1	F	M	N	Not Used

Operation: If the dual programming feature is installed, the condition specified by the N portion of the Q byte is tested. If the condition exists, the address of the next instruction is taken from the instruction address register of the program level that is *not* active at the time the APL instruction is encountered. The program on this level now becomes the active program level, and the program level from which the advance occurred becomes the inactive program level. If the condition is not present, the next sequential instruction is taken and no program level advance occurs. If a program level advance occurs, the return point to the program level advanced from is the address of the start of the advance program level instruction.

If the dual programming feature is not installed, the program loops on the advance program level instruction until the specified condition is not present, then executes the next sequential instruction. An unconditional advance program level instruction results in execution of the next sequential instruction.

The Q byte contains the device address, the M bit, and the N code. Bits 0-3 of the Q byte contain the device address (always F for the MFCU). Bit 4 is the M bit. When bit 4 is 0, the primary feed is tested; when the bit is 1, the secondary feed is tested.

Bits 5, 6, and 7 of the Q byte constitute the N code. The N code specifies the conditions that are to be tested as follows:

N Code	*Condition*
000	Specified feed not ready/check condition exists.
001	Read/feed busy.
010	Punch data busy.
011	Either or both read/feed or punch data is busy.
100	Card printer busy.
101	Read/feed is busy, card printer is busy, or both read/feed and card printer are busy.
110	Punch data is busy, card printer is busy, or both punch data and card printer are busy.
111	Any or all of the following are busy: read/feed, punch data, or card printer.

Read/feed becomes busy as soon as a start I/O instruction for the MFCU is accepted by the MFCU. Punch data becomes busy when the MFCU accepts a start I/O instruction that specifies punching. Acceptance of an MFCU instruction that specifies printing causes a card printer busy indication. The card printer becoming not busy does not indicate that the print operation is complete, because this indication drops (to allow another print instruction to be issued) before the print operation is completed. The occurrence of a feed check while any one of the busy conditions is active turns off the busy condition immediately. Otherwise, the busy condition is turned off at the end of the I/O operation (except as noted for the card printer busy indication).

Example:

Instruction (for program level 1)

F1	F0	00
0400	0401	0402

Resulting Operation:

If the primary feed is not ready to feed cards or if an error condition exists in the MFCU, the address of the next instruction is taken from P2 IAR. If the primary feed is ready, the next instruction will be taken from location 0403 and following bytes.

Load I/O

Mnemonic: LIO

Op Code	Q Byte	Operand Address	
Y1	F M N		

Operation: The contents of the two-byte field addressed by the operand address are moved to the local storage register designated by the Q byte. If the selected register is busy, an unconditional program advance occurs if the system has dual programming feature installed. If the dual programming feature is not installed and the selected register is busy, the program loops on the load I/O instruction until the register becomes not busy.

The Q byte contains the device address, M bit, and N code. Bits 0-3 are the device address (always F for the MFCU). The M bit designates whether the start I/O operation that follows the load operation is to be performed in normal mode or in diagnostic mode. If bit 4 is 0, the operation is performed in normal mode; if the bit is 1, the operation is performed in diagnostic mode.

The N code (bits 5, 6, and 7) specifies the register to be loaded. Only the following bit patterns are valid:

Bits *5 6 7*	*Register*
1 0 0	MFCU print data address register.
1 0 1	MFCU read data address register.
1 1 0	MFCU punch data address register.

Any other bit patterns are invalid and cause processor check from an invalid Q byte.

If diagnostic mode is specified (normally only for CE purposes) when loading the read data address register, read check data will be placed in storage starting with an address 128 locations higher than the read address when the next start I/O instruction specifying reading is executed. Loading the punch data address register in diagnostic mode results in punch check data entering storage on the next start I/O instruction that specifies punching, starting 128 bytes above the punch data location.

Program Note:

1. A Q byte of 00 results in a no-op condition.

Example:

Instruction

31	F5	02	77

Operand

2F	10
0276	0277

MFCU Read Data Address Register After Operation

2F	10

Sense I/O

Mnemonic: SNS

Op Code	Q Byte	Operand Address	
Y0	F M N		

Operation: Two bytes of status information presented to the processing unit by the I/O attachment circuitry are placed in storage in the field specified by the operand address. The field is addressed by its rightmost byte.

The Q byte contains the device address (always F for the MFCU), an M bit that is not used in this instruction and should be 0, and an N code. The N code specifies the information to be stored as follows:

N Code	*Information*
000	Special indicators for CE use.
001	Special indicators for CE use.
010	Invalid.
011	Status indicators.
100	MFCU print data address register.
101	MFCU read data address register.
110	MFCU punch data address register.
111	Invalid.

Use of an invalid code results in a processor check caused by invalid Q byte.

Figure 4-3 shows the meaning of the status bits in the status bytes. The conditions that set the MFCU status bits are:

1. Print buffer 1 busy: This indicator is turned on when a start I/O instruction that specifies printing from buffer 1 is accepted. The bit is reset when the printer has finished printing on that card. See *Start I/O* for the method of buffer selection.
2. Print buffer 2 busy: This indicator is turned on when a start I/O instruction that specifies printing from buffer 2 is accepted. The bit is reset when the printer has finished printing on that card. See *Start I/O* for the method of buffer selection.
3. Card in wait 1: This indicator is set when read/feed becomes not busy if a card was fed or read from the primary hopper. This indicator is not reset if a document is manually removed from the wait station.

Bit	Status Byte 2	Status Byte 1
0	Print Buffer 1 Busy	Read Check
1	Print Buffer 2 Busy	Punch Check
2	Card in Wait 1	Punch Invalid
3	Card in Wait 2	Print Data Check
4	Reserved	Print Clutch Check
5	Hopper Cycle Not Complete	Hopper Check
6	Card in Transport/Counter Bit 2	Feed Check
7	Card in Transport/Counter Bit 1	No-Op

Figure 4-3. MFCU Status Bytes

4. Card in wait 2: This indicator is set when read/feed becomes not busy if a card was fed or read from the secondary hopper. This indicator is not reset if a document is manually removed from the wait station.
5. Reserved: Should be 0.
6. Hopper cycle not complete: This indicator is set when a start I/O command is accepted for execution. It is reset by the card exiting from the hopper.
7. Card in transport counter, bits 1 and 2: These two bits constitute a counter that keeps track of the number of cards between the wait station and the stackers. Every card that leaves the wait station adds 1 to the counter. Every card that is directed to a stacker, except those stacked after a machine check, subtracts 1 from the counter. When a feed check occurs, the counter indicates the number of cards that were in the transport when the feed check occurred. These bits are reset to zero by turning power on and by non-process runout.
8. Read check: This indicator is set if data is read from the card incorrectly. This check also sets the check condition that can be tested by a test-I/O-and-branch instruction. The read check indicator is reset by the next start I/O instruction, system reset, non-process runout, or check reset.
9. Punch check: This indicator is set if the correct punches for the specified data are not selected. This check also sets the check condition that can be tested by the test-I/O-and-branch instruction. The punch check indicator is reset by the next start I/O instruction, system reset, non-process runout, or check reset.
10. Punch invalid: This indicator is set if the processing unit sends the MFCU a character that is not one of the 64 card code characters during a punch operation. If this bit is turned on, punch checking is not performed for the rest of the card. The indicator is reset by the next start I/O instruction, system reset, non-process runout, or check reset. This bit also sets the check condition that can be tested by a test-I/O-and-branch instruction.
11. Print data check: This indicator is set if the print wheel loses synchronism with the processing unit. This check also sets the check condition that can be tested by the test-I/O-and-branch instruction. This indicator is reset by a sense I/O instruction that specifies the status indicators, system reset, check reset, or non-process runout.
12. Print clutch check: This indicator is set when the card is printed on the wrong line, either too high or too low. This bit also sets the check condition that is tested by the test-I/O-and-branch instruction. Print clutch check is reset by a sense I/O instruction that specifies the status indicators, system reset, check reset, or non-process runout.
13. Hopper check: Hopper check is set when the MFCU is instructed to feed a card and a card fails to leave the specified hopper. Hopper check causes the MFCU to become not ready. The hopper check bit is reset by non-process runout or by pressing the start key.
14. Feed check: Feed check is set by any improper movement of the card through the feed and transport sections of the MFCU. Feed check causes the MFCU to become not ready and lights the NPRO light. Feed check is reset by non-process runout.
15. No-op: This indicator is set when the MFCU is issued a command it is unable to execute. This bit sets the check condition that can be tested by the test-I/O-and-branch instruction. The no-op indicator is turned off by the sense I/O instruction that specifies the status indicators, system reset, check reset, or non-process runout.

Start I/O

Mnemonic: SIO

Op Code	Q Byte	Control Code
F3	F M Func	

Operation: The start I/O instruction is used to initiate any MFCU operation. If the MFCU is busy for that instruction or is not ready for any reason except unit check, the program will loop on the start I/O instruction until the MFCU becomes not busy or is made ready. In systems with the dual programming feature a start I/O instruction issued to an MFCU that is busy or not ready causes an automatic program level advance. If the start I/O instruction is issued when the MFCU is in the not ready condition, the I/O attention light on the system control panel will light. Correcting the not ready condition causes the instruction to be executed. If the MFCU has a feed check when the start I/O instruction is issued, the instruction is no-oped and the no-op status bit is set. (Status bits are discussed in *Sense I/O.*)

The Q byte defines the unit to operate and the operation to be performed. Bits 0-3 of the Q byte are the device address. For the MFCU this is always F. Bit 4 is a modifier bit that determines if the operation is to be performed on the primary card or the secondary card. If bit 4 is 0, the operation is performed on the primary card; if bit 4 is 1, the operation is performed on the secondary card.

Bits 5 through 7 of the Q byte are called the N code. Each of these bits specifies one of the data functions the MFCU can perform (read, punch, or print). A card will be fed from the feed specified by the M bit for each start I/O instruction. If none of bits 5 through 7 is 1, only feeding will be accomplished; no data will be transferred. The bit patterns cause operation as follows:

Bits *5 6 7*	*Operation*
0 0 0	Feed.
0 0 1	Feed and read.
0 1 0	Punch and feed.
0 1 1	Punch, feed, and read.
1 0 0	Print and feed.
1 0 1	Print, feed, and read.
1 1 0	Punch, print, and feed.
1 1 1	Punch, print, feed, and read.

The third byte in the instruction is a control code. It furnishes controls on reading and printing, and provides for stacker selection. The control code is bit significant as follows:

Bit	*Meaning*
0	Print buffer address. When this bit is 0, print buffer 1 is used; when it is 1, print buffer 2 is used. See *Print Operations.*
1	IPL Read if 1.
2	Print four lines if 1. See *Print Operations.*
3	Reserved.
4	Reserved.
5-7	Select stacker according to the following:

Bits *5 6 7*	*Select*
1 0 0	Stacker 4.
1 0 1	Stacker 1.
1 1 0	Stacker 2.
1 1 1	Stacker 3.

Program Note: If an MFCU check that would prevent the execution of the start I/O instruction exists, the instruction is ignored (no-oped) and a no-op status bit is set in the device attachment. If a check that will not prevent the execution of the instruction exists, the instruction will be executed and the check will be reset. Conditions causing no-ops are: (1) feed check, or (2) either a punch or print instruction has been issued without a card in the wait station.

Example:

Instruction

F3	FF	26

Result:

96 columns read into storage beginning at the address specified by the MFCU Read Data Address Register.

96 columns punched into card. Data taken from storage beginning at the location specified by the MFCU Punch Data Address Register.

128 print positions printed (blank is considered a printed character). Data taken from print buffer 1 of the addresses beginning at the address specified by the MFCU Print Data Register. See Print Operation.

Card in the secondary position of the wait station is punched, printed, and stacked in stacker 2.

The card to be read is fed from the secondary hopper and after reading, is transported to the secondary wait station.

CARD READ OPERATIONS

The read/feed functions of start I/O instructions move a card from the specified hopper to the corresponding wait station. If read is specified, the data contained in all 96 columns of the card is transferred to storage at a field specified by a load I/O instruction. The read data is checked to ensure that it is read correctly. An error in reading causes a read check.

A load I/O instruction must be executed before each start I/O instruction that specifies card reading. This load I/O instruction must load the address of the high-order byte of the read data field into the MFCU read data address register. To meet performance specifications the addresses must be on 128 byte boundaries.

The card feeding and reading rate is determined by the operations being performed. The rated reading speeds (250 cards per minute for model A1 and 500 cards per minute for model A2) are for read operations only. If punching or printing is performed at the same time, the reading rate will be reduced to the rate at which punching and printing are performed. To maintain the rated reading rate, successive start I/O instructions specifying reading must be issued within 44 milliseconds (model A1) or 22 milliseconds (model A2) after the read/feed busy indicator indicates not busy. The read/feed busy indicator can be tested with a test-I/O-and-branch instruction.

Program Notes:

1. There are three MFCU print busy indicators. The card printer busy (testable with the TIO or APL instruction) comes on with the SIO instruction including print and goes off with the start of printing on the actual card. For maximum hardware overlap for rated throughput, the next SIO instruction including print can be issued and will be accepted by the hardware at this time. Because printing for the first card has not been completed, error checking (for print errors) cannot be done at this time. When the next APL or TIO instruction is issued (after the second SIO), it will indicate any errors on the first card, since the first card is now complete and the second card has arrived at the print station. However, printing may have been started or even completed on the second card. Therefore, an error indicated at this time may have occurred on either of the two cards.

   Print Buffer 1 Busy and Print Buffer 2 Busy (testable by SNS and TBN or TBF instructions) can be used to determine which MFCU print buffer (or buffers) is available. However, this busy indication drops just prior to the completion of the print operation. Consequently, an error condition can come up after this indication drops.

2. After the last I/O operation in a program, a final wait operation should be performed in which a wait is done on the card in transport/counter bits to become 0. This is to ensure that all cards have cleared the transport without feed checks and that no errors have occurred during the last I/O operations.

IPL Read

Pressing the program load key causes the following reader actions to occur:

1. The MFCU read data address register is set to 0000.
2. A read operation is performed from the primary hopper of the MFCU without a start I/O instruction being executed.

The read operation is performed in the IPL card reading mode described in the introductory chapter of this manual. Reading in this mode (C and D bits taken from tier 3) can be continued by setting bit 1 of the start I/O instruction control code to 1 for each read start I/O instruction in which IPL mode reading is desired. IPL read can also be initiated by a start I/O operation.

PUNCH OPERATIONS

Start I/O instructions that specify punching initiate moving a card from one of the wait stations, through the punch station and transport, to the stackers. As the cards pass through the punch station, data from storage is recorded in them in the form of punched holes. The punching is checked to ensure that the correct data is punched. An error causes a punch check. The punch data is checked to ensure that the data to be punched is valid for the 64 characters allowed in the card code. An error causes a punch invalid check. No punch checking is performed after a punch invalid check.

A load I/O instruction must be executed before each start I/O instruction that specifies a punch operation. This load I/O instruction places the address of the high-order byte of the punch data field in the MFCU punch data address register. Column 1 of the card is punched with the data contained in storage at this address. Column 2 of the card is punched with the data contained in storage at the next higher address. The punch data fields must be on 128 byte boundaries.

If a punch start I/O instruction is given with no card in the wait station, the instruction will be ignored and the no-op status bit will be set.

Card punching is performed at a single rate for each model of MFCU, model A1 at 60 cards per minute and model A2 at 120 cards per minute. To maintain this throughput, successive punch start I/O instructions must be executed within 90 milliseconds (A1) or 45 milliseconds (A2) of the end of punch busy indication to the test-I/O-and-branch instruction.

PRINT OPERATIONS

The start I/O print and feed or print and read operation initiates card motion from the selected wait station, through the punch and cornering stations, and into the print station where three or four lines of 32 characters each are printed on the card. If there is no card in the wait station, the instruction is ignored and the no-op bit is turned on in the status indicators.

The print data area must be loaded before the start I/O print instruction is issued. The print data area consists of two print buffers each of which is always 128 bytes in length even though only 96 bytes are required when three lines are printed. The buffers are located in main storage. They are defined to the MFCU attachment with a load I/O instruction that loads the address of the high-order byte of print buffer 1 into the MFCU print data address register. The print data buffer address must be on a 256 byte boundary.

The load I/O instruction should be given only once for each job or each time the print data address area changes. If the load I/O instruction is given while either print buffer is busy, an unconditional program advance (or loop on the load I/O instruction) occurs until both buffers are free. This causes a loss of throughput. If power is lost for any reason, the print load I/O instruction must be re-executed before a start I/O instruction specifying printing is executed, or processor checks will occur if printing is attempted.

The 128 byte print data area is printed on the card in the following manner:

Line 1–Leftmost address to byte 32.
Line 2–Bytes 33 through 64.
Line 3–Bytes 65 through 96.
Line 4–Bytes 97 through 128 if the fourth line of print is called for.

The print buffer to be used for the print command is selected by setting bit 0 of the control code portion of the start I/O instruction to 0 for print buffer 1 and to 1 for print buffer 2.

The MFCU prints any of the 64 characters in the card code. Any of the characters in the 256 character EBCDIC set that is not included in the card code prints as a blank *without signaling the program.*

The rated throughput in print operations printing three lines is 60 cards per minute for the model A1, 120 cards per minute for the model A2. To maintain rated throughput, successive print operations must be initiated within 600 milliseconds (A1) or 300 milliseconds (A2) after the end of print data busy indication to the sense and test bits operation.

COMBINED OPERATION

Start I/O punch, print, read, or punch, print, feed operations proceed in the same manner as described for individual operations except that one card is fed from the wait station, punched into, and printed on before stacking. The next card is fed from the specified hopper into the wait station during punching. If read was specified, the data in the card is read into storage. To maintain rated throughputs, successive punch, print operations must be initiated within 20 milliseconds after the end of the later of punch busy or print data busy indicators.

STACKER SELECTION

Primary cards are selected to stacker 1 and secondary cards are selected to stacker 4 unless another stacker is specified. Stacker select is given by including the stacker select information in the start I/O control code of any of the start I/O instructions previously described. Stacker selection is performed on the card in the wait station when the start I/O instruction is executed, not the card that leaves the hopper. For programmed stacker select to operate, the stacker bit (bit 5) of the control code must be 1. Selection is made as specified under *Start I/O.*

CHECK CONDITIONS

Read Check

Reading is checked by reading each set of three tiers twice and checking that both readings are the same. When a read check occurs, the incorrect card (assuming that the read check was discovered by a sense I/O command before another start I/O instruction is executed) can be found in the wait station for the feed that produced the read check. The hopper that fed the card in error can be determined from the lights on the MFCU operator's panel.

Punch Check

Punching each hole in a column generates a signal that represents that hole. After the column has been punched, all the signals for holes in the column are compared with the character code specified for that column. If the two do not match, a punch check occurs. The card that contains the incorrectly-punched information moves to the stacker that was selected by the start I/O instruction that initiated the punch operation.

Punch Invalid

This condition occurs whenever the processing unit attempts to send an EBCDIC character that contains a C-bit or a D-bit to the MFCU for punching. The MFCU punches the B,A,8,4,2, and 1-bits, but not the C or D-bits, into the card for that character. Subsequent characters are punched into that card without punch-checking. The card containing the invalid character enters the stacker designated by the start I/O instruction that initiated the punch operation.

Print Data Check

An error in the synchronization between the print wheels and the MFCU attachment circuitry causes a print data check. The card in error is in the stacker selected by the start I/O operation that initiated the print operation.

Print Clutch Check

An error in the synchronization between the MFCU attachment circuitry and the printer stepper clutch causes print clutch check. The card in error is fed to the stacker designated by the start I/O instruction that initiated the print operation.

Hopper Check

Failure of a card to feed from the selected hopper causes a hopper check. The hopper that failed can be determined from the lights on the MFCU operator's panel. The card that failed to feed can be found in this hopper.

Either an IBM 5203 or an IBM 1403, but not both, can be used with the system.

IBM 5203 PRINTER

The IBM 5203 Printer (Figure 5-1) provides hard copy output from the system. This unit is also referred to as the line printer. The printer is available in three models.

Model 1 – 100 lines per minute
Model 2 – 200 lines per minute
Model 3 – 300 lines per minute

The standard print line is 96 print positions wide. Paper movement is controlled by the program. Interchangeability of type font, styles, or character arrangement is available on all models. All models come equipped with one interchangeable character set cartridge.

A variety of features are available to provide:

1. 120 print positions
2. 132 print positions
3. Dual feed carriage
4. Universal character set
5. Additional character set cartridges

The printer uses a type cartridge with 240 characters on the cartridge. The standard set of 48 characters, repeated five times on the cartridge, permits the rated throughput of 100, 200 or 300 lines per minute. The character set can be expanded from 48 to as many as 120 characters by using the universal character set special feature. However, when this feature is used, throughput will decrease depending on the text being printed.

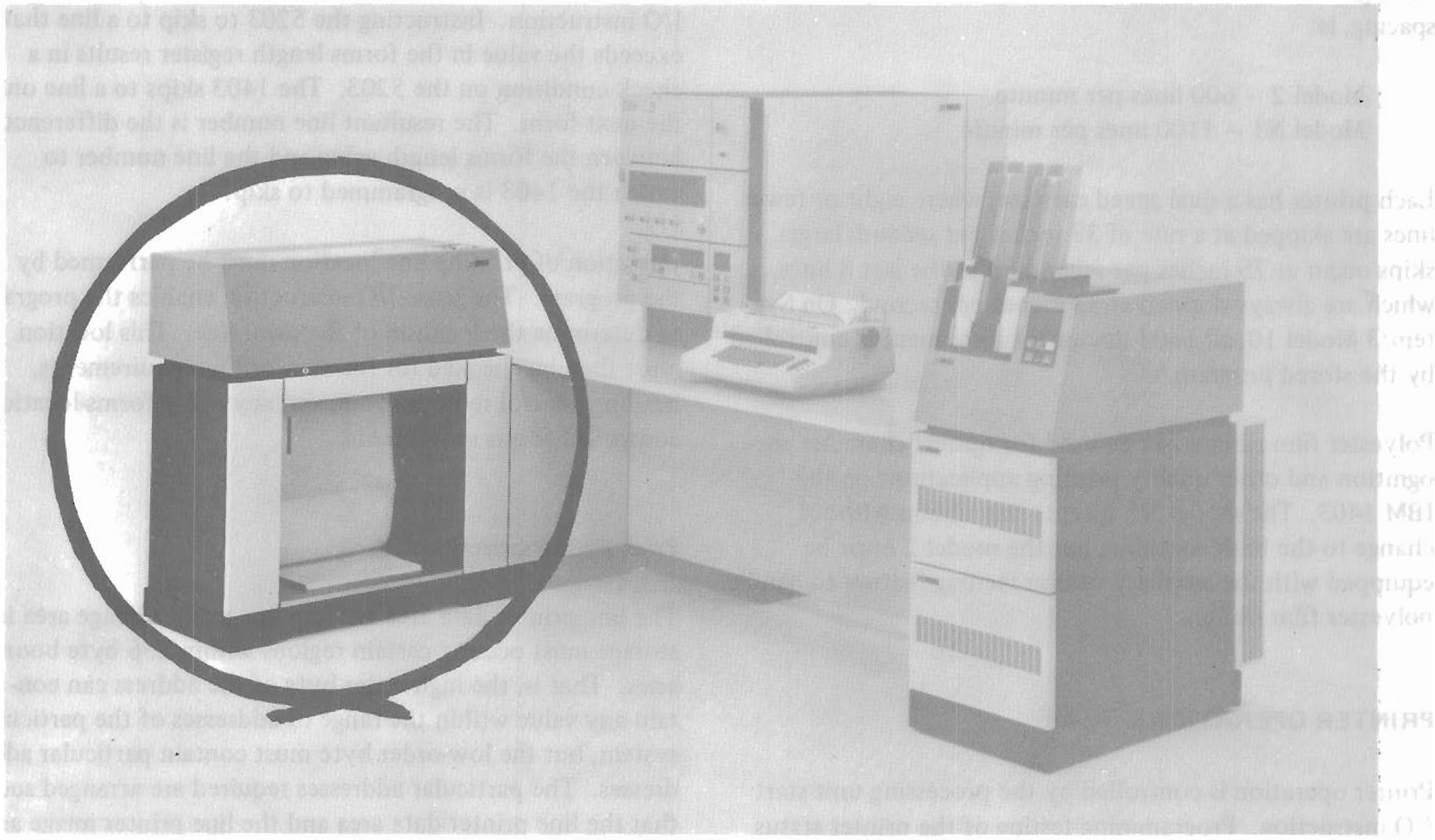


Figure 5-1. IBM 5203 Line Printer

5203 Operational Limitation

Because of its data transfer rate requirements the,5203 is subject to data overrun when its operations are overlapped with other devices in certain system configurations. This condition is not detected by the 5203 and may result in loss of data. Refer to Chapter 2, "Channel Limitations" for allowable overlapped device configuration that will not cause overrun in the system.

IBM 1403 PRINTER

The IBM 1403 Model 2 or Model N1 can be attached to the system via an IBM 5421 Printer Control Unit. Each model produces a print line with 132 print positions. The character set can be expanded from 48 characters (basic) to as many as 120 characters by using the universal character set special feature.

Model 2 requires an interchangeable chain cartridge adapter special feature for installation of the universal character set. Various type fonts, styles, and character arrangements are available.

The printers use a type cartridge with 240 characters. The standard set of graphics, repeated five times on the cartridge, permits the rated throughput of the standard models. Rated throughput, based on a 48-character set with single-line spacing, is:

Model 2 – 600 lines per minute
Model N1 – 1100 lines per minute

Each printer has a dual speed carriage, where eight or fewer lines are skipped at a rate of 33 inches per second; larger skips occur at 75 inches per second up to the last 8 lines, which are always skipped at 33 inches per second. On System/3 Model 10, all 1403 document movement is controlled by the stored program.

Polyester film ribbon can be used for optical character recognition and other quality printing applications on the IBM 1403. The model N1 accepts this ribbon without change to the basic machine, but the model 2 must be equipped with the auxiliary ribbon feeding feature to handle polyester film ribbon.

PRINTER OPERATIONS

Printer operation is controlled by the processing unit start I/O instruction. Programming testing of the printer status to establish program branch decisions is performed by the test I/O and branch instruction and the sense I/O instruction.

Output data flow to the printer is from an I/O area in storage, designated as the line printer data area. The program must fully prepare the area before issuing a start I/O line print instruction.

A character set image is defined as the sequence of print characters as they appear on the print cartridge. Before line printer operations are begun, a given character set image must be loaded in an I/O area in main storage, designated as the line printer image area, for reference by the line printer I/O attachment. Thus, line printer flexibility is achieved with the ability to alter the character set image at the printer via interchangeable cartridges and preloading of the altered character set image in storage.

The line printer image and data areas in main storage are specified by the programmer. Load I/O instructions are used to specify to the printer attachment the location of the image and data in storage.

Forms movement is also controlled by the start I/O instruction. Forms length must first be defined by a load I/O instruction. The maximum length of forms is 14 inches (112 spaces at 8 lines per inch or 84 spaces at 6 lines per inch spacing). Forms can be moved at either 6 lines per inch or 8 lines per inch. Spacing can be performed in increments of 0, 1, 2, or 3 lines. Skips can be any length up to the value established in the forms length register by the load I/O instruction. Instructing the 5203 to skip to a line that exceeds the value in the forms length register results in a check condition on the 5203. The 1403 skips to a line on the next form. The resultant line number is the difference between the forms length value and the line number to which the 1403 is programmed to skip.

Detection of printing line location must be performed by the program. The sense I/O instruction enables the program to determine the location of the print line. This location must then be checked for forms overflow requirements, heading control requirements, and any other forms-location-controlled forms movements.

Print Area Restrictions

The line printer data area and the line printer image area in storage must occupy certain regions within 256 byte boundaries. That is, the high-order byte of the address can contain any value within the range of addresses of the particular system, but the low-order byte must contain particular addresses. The particular addresses required are arranged such that the line printer data area and the line printer image area can (but are not required to) occupy regions within the same 256 byte area of storage. The following requirements must be met:

1. The 48-character set image must be in the 48 bytes having low-order address bytes of 00 through 2F.
2. The 120-character set image must be in the 120 bytes with low-order address bytes of 00 through 77.
3. The line printer data for 96 print positions (5203 only) must occupy the 96 bytes with low-order address bytes of 7C through DB.
4. The line printer data for 120 print positions (5203 only) must occupy the 120 bytes with low-order address bytes of 7C through F3.
5. The line printer data for 132 print positions must occupy the 132 bytes with low-order address bytes of 7C through FF.

The line printer data area in storage beginning at location XX7C corresponds character for character to the print line beginning at print position 1.

Dual Feed Carriage Print Considerations (5203 Only)

When dual feed carriage is installed, carriage instructions are referenced to the left and right carriages. When the dual feed carriage feature is not installed, only the left carriage commands are effective.

When dual feed carriage is used, a minimum of 17 positions is lost between the last character on the left form and the first character on the right form (assuming carrier strips are used).

For best print quality in dual feed carriage systems, the forms thickness should be the same in both carriages.

INSTRUCTIONS

Start I/O

Mnemonic: SIO

Op Code	Q Byte			
F3	E	M	N	Control Code

Operation: This instruction can initiate either or both forms movement and printing. If printing is specified, the data contained in the printer data area of storage is printed as a single line, beginning at the address specified in the line printer data address register. Unprintable characters and coded blanks (hex 40) print as blanks. Unprintable characters set a testable indicator and remain in the data area. All positions in which characters are printed are set to hex 40. If forms movement is specified, the printer spaces or skips to the next print line as specified by the Q byte.

The Q byte contains the device address (always E for the line printer), an M bit, and an N code. The M bit function depends on whether the printer is a 5203 or a 1403.

5203 M-bit: Controls carriage selection in systems with the dual carriage feature. An M bit of 1 refers to the right carriage. An M bit of 1 when the dual carriage feature is not installed results in a processor-check stop caused by invalid device address.

1403 M-bit: An M bit of 0 is required for normal print operations. An M bit of 1 normally results in a processor-check stop with the invalid Q-byte indicator on the register display unit lighted. However, if an M bit of 1, an N code of 1, and a control code of 80 is used, the attachment enters a diagnostic mode, requiring all diagnostic instructions to carry an M bit of 1. When the attachment is in the diagnostic mode, the 1403 is logically disconnected from the attachment.

The N code specifies the print, space, and skip functions as follows:

N Code	*Function*
000	Space only.
001	Invalid.
010	Print and space.
011	Invalid.
100	Skip only.
101	Invalid
110	Print and skip.
111	Invalid.

Specifying an invalid N code results in a processor-check stop caused by invalid Q code.

The third byte of the instruction is a control code that specifies the number of spaces a form is to be moved. For space operations the form is moved the number of spaces corresponding to the decimal value of the binary number in the control code. The control code must specify only 0, 1, 2, or 3 for spacing operations. A space control code of 4 or more results in a space 0 operation. In skip operations, the control code specifies the line number that is to end the skip. This number can be any number from 0 through 112. If the number exceeds the number of the last line of the form in a 5203, a check condition occurs. If the number exceeds the number of the last line on a 1403 form, the 1403 skips to a line equal to the specified destination less the forms length. A control code of 00 results in no carriage motion. A skip to a line number less than that at which the carriage is located results in a skip to the following page. A skip to the line at which the carriage is located results in no carriage motion.

A parity error detected by the attachment results in a processor-check stop and lights the DBO parity check light. The attachment will no-op the instruction and set the no-op status bit if a device error exists when the start I/O is executed.

If the printer is busy or intervention is required when the start I/O instruction is executed, the program loops on the start I/O instruction if the dual programming feature is not installed, or automatically program level advances if the dual programming feature is installed.

In a system using a 5203 with a dual feed carriage, a control instruction for a specific carriage will be accepted if that carriage is not busy, but execution is delayed until any printing from that or a previous instruction is completed. Forms motion of both carriages can be accomplished by giving a print and forms motion instruction to one carriage followed by a forms motion instruction to the other carriage.

The no-op indicator indicates that the last SIO instruction issued was accepted but was not executed because of a printer check condition. The no-op indicator is reset by a system reset, a system check reset, or an SNS instruction.

Programming Note: The first TIO for ready instruction issued after the no-op bit is set causes the program to branch. If the no-op bit is on, the program should issue the last SIO instruction used, because no data has been lost.

Example:

Instruction

F3	E6	16

The printer prints one line of information and skips to line 22 on the form.

Test I/O and Branch

Mnemonic: TIO

Operation: The printer attachment is tested for conditions specified in the Q byte. If the condition exists, the next instruction is taken from the address contained in the operand address portion of the instruction and the next sequential instruction address is placed in the address recall register.

If the condition does not exist, the next sequential instruction is used and the address from the operand address of the test I/O and branch instruction is placed in the address recall register. The address recall register will not then be changed until the next decimal, insert-and-test-characters, or branch instruction is executed.

The Q byte contains the device address (always E for the line printer), an M bit, and an N code.

The M bit function depends on whether the printer is a 5203 or a 1403.

5203 M-bit refers to the carriage in a dual feed carriage system. An M bit of 0 refers to the left carriage; an M bit of 1 refers to the right carriage. An M bit of 1 in a system without dual feed carriage results in a processor-check stop with an invalid device address indication.

1403 M-bit of 0 specifies a printer condition to be tested. An M bit of 1 with an N code of 001 specifies a test for diagnostic mode off; an M bit of 1 with any other N code is invalid.

The N code controls the condition tested by the instruction as follows:

N Code	*Condition*
000	Not ready/check.
001	Invalid.
010	Print buffer busy.
011	Invalid.
100	Carriage busy.
101	Invalid.
110	Printer busy.
111	Invalid.

The specification of an invalid N code results in a processor-check stop with an invalid Q code indication.

Not ready/check condition becomes active any time the printer becomes not ready for any reason. It becomes inactive when the reason for the not ready condition is removed.

Print buffer busy becomes active when the printer accepts a start I/O instruction that specifies printing. It becomes inactive when the line has been printed but before carriage motion stops.

Carriage busy becomes active when the printer accepts a start I/O instruction that specifies carriage motion. It becomes inactive when carriage motion stops.

Printer busy becomes active as soon as the printer accepts any start I/O instruction and becomes inactive when the instruction has been completely executed.

A parity error detected by the attachment results in a processor-check stop and lights the DBO parity check light.

Example:

Instruction

D1	EO	21

If the printer is not ready or has an error the next instruction will be taken from an address developed by adding Hex 21 to the contents of Index Register 1.

Advance Program Level

Mnemonic: APL

Operation: The printer is tested for conditions specified by the Q byte. If the condition exists, systems with the dual programming feature advance the program level; systems without the dual programming feature loop on the advance program level instruction until the tested condition no longer exists. If program level advance occurs, the reentry point of the program advanced from is the advance program level instruction.

The Q byte contains a device address (always E for the line printer), an M bit, and an N code.

The M bit function depends on whether the printer is a 5203 or a 1403.

5203 M-bit refers to the dual feed carriage feature. When the M bit is 0, the left carriage can be tested; when the M bit is 1, the right carriage can be tested. If an M bit of 1 is used when the dual feed carriage is not installed, a processor-check stop results with an invalid device address indication.

1403 M-bit of 0 specifies a printer condition to be tested. An M bit of 1 with an N code of 001 specifies a test for diagnostic mode off; an M bit of 1 with any other N code is invalid.

The N code defines the conditions to be tested as follows:

N Code	*Condition*
000	Not ready/check.
001	Invalid.
010	Print buffer busy.
011	Invalid.
100	Carriage busy.
101	Invalid.
110	Printer busy.
111	Invalid.

Specification of an invalid N code results in a processor-check stop with an invalid Q byte indication.

Not ready/check condition becomes active any time the printer becomes not ready for any reason. It becomes inactive when the reason for the not ready condition is removed.

Print buffer busy becomes active when the printer accepts a start I/O instruction that specifies printing. It becomes inactive when the line is printed but before carriage motion stops.

Carriage busy becomes active when the printer accepts a start I/O instruction that specifies a carriage operation. It becomes inactive when carriage motion stops.

Printer busy becomes active as soon as the printer accepts any start I/O instruction and becomes inactive when the instruction has been completely executed.

Program Note: The third byte of this instruction is not used. Care should be exercised in punching program cards to ensure that the op code byte for the following instruction is not inadvertently punched in the column that should be occupied by the third byte of this instruction.

Example:

Instruction

F1	EC	8A

The right carriage will be tested for carriage busy. If the carriage is busy, the other program level will become active.

Load I/O

Mnemonic: LIO

Op Code	Q Byte	Operand Address
Y1	E M N	

Operation: The contents of the two-byte field addressed by the operand address are transferred to the local storage register specified by the Q byte. The operand is addressed by its rightmost byte and remains unchanged. If the no-op status bit is set in the printer, the load I/O instruction is no-oped. If the addressed register is busy, systems with the dual programming feature will advance the program level; systems without dual programming feature will loop on the load I/O instruction until the register is not busy.

The Q byte contains the device address (always E for the line printer), an M bit, and an N code. The 5203 M bit has no significance for this instruction, but should be 0. The 1403 M bit either is used for customer engineering diagnostics or is invalid. It should always be 0 for non-diagnostic programming.

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

N Code	*Register*
000	Load forms length register.
001	Invalid.
010	Invalid.
011	Invalid.
100	Line printer image address register.
101	Invalid.
110	Line printer data address register.
111	Invalid.

Two bytes are loaded with each load I/O instruction for load forms length. The high-order byte is the forms length for the left carriage, and the low-order byte is the forms length for the right carriage. If the dual feed carriage feature is not installed, the low-order byte can contain any value. Specification of an invalid N code results in a processor-check stop with an invalid Q byte indication.

Program Note: End of page can be sensed only by programming.

Example:

Instruction

31	E4	2C	44

Operand

1F	00
2C43	2C44

Line Printer Image Address Register Before Operation

2A	CF

Line Printer Image Address Register After Operation

1F	00

Sense I/O

Mnemonic: SNS

Op Code	Q Byte	Operand Address
Y0	E M N	

Operation: The contents of the specified data source from the printer attachment are placed in a two-byte field at the storage location specified by the operand address. The operand is addressed by its rightmost byte. The Q byte specifies the data source. The sense I/O instruction is executed even if the printer is busy.

The Q byte contains the device address (always E for the line printer), an M bit, and an N code. The 5203 M bit is not used by this instruction but should be 0. The 1403 M bit is used for diagnostic programming and should be 0 for all non-diagnostic programming.

The N code specifies the sense data that is to be transferred by the sense I/O instruction as shown in Figure 5-2. Some of these conditions are useful only as diagnostic aids to the CE and will not be discussed here.

N Code *Data Transferred*

000 Carriage print line location counter. A test I/O instruction should be executed before sensing this condition to ensure that the carriage is not moving when the print line location is sensed.

(5203) 001 First byte contains an incrementing factor for the line printer data address register. The second byte contains (when printing) the value of the chain character counter, specifying the next print position to be addressed.

(1403) 001 Invalid code.

010 Printer timing. Used for diagnostics.

N Code *Data Transferred*

011 Printer check status. These bytes provide information concerning the kind of error that has occurred when an error check is detected as shown in Figure 5-3. The causes of the errors are discussed in *Error Conditions.* The bits labeled as CE bits are of no interest when operating the system.

100 LPIAR. This code selects the line printer image address register.

110 LPDAR. This code selects the line printer data address register.

Codes 101 and 111 are invalid. Selection of an invalid N code causes a processor-check stop and lights the invalid Q indicator.

N Code	Byte 1	Byte 2
000	5203 Right Carriage Line Location 1403 Character Count	5203 Left Carriage Line Location 1403 Carriage Line Location
001	5203 Chain Character Counter 1403 Invalid	5203 Incrementing factor of LPDAR 1403 Invalid
010	Printer Timing — Byte 1	Printer Timing — Byte 2
011	Printer Check Status — Byte 1	Printer Check Status — Byte 2
100	LPIAR — Low Byte	LPIAR — High Byte
101	Invalid	Invalid
110	LPDAR — Low-Order Byte	LPDAR — High-Order Byte
111	Invalid	Invalid

Figure 5-2. Line Printer Sense Data *

*Operand 1 address defines byte 1, operand 1 address minus 1 defines byte 2.

Bit	Byte 1	Byte 2
0	Chain Synchronization Check	Carriage Sync Check
1	5203 Incrementer Sync Check 1403 Not Used	5203 Carriage Space Check 1403 Not Used
2	5203 Thermal Check 1403 Not Used	Forms Jam Check
3	5203 Not Used 1403 Echo Check of Set Address	5203 Incrementer Failure Check 1403 Print Data Check
4	5203 Not Used (always on) 1403 Interlock Check	CE SNS Bit Latched
5	48-Character Set Chain	Hammer Echo Check
6	Unprintable Character	Any Hammer On Check
7	CE SNS Bit	No-op

Figure 5-3. Line Printer Check Status Bytes *

*Operand 1 address defines byte 1, operand 1 address minus 1 defines byte 2.

ERROR CHECKS

The following checks can be detected by the sense I/O instruction unless otherwise indicated. These checks light the printer check light or I/O attention light. They are reset by pressing the printer start key (check reset key on 1403) or the processing unit check reset key unless otherwise noted.

1. Incrementer failure check (5203 only). This check is caused by the incrementing hammer unit failing to move. Re-executing the last printer start I/O instruction will result in printing the remaining information on the line with no loss of data.
2. Incrementer.sync/slip check (5203 only). This check is caused by the incrementing hammer unit getting out of synchronism with the printer attachment or by a failing roller clutch in the incrementer cam.
3. Chain sync check. This check can be caused by the chain getting out of synchronism with the printer attachment. The data printed in error from these sync checks is no longer available because printing a character results in a blank being stored in that position of the printer data area.
4. Print Check. This is caused by either an echo check caused by the hammer circuitry not responding properly to a print signal or by an any hammer on check caused by a hammer being on outside of print time. Characters in the last line of printing may be incorrect. For the 5203, re-executing the last printer start I/O instruction results in printing all the characters not printed after the character in error. The character in error is replaced by a blank during the printing process. For the 1403, re-executing the last SIO reprints the last line without loss of any data.
5. Thermal check on the 5203. This check occurs because something overheated in the hammer unit. Processing can continue as soon as the hammer unit has cooled. Successive thermal checks indicate that the CE should be called.
6. Forms check. This is caused by the forms crumpling or tearing in the forms tractor area. The remainder of the last destroyed form will print on the new form. On the 1403 the forms check light also comes on whenever the carriage stop key is pressed.
7. Carriage check. This check occurs when loss of synchronism between the carriage and the attachment causes a carriage sync check. Skipping or spacing farther than the instruction called for causes a carriage space check in the 5203 and a sync check in the 1403.
8. Unprintable character. This is caused by a character in the printer data area that is not available on the chain. This check does not light the printer check light and is reset by the next start I/O print instruction or a system reset.
9. End of forms. This check does not have a status bit. It is indicated by the I/O attention and forms lights.
10. Interlock conditions. These conditions do not have status bits. They are indicated by the I/O attention and interlock lights. On the 1403, interlock conditions are indicated by the I/O attention and the print check light or the forms check light. An internal indicator panel shows the appropriate interlock condition.
11. Print data checks (1403 only). A parity error condition has been detected during data access from the print buffer during a print operation.

5

Channel Overrun Considerations

If a System/3 has a 5444 or 5445 attached and is not using IBM program products, certain system configurations can result in I/O channel overruns and resulting possible loss of data. Non-overrunable configurations are shown under the heading "Channel Limitations" in the second chapter, "5410 Processing Unit."

IBM 5444 Disk Storage Drive

The IBM 5444 Disk Storage Drive (Figure 6-1) provides the system with a large capacity, direct access storage device with capacities ranging from 2,457,600 bytes through 9,830,400 bytes.

The 5444 is available in six models:

Model	Tracks/ Surface*	No. of Disks**	Total Capacity (in bytes)	Avg Access Time
1	104	2	2,457,600	153 ms
A1	104	2	2,457,600	86 ms
2	204	2	4,915,200	269 ms
A2	204	2	4,915,200	126 ms
3	204	1	2,457,600	269 ms
A3	204	1	2,457,600	126 ms

*IBM resident control program requires one track per surface for customer engineers. IBM disk systems programming support requires three tracks per surface for alternate data tracks. Systems using these programs are therefore limited to 100 or 200 tracks per surface, according to the model selected.

**Each model has one removable disk (Figure 6-2). Models 1, A1, 2, and A2 also have 1 permanent disk. Both surfaces of each disk are used.

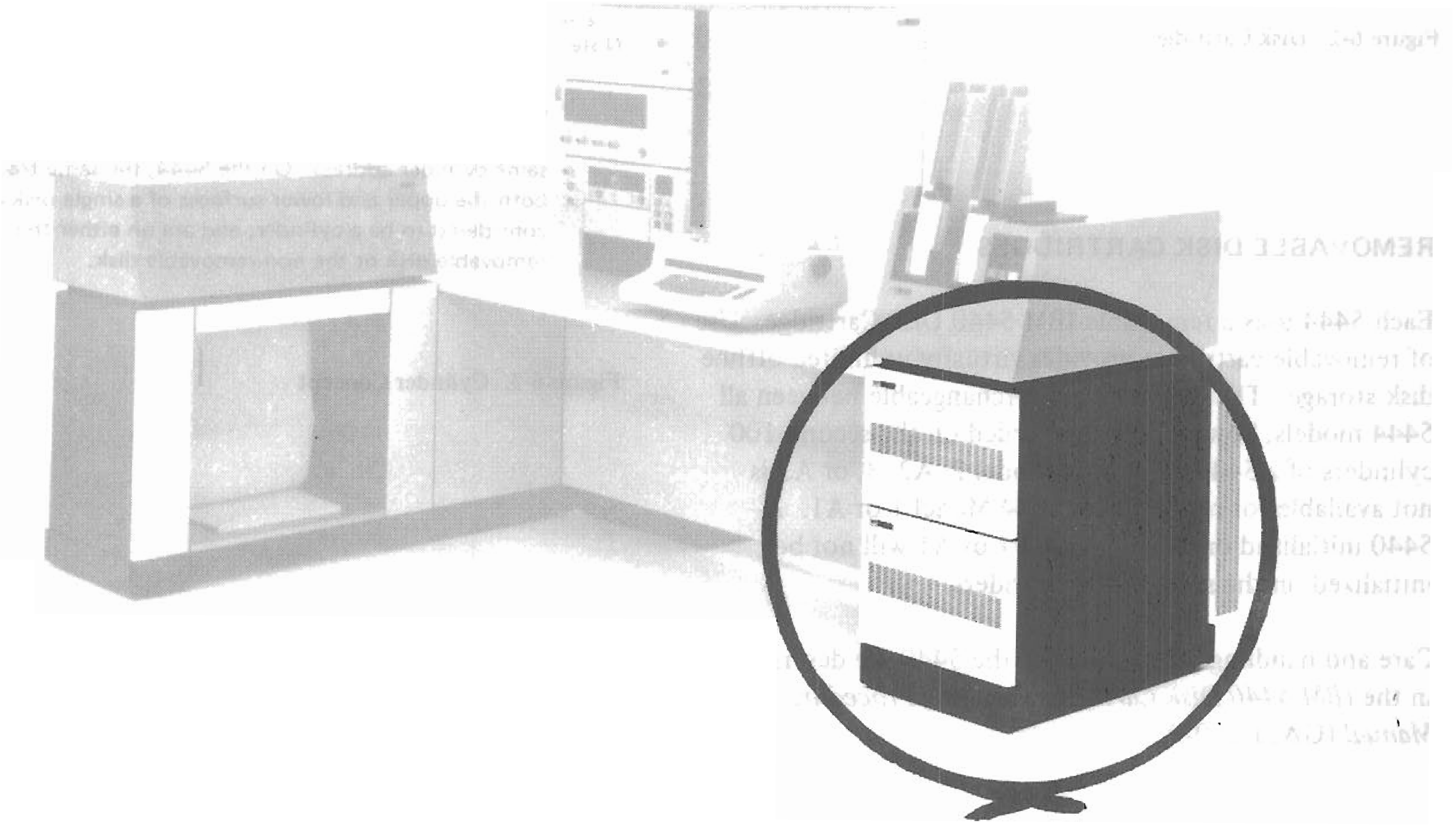


Figure 6-1. IBM 5444 Disk Storage Drive

The 5444 can be ordered with the following configurations of models:

Standard Speed Access

- 1 Model 1
- 1 Model 2
- 2 Model 2s
- 1 Model 2 and 1 Model 3

High Speed Access

- 1 Model A1
- 1 Model A2
- 2 Model A2s
- 1 Model A2 and 1 Model A3

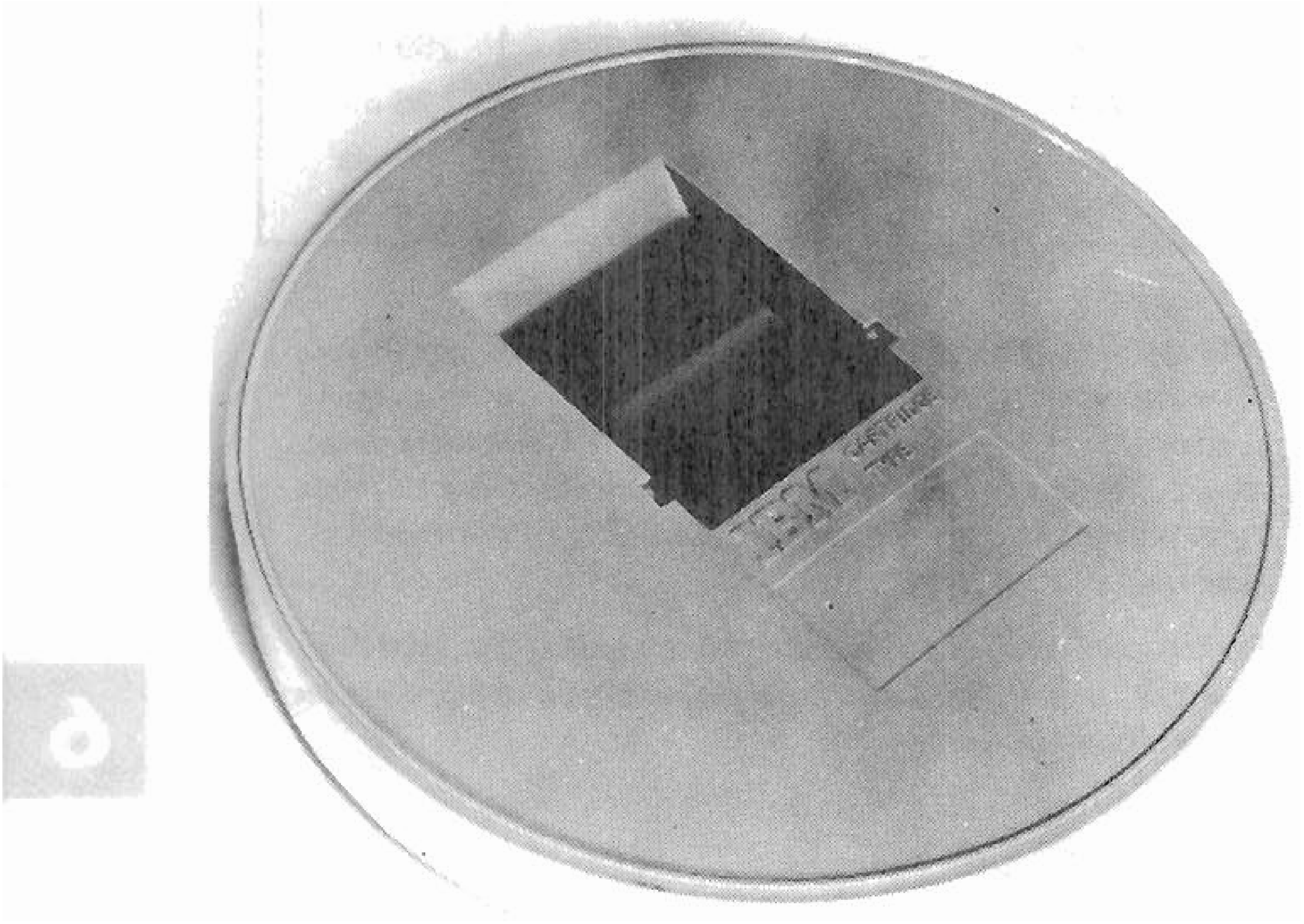


Figure 6-2. Disk Cartridge

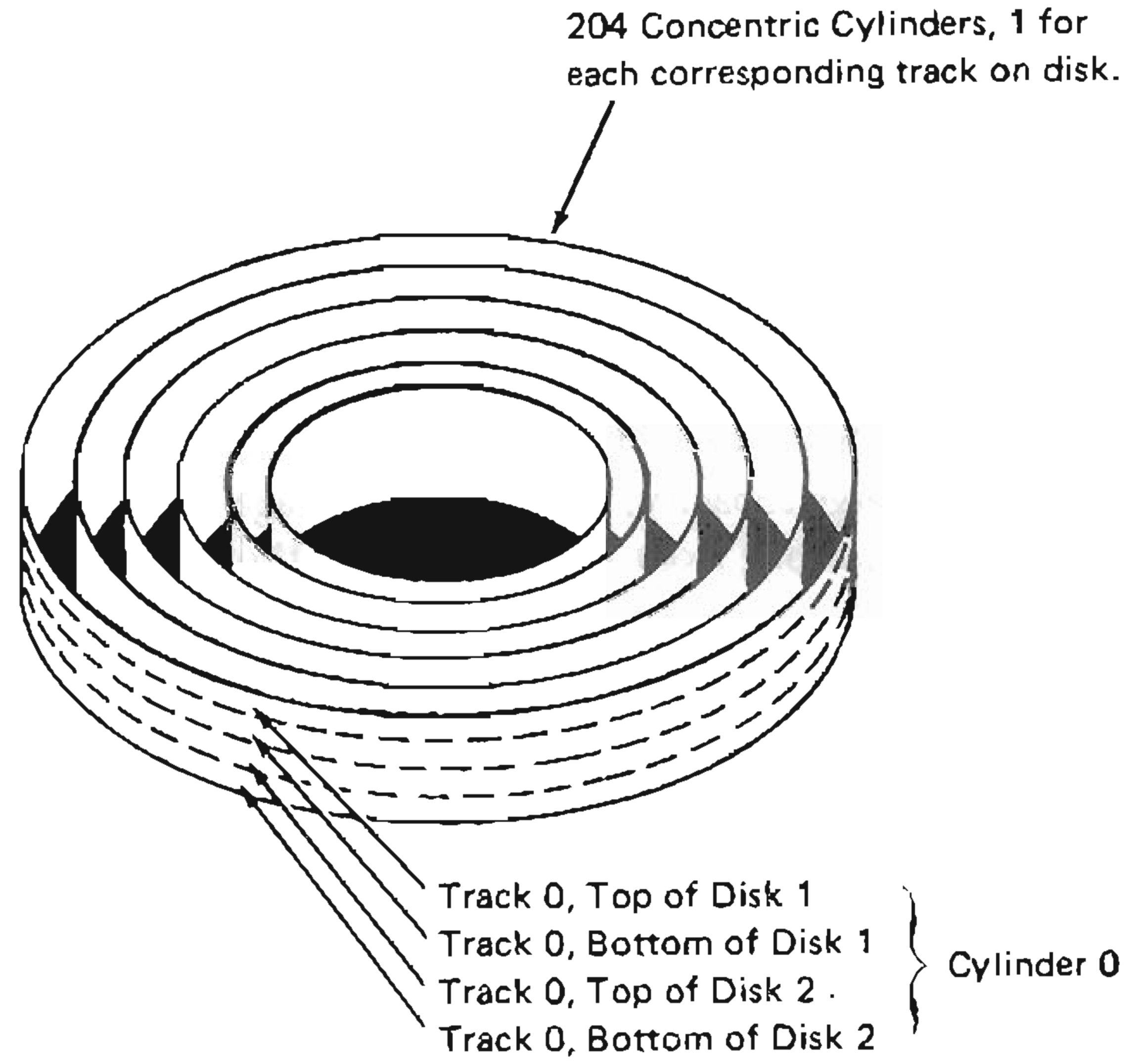
REMOVABLE DISK CARTRIDGES

Each 5444 uses a removable IBM 5440 Disk Cartridge. Use of removable cartridges provides virtually unlimited offline disk storage. The cartridge is interchangeable between all 5444 models; however, data recorded on the second 100 cylinders of a 5440 by a 5444 Model 2, A2, 3, or A3 is not available for reading by a 5444 Model 1 or A1. A 5440 initialized on a 5444 Model 1 or A1 will not be initialized on the second 100 cylinders.

Care and handling procedures for the 5440 are described in the *IBM 5440 Disk Cartridge Handling Procedure Manual* (GA21-1598).

5444 DISK ORGANIZATION

Each surface of each disk contains either 104 or 204 tracks. The tracks that are related to each other in the vertical plane on a single disk are considered to form a cylinder as shown in Figure 6-3. On drives with two disks the corresponding cylinders on both disks have the same cylinder number.


204 Concentric Cylinders, 1 for each corresponding track on disk.
Track 0, Top of Disk 1
Track 0, Bottom of Disk 1
Track 0, Top of Disk 2
Track 0, Bottom of Disk 2
Cylinder 0

Note: The same cylinder address is used for all corresponding tracks on the disks. For example, track 15 on both top and bottom of disks 1 and 2 are all considered to be bands of data on one cylinder, so all four bands have the same cylinder address. On the 5444, the same track on both the upper and lower surfaces of a single disk are considered to be a cylinder, and are on either the removable disk or the non-removable disk.

Figure 6-3. Cylinder Concept

5444 Track Format

Each track is divided into 24 sectors as shown in Figure 6-4. Each sector has its own individual address. A sector is made up of an address marker, sector identifier, data field, and some gaps.

Index Marker — A mark that is fixed for each disk and provides orientation information to the controlling circuits. It is the starting point for every track.

AM — Address marker is a specially written group of bits used to indicate the start of a new sector.

ID — The sector identifier. This group of six bytes contains three bytes for unique identification of that sector for that disk, and three bytes of check characters.

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

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

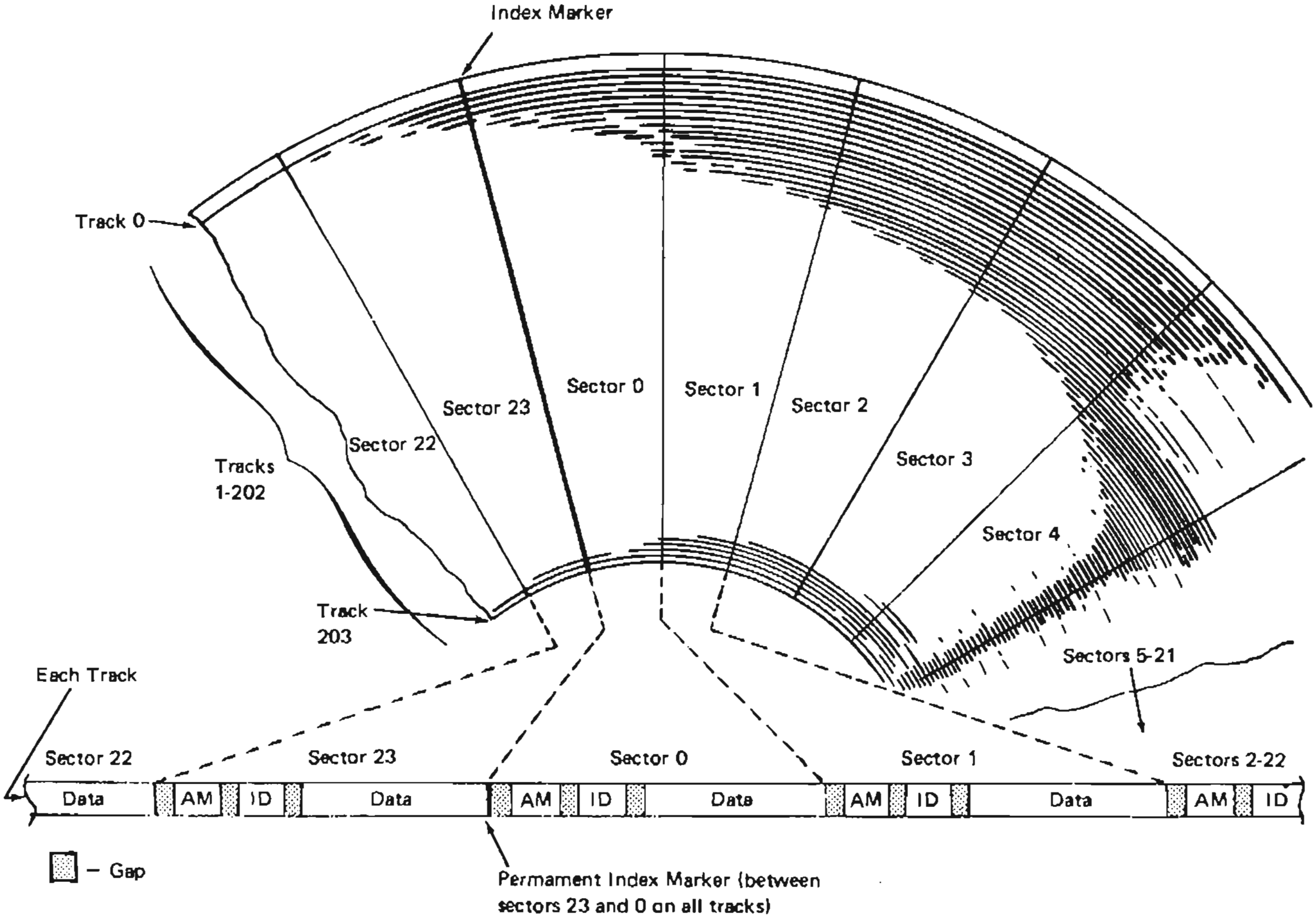


Figure 6-4. Sector Layout

5444 SECTOR IDENTIFIER FORMAT AND ADDRESSING

The identifier area of a sector (ID) contains a flag byte, two bytes of address information, and three bytes of check information as shown below.

Flag	Address		Check Characters		
F	C	S	CC	CC	BCA

F Flag byte. This byte contains the flagging information in bits 6 and 7. All other bits in this byte should be 0 (see "Flagging").

C Cylinder byte. This byte contains the binary number that corresponds to the physical location of the track on the disk.

S Sector byte. The six leftmost bits in this byte hold the binary number of the sector. Sectors on top of the disk have sector numbers from 0 through 23. Sectors on the lower surface of the disk have sector numbers from 32 through 55.

CC Cyclic check. These bytes are automatically generated and used for checking purposes.

BCA Bit count appendage. Another automatically generated checking byte.

The address of any individual sector is contained in the second and third bytes of the identifier. Sectors occupying the same physical location on the fixed disk and on all of the removable disks *have identical binary numbers in the cylinder and sector bytes.* Use of a sector requires that the drive (1 or 2) and the disk (fixed or removable) containing the desired sector must be specified.

Cylinders are numbered 0 through 203, counting from the outer cylinder. IBM customer engineers use cylinder 203 for diagnostic functions, so this cylinder should not be used for permanent storage. Tracks in cylinders 1, 2, and 3 are used by IBM program products as alternate tracks whenever tracks in cylinders 1 through 202 are found to be defective; therefore, if IBM program products are being used, cylinders 1, 2, and 3 are to be reserved for use as alternate cylinders. Cylinders 0 and 4 through 202 can always be used as standard data cylinders.

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

5444 DISK OPERATING RESTRICTIONS

1. The disk drive drawers cannot be opened unless system power is on and the disk start/stop switch on the system control panel is in the stop position. The OPEN light on the system control panel will light when it is safe to open the drawers. We recommend that the drawers be kept closed at all times unless a disk cartridge is being inserted or removed. A cartridge should always be stored on the drive to prevent dust from entering the drive.
2. The 5440 disk cartridge must be stored in the operating environment for at least two hours before the cartridge is used for processing.

5444 DISK OPERATIONS

Two things must be done to prepare for each disk operation. The address of the disk control field must be stored in the disk control address register and the address of the first byte of the disk data field must be stored in the disk read/write address register (see "Local Storage Registers").

Disk Control Field

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

Byte *Use*

F This is the first byte in the field and the byte addressed by the disk control address register. In seek operations this byte is not used. In other disk operations it contains flag bits in bits 6 and 7.

C This second byte of the field contains a binary number that designates a cylinder number. This byte is not used on a seek operation.

S The function of this byte (the third byte in the field) depends on the operation to be performed:

Seek Operation: Bit 0 selects the head to be used (0 = head 0, for upper surface; 1 = head 1 for lower surface).
Bits 1 through 6 are not used.
Bit 7 selects direction of seek (0 = toward decreasing cylinder numbers; 1 = toward increasing cylinder numbers).

All Other Operations: Bits 0 through 5 hold the binary representation of the sector ID number.
Bits 6 and 7 are not used; bit 7 must be 0.

N This last byte in the field specifies either the number of cylinders to move the access mechanism for a seek operation or the number of sectors to operate on for any other operation. For operations other than seek, this binary number must be one less than the actual number of sectors desired. For example, if one sector is to be handled, the N byte must hold a zero; if ten sectors are to be acted on (a multiple-sector operation), the N byte must hold 9.

5444 Seek Operation

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

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

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

The recalibration function is executed by specifying a seek in the reverse direction and a number of cylinders to be moved that is greater than or equal to 224. The recalibrate function causes the access mechanism to return to cylinder 0 and selects read/write head 0, regardless of the condition of bit 0 of the S byte.

Note: On high performance disk drives, recalibration should be used only for error recovery, because recalibration forces a low speed seek in a reverse direction.

6

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

Seek operation is begun by issuing an SIO instruction. A second SIO instruction can be issued to the same disk drive if a read, write, or scan operation is specified. The second instruction will be accepted provisionally and executed if no errors occur in the operation of the seek instruction. A subsequent SIO instruction to either disk causes the CPU to loop on that instruction until the read, write, or scan operation ends. However, seek commands to both drives can be executed concurrently if there is no intervening SIO read, write, or scan instruction.

No data in storage will be changed by this operation. Test I/O for busy or advance program level on busy will not detect busy unless a read, write, or scan instruction has been provisionally accepted. The sense bit for seek busy will be on, however, for interrogation by the sense I/O instruction. A seek instruction to an access mechanism that is already seeking results in an automatic program level advances if the dual programming feature is installed.

A seek to the cylinder at which the access mechanism is located is completed immediately because no access mechanism motion is required. However, the head is selected according to bit 0 of the S byte.

5444 Access Time

Access time is the interval from the receipt of a seek command until read/write head movement stops.

Access Time for Models 1, 2, and 3

Figure 6-5 shows the approximate time required to seek across any number of tracks from 1 to 200. Access time can also be determined from the following formula:

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

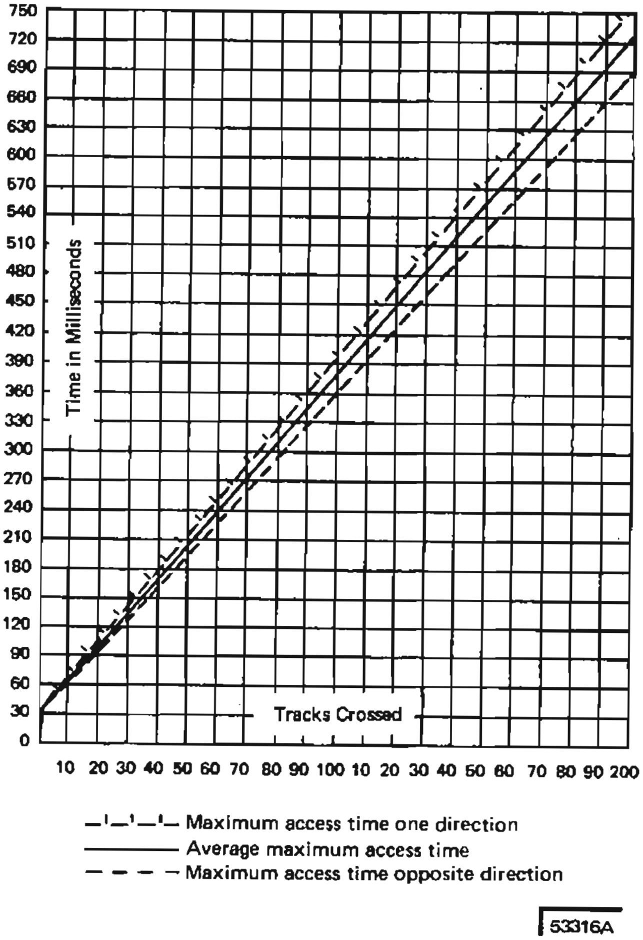


Figure 6-5. Access Timing (Models 1, 2, and 3)

Access Time for Models A1, A2, and A3

For the high performance disk storage drive models, access times are not necessarily the same for both forward and reverse seek operations. The more important access times (for both forward and reverse directions) for these models of the 5444 are:

- The access time for a one-cylinder movement is 28 milliseconds for all three models.
- The normal average access time for model A1 is 86 milliseconds; for models A2 and A3, the normal average access time is 126 milliseconds. This is the average access time across 67 cylinder addresses with the exception of when a forward seek terminates in cylinder address 170 through 203.
- The maximum access time (the time taken for the access mechanism to cross the maximum number of cylinders available on each model) is 163 milliseconds for 99 cylinders on model A1; for models A2 and A3 the access time is 255 milliseconds for 199 cylinders.

To Determine Approximate Maximum Access Forward Time: Access times for access forward operations are not dependent solely on the number of cylinders traveled, but also depend on where the access forward operations terminate. For this reason, no simple graph can be drawn showing access times for all possible access forward operations.

Figure 6-6 shows maximum access time curves for access forward operations starting from several different cylinder addresses. Each curve is labeled with its appropriate starting cylinder address. Intermediate values may be determined by interpolation.

To determine the access time for any forward operation follow the curve corresponding to the starting cylinder address until the curve coincides with the cylinder address that is being accessed (horizontal axis). The corresponding access time is then read from the vertical axis in milliseconds. For example, to determine the access time for an access operation from cylinder address 040 to cylinder address 120, follow the curve corresponding to cylinder address 040 until the curve is aligned with cylinder address 120 on the horizontal axis. The required access time indicated on the vertical axis is 140 milliseconds.

If more accurate information is required, refer to "Maximum Access Forward Calculations".

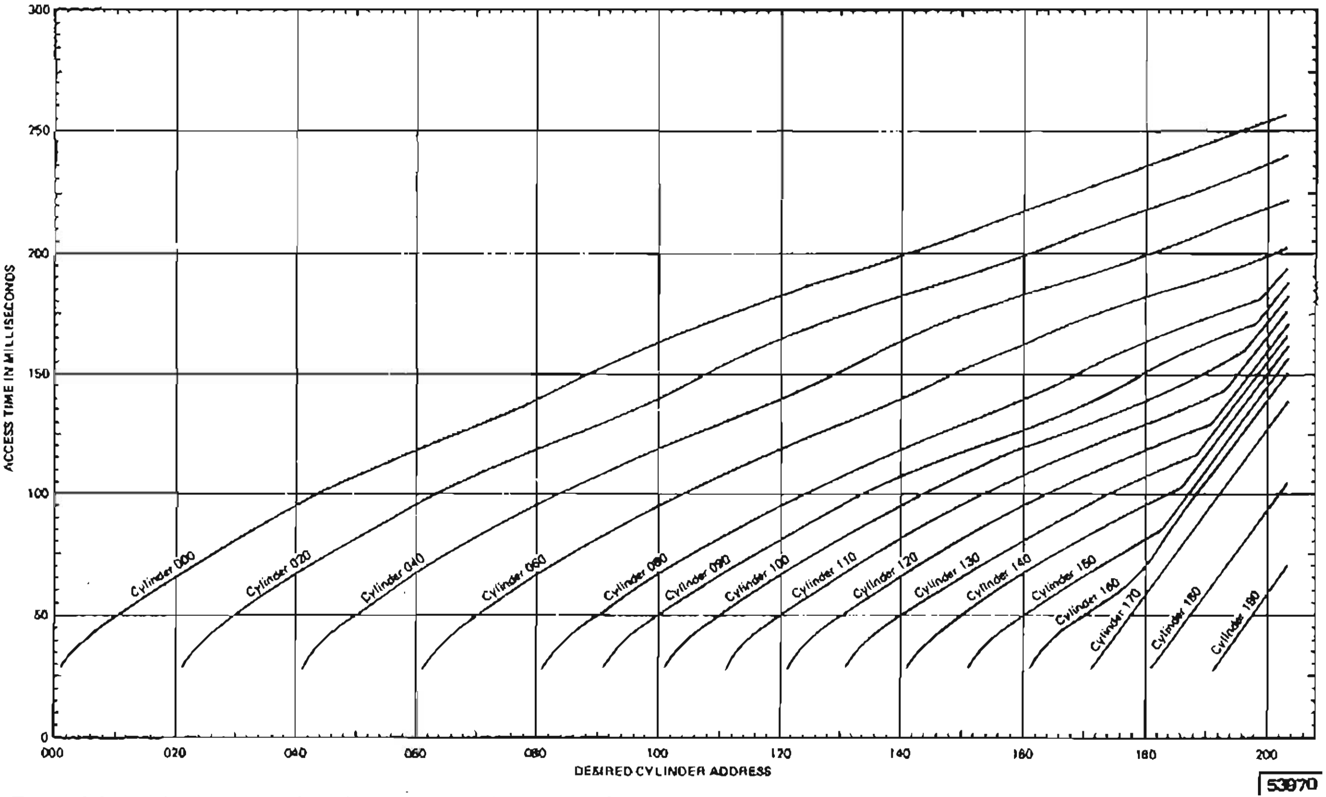


Figure 6-6. Maximum Access Time for Models A1, A2 and A3 (Forward Direction)

6

To Calculate Maximum Access Forward Time: Approximate maximum access times for access forward operations (models A1, A2, and A3) are shown in Figure 6-7. However, some access forward operations finishing above cylinder address 170 have access times greater than that indicated in Figure 6-7. (Reverse operations are not affected; see "Access Reverse Operations".)

Figure 6-8 shows all possible access operations. The chart is divided by a diagonal line into two regions covering access forward operations and access reverse operations. The access forward region is further sub-divided into three areas.

To determine the maximum access time for any access forward operation,find the point where the from cylinder address (horizontal axis) and the to cylinder address (vertical axis) intersect. The area in which this point of intersection occurs defines how the access time is calculated,as follows:

1. Unshaded area—access time determined directly from Figure 6-7.

2. Shaded area—access time determined by Figure 6-7 plus an additional time as indicated by the chart (Figure 6-9).

3. Cross-hatched area—access time shown in Figure 6-10.

For example, to determine the maximum access time for an access operation from cylinder address 150 to cylinder address 200:

1. From Figure 6-8 locate the point of intersection of the present cylinder address (cylinder address 150) and the new cylinder address (cylinder address 200). The point of intersection is in the shaded area.

2. Figure 6-7 shows that maximum access time for a 50-cylinder address difference is 107 milliseconds.

3. Figure 6-9 shows that the additional time that must be added is 39 milliseconds.

4. Therefore, the total maximum access time for this access operation is 146 milliseconds (107 + 39 = 146 milliseconds).

Access Reverse Operations: Figure 6-7 shows the maximum access time for the number of cylinders that the access mechanism crosses during an access reverse operation.

Note: Ready may be dropped if an access reverse operation specifies more tracks than the actual number of tracks from the present track to the home position. If ready is dropped and no permanent hardware fault exists, stop the disk drive and then restart the disk drive to establish a file ready condition.

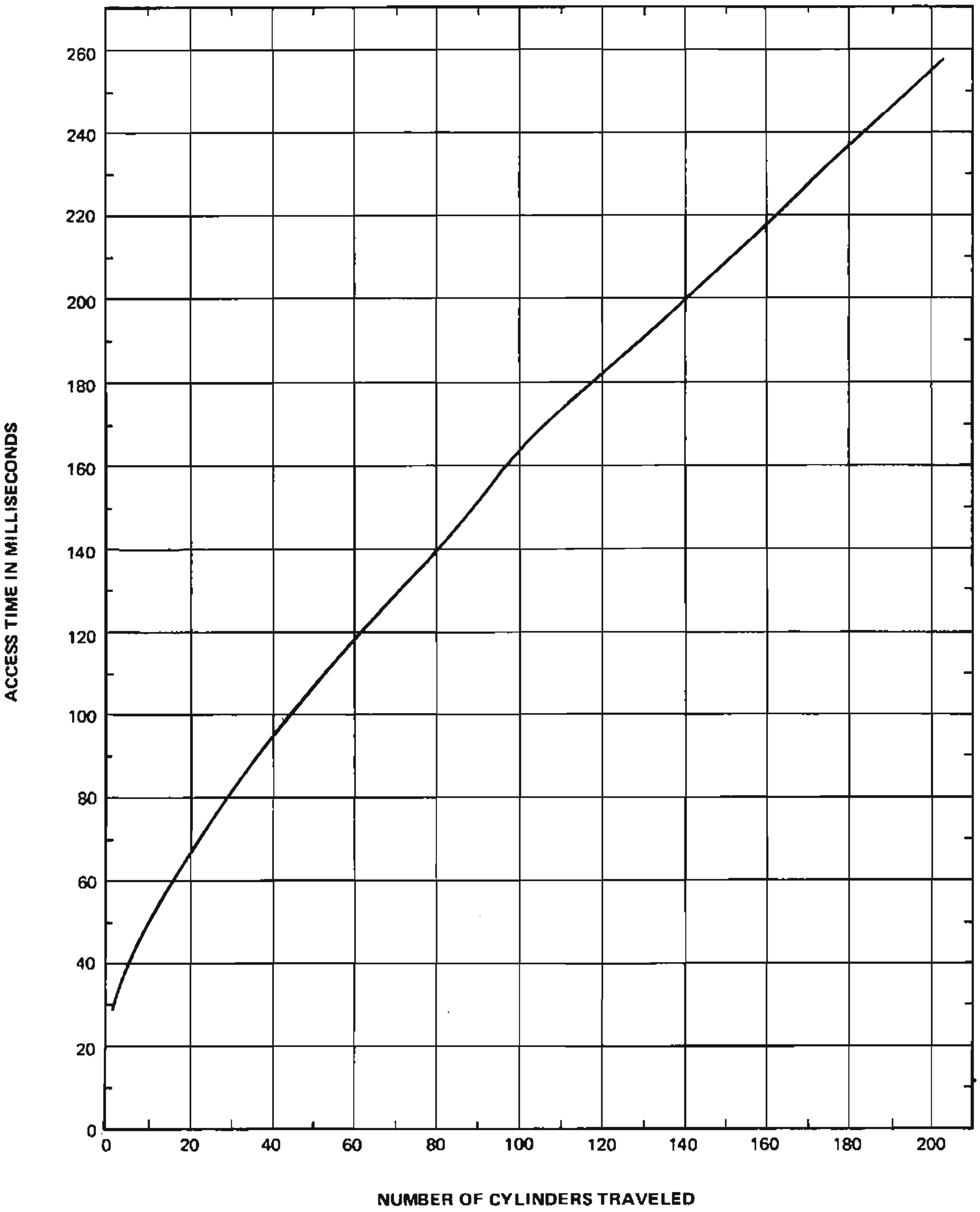


Figure 6-7. Maximum Access Times for Models A1, A2 and A3 (Forward or Reverse Direction)

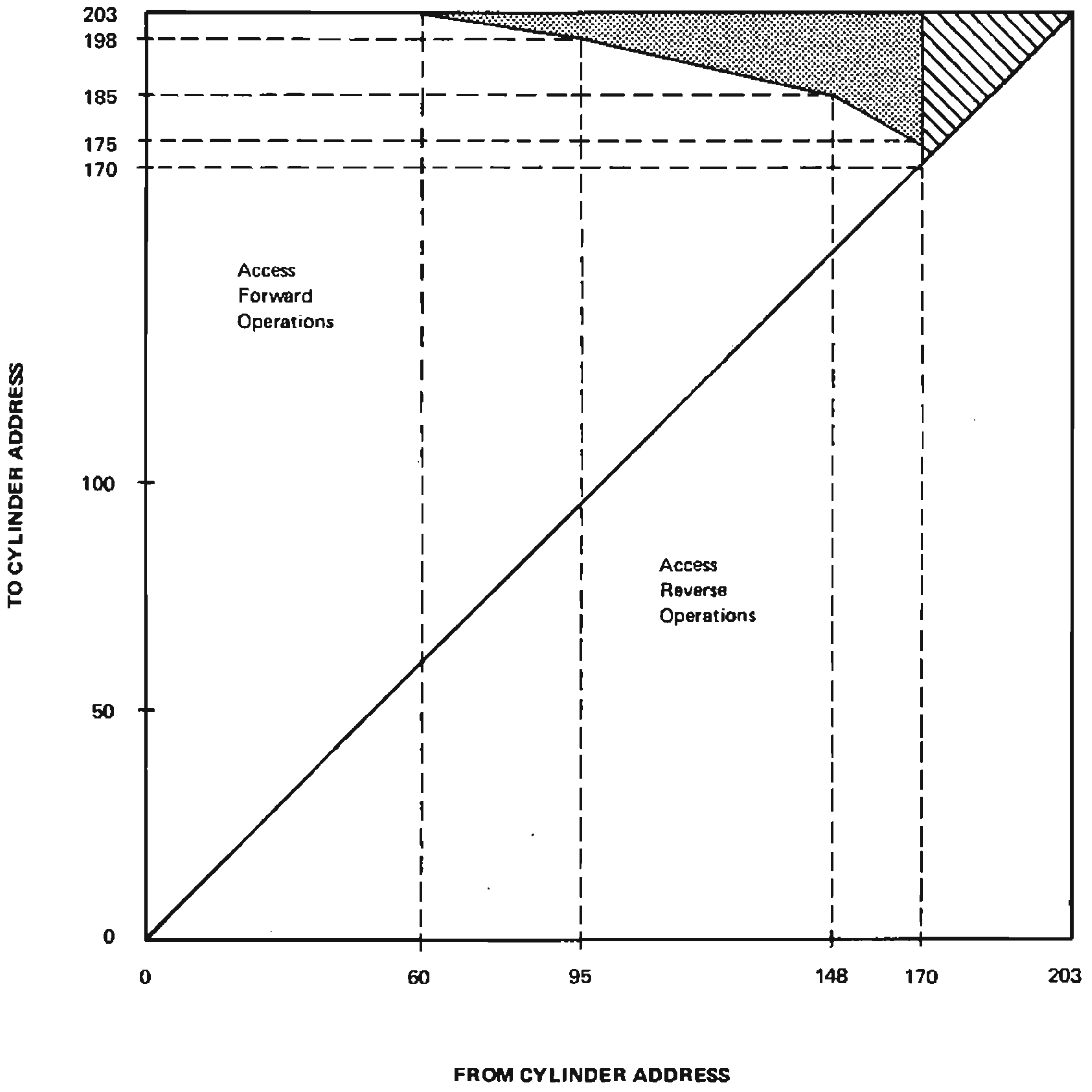


Figure 6-8. Access Time Chart for Models A1, A2 and A3

Note: The "From Cylinder Address" axis is not continuous in order to reduce the size of the chart. If the required "From Cylinder Address" is not listed, use the next higher "From Cylinder Address". In some cases this will mean that the "Additional Access Time" obtained from the chart is a maximum of 3 msec greater than the true "Additional Access Time".

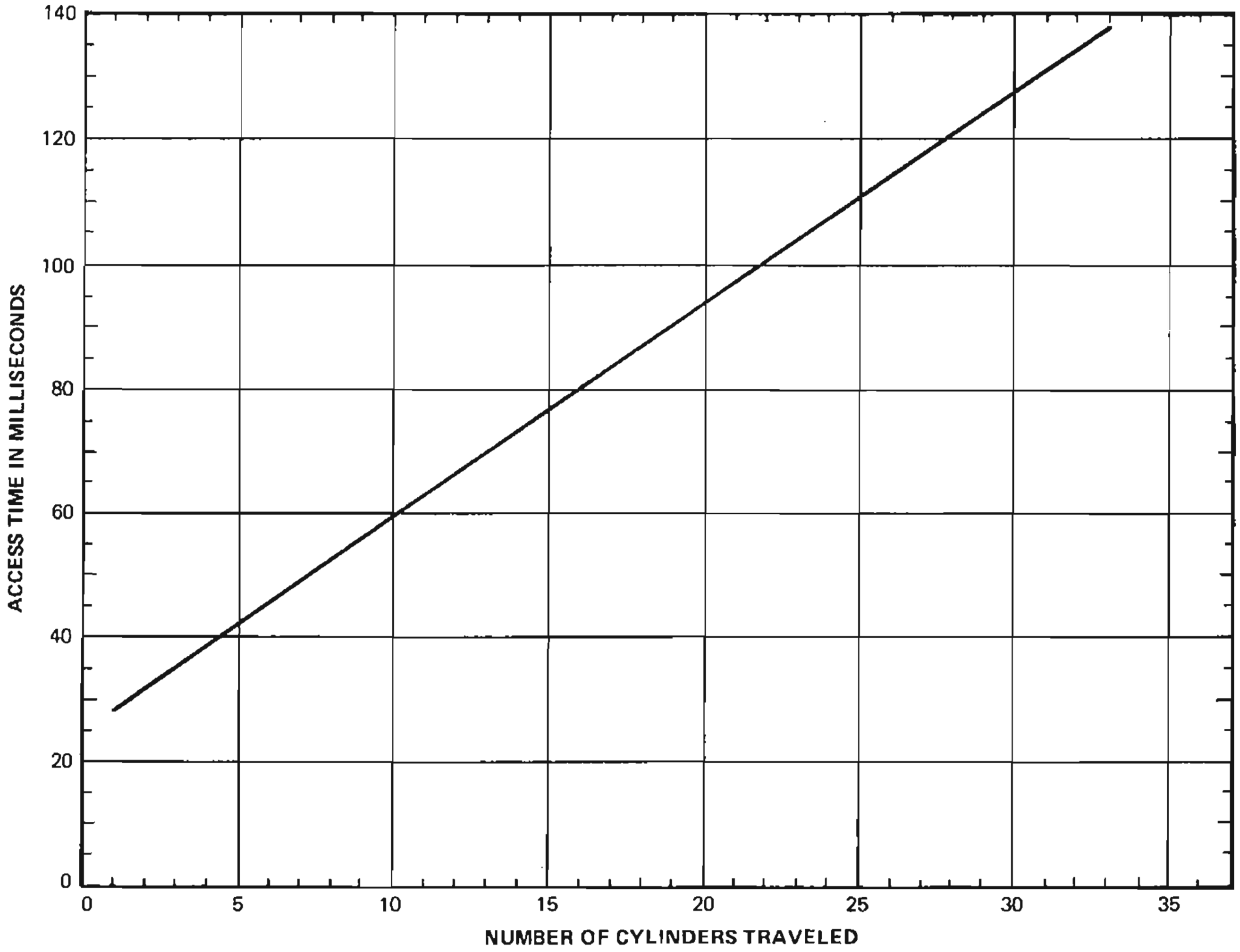
The chart specifies the number of milliseconds that must be added to the access time given by the general curve in Figure 6-8, for the range of cylinder addresses indicated.

FROM CYLINDER ADDRESS

TO CYLINDER ADDRESS

	60	68	76	84	92	96	100	104	108	112	116	120	124	128	132	136	140	144	148	150	152	154	156	158	160	162	164	166	168	169
203	3	3	6	9	12	15	15	18	21	24	27	27	30	33	36	39	39	42	45	45	48	48	51	51	51	54	54	57	57	60
202	0	3	3	6	9	12	15	15	18	21	24	27	27	30	33	36	39	39	42	45	45	45	48	48	51	51	51	54	57	57
201		0	3	3	6	9	12	15	15	18	21	24	27	27	30	33	36	39	39	42	42	45	45	45	48	48	51	51	54	54
200			0	3	3	6	9	12	15	15	18	21	24	27	27	30	33	36	39	39	39	42	42	45	45	45	48	51	51	54
199				0	3	3	6	9	12	15	15	18	21	24	27	27	30	33	36	36	39	39	39	42	42	45	45	48	51	51
198					0	3	3	6	9	12	15	15	18	21	24	27	27	30	33	33	36	36	39	39	39	42	45	45	48	48
197						0	3	3	6	9	12	15	15	18	21	24	27	27	30	33	33	33	36	36	39	39	42	45	45	48
196							0	3	3	6	9	12	15	15	18	21	24	27	27	30	30	33	33	33	36	39	39	42	45	45
195								0	3	3	6	9	12	15	15	18	21	24	27	27	27	30	30	33	33	36	39	39	42	42
194									0	3	3	6	9	12	15	15	18	21	24	24	27	27	27	30	33	33	36	39	39	42
193										0	3	3	6	9	12	15	15	18	21	21	24	24	27	27	30	33	33	36	39	39
192											0	3	3	6	9	12	15	15	18	21	21	21	24	27	27	30	33	33	36	36
191												0	3	3	6	9	12	15	15	18	18	21	21	24	27	27	30	33	33	36
190													0	3	3	6	9	12	15	15	15	18	21	21	24	27	27	30	33	33
189														0	3	3	6	9	12	12	15	15	18	21	21	24	27	27	30	30
188															0	3	3	6	9	9	12	15	15	18	21	21	24	27	27	30
187																0	3	3	6	9	9	12	15	15	18	21	21	24	27	27
186																	0	3	3	6	9	9	12	15	15	18	21	21	24	24
185																		0	3	3	6	9	9	12	15	15	18	21	21	24
184																			0	3	3	6	9	9	12	15	15	18	21	21
183																				0	3	3	6	9	9	12	15	15	18	18
182																					0	3	3	6	9	9	12	15	15	18
181																						0	3	3	6	9	9	12	15	15
180																							0	3	3	6	9	9	12	12
179																								0	3	3	6	9	9	12
178																									0	3	3	6	9	9
177																										0	3	3	6	6
176																											0	3	3	6
175																												0	3	3
174																													0	0

Figure 6-9. Additional Access Time Chart (Models A1, A2 and A3)



For a one-cylinder access:

Maximum Access Time = 28 Milliseconds

For an access of more than one cylinder address above cylinder 170:

Maximum Access Time in Milliseconds = 32 + 3.42 (N-2)

$2 \leq N \leq 33$

N = Number of tracks to be crossed

53971

Figure 6-10. Maximum Access Times for Accesses Above Cylinder 170 (Forward Direction)

5444 Read Data Operation

This instruction initiates the transfer of data from the selected disk to main storage. Data is transferred in multiples of 256 bytes (the contents of an individual disk sector).

If reading is started at sector 0, all 48 sectors from corresponding upper and lower tracks on the same disk can be read as the result of a single read operation. Only consecutive sectors are read when multiple sector reading is indicated.

Reading begins with the sector specified by the S byte of the disk control field in main storage. (Bit 0 of the S byte for this instruction does not select the head, but is used for comparison only. Head selection can be accomplished only by a seek operation.) The data is transferred to CPU storage, starting at the CPU storage address specified by the disk data address register. Succeeding bytes are stored in progressively higher locations, because the 5444 automatically updates the disk data address register so that it points to the storage address where the next byte of data is to be stored.

When the disk control field specifies that more than one sector is to be read, the 5444 automatically updates the S byte of the disk control field each time a sector is read so that it contains the address of the next higher sector on that cylinder and disk. After the 5444 has read sector 23 and stored its data, the 5444 automatically switches heads to read sector 32 from the associated track on the lower surface of the disk (the other track on the same cylinder on that disk). The read operation then continues. (Sector addresses cannot overflow from disk to disk because each disk contains identical addresses for common cylinders—that is, for cylinders with the same track number.

During read operations, the 5444 compares the disk control field with the sector identifier fields on the disk track to find the first sector to be read. The comparison is repeated for each additional sector to be read. If the disk control field and the sector identifier fail to match, the operation terminates after the data portion of that sector is transferred to main storage even if other sectors remain to be read.

Two other abnormal conditions cause termination of the reading operation. Reading will be terminated at the end of any sector in which an error is detected or if the sector read is the last sector (sector 55) in the cylinder.

During a read operation, the attachment generates two cyclic check (CC) bytes and a one-byte bit count appendage from the data that has been read and compares these to the CC bytes and bit count appendage read back with the data, providing a data check for read errors. During multiple sector reads, the operation will be terminated at the end of any sector in which an error is detected except that an equipment check causes immediate termination. The data portion of the error sector is stored in storage and the 5444 disk data address register is updated.

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

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

The disk control unit will be busy to all other operations except sense I/O during a read data operation.

5444 Read Identifier Operation

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

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

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

At the end of the operation, the disk control address register contains the original address unless there is an equipment check. With an equipment check, the contents of the register may or may not contain the original address.

5444 Read Data Diagnostic Operation

This operation is similar to a read data operation. Reading always begins at index. Up to 48 sectors *can* be read (the entire contents of the cylinder), but no more than 24 sectors *should* be read. Exceeding the 24 sector limit increases the chances of reading the wrong data field into storage. The data portion of the record is read and placed in storage beginning at the address specified in the disk data address register. One is subtracted from the N byte and added to the S byte of the disk control field for each sector read. The data address in the disk data address register is returned to its original value at the beginning of each sector so that successive data fields overlay each other in storage. The operation ends at the end of the sector in which the N-byte is reduced to FF, the end of the cylinder is detected, or equipment check is detected. (No other conditions terminate the operation.) When the operation is terminated, data from the last sector read is in the disk data field in CPU storage.

This operation functions with reduced address marker requirements so that data that cannot be read by a read data operation because of a missing address marker possibly may be recovered.

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

Errors that do not terminate the read operation are reset at the end of the sector in which they occur unless they occur in the last sector to be read.

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

The control unit will be busy to any new operation except sense I/O while performing a read data diagnostic operation.

5444 Read IPL Operation

This operation is initiated by pressing the load key on the system control panel. In order for the load key to cause initial program loading from disk (drive 1 only) the IPL selector switch on the system control panel must be set either to FIXED DISK or REMOVABLE DISK. The read IPL operation causes the 256 bytes of data contained in the first record after the index mark on track 0 of the selected disk to be transferred to storage starting with storage address 0000. Control is then passed to the processing unit to begin executing the instructions starting at address 0000.

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

5444 Verify Operation

The verify operation is performed for write checking. It should be performed after every write operation to ensure data integrity. (If the write was a multiple-sector operation that crossed a track boundary, the head select must be reset to 0 by a seek operation before issuing the read verify instruction.)

This operation is performed in the same way as the read data operation except that no data is transferred to main storage and the disk read/write address register is not updated. No cycle steals are required except for updating the sector and N-bytes in the disk control field.

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

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

5444 Write Data Operations

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

Writing begins with the sector specified by the identifier portion (F, C, and S bytes) of the disk control field located in storage and addressed by the disk control address register. The identifier from the disk control field is compared with the sector identifiers read from the selected disk track. (The head selection is the result of the last seek unless a multiple sector operation that caused a track boundary crossing was performed subsequently. The 5444 automatically switches from head 0 to head 1 when it crosses the track boundary on the upper surface of the disk.)

Comparing begins with the first sector identifier to come under the head. An equal condition between the disk control field identifier and the sector identifier enables the writing of the 256 bytes of data. The data is fetched from storage using the disk data address register for addressing.

When multiple sectors are indicated, one is added to the S byte and one is subtracted from the N byte of the disk control field for each sector written (except for switching heads, when 9 is added to the S byte).

This updated disk control field identifier is then compared with the next identifier read from the disk. An equal comparison of all succeeding addresses must occur before their corresponding data fields are written on the disk. The data field of a sector will not be written if an error is found before writing of data begins.

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

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

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

At the end of the operation the disk control field contains information about the progress of the operation. The identifier portion of the disk control field indicates the last sector written or where writing was attempted. The number of records processed can be determined by subtracting the contents of the N byte from the original value of the N byte unless all sectors were written. If all sectors were written, the N byte contains FF. If a track boundary is crossed in a multiple-sector write operation, head 1 remains selected.

5444 Write Identifier Operations

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

The identifier portion of the disk control field is written as the sector identifier of the first sector after the index marker. The N byte of the disk control field is forced to a value of decimal 23 by this operation so that exactly 24 sectors will be written on the track. As each identifier is written on the disk, one is added to the S byte and one is subtracted from the N byte of the disk control field.

The data field of each sector is filled with the characters stored at the address contained in the disk data address register. The register is *not* updated during the operation, so the same character is propogated in all data byte positions of all the sectors. During writing of each identifier and data field, the control unit generates two cyclic check and one bit count appendage bytes and automatically writes them as the last three bytes of both the identifier and the data fields. The check data for the identifier applies only to the identifier, and the check data for the data applies only to the data.

At the end of the operation the disk control field contains information about the progress of the operation. The identifier portion of the disk control field indicates the last sector written or where writing was attempted. The number of records processed can be determined by subtracting the contents of the N byte of the disk control field from 23, the original value of the N byte, unless all records have been processed. If all records have been processed, the N byte contains FF.

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

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

5444 Scan Operations

The scan operation searches the data fields on the disk to find one that meets certain specified conditions when compared to a 256-byte data field in storage. Up to one cylinder of data (48 sectors) can be scanned in one operation. The scan operation can specify one of the following conditions to satisfy the scan.

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

The data in the sectors on the disk is compared with the 256 characters in the disk data field in storage. The disk data field is addressed by the disk 5444 data address register. The comparison of individual characters within the sector can be masked off by placing a mask character consisting of all bits (hexadecimal FF) in each non-compare byte in the disk data field in storage. If only ten bytes are to be compared, the field must contain 246 mask characters in the byte positions of the characters that are not to be compared.

Scanning of the data begins with the sector specified by the identifier portion of the disk control field. Bit 0 of the S byte for this instruction does not select the head but is used for comparison only. Head selection can only be accomplished by a seek instruction. Comparing of sector addresses begins with the first sector identifier to come under the head. After the beginning sector is scanned, the S byte is updated to the identifier of the next sector (by adding one to the S-byte value) and the N byte is decreased by 1 for each sector scanned. The S-byte-updating and head switching from 0 to 1 are automatic when a track boundary is crossed in a multiple sector.

The operation terminates under the following conditions:

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

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

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

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

The 5444 data disk address register will contain the original address at the end of the operation unless equipment check occurs. The contents of the register are unpredictable in the even of an equipment check.

5444 DISK INSTRUCTIONS

Start I/O

Mnemonic: SIO

Op Code	Q Byte			Control Byte
F3	DA	M	N	Control Code

Operation: This instruction selects a drive and disk and specifies the operation that is to be performed by that drive and disk.

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

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

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

N Bits	*Control Bits 6 and 7*	*Operation*
000	--	Seek
001	00	Read Data
001	01	Read Identifier
001	10	Read Data Diagnostic
001	11	Verify
010	-0	Write Data
010	-1	Write Identifier
011	00	Scan Equal
011	01	Scan Low or Equal
011	1-	Scan High or Equal

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

Issuing a start I/O instruction to a control unit that is busy, issuing a seek start I/O instruction to a drive that is seeking, or issuing a seek start I/O instruction to a drive that is not ready causes an automatic program level advance in systems with dual programming feature installed. If the feature is not installed, the program loops on the start I/O instruction until the condition is corrected.

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

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

The start I/O instruction uses the contents of the 5444 disk data address register as the initial address of all sector data fields. It uses the contents of the disk control address register as the address of the disk control field.

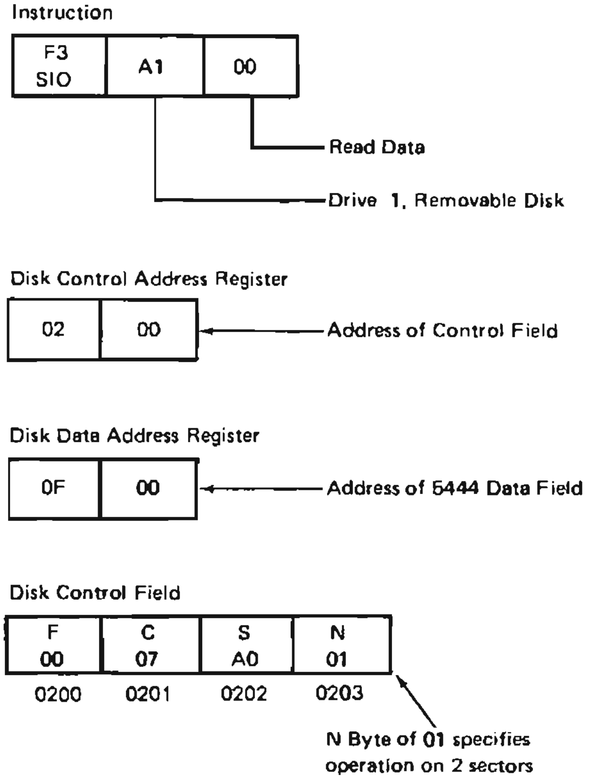
A start I/O addressed to an unsafe drive is no-oped and the no-op bit in the status bytes is set.

Any start I/O that is executed resets all previously generated device status except:

Seek check.
Equipment check caused by an unsafe condition.
Cylinder zero.
No-op.
Intervention Required.

Seek check is also reset by start I/O if it is associated with the drive that is addressed. Equipment check caused by an unsafe condition is not reset by any instruction. No-op is reset by sense I/O. Cylinder zero resets when the access arm moves away from cylinder 0.

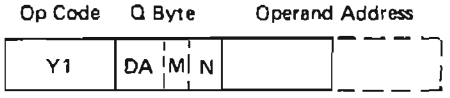
Example:



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

Load I/O

Mnemonic: LIO



Operation: This instruction loads the two bytes of data contained in the operand addressed by the operand address into a local storage register specified by the N field in the Q byte. The operand is addressed by its low-order (right-most) byte. The M bit is not used.

The device address portion of the Q byte can specify either drive (A for drive 1 or B for drive 2) because the LIO instruction is not drive sensitive.

The N code can specify only three values:

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

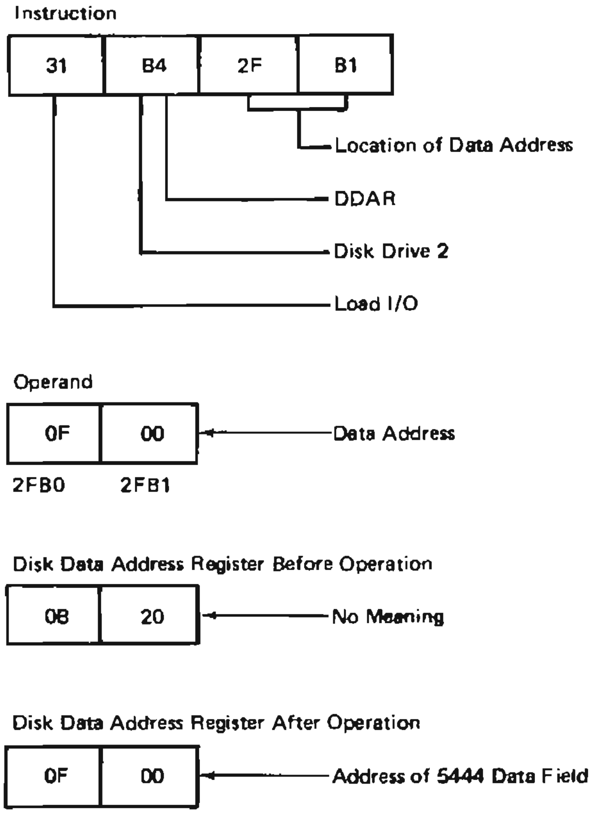
Any N code other than the ones specified causes the processing unit to stop with a processor check and an invalid Q byte indication.

A load I/O instruction issued to a busy control unit causes an automatic program level advance if the system has dual programming feature installed. If the feature is not installed, the program loops on the load I/O instruction until the control unit is no longer busy.

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

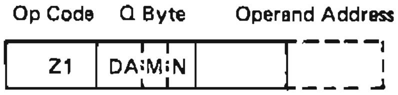
The load I/O instruction is executed if the addressed drive is executing a seek or recalibrate operation and a read, write, or scan has not been accepted or provisionally accepted. The load I/O instruction is executed regardless of ready status.

Example:



Test I/O and Branch

Mnemonic: TIO



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

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

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

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

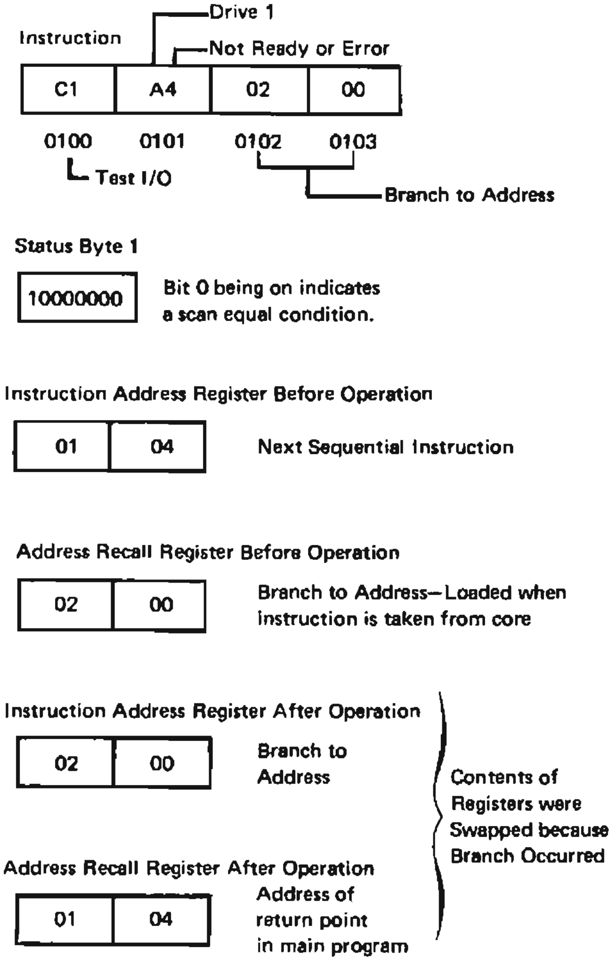
   Data check.
   Track condition check.
   Missing address marker.
   End of cylinder.
   No record found.
   Equipment checks not caused by unsafe.
   No-op.
   Overrun.

   Check condition is also indicated if a seek check or unsafe exists for the addressed drive. The drive that has the check condition can be determined from status byte 1, bits 6 and 7.
2. N code 010—busy. The disk drive control unit either is executing a read, write, or scan operation or has provisionally accepted one of these operations for execution at the end of the seek operation in progress.
3. N code 100—scan found. Scan found is indicated when either drive is addressed. The sense byte indicates which drive contained the scan found condition. Scan found indication is reset by the next start I/O instruction.

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

When addressing a model 3 or A3 an M bit, used with the N code of 000, will indicate a not ready condition.

Example:


Instruction
Drive 1
Not Ready or Error
C1
A4
02
00
0100
0101
0102
0103
Test I/O
Branch to Address
Status Byte 1
10000000
Bit 0 being on indicates a scan equal condition.
Instruction Address Register Before Operation
01
04
Next Sequential Instruction
Address Recall Register Before Operation
02
00
Branch to Address—Loaded when instruction is taken from core
Instruction Address Register After Operation
02
00
Branch to Address
Address Recall Register After Operation
01
04
Address of return point in main program
Contents of Registers were Swapped because Branch Occurred

Advance Program Level

Mnemonic: APL

Op Code	Q Byte			
F1	DA	M	N	Not Used

Operation: This instruction tests for the conditions specified in the Q byte. If the condition tested for is present (and program level has been enabled by the start I/O instruction), a system with dual programming feature installed activates the inactive program level; a system without the dual programming feature installed loops on the advance program level instruction until the condition no longer exists. If the condition is not present, systems with and without the dual programming feature take the next sequential instruction in the active program level.

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

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

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

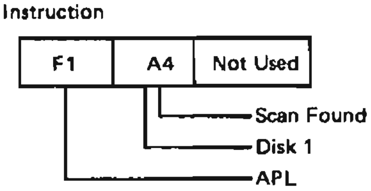
   Data check.
   Track condition check.
   Missing address marker.
   End of cylinder.
   No record found.
   Equipment check not caused by unsafe.
   No-op.
   Overrun.

   Check condition is also indicated if seek check or unsafe exists for the addressed drive. Seek check or unsafe for the drive not addressed will not be indicated. The drive with the check condition can be determined from status bytes 1, bits 6 and 7.
2. N code 010–busy. The disk drive control unit is executing or has accepted provisionally for later execution a read, write, or scan operation.
3. N code 100–scan found. Scan found is indicated when either drive is addressed and a scan has been matched in one of the drives. The sense byte indicates which drive contained the scan found condition. Scan found indication is reset by the next start I/O instruction.

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

The third byte of the instruction is not used. The M bit is ignored except for the Model 3 and A3 not ready indication.

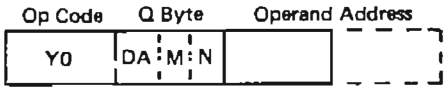
Example:



The next instruction address is taken from the instruction address register of the program level that did not execute this instruction.

Sense I/O

Mnemonic: SNS



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

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

1. N code 010—status bytes 0 and 1 (1 is sent first).
2. N code 011—status bytes 2 and 3 (3 is sent first).
3. N code 100—disk read/write address register.
4. N code 110—disk control address register.

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

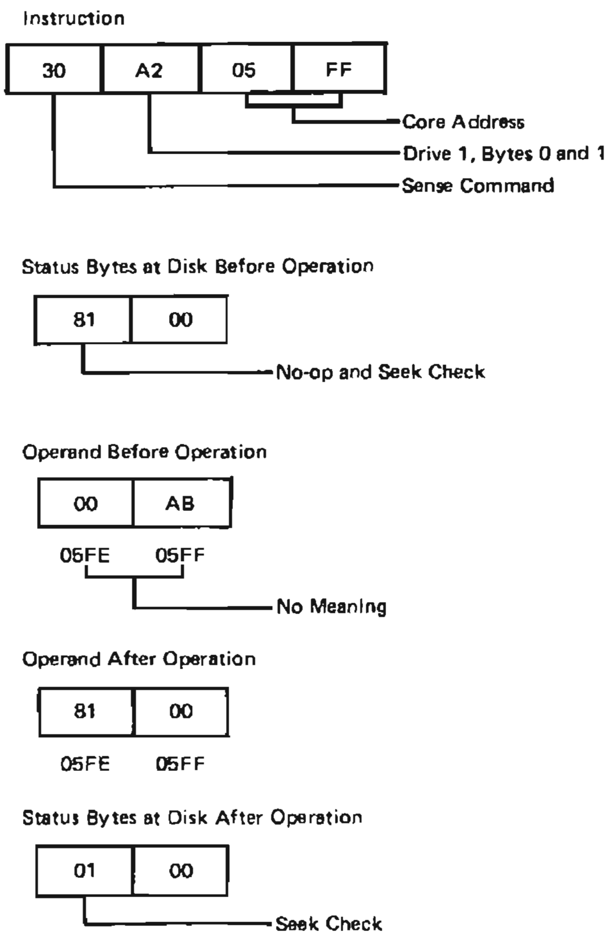
The status bytes are bit significant as illustrated in Figure 6-11. The higher numbered status byte is stored in the low-order position of the field. An explanation of each status bit is provided in *Check Conditions and Status.*

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

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

All the status bits not discussed in the preceding paragraph are presented with the status bytes to a sense I/O for either drive. All except no-op are reset by the next start I/O instruction issued to either drive. No-op is reset by the sense I/O instruction to either drive that transfers it to storage.

Example:



Bit	Byte 0 (Low Address)	Byte 1 (High Address)	Byte 2 (Low Address)	Byte 3 (High Address)
0	No-op	Scan Equal Hit	Unsafe*	CE Sense Bit
1	Intervention Required*	Cylinder Zero*	Timing Analysis Program Line A	CE Sense Bit
2	Missing Address Mark	End of Cylinder	Timing Analysis Program Line B	CE Sense Bit
3	Equipment Check*	Seek Busy*	Timing Analysis Program Line C	Not Bit Ring Inhibit
4	Data Check	100 Cylinder	Index*	Standard Write Trigger
5	No Record Found	Overrun	Head Settling*	Condition Priority Request
6	Track Condition Check	Status Address A	CE Sense Bit	Bit Ring 0
7	Seek Check*	Status Address B	Model 6	Not CC Register Position 17

*These bits apply only to the drive addressed.

Figure 6-11. Disk Storage Status Byte Information

CHECK CONDITIONS AND STATUS

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

Status Byte 0

Bit 0—No-Op

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

Bit 1—Intervention Required

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

Note: Ready may be dropped if an access reverse operation (high performance disk drives only) specifies more tracks than the actual number of tracks from the present track to the home position. If ready is dropped and no permanent hardware fault exists, stop the disk drive and then restart the disk drive to establish a file ready condition.

Bit 2—Missing Address Marker

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

Bit 3—Equipment Check

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

Bit 4—Data Check

This status indicates that a cyclic check or bit count appendage error was detected while reading the identifier or data fields from the file.

Bit 5—No Record Found

This bit indicates that a sector called for by a read, write, verify, or scan instruction could not be found on the track specified by the previous seek operation or that it did not match the identifier in the disk control field. This could also occur if the previous operation was a multiple sector operation that switched the head selection to head 1 while the current operation referenced head 0.

Bit 6—Track Condition Check

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

Bit 7—Seek Check

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

Byte 1

Bit 0—Scan Equal Hit

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

Bit 1—Cylinder Zero

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

Bit 2—End of Cylinder

This bit indicates that, on a multiple-sector or scan operation, one of the following has occurred:

1. The last sector on the disk (sector 55) has been operated on and the number of sectors specified by the instruction still has not been satisfied. That is, the instruction attempted to operate beyond the end of the cylinder.
2. Head 1 IDs were written on the upper surface of the disk (to identify lower tracks on upper surface alternate tracks) and the instruction tried to operate beyond the end of the track (head-switching).

Whenever an end-of-cylinder condition occurs, all sectors up to and including the last one on the cylinder (or track, for alternate tracks) are successfully processed.

Bit 3—Seek Busy

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

Bit 4—100 Cylinder

This bit indicates that the drive installed in the system has 100 cylinders available to the customer.

Bit 5—Overrun

This bit is set when the processing unit fails to allow a cycle steal to the disk unit in time to transfer data before it is lost. This occurs during processor check stop in the processing unit that stops the processing unit clock.

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

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

Byte 2

Bit 0—Unsafe

This bit indicates that one of the following checks has been detected by the disk unit.

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

This check also causes equipment check.

This check must have a unique program halt indicator.

Bits 1, 2, and 3—Timing Analysis Program (TAP) Lines A, B, and C

These three bits are used by the CE diagnostic programs. The bits may be jumpered to various file control unit signals for sensing. These three bits are normally jumpered to the unsafe latches that are used by the CE to further define the unsafe condition.

Bit 4—Index

This bit is on for about 43 microseconds, starting when the index mark passes the read head. It turns off when the address marker for the first sector (sector 0) passes the read head.

Bit 5—Head Settling

This bit indicates that the seek operation is not complete because the head is not ready for operation.

Bit 6—CE Sense Bit

This bit is used by the CE diagnostic programs. The bit may be jumpered to various file control unit signals for sensing by the program.

Bit 7—Unused

Byte 3

Bits 0, 1, and 2—CE Sense Bits

These three bits are used by the CE diagnostic programs. The bits may be jumpered to various file control unit signals for sensing.

Bits 3, 4, 5, 6, and 7—CE Sense Bits

These bits are used by the CE diagnostic programs and represent the condition of the following file control unit signals:

- Bit 3—Not bit ring inhibit.
- Bit 4—Standard write trigger.
- Bit 5—Condition priority request.
- Bit 6—Bit ring 0.
- Bit 7—Not CC register position 17.

FLAGGING

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

The flagging procedure uses bits 6 and 7 of the flag byte of the identifier of each sector recorded on the disk. Bit 6 alone indicates that the track is a defective track, and bit 7 alone indicates that the track is an alternate track. Both bits 0 indicates that the track is an original good track. Both bits 1 indicates a defective alternate track.(which has its own address in the C byte when IBM program products are used).

A track with a bad spot is marked defective and an alternate is assigned to replace the whole track. When a track is found to be defective, a write identifier operation must be performed to write the flag bytes with bit 7 = 1 and the C and S bytes of the identifiers from the defective track on the alternate track. Then the recoverable data from the defective track must be written on the corresponding sectors on the alternate track. Finally, the defective track must be written with a write identifier operation to write flag bytes with bit 6 = 1 and the C and S bytes of the identifiers from the alternate track on the defective track.

Track 203 (used by IBM customer engineers for diagnostics) can be flagged with bit 6 on, bit 7 off if the track is defective. However, the address of an alternate track should not be assigned to track 203 if the track is used for CE diagnostics.

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

The identifier fields of the tracks are:

Good—Bits 6 and 7 of the F byte are both 0 and the C and S bytes contain the cylinder and sector numbers that are correct for that track.

Defective—Bit 6 is 1 and bit 7 is 0 in the F byte. The C and S bytes contain the cylinder and sector address from the alternate track.

Alternate—Bit 6 is 0 and bit 7 is 1 in the F byte. The C and S bytes contain the cylinder and sector addresses from the defective track replaced by the alternate.

Defective Alternate—Bit 6 is 1 and bit 7 is 1 in the flag byte. Bytes C and S contain the cylinder, head, and sector numbers of the respective sectors.

TRACK INITIALIZATION PROCEDURES

The following procedures must be followed by track initialization programs for the 5444 disk storage drive for System/3. They analyze the condition of the surface and format the tracks.

1. Read identifier to determine that the track has not been previously flagged. This step *must not* be performed when initializing a previously *unused* disk.
2. Write identifier with a data field of hexadecimal 55. Write appropriate code into bits 6 and 7 of the flag byte: hex 00 for original tracks, hex 01 for tracks assigned as alternate tracks.
3. Read data of all the sectors to ensure that it can be recovered. If an error occurs, go to step 10.
4. Repeat step 2 with a data field of hexadecimal 00.
5. Repeat step 3.
6. Seek to the next track and repeat steps 1 through 5.
7. Repeat steps 1 through 6 until all tracks have been processed.
8. Read identifier on all tracks to check for seek errors. If a seek error on the writing operation is detected, initialization must repeat steps 1 through 7. A seek error on the writing operation causes two different tracks to contain the same identifiers or the identifiers for one track to be missing.
9. Performs steps 1 through 8 at least once.
10. If an error occurs, the device status must be analyzed. If a missing address mark or data check occurs, retry a read data instruction at least 10 times. On the first unsuccessful retry that indicates missing address marker or data check, flag the track as defective and go to step 11. If all ten retries are successful proceed with the initialization procedure from the point at which it was interrupted.

    For any error other than missing address marker or data check follow the normal error recovery procedures.
11. Assign an alternate track unless this is an alternate track.
12. Write identifier on the defective track with the address of the alternate track in the identifier and a hex value of 02 in the flag byte. A defective alternate track should contain its own address and a hex value of 03 in bits 6 and 7 of the flag byte.
13. Set the flag byte in the disk control field to hex 02. Perform a read identifier operation. If the address of the alternate track is not recoverable, the disk must be repaired unless this is an alternate track.
14. Seek to the alternate track.
15. Set the flag byte in the disk control field to hex 01. Write identifier on the alternate track with the identifiers of the defective track in the disk control field. Alternate tracks must be proved reliable by steps 1 through 5 before they are used as alternates.
16. Continue with initialization on the next track.

The basic requirement is for one pass through steps 2 through 8. An option must be provided to allow any number of passes up to 255.

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

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

6

SUGGESTED ERROR RECOVERY PROCEDURES

The following minimum error recovery procedures are defined for the disk and attachment. A test I/O for not ready/check condition must be performed. If not ready/check is present perform action III from Figure 6-12. The status bytes and bits must be tested in the following order and the actions from Figure 6-12 performed when the bits are set.

Priority	*Byte*	*Bit*	*Condition*	*Action*
1	0	3	Equipment Check	II
2	0	1	Intervention Required	VII
3	1	5	Overrun	III
4	0	5	No Record Found	III
5	0	2	Missing Address Mark	III
6	0	4	Data Check	III
7	0	6	Track Condition Check	III
8	0	7	Seek Check	V
9	1	2	End of Cylinder	IV

Action I

1 If there is no additional error recovery procedure, perform an operator message and stop.

2 If there is an additional error recovery procedure, exit to it.

3 If the additional error recovery procedure fails, perform an operator message and stop.

Action II

Retry the original operation or sequence of operations once. On the second occurrence of this error condition, perform an operator message and stop. Upon operator restart, do Action II.

Action III

1 Perform a Read ID operation. If there is an error, do Action IV for the original operation. If no error, update the residual DFDR and N field values.

2 If the present track is defective, indicate an alternate is being used, determine the alternate track from the ID field, and go to Action IV, part 2.

3 Check to determine if head switching from an alternate track has just taken place. If so and no true error exists, go to Action IV, part 2

4 Go to Action VII.

Action IV

1 Update the residual disk address to the next track.

2 Use the residual to obtain the next track address. Set the flag byte to zero.

3 Continue the operation with updated values.

Action V

1 If 16 recalibrate retries were attempted, go to Action I.

2 Issue a Recalibrate.

3 Retry the original operation.

Action VI

1 If 16 retries have been attempted since the last recalibrate, go to Action V, step 1.

2 Go to Action V, step 3.

Action VII

Perform an operator message and stop. After restart, repeat the original operation or sequence of operations.

CAUTION: If a nonrecoverable disk error occurs, have a qualified customer engineer examine both the disk drive and the disk cartridge for damage before using the drive or cartridge for any subsequent disk operations.

Figure 6-12. Disk Error Recovery Procedures

SUMMARY OF INSTRUCTION HANDLING

Figure 6-13 summarizes how the system handles disk instructions under various operating conditions.

Instruction Issued to a Disk Drive That:	Is Not Ready	Has an Equipment Check (Not Caused by Unsafe)	Is Unsafe	Has the No-op Bit Active	Is Ready and Not Busy	Is Executing a Seek	Is Busy**
Start I/O **Read** **Write** **or** **Scan**	Is accepted and executed	Accepted and executed (resets the equipment check)	No-Op'ed	No-Op'ed	Accepted and executed (brings up FCU busy)	Accepted (will be executed if and when the seek is completed w/o error)	Causes an APL or is rejected*
Start I/O **Seek**	Causes an APL or is rejected*	Accepted and executed (resets the equipment check)	No-Op'ed	No-Op'ed	Accepted and executed (brings up seek busy)	Causes an APL or is rejected*	Causes an APL or is rejected*
Load I/O	Causes an APL or is rejected*	Accepted and executed	Accepted and executed	Accepted and executed	Accepted and executed	Accepted and executed	Causes an APL or is rejected*
Test I/O							
Error		Branch	Branch	Branch			
Not Ready	Branch		Branch	Branch***			
Busy							Branch
Sense I/O	Executed	Executed	Executed	Executed (and resets no-op)	Executed	Executed	Executed

* APL for Dual Program Level Systems, and rejected and looped on for One Program Level Systems.

**FCU busy will become active when, and only when, a read, write, or scan operation is accepted by the FCU.

***Branch occurs only if no-op was set due to unsafe.

Figure 6-13. Summary of Instruction Handling

IBM 5445 DISK STORAGE DRIVE

The IBM 5445 (Figure 6-14) provides large capacity, high speed, direct access storage capability for System/3. The 5445 is available in two models–Model 1 and Model 2. Each model contains the mechanism needed to drive one removable IBM 2316 Disk Pack (Figure 6-15). Model 1 contains the electric power supply for both models, and must be attached to a storage control special feature within the 5410. Model 2 must be attached to Model 1.

Each 5445 provides an online data capacity of 20.48 million bytes for a total of 40.96 million bytes when both drives are attached. (This is in addition to any disk storage capacity provided by an IBM 5444.)

Physical Characteristics

Maximum 5445 Drives per system	2
Data rate	312 kilobytes/second
Disk rotation speed	2400 RPM
Average rotational delay	12.5 milliseconds
Maximum access time	130 milliseconds
Average random access time	60 milliseconds
Minimum access time (single track movement)	25 milliseconds
Capacity per drive	20.48 megabytes
Number of data cylinders *	200
Alternate (spare) cylinders *	3
Data tracks per cylinder	20
Number of maximum-size data records per track	20
Capacity per record (maximum)	256 bytes (key and data)

* As used with IBM programming systems support

IBM 2316 Disk Pack

The IBM 2316 Disk Pack is a compact disk assembly, 15 inches in diameter (with cover), and weighs about 13 pounds. The disk pack contains 11 disks, each 14 inches in diameter. Disks are mounted one-half inch apart on a vertical shaft. The disks provide 20 surfaces on which data can be recorded (the top of the upper disk and the bottom of the lower disk are not used). The entire assembly rotates once every 25 milliseconds.

Care and handling procedures for 2316 disk packs are described in *IBM Disk Pack and Cartridge Handling Procedures,* GA26-5756.

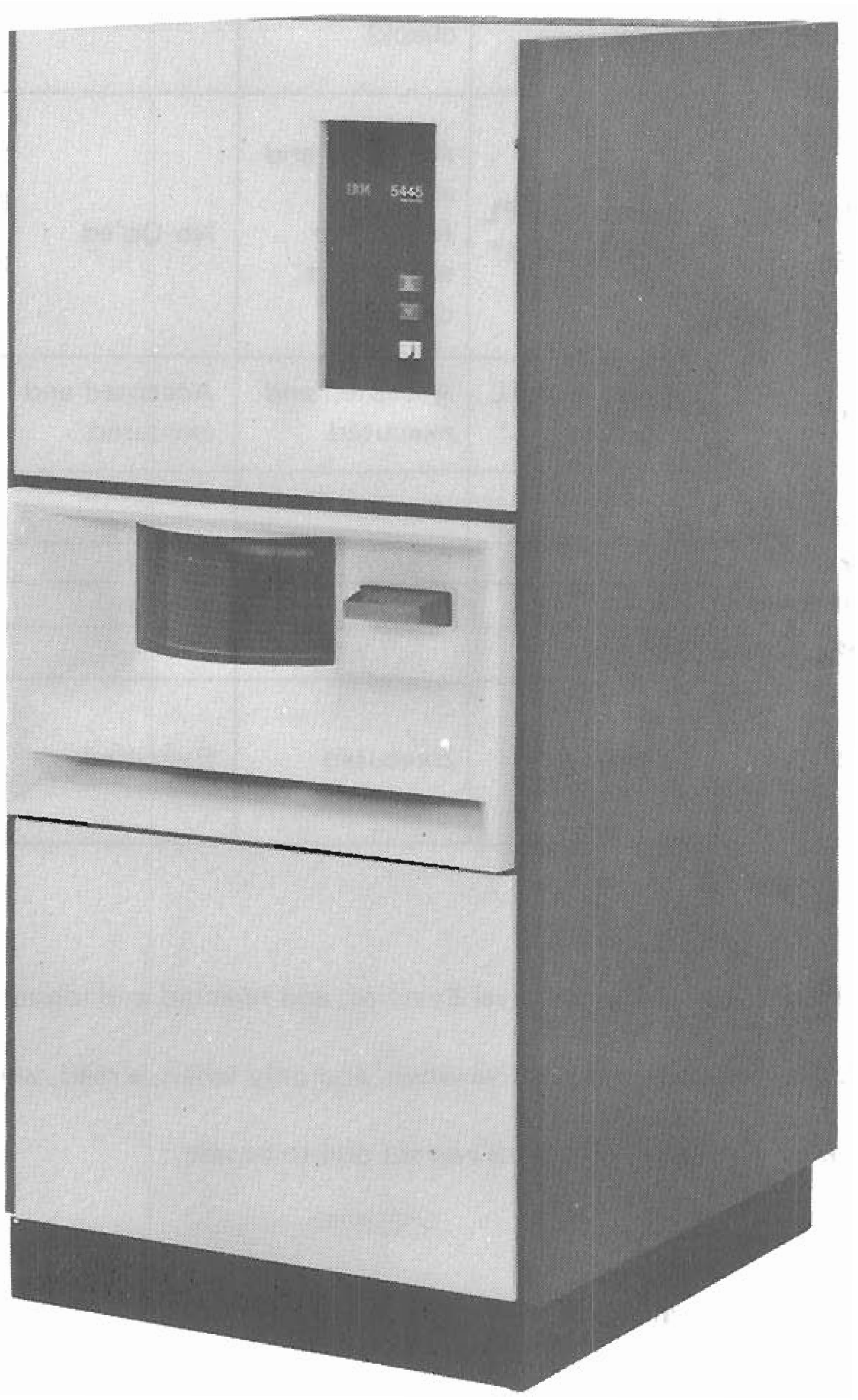


Figure 6-14. IBM 5445 Disk Storage Drive

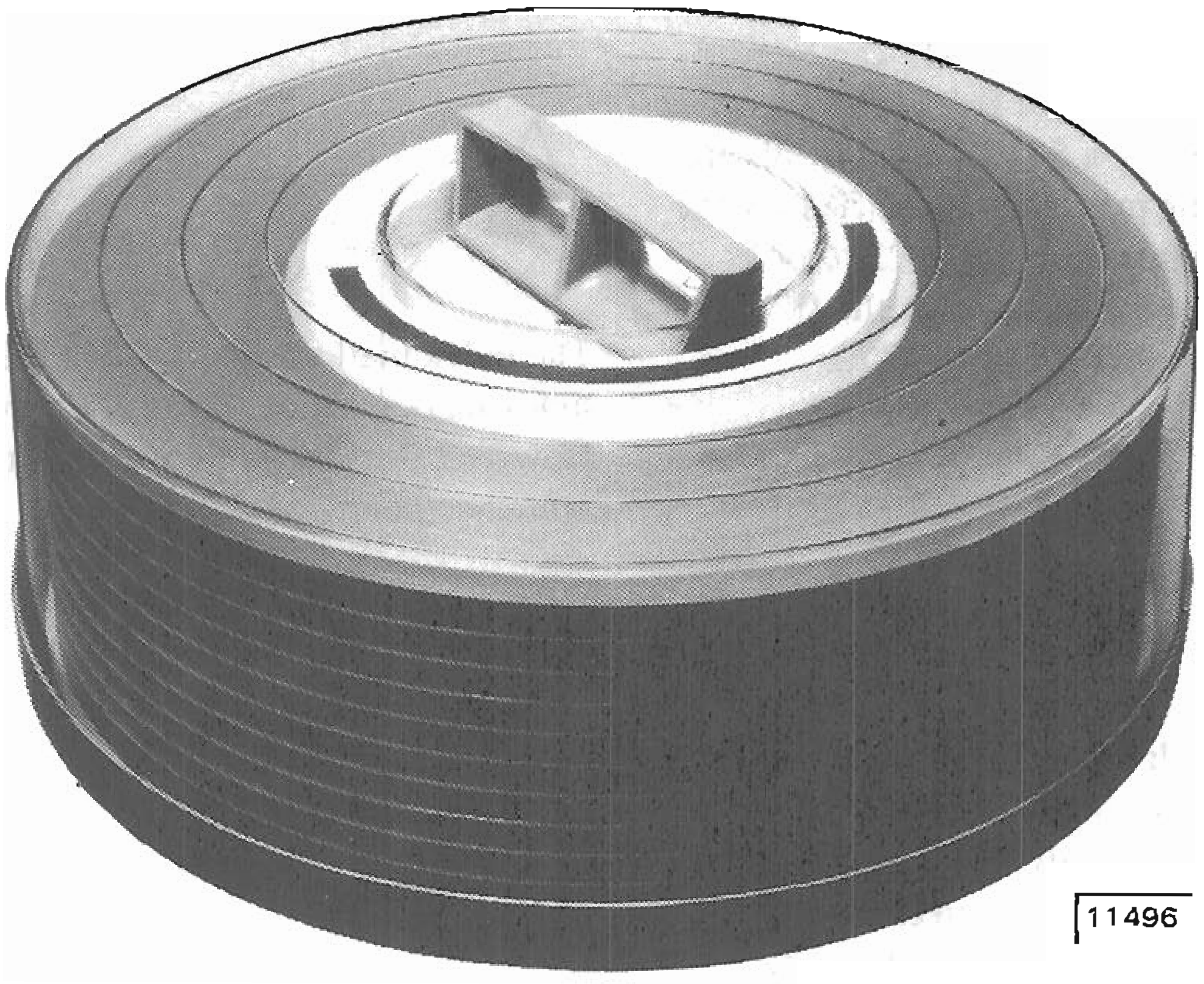


Figure 6-15. IBM 2316 Disk Pack

Access Mechanism and Disk Organization

Information is written on and read from disk surfaces of the 2316 disk packs by read/write heads in the 5445. The 20 read/write heads are positioned by a movable, comb-like access mechanism. Two read/write heads are attached to each arm of the access mechanism, and the heads are numbered 0 to 19 from top to bottom. Therefore, heads 0 and 1 are attached to the upper arm, 2 and 3 to the second arm, etc. The heads are held just off the disk surfaces by a cushion of air while the disk drive is operating.

The 20 read/write heads always occupy a common vertical plane; that is, all 20 heads are always aligned one above the other, so that any movement of the access mechanism causes identical movement of all heads. Therefore, 20 different tracks—one for each of the 20 disk surfaces used—are always under the read/write heads (one head for each track) at each access arm position. This means that 20 tracks are available for read/write operations without moving the access mechanism.

Figure 6-16 shows how the entire disk pack constitutes 203 concentric cylinders of information. Cylinder numbering is from 000 (outermost cylinder) to 202. Tracks in cylinders 200, 201, and 202 are specified by IBM programming products as alternate tracks, and tracks in cylinders 000 through 199 as primary tracks. If one of the primary tracks is defective, the program assigns an alternate track in place of the defective track.

Each unique track has an address that consists of the track cylinder number followed by its read/write head number.

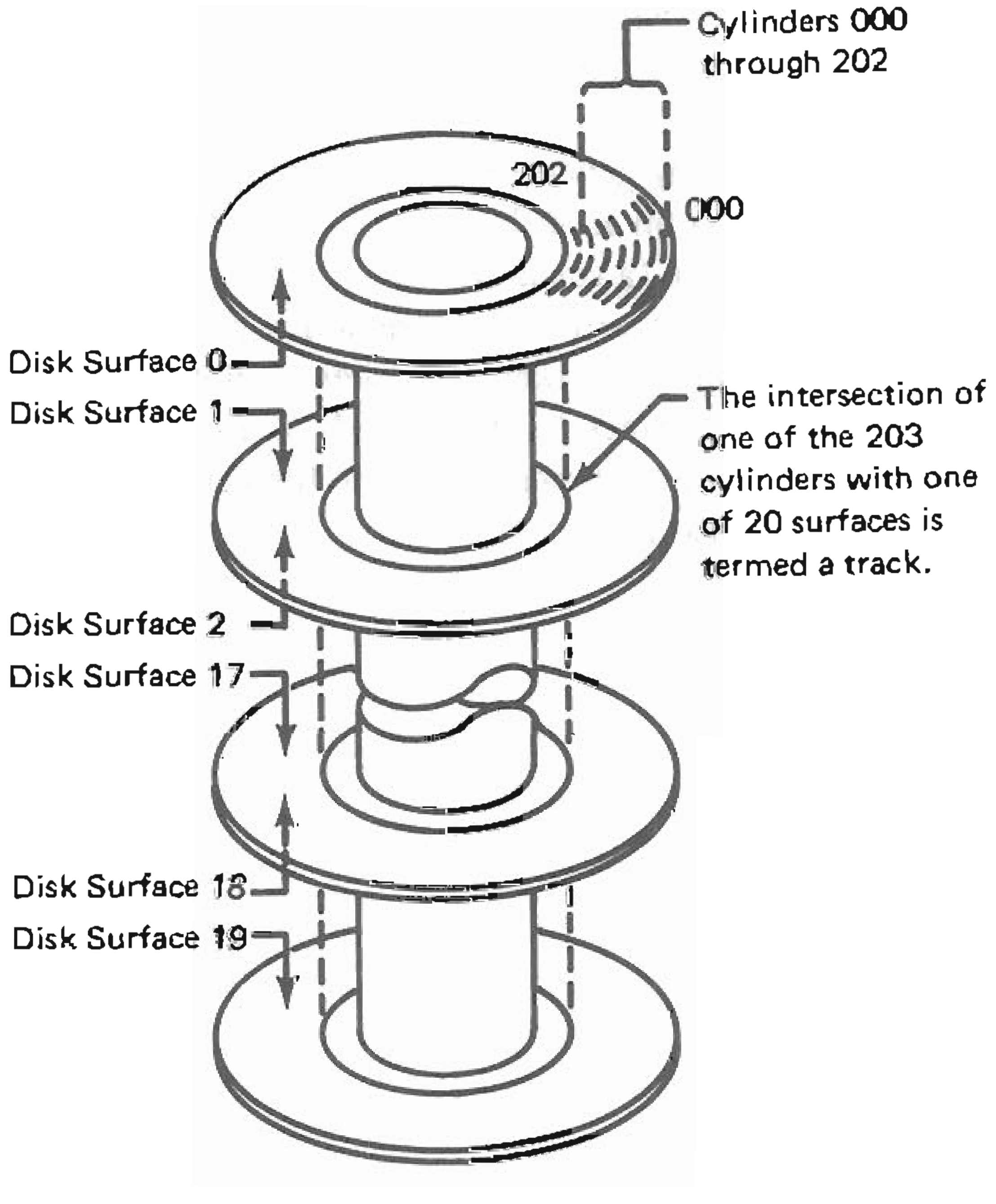


Figure 6-16. Cylinder Concept

DATA COMPATIBILITY

Data written on a 2316 disk pack by the 5445 can be read by any IBM 2314 or 2319 Disk Storage Drive; data recorded by a 2314 or 2319 can be read by a 5445 if the records are formatted using the formatting procedures specified for the 5445. When such formatting is followed, the 2316 disk packs provide data interchangeability between the IBM System/3, IBM System/360, and IBM System/370.

DATA FORMAT

Data is recorded on the 2316 disk pack in variable-record-length format, with a maximum length of 256 bytes for the combined key and data fields. Twenty maximum length records can be recorded on one track. Decreasing the record length increases the number of records that can be recorded on a track. (When IBM Program products are used, the key length is always 0, and the data length is always 256.)

TRACK FORMAT

Figure 6-17 shows disk organization and track format. The format for each track written on the disk pack starts at a point on the disk called the index marker. (This point is specified by a signal emitted by the disk spindle as it turns all the disks, effectively generating synchronized index markers for all tracks on all disks in the pack.) A home address, then record zero (a track descriptor record), then sequentially-numbered records follow the index marker on the track until the index marker is again encountered, signalling that the entire track has been used. Gaps, automatically written by the attachment, separate the various unique format units and areas in the records.

Index Marker

△

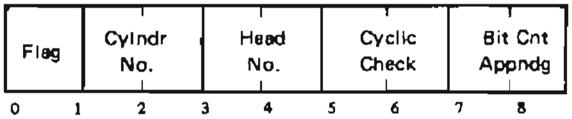
The index marker signals the initial point of each track. The index marker is not recorded on the track or in storage. However, it is shown in figures in this manual as a track reference point.

Gap

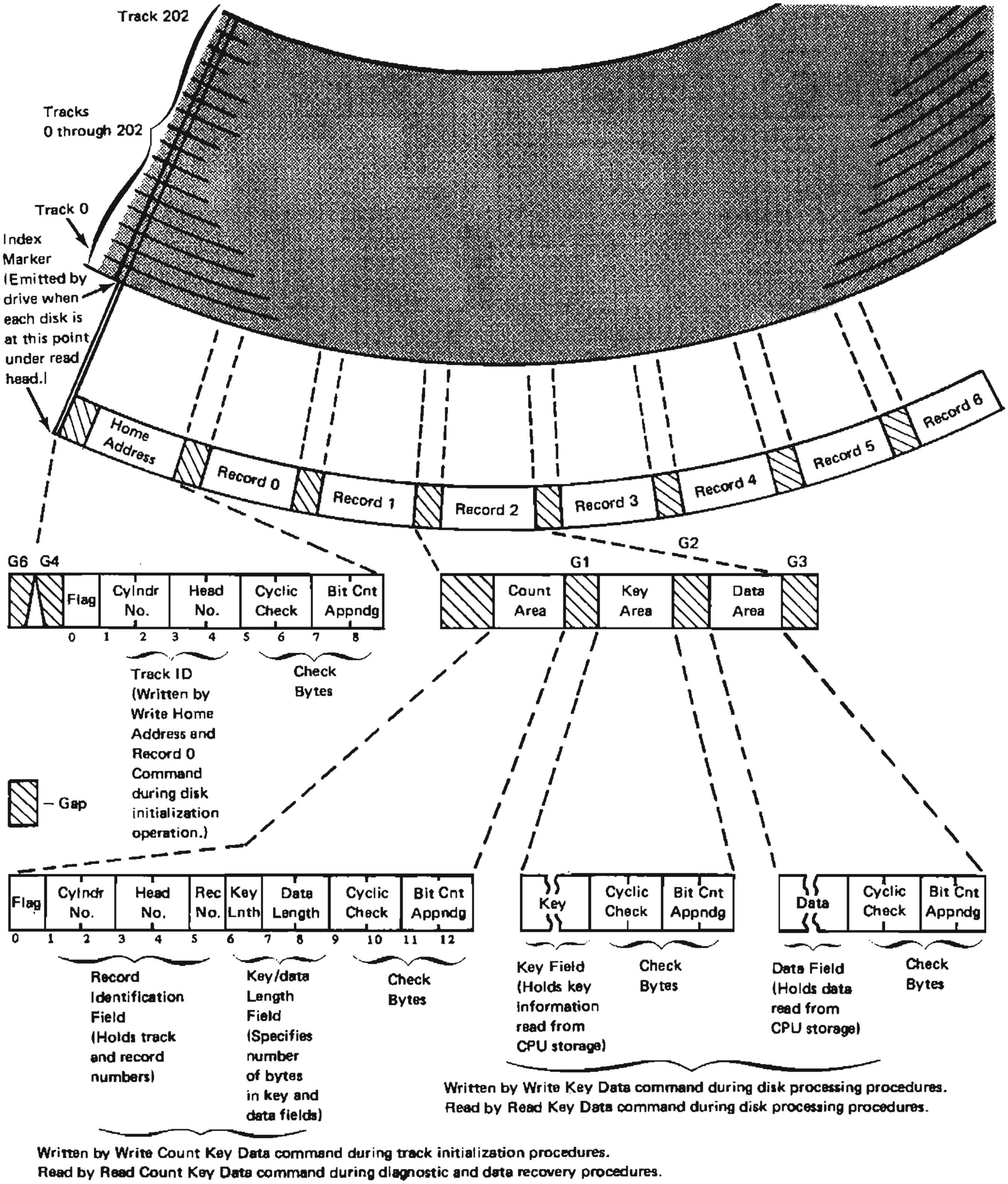
▨

A gap is an area written on the track by the attachment to separate two adjacent groups of data and to identify the group that follows the gap. This information is used by the attachment only.

Home Address (HA)



The home address gives each track a unique track identity that is not affected by normal programming operations. Each track in a storage drive can be located directly by cylinder number and head number. Normal programming operations can use the home address area without changing its contents. Home addresses are transferred from the processing unit to the 5445 by a write home address command, and from the 5445 to the processing unit by a read home address command. Home addresses are usually written by utility programs during file initialization.

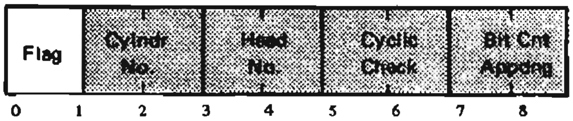


Notes:

1. Records are numbered consecutively from zero for correct machine operation.
2. Key and data fields are variable length; therefore, track formats are not identical in record locations. If key length of zero is specified, the key field and its preceding gap are not included in format.

Figure 6-17. IBM 5445 Track Format and Disk Layout

Home Address Flag Byte (F)



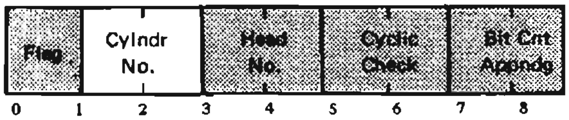
The home address flag byte indicates track condition and whether the track is a primary track or an alternate track. The flag byte can be transferred to the CPU by a read home address command.

Normally, all eight bits of the flag byte are zero when the home address is first written by a write home address command. Thereafter, the flag bits assume significance:

Bit	*State*	*Meaning*
0	0 or 1	Internal control bit
1	0 or 1	Special control bit in Write HA and R0 operation.
2	– – –	Not used.
3	– – –	Not used.
4	– – –	Not used.
5	– – –	Not used.
6	0	Track is operative.
	1	Track is defective.
7	0	Track is a primary track.
	1	Track is an alternate track

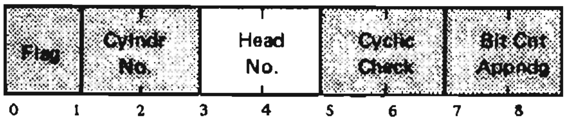
Bits 6 and 7 must be program-propogated into the flag byte of each record on the track; otherwise, a check occurs.

Home Address Cylinder Number Bytes (CC)



The group of tracks available to the twenty read/write heads at each access mechanism position comprise a cylinder. The cylinder number identifies the cylinder within which the track is situated. All bits in the first byte must be zero; the next byte holds the cylinder number (000 through 202 decimal, or 00 through CA hexadecimal).

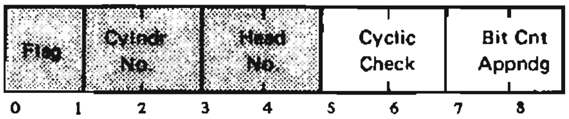
Home Address Head Number Bytes (HH)



This two-byte field identifies the read/write head associated with the specified track. The head number, together with the cylinder number, identify a single track to be acted upon. The bits in the first head-number byte all must be zeros; the second byte must contain the head number (00 through 19 decimal, or 00 through 13 hexadecimal).

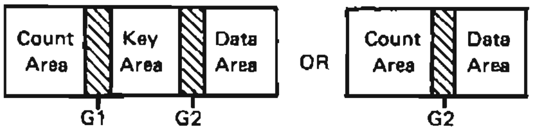
Note: The disk module holding the disk pack is specified by the M-bit in the program instruction used to initiate the I/O operation.

Home Address Cyclic Check and Bit Count Appendage Bytes (Check Bytes)



The two cyclic check and two bit count appendage bytes are generated by the attachment and used by the attachment for error detection and recovery. The leftmost bit count appendage byte is called the BCI byte, and indicates which disk drive wrote the record: hexadecimal C1 = 5445 drive 1, hexadecimal C2 = 5445 drive 2.

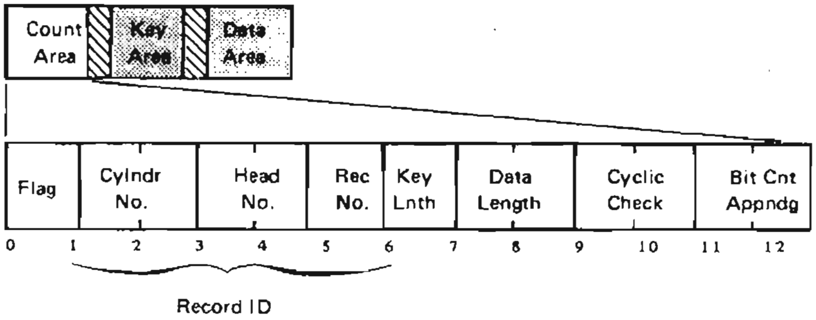
Records (R_0, R_1, R_2, etc.)



Records, consecutively numbered from record zero (R0) upward, fill the track from the home address to the end of the track (detected by encountering the index marker).

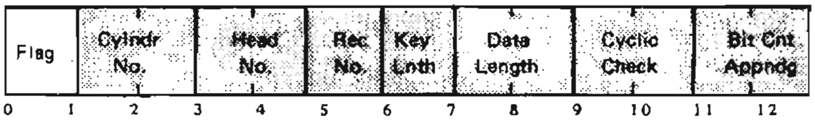
Each record contains a count area and either (1) a data area only, or (2) both a key area and a data area. The number of records that can be formatted on a track is a function of the assigned lengths of the key areas and data areas for the records being formatted.

Record Count Area



The count area identifies the record and defines the number of bytes in the key and data areas of the record. During record operations, the attachment compares the record identification data (cylinder number, head number, and record number) in the disk drive control field in CPU storage with the cylinder number, head number, and record number bytes in the count area of records passing under the read head. A compare equal condition indicates that the desired record is under the read head: this is called *record orientation*. If no orientation occurs, the attachment posts a no-record-found indication.

Record Count Area Flag Byte (F)

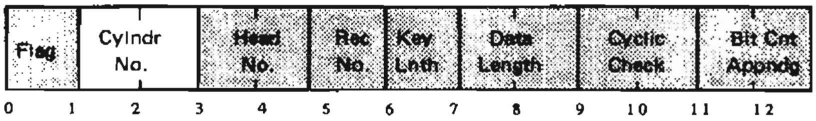


The record count area flag byte is formatted by the 5445 attachment from information stored in the disk drive control field in the CPU. Flag bit significance and their settings are:

Bit	*State*	*Meaning*
0	0	Indicates even-numbered record.
	1	Indicates odd-numbered record.
1	– – –	Not used; should be zero.
2	– – –	Not used; should be zero.
3	– – –	Not used; should be zero.
4	–––	Not used; should be zero
5	–––	Not used; should be zero.
6	0	Indicates operative track.
	1	Indicates defective track.
7	0	Indicates track is a primary track.
	1	Indicates track is an alternate track.

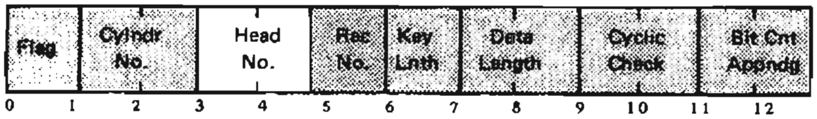
The attachment causes bits 6 and 7 for all records on the track to be set to the values of the corresponding bits in the home address flag byte.

Record Count Area Cylinder Number Bytes (CC)



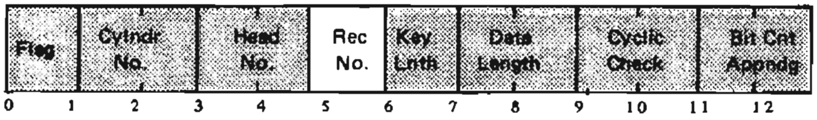
The cylinder number identifies the cylinder within which the record is stored. All bits in the first byte must be zero; the next byte holds the cylinder number, which is assigned by the program. The cylinder number is written from the disk drive control field during a write count key data operation; it is not checked by the 5445.

Record Count Area Head Number Bytes (HH)



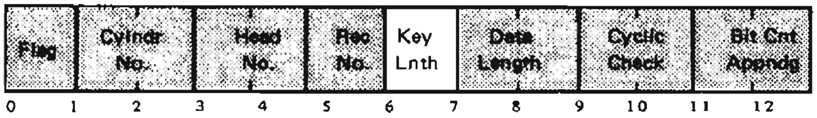
This two-byte field identifies the read/write head associated with the track on which the record is to be placed or from which the record is to be read. The head number, together with the cylinder number, identify the track associated with the record. The bits in the leftmost head-number byte must be all zeros; the second head-number byte holds the head number, which is assigned by the program. The head number is written from the disk drive control field during a write count key data operation; it is not checked by the 5445.

Record Count Area Record Number (R)



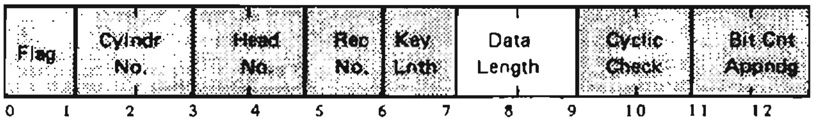
This byte identifies a particular record on the specified track. Records are numbered sequentially on the track, starting with the number assigned by the program to record zero. The number assigned to record zero is not checked by the 5445, but must be hexadecimal 00 for correct disk drive operation. The number of records per track is limited by the addressing capability and by the track capacity. (When the read/write head encounters the index marker point on the disk track during a write count key data operation, the track capacity has been exceeded.) The record number is written on the track from the disk drive control field during a write count key data operation.

Record Count Area Key Length Byte (K_L)



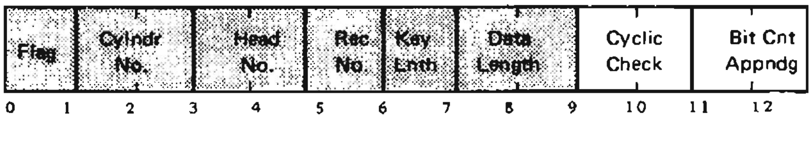
The key length byte specifies the number of bytes in the key area of the record (excluding the cyclic check and bit count appendage bytes, which are check bytes). Valid key lengths are 0 through 255 decimal, or 0 through FF hexadecimal. However, in System/3, the key length is also conditioned by the data length specified, because the total value of the key area plus the data area on the record cannot exceed 256 bytes (decimal). For those installations using IBM programming systems support, the key length must be zero.

Record Count Area Data Length Bytes (D_L)



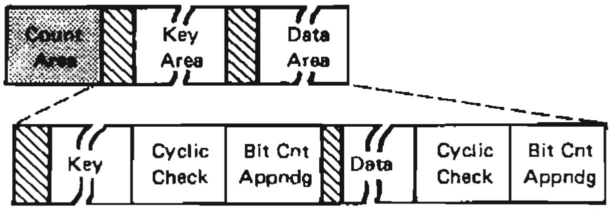
The data length bytes specify the number of bytes in the data area of the record (excluding the cyclic check and bit count appendage bytes, which are check bytes). Valid data lengths are 0 through 256 decimal, or 0 through 100 hexadecimal. However, in System/3, the data length is also conditioned by the key length specified, because the total value of the data area plus the key area on the record cannot exceed 256 bytes (decimal). For those installations using IBM programming systems support, the data length must be 256 bytes for all records except record zero, which is assigned a data length of 8 bytes.

Record Count Area Cyclic Check and Bit Count Bytes



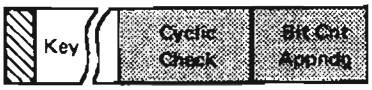
The two cyclic check and two bit count appendage bytes are generated by the attachment and used by the attachment for error detection and recovery. The leftmost bit count appendage byte is called the BCl byte, and indicates which disk drive wrote the record: hexadecimal C1 = 5445 drive 1, hexadecimal C2 = 5445 drive 2.

Record Key Area and Data Area (Key/Data Area)



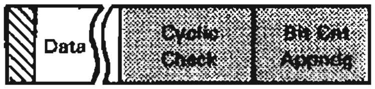
These two record format areas hold the application-oriented information in the record, and should always be considered as a single entity for System/3 programming and operations. For example, the total number of bytes of combined key and data information in a record cannot exceed 256. If the key area is omitted from the record (a key length byte of zero), the gap preceding the missing area is also omitted from the record. (Note: When counting data bytes for the areas, do not consider the cyclic check and bit count appendage bytes as part of the areas.)

Record Key Area Key Bytes



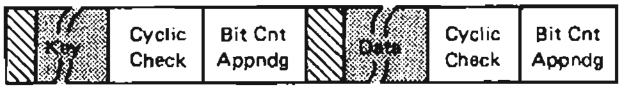
The key-area key bytes can contain record identifying information such as serial number, social security number, or policy number. The number of key bytes in the key area is specified by the key-length byte in the count area, but can never exceed 255.

Record Data Area Data Bytes



The data-area data bytes can contain the information identified by the count and key areas of the record. Data information is organized and arranged by the programmer. The number of data bytes in the data area is specified by the data-length bytes in the count area, but can never exceed 256.

Record Key/Data Area Cyclic Check and Bit Count Bytes

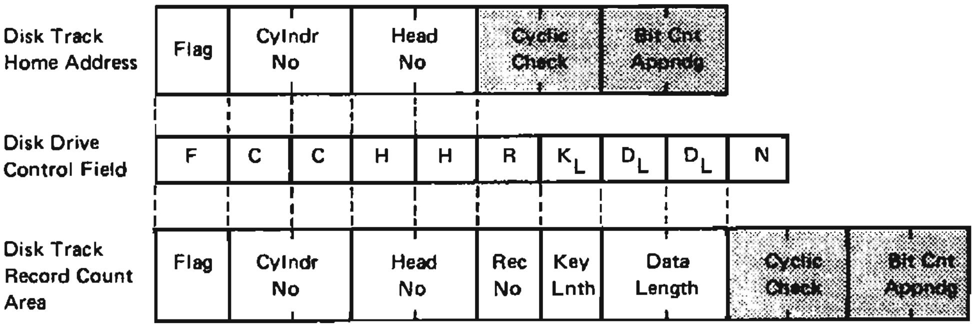


The cyclic-check and bit-count-appendage bytes are generated by and used by the attachment for error detection and recovery. The leftmost byte of each set of bit count appendage bytes is called a BCI (bit count indicator) byte, and indicates which disk drive wrote the record: hexadecimal C1 = 5445 drive 1, hexadecimal C2 = 5445 drive 2.

DISK DRIVE CONTROL FIELD (DDCF)

F	C	C	H	H	R	K_L	D_L	D_L	N

The disk drive control field is a program-defined field in main storage that contains a ten-byte control argument for all start I/O instructions. The DDCF can start on any byte boundary addressed by the disk drive control register (DDCR). As shown below, all the bytes except the N-byte in the DDCF have directly-related bytes in the disk home address and record count areas.



It is generally necessary to preload the defined DDCF with the control argument for the operation before issuing a disk-related start I/O command. Program modification of the DDCF must not be attempted while the disk drive attachment is busy. The functional significance of each DDCF byte except the R-byte and the N-byte is identical to that of the corresponding byte in the disk track record count area.

DDCF R-Byte

F	C	C	H	H	R	K_L	D_L	D_L	N

This byte specifies the sequential number of the record on the track. Valid record numbers are 0 through 255 decimal (00 through FF hexadecimal). The R-byte must match the corresponding byte in the disk count area before record orientation can occur. (Also see "Multiple Fixed-Format Records".)

DDCF N-Byte

F	C	C	H	H	R	K_L	D_L	D_L	N

This byte specifies the number of additional fixed-format records to be operated on. Therefore, a control field with an N-byte of 5 specifies an operation on the addressed record and the following five records. (For additional information about the N-byte, see "Multiple Fixed-Format Records".)

Multiple Fixed-Format Records (Multiple Records)

Fixed format records are defined as contiguous records having equal length key areas and equal length data areas. Therefore, a control field with an N-byte specifying other than zero causes a multiple fixed-format record (often referred to simply as multiple record) operation.

After a record has been successfully operated on, the attachment:

1. Decrements the N-byte by 1. (Decrementing by 1 from an N-byte value of zero places a hexadecimal FF in the N-byte.)
2. Examines the contents of the N-byte. If the N-byte now holds FF, there are no more records to be operated on and the operation ends. If the N-byte contains other than FF, the value contained in the N-byte, plus 1, specifies how many more records must be operated on, and the attachment continues with step 3.
3. Increments the DDCF R-byte by one to specify the next sequential record as the record to be operated on, and
4. Performs the operation specified by the instruction on the record specified by the updated R-byte.

Note: These functions are slightly modified if head switching occurs during the operation (see "Head Switching").

Head Switching

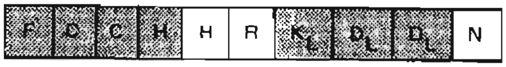
During multiple record operations, a single start I/O instruction can cause as many as 256 records to be operated upon (the record specified by the R-byte, plus another 255 identically-formatted records trailing the specified record in the disk drive, as specified by an N-byte of FF in the DDCF). In many cases, some of the multiple records must be read from the originally-specified track, and the next records must be read from a second track. To operate on records from two different tracks as the result of a single instruction, the attachment switches read/write heads, switching from the presently-active head to the next-higher numbered head after the last record on the original track has been operated upon. During head-switching, the attachment increments the head number by one and resets the record number to one. Therefore, the next record operated on is record one (note that record zero is bypassed) of the newly-selected track.

Note: If head switching occurs from head 19 to head 20 (which is a non-existent head) the file stops with an end-of-cylinder condition posted.

RESIDUAL VALUES

The data held by the DDCF, DDCR, DDDF, and DDDR at the end of each start I/O operation is particularly important for error recovery. These residual values at the end of each normal-end I/O operation are discussed with the write-up about the operation. This section defines residual values when check ending status is posted.

Disk Drive Control Field (DDCF) Residuals



The unshaded portions of the DDCF are updated by the attachment as each record is operated on.

N-Byte: Decremented by one after record orientation.

R-Byte: Incremented by one after record orientation if the last record specified has not been operated on and if check status was not posted for the last record that was operated on. If head-switching occurs, the R-byte is forced to a one so that the first record read from the next track is record one. (Note that record zero is bypassed during head-switching.)

H-Byte: The head number (second H-byte) is incremented by one at the index marker after record orientation if: (1) the last record specified by the instruction has not been operated on by the drive (that is, if the N-byte does not hold FF) and (2) no check status except end-of-cylinder is posted.

- If any of the following check bits are posted, the record identifier portion of the DDCF contains the address of the last record being operated on:

	Byte	*Bit*
Format Error	0	0
Missing address marker	0	2
Data Check	0	4
No record found	0	5
Data overrun	0	7

- If end-of-cylinder status is posted, the R-byte and N-byte residuals are valid and can be re-used when the data operation is restarted on a new cylinder.

The number of records processed can be derived from the residual value of the N-byte:

1. If N= hexadecimal FF, the specified number of records, $N_O + 1$, have been operated on (where N_O represents the value in the N-byte at the start of the operation).

2. If N= hexadecimal FF, the number of records operated on equals $N_O - N$ (where N_O represents the value in the N-byte at the start of the operation, and N represents the residual value in the N-byte).

- If equipment check status is posted, the integrity of the DDCF cannot be guaranteed.

Disk Drive Control Register (DDCR) Residuals

The disk drive control register is returned to its initialized value at the end of any operation in which it is used.

- If an equipment check is posted for the operation, the contents of the register are not guaranteed.

- If end-of-cylinder status (status byte 1, bit 5) is posted during a multiple record operation, then the contents of the DDCR are equal to the initialized value plus 2.

Disk Drive Data Field (DDDF) Residuals

DDDF residuals for any normal operation except read are identical with the initialized data. At the end of a write operation, the DDDF contains the data from the key and data fields of the specified record. If the instruction executed specified the reading of multiple records, then the key and data fields of sequentially-read records will occupy contiguous positions of the DDDF without any indication of where one record ends and the next record starts.

Disk Drive Data Register (DDDR) Residuals

At the end of scan and write count key data operations, the DDDR contains the initial value. At the end of data overrun operations, the DDDR contains the address of the last DDDF position acted upon. At the end of all other operations, the DDDR contains the address of the last DDDF position acted upon, plus one.

5445 TIMINGS

IBM 5445 Disk Access Times

- Minimum: 25 ms
- Average: 60 ms
- Maximum: 130 ms

For more exact access timings, see Figure 6-18.

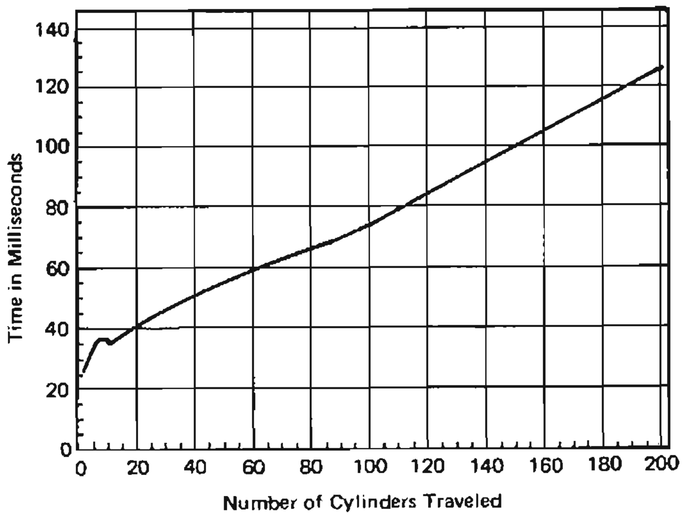


Figure 6-18. 5445 Disk Access Times

Command Execution Times

Generally, command execution time represents that period of time during which the response to a test for I/O attachment busy is positive. The start I/O control commands specifying seek and recalibrate operations require additional seek busy time to complete mechanical motion and head switching. Head switching during multiple record operations also requires additional time. For rough timings, assume an average rotational delay time of 12.5 milliseconds, and a factor of 3.2 microseconds for each byte acted upon. See Figure 6-19 for the command execution timings formula, which allows you to derive more exact command execution timings.

Command Type	Notes	Busy Times	
Control		Seek Busy (Max/Min)	Attachment Busy (Max/Min)
Seek			
Head Select	3	= 30/20 microseconds	30/20 microseconds
Head Motion		= .53(C-1)+ 25 milliseconds where: $1 \leqslant C \leqslant 202$	34/26 microseconds
Recalibrate		130/25 microseconds	50/25 milliseconds
Read		Attachment Busy	
HA and R0 Count	1, 4	= 269 microseconds	
KD	1, 2, 4	$= 3.2 \left\{ \sum_{\text{Record 1}}^{N \div 1} [D_L + (K_L + 45)] + 112 N + 77 \right\}$ microseconds	
CKD	1, 4	$= 3.2 [189 + 2D_L + 2 (K_L + 45)]$ microseconds	
Verify	1, 2, 4	Same as Read KD	
Write			
HA and R0	1	= 25 milliseconds	
C-K-D		Assume an average execution time of 12.5 milliseconds	
K-D	1, 2, 4	Same as Read KD	
Scan	1, 2, 4	Same as Read KD	

Notes:

1 Average rotational delay time of 12.5 milliseconds is not considered.

2 522 microseconds must be added for each head switching action required for multiple record operations.

3 Seek busy for head motion is an approximate formula.

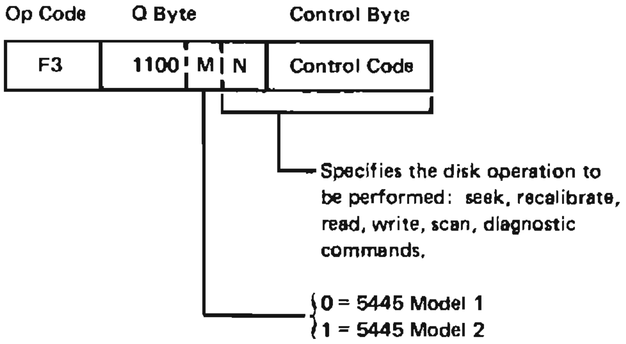
4 The term (K_L+46) must be set equal to zero when $K_L = 0$.
N = Number of records in excess of one to be operated on.

Figure 6-19. 5445 Command Execution Timings

5445 INSTRUCTIONS

Start I/O

Mnemonic: SIO



Op Code
Q Byte
Control Byte
F3
1100
M
N
Control Code
Specifies the disk operation to be performed: seek, recalibrate, read, write, scan, diagnostic commands.
0 = 5445 Model 1
1 = 5445 Model 2

Operation: This instruction selects a drive and disk track and specifies the operation that is to be performed. The N code of the Q byte and bits 5, 6, and 7 of the control code byte specify the operation to be performed.

N Bits 5, 6, 7*	Control Bits 5, 6, 7**	Operation
000	000	Seek.
000	001	Recalibrate.
001	000	Read Key Data.
001	001	Read Home Address and Record Zero.
001	010	Read Count Key Data Special.
001	011	Read Verify Key Data.
001	111	Read Buffer Diagnostic (CE diagnostic).
010	000	Write Key Data.
010	001	Write Home Address and Record Zero.
010	010	Write Count Key Data.
011	000	Scan Key Data Equal.
011	001	Scan Key Data Low or Equal.
011	010	Scan Key Data High or Equal.

* Any N code not listed causes the processing unit to stop with a processor check and an invalid Q byte indicated.

** Control byte bits 0, 1, 2, 3, and 4 are not used; they should be 0.

Issuing any start I/O instruction to a control unit that is busy or issuing a start I/O seek instruction to a drive that is not ready causes an automatic program level advance in systems with the dual programming feature installed; if the feature is not installed, the program loops on the start I/O instruction until the condition is corrected. If the program addresses drive 2 and drive 2 is not installed in the system, a processor check with an invalid Q code indication occurs.

A single start I/O specifying read, write, or scan is provisionally accepted by the 5445 for later execution if either drive is executing a seek. If error conditions are set at the end of the seek when a read, write, or scan has been provisionally accepted, the read, write, or scan command is ignored (no-oped) and the no-op bit is set.

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

The start I/O instruction uses the contents of the disk drive data (address) register as the initial CPU storage address of all disk record data fields. It uses the contents of the disk drive control (address) register as the address of the disk drive control field in CPU storage.

A start I/O addressed to an unsafe drive is not accepted; both the no-op bit (status byte 0, bit 6) and unsafe (byte 1, bit 1) are set.

Example:

Instruction (to read key and data from file 1)

F3	1100	0	001	0000 0000

Disk Drive Control Register

02	00

Disk Drive Data Register

04	00

Disk Drive Control Field

F	C	C	H	H	R	K_L	D_L	D_L	N
00	00	14	00	06	10	00	00	28	00

2000 (CPU Storage Address

The disk drive data field starts at CPU storage address 0400 and ends at 0428. Before the operation, the DDDF should be initialized to blanks; at the end of the operation, the DDDF will contain data from cylinder 14, head 6, record 10 (hexadecimal).

5445 SEEK OPERATION

The seek control command selects one of 4000 primary tracks or one of 60 alternate tracks on the disk drive specified by the DA and N bit portion of the Q byte. After a seek operation, a *cylinder* remains selected until a different cylinder is selected by a subsequent seek or recalibrate operation. A *track* remains selected until a different track is selected by a new seek or recalibrate operation or until automatic head switching occurs. (A *track* that is initially selected by a seek or recalibrate operation will be changed by any subsequent multiple-record read, write, or scan command that causes automatic head-switching to occur.)

The seek command does not verify that the correct track has been selected. Invalid cylinder and head number checking is not performed.

A "zero cylinder seek" (that is, seek to the same cylinder) is provisionally accepted (stacked) while a seek or recalibrate command is being executed. The system executes the command at the end of the seek operation unless equipment check status (byte 0, bit 3) has been posted.

Initial Conditions

DDCF--contains the five-byte "seek" address format (FCCHH) used to specify the "seek to" cylinder and head number. The five remaining bytes in the DDCF are not used.

Not Used	"Seek to" address		Not used for seek operation			
Flag	Cylndr No.	Head No.	Rec No.	Key Lnth	Data Length	N

F--Not used.

CC--A two-byte cylinder number field that specifies the cylinder number. Byte 1 should be hexadecimal 00, and the second byte must be the hexadecimal number of the cylinder. Cylinders are identified by decimal numbers 000 through 202, or hexadecimal 00 through CA. Cylinder numbers are not hardware checked.

HH--A two-byte head number field which specifies the head number. The first byte should be set to zero. The second byte is set to the binary number of the "seek to" head. Decimal head numbers for byte 2 are 00 through 19, or hexadecimal 00 through 13. Head numbers are not hardware checked.

DDCR--Must contain the address of the left-most (high-order) byte of the DDCF.

DDDF--Unchanged.

DDDR--Unchanged.

In-Process Conditions

Test I/O--Selected device seek-busy response is positive until the seek operation has been completed and the read head has settled enough to read data without errors.

Test I/O--Attachment busy is positive:

1. From the time the seek command is issued until the drive has accepted the seek information, or

2. From provisional acceptance of a read, write, or scan command until the operation has been completed.

An overlapped seek operation can be initiated if the selected device is not seek-busy or attachment-busy.

Ending Conditions

DDCF--Remains unchanged.

DDCR--Contains the initialized address. The contents of the register are unpredictable if equipment check status is posted. See "Disk Drive Control Register" description.

Ending Status

See "Ending Status Conditions."

5445 RECALIBRATE OPERATION

The recalibrate control command starts a direct seek to cylinder zero and head zero. Execution is the same as that for the seek command except for command execution times. Initial control and register fields need not be specified and therefore remain unchanged.

READ HOME ADDRESS AND RECORD ZERO OPERATION

Read home address (HA) and record zero (R0), transfers all data from the five-byte home address field (FCCHH) and all data from record zero on the track under the active read head into main core storage. The 5445 locates the home address area, then reads the home address into the disk drive control field in CPU storage and reads record 0 into the disk drive data field in CPU storage. Record 0 key length and data length are obtained from the R0 count area on the actual disk.

Initial Conditions

DDCF--Destination field for the data in the first five bytes (FCCHH) of the home address area of the active track. (These bytes hold the flag data and the track address.)

DDCR--Must contain the address of the left-most byte of the DDCF.

DDDF--Destination field for data from record zero (R0) of the active track.

Field length--(key length + data length + 9).

DDDR--Must contain the address of the left-most byte of the DDDF.

In Process Conditions

Busy to all commands except sense I/O until test I/O "attachment busy" is negative.

Ending Conditions

DDCF--Contains flag byte and track number from the home address area of the track.

DDCR--Contains the initial address. (If an equipment check is posted, the contents of the register are not guaranteed.)

DDDF--Contains data from record zero count field.

DDDR--Contains the starting DDDR value, plus nine.

Ending Status

See "Ending Status Conditions."

READ KEY DATA OPERATION

The read key-data operation transfers one or more disk records from the selected 5445 track into main CPU storage. Reading begins at the record specified by the identifier field (CCHHR) in the disk drive control field in CPU storage. Record orientation is conditioned (that is, the correct record is assumed to have been found on the track) when the flag and identifier fields of a record on the disk track exactly match those fields in the disk drive control field (DDCF) located in CPU storage.

The key and data lengths need not be specified in core, because these lengths are automatically read from the actual disk record by the attachment.

The attachment reads key and data fields into contiguous positions of the disk drive data field (DDDF) in CPU storage. The drive reads one more than the specified number of multiple fixed-format consecutive records (up to a maximum of 256 records) during this operation if the disk drive control field N byte specifies a number greater than zero. As soon as each record is read, the attachment increments the DDCF record number byte (R) by 1, and decrements the DDCF N-byte (N) by 1.

When properly specified by the disk drive control field, record zero (R0) on a track can be read. However, the drive bypasses R0 whenever R0 is encountered after head switching during multiple-record operation. During head switching operations, the attachment selects record one on the next sequential track as the next record to be read, thereby bypassing record zero.

Note: Head switching occurs at index time if record orientation was successful and multiple records are being read.

Initial Conditions

DDCF--Must contain the starting disk record address.

DDCR--Must contain the address of the left-most byte of the DDCF located in CPU storage.

DDDF--CPU storage area to receive the contiguous key and data fields from disk storage. Field Length = (N+1) (key length + data length).

DDDR--Must contain the address of the left-most byte of the DDDF.

In Process Attachment Status

Attachment returns busy to all instructions except the sense I/O.

Ending Conditions

DDCF–Identifier portion contains the address of the last record read. The N-byte portion residual equals hexadecimal FF if all records have been read.

DDDF--Contains contiguous key and data fields read from the disk.

DDDR--Contains the address of the last DDDF location operated on, plus one. That is: (disk drive data record zero + (N+1) (key length + data length) where the disk drive data record zero is the initialized contents.

Ending Status

See "Ending Status Conditions."

READ COUNT-KEY-DATA OPERATION

This instruction recovers a single record under the following circumstances:

- The record being read has a defective count area.
- The key and data lengths of the record being read are unknown.

If record (Rn) is being read, the attachment starts the operation by orienting on record Rn-1 and then spaces over the following key and data fields of Rn-1. Reading begins at the next Rn count area. The drive transfers the first nine bytes of the Rn count area into the DDCF in the CPU. Reading continues with the attachment using- the key and data lengths extracted from the Rn count area, and transferring the contents of the key and data fields from disk record Rn into the DDDF.

Reading can begin at record R0 with the appropriate DDCF specification.

Initial Conditions

DDCF--Must specify the address of the disk record (Rn) to be recovered. The attachment orients on record Rn-1.

DDCR--Must contain the address of the left-most byte of the DDCF.

DDDF–CPU field that will receive the contents of contiguous key and data fields from the disk drive. Field length = (key length + data length).

DDDR--Must contain the address of the left-most byte of the DDDF.

In Process Conditions

The attachment is busy to all commands except sense I/O. For command execution timings, see "Command Execution Times".

Ending Conditions

DDCF--Identifier portion contains the address of the last record read.

DDCR–Contains the initialized left-most byte address of the DDCF.

DDDF--Contains data read from count, key, and data fields on contiguous disk records.

DDDR--Contains the address of the last DDDF location operated on (That is: disk drive data record zero + key length + data length + 9) where disk drive data record zero is the initialized contents.

Ending Status

See "Ending Status Conditions."

VERIFY KEY-DATA OPERATION:

The verify key-data operation performs a "read back" check of the key and data fields. This operation is the same as a normal read key-data operation, except that data transfer does not take place. The attachment performs the "read back" check by comparing generated cyclic-check and bit-count fields with the corresponding fields read from the selected disk. Key and data fields that are read are not compared.

To ensure that data has been written accurately, issue a verify key data instruction immediately after any write command that modifies the key or data fields. Verification begins at the record specified by the identifier portion of the DDCF. The attachment reads the key length and data length from the count field of the record on the disk, so these lengths do not have to be supplied by the program. To verify multiple consecutive records, specify the number of records to be verified, plus one, in the DDCF.

A maximum of 256 records can be verified without reissuing a new command.

Head switching can occur during command execution. However, during head switching operations, the drive starts examining records on the new disk at the index marker and searches until it encounters the record assigned the hexadecimal number 01 before it restarts the verification function. This means that record zero is not verified.

Initial Conditions:

DDCF–Must contain the address of the first record to be verified, and the number of records (N+1) to be verified.

DDCR–Must contain the address of the left-most byte of the DDCF.

DDDF--Not used.

DDDR--Not used.

In Process Conditions:

The attachment is busy to all commands except sense I/O. See "Command Execution Times" for command timings.

Ending Conditions:

DDCF–Identifier portion contains the address of the last record verified. The N-byte portion contains hexadecomal FF if all records have been verified.

DDCR–Contains the initialized left-most byte address of the DDCF.

DDDF--Remains unchanged.

DDDR--Remains unchanged.

Ending Status:

See "Ending Status Conditions."

WRITE HOME ADDRESS AND RECORD ZERO OPERATION

A write HA and R0 operation usually establishes track identify. Each track must be initialized with a write home address and record zero operation before a data operation that involves Record 0 can be performed. Thereafter, records written on the track must be numbered consecutively as the records are first written.

The write HA and R0 operation starts with the disk drive examining bit 1 of the DDCF (disk drive control field) flag byte. Then, when the drive senses the index marker, the drive writes the home address, record zero, and their associated gaps in the following sequence:

1. Gap 4 (G4). This gap contains 73 bytes if the flag byte bit 1 is 0, or 778 bytes if bit 1 is 1. (This data is generated by the drive.)

2. Data from the F, CC, and HH bytes of the DDCF.

3. Two cyclic check bytes, then a BCI (bit count indicator) and a BCA (bit count appendage) byte that are generated by the drive.

4. Gap 5 (G5), which is generated by the drive.

5. Record zero. The FCCHHR portion of the count field in the DDCF is used to format the count area of record zero. Then, the key and data fields for record zero are written onto the disk track. As record zero is written, the drive generates and writes gaps 1, 2, and three, as required.

Note: R must be assigned the hexadecimal number 00 by the program to ensure correct disk operation. However, the drive does not check the program-assigned number during this operation.

6. After record zero has been written, the drive fills the remainder of the track with hexadecimal FF bytes.

Programmers Note:

After the operation is complete, the program should issue a read home address and record zero command. If a check status results during each of several successive rereads, the program should set the flag byte bit 1 to a 1, and re-issue the write home address and record zero command. The program should then assign an alternate track for the defective track, load the alternate track address into the count area of record zero with a write count key data command (that indicates the primary track is defective), then write the entire record zero onto the alternate track specifying the address of the defective track in the record zero count area along with the indication that this is an alternate track. Flag byte bit 1 is not written on the disk record when the bit is being used for displacement control.

Initial Conditions

DDCF–Contains the FCCHHR KL DL DL N field specifications for HA and R0.

FCCHH--HA flag and home address.

FCCHHR-R0 count- area flag and identifier.

KL DL DL--R0 key length and data length specifications.

N-Not used.

DDCR-Must contain the address of the left-most byte of the DDCF.

DDDF-Contains contiguous R0 key and data fields.

DDDR-Must contain address of the left-most byte of the DDDF.

In Process Attachment Status

The attachment is busy to all instructions except sense I/O.

Ending Conditions

DDCF--Contains the original contents with N unchanged.

DDCR-Contains the initialized address.

DDDF--The original contents are unchanged.

DDDR--Contains the address of the last DDDF position operated on plus one.

Ending Status

See "Ending Status Conditions."

WRITE COUNT-KEY-DATA OPERATION

This is a single track initialization operation used to format single or multiple fixed-format records (R0 through Rn). The disk drive starts formatting records at the record specified by the record identifier in the DDCF and formats N+1 records. The drive formats the count, key, and data areas as specified by the DDCF. The FCCHR of the count area is obtained from the DDCF. Key and data fields to be written are obtained from contiguous positions within the DDDF. Corresponding field length counts, KL and DL, are obtained from the DDCF. As the drive writes on the track, the attachment accumulates a KL + DL sum. A sum greater than 256 sets wrong length record (WLR) status and terminates the operation.

If record Rn is to be formatted, the attachment starts the operation by orienting on record (Rn-1), then spaces over (but ignores) Rn-1. The drive then formats record Rn. After (N+1) records have been formatted, the remainder of the track is filled with hexadecimal 'FF' bytes. For orientation on record Rn-1, corresponding CCHHR fields contained in the DDCF and the count area read from disk must compare. (The R byte of the FCCHHR field contained in the DDCF is initially decremented by one for comparison with the corresponding ID field contained in Rn-1.) When record R0 is specified as the starting record, the drive orients on the last two bits of the home address flag byte. If the flag in core equals the flag on the disk, orientation occurs.

The attachment obtains track condition bits 6 and 7 from the flag byte in the DDCF. Bit 0 of the flag byte is always written as a zero in R0, and alternates to one and zero in subsequent records.

WRITE COUNT-KEY-DATA-(FORMATTING) OPERATION

Multiple consecutive fixed-format records can be written on a single track by specifying an N-byte greater than zero. A write C-K-D command must be re-issued for each track to be formatted. Track overrun status is posted if the read head encounters the index pointer before execution all the specified information has been written on the track. The record number (R) in the DDCF is automatically incremented by one and the N-byte is decremented by one as each record is written. The source program is responsible for observing track capacity limitations. The program must verify initialization by issuing an independent read verify key data command in order to meet file performance specifications.

The key and data fields of one record are identical with those of all other records, because the DDDR contains its initial value at the end of formatting each record.

Initial Conditions

DDCF-Contains the initial control field bytes (FCCHHR KL DL DL N) used to specify the starting record address, key and data length counts and the number of records (N+1) to be written.

DDCR--Must contain the address of the left-most byte of the DDCR.

DDDF--Contains the information for contiguous key and data fields of the record to be written.

DDDR--Must contain the left-most byte address of the DDDF.

In Process Conditions

The attachment is busy to all instructions except sense I/O.

Ending Conditions

DDCF--Unchanged.

DDCR--Contains the intiialized DDCF address.

DDDF--Contents remain unchanged.

DDDR--Contains the initialized DDDF address.

Ending Status

See "Ending Status Conditions."

WRITE KEY-DATA OPERATION

The write key-data operation transfers specified key and data fields from main storage to the selected disk drive and track. The attachment compares the flag and identifier field (FCCHHR) of the DDCF with the same flag, and identifier field of the count area read from the selected track. Comparison begins with the first count area read. A successful comparison is called *record orientation.* Following record orientation, the result of the count field comparison and field checking determines how the write operation proceeds.

- *If the DDCF Counts are equal and field checking shows no errors,* then writing begins in the key and data areas of the oriented record. A mismatch sets the no record found status and terminates the operation after field checking.

As the drive writes each record, it generates check field bytes and appends them to each key or data field, as required.

The drive writes multiple fixed-format consecutive records if the DDCF N-byte is greater than zero. After initial orientation, the attachment decrements the N-byte by 'one'. When a multiple-record operation is specified, the attachment updates the DDCF by adding 'done' to the record number (R-byte) and subtracting 'one' from the N-byte as each record is operated on.

Writing can begin at record R0 if the DDCR R-byte in the DDCR specifies 0. However, the drive bypasses R0 if R0 passes the read head after head switching during a multiple-record operation.

Initial Conditions

DDCF--Contains the initial control field bytes (FCCHHR KL DL DL N). Specifies the starting record address, key and data length counts, and the number of records (N+1) to be written.

DDCR--Must contain the address of the left-most byte of the DDCF.

DDDF--Contains contiguous key and data fields to be written onto disk storage.

Length = (N+1) (KL + DL)

DDDR--Must contain the address of the left-most byte of the DDDF.

In Process Conditions

The attachment is busy to all instructions except sense I/O.

Ending Conditions

DDCF--Contains the address of the last record written or attempted to be written.

DDCR--Contains the initialized left-most byte address of the DDCF.

DDDF--Contents remain unchanged.

DDDR--Contains the address of the last DDDF position operated on plus one, or:

Disk drive data record R0 + (N+1) (KL + DL)

Ending Status

See "Ending Status Conditions."

SCAN OPERATIONS

A scan operation compares a record in main storage with a record stored on the disk drive. A "scan under mask" is implemented by inserting hexadecimal 'FF' mask characters into positions of the storage argument that are to be masked out (that is, that are not to be compared).

6

Scan equal, scan high or equal, and scan low or equal operations are provided. A scan hit is a testable state (state--3) within the test I/O instruction. A sense I/O instruction must be issued to determine if a scan equal condition is found during a scan equal operation or scan high or equal operation.

SCAN KEY-DATA EQUAL

The scan key-data equal operation compares the contents of the key and data fields read from the selected disk drive with a corresponding key and data comparison field argument in CPU storage. Comparison begins at the record specified by the identifier field (CCHHR) in the DDCF. Single or multiple-byte fields can be "scanned under mask" by inserting a mask character (hexadecimal 'FF') in byte positions of the CPU storage argument *not* to be compared. N+1 consecutive records can be scanned if the appropriate N-byte in the DDCF is specified. A maximum of 256 records can be scanned by a single instruction. After identifier orientation, the key-and-data-length-count fields specified determine how the scan proceeds: non-zero DDCF counts cause both key-and-data fields to be scanned. A mismatch between count fields sets no record found status and terminates the operation after field checking.

Scanning can begin at record R0 with the appropriate DDCF specification. However, R0 is bypassed if encountered after head switching during a multiple-record operation.

The scan operation proceeds until:

- A scan-equal condition is found.
- N+1 records have been scanned.
- An end-of-cylinder condition is detected.
- An equipment check or a data check is detected.

During the operation, the DDCF record number (R) is incremented by one and the N-byte decremented by one after each record has been scanned. Multiple-record head-switching occurs at index time provided record orientation is successful.

Initial Conditions:

DDCF--Contains the starting record address.

DDCR--Must contain the address of the leftmost byte of the DDCR.

DDDF--Contains the comparison-field argument. The DDDF is partitioned into key-and data fields using the length counts specified in the DDCF.

DDDR--Must contain the address of the leftmost byte of the DDDF.

In-Process Conditions:

The attachment is busy to all instructions except sense I/O.

Ending Conditions:

DDCF--Contains the address of the record in which a scan hit was found.

Contains the address of the next record to be scanned if N+1 records are scanned and a scan hit is not found.

DDCR--Contains the address of the initialized leftmost byte of the DDCF.

DDDF--Remains unchanged.

DDDR--Contains the address of the initialized DDDRo.

Ending Status:

See "Ending Status Conditions." End-of-cylinder status is *not* posted if the operation ends prior to EOC detection.

SCAN KEY-DATA LOW OR EQUAL

This is a scan operation that is similar to scan equal. Field comparison results are based on low or equal conditions. A scan hit condition is set when the specified key and data fields read from the selected disk drive are lower than, or equal to, the masked argument in the DDDF. A scan hit can be tested for by a test I/O (TIO) instruction. If a scan equal condition is found, the scan-equal status bit (byte 1, bit 6) is set.

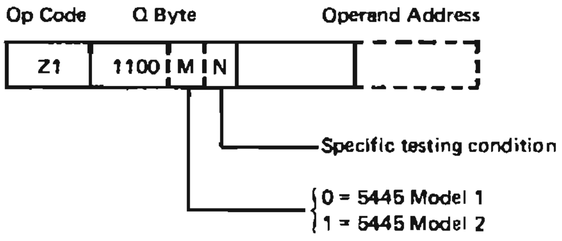
SCAN KEY-DATA HIGH OR EQUAL

This is a scan operation similar to scan equal. Field comparison results are based on high or equal conditions. A scan hit condition is set when the specified key and data fields read from the selected disk drive are higher than, or equal to, the masked argument in the DDDF.

A scan hit can be tested via the TIO instruction. If a scan-equal condition is found, the sacn-equal status bit (byte 1, bit 6) is set.

Test I/O and Branch

Mnemonic: TIO



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

The Q byte specifies the drive to be tested and the condition to be tested. The device address (DA) and the M bit portion of the Q byte specify the disk drive.

The N code of the Q byte can specify testing for any of these conditions.

N Code Condition

000 Not ready/unit check. Not ready state indicates that the addressed disk drive is either:

- Power down.
- In a disk start-up transition.
- Under remote off-line control.

Unit check state indicates that the addressed disk drive has either a disk drive check status outstanding or a common check status. A common check relates to those sections of the attachment that are shared by both disk drives. The common checks are:

- Format error.
- Intervention required
- Missing address mark
- Equipment check
- Data check
- No record found
- No-op
- Data overrun
- Disk drive (file) error
- Unsafe
- End of cylinder
- Scan equal condition.
- Disk drive identification.

A disk drive error (unsafe) or seek incomplete check condition is also indicated if a seek check or unsafe exists for the addressed drive. A seek check or unsafe for the drive not addressed is not indicated. The drive that has the check condition can be determined from the status byte.

001 Seek busy. Indicates that the addressed disk drive is performing a seek or recalibrate operation.

010 Attachment busy. Indicates that the addressed disk drive/attachment is either:

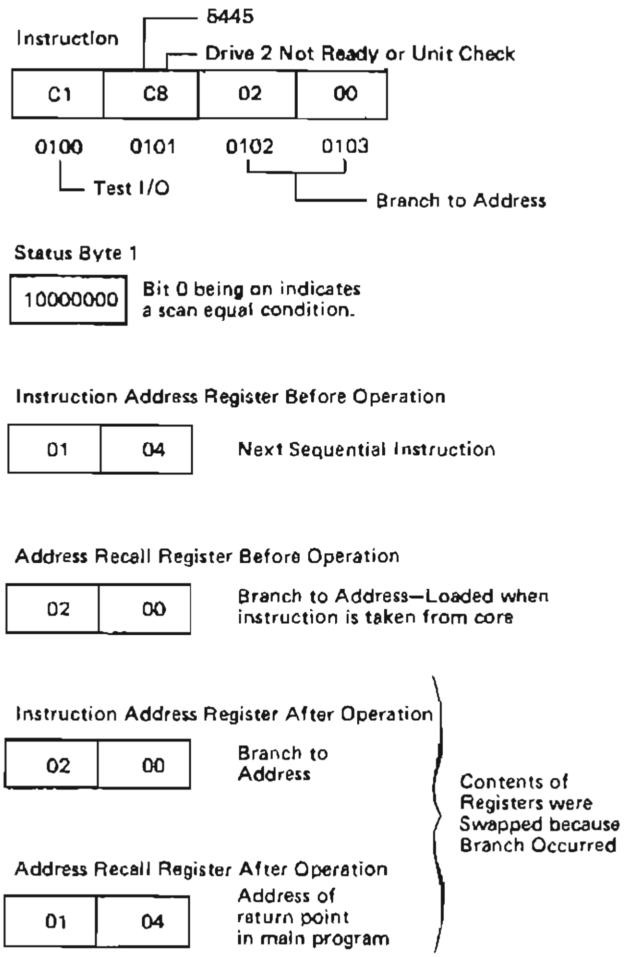
- Executing a read, write, or scan command.
- Has provisionally accepted a read, write or scan command for subsequent execution.
- Is in the starting phase of the seek operation that requires additional CPU cycle steal requests.

011 Scan hit. Indicates that a previously issued scan command was successful. Scan hit is reset at the beginning of the next SIO instruction.

Programming Note: Scan hit is a common state in that an address to any disk drive will give positive indication if either drive has a scan hit. The correct disk drive address can be verified by issuing an SNS instruction and checking status.

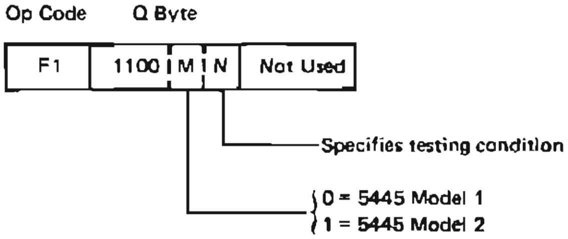
Any N code not listed causes a processor stop with an invalid Q byte indicated.

Example:


Instruction
5445
Drive 2 Not Ready or Unit Check
C1
C8
02
00
0100
0101
0102
0103
Test I/O
Branch to Address
Status Byte 1
10000000
Bit 0 being on indicates a scan equal condition.
Instruction Address Register Before Operation
01
04
Next Sequential Instruction
Address Recall Register Before Operation
02
00
Branch to Address—Loaded when instruction is taken from core
Instruction Address Register After Operation
02
00
Branch to Address
Contents of Registers were Swapped because Branch Occurred
Address Recall Register After Operation
01
04
Address of return point in main program

Advance Program Level

Mnemonic: APL



Operation: This instruction tests for the conditions specified in the Q byte. If the condition tested for is present, a system with dual programming feature installed activates the inactive program level; a system without the dual programming feature installed loops on the advance program level instruction until the condition no longer exists. If the condition is not present, systems with and without the dual programming feature take the next sequential instruction in the active program level.

The Q byte specifies the drive to be tested and the condition to be tested. The device address (DA) and the M bit portion of the Q byte specify the disk drive.

The N code of the Q byte can specify testing for any of these conditions.

N Code Condition

000 Not ready/unit check. Not ready state indicates that the addressed disk drive is either:

- Power down
- In a disk start-up transition
- Under remote offline control

Unit check state indicates that the addressed disk drive has either a disk drive check status outstanding or a common check status. A common check status relates to those sections of the attachment that are shared by both disk drives. The common checks are:

- Format error
- Intervention required
- Missing address mark
- Equipment check
- Data check
- No record found
- No-Op
- Data overrun
- Disk drive (file) error
- Unsafe
- End of cylinder
- Scan equal condition
- Disk drive identification.

A disk drive error (unsafe) or a seek incomplete check condition is also indicated if a seek check or unsafe exists for the addressed drive. A seek check or unsafe for the drive not addressed will not be indicated. The drive that has the check condition can be determined from the status byte.

001 Seek busy. Indicates that the addressed disk drive is performing a seek or a recalibrate operation.

010 Attachment busy. Indicates that the addressed disk drive/attachment is either:

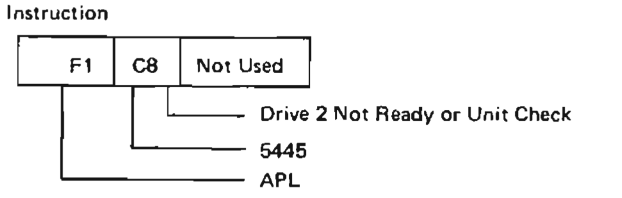
- Executing a read, write, or scan command.
- Has provisionally accepted a read, write or scan command for subsequent execution.
- Is in the starting phase of the seek operation that requires additional CPU cycle steal requests.

011 Scan hit. Indicates that a previously issued scan command was successful. Scan hit is reset at the beginning of the next SIO instruction.

Programming Note: Scan hit is a common state in that an address to any disk drive will give positive indication. The correct disk drive address can be verified by issuing an SNS instruction and checking status.

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

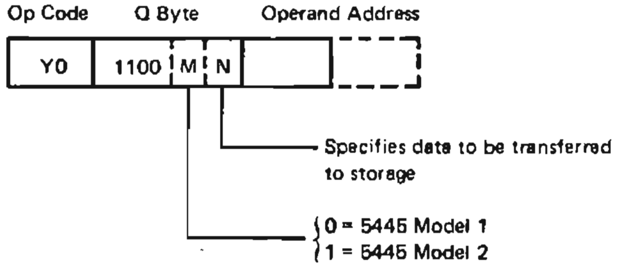
Example:



The next instruction address is taken from the instruction address register of the program level that did not execute this instruction.

Sense I/O

Mnemonic: SNS



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

The Q byte specifies the drive to be sensed and the register or status bytes to be transferred. The device address (DA) and the M bit portion of the Q byte specifies the disk drive.

The N code specifies a certain status byte, the DDDR or the DDCF is to be transferred to storage as follows:

N Code	*Transferred to Storage*
000	Status bytes 0 and 1.
001	Status bytes 2 and 3.
010	Status bytes 4 and 5.
011	Status bytes 6 and 7.
100	DDDR Local Storage Register.
101	Status bytes 8 and 9.
110	DDCR Local Storage Register.
111	Invalid (causes processor check).

The status bytes are bit significant as illustrated in Figure 6-20. The higher numbered status byte is stored in the low-order position of the field. An explanation of each status bit is provided in *Check Conditions and Status.*

Bit	Byte 0	Byte 1
0	Format Error	Disk Drive (File) Error
1	Intervention Required	Unsafe
2	Missing Address Mark	Spare (Not Used)
3	Equipment Check	Spare (Not Used)
4	Data Check	Spare (Not Used)
5	No Record Found	End of Cylinder
6	No-Op	Scan Equal Condition
7	Data Overrun	Disk Drive ID 0=Drive 1 1=Drive 2

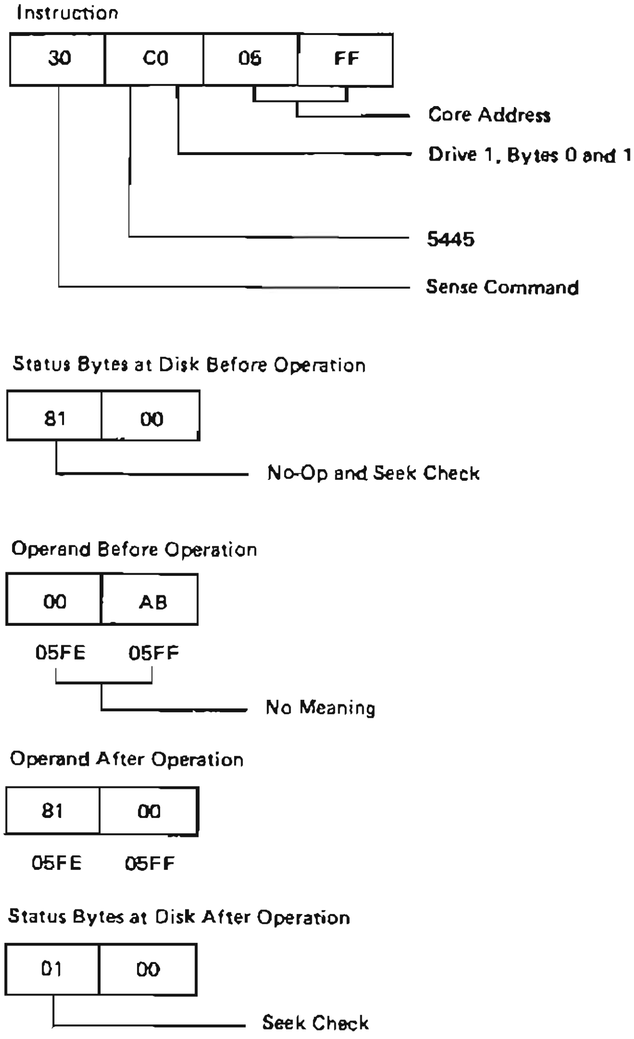
Figure 6-20. IBM 5445 Status Byte Information

The sense I/O instruction is accepted by the 5445 no matter what other operation is in progress at the time.

Some bits of the status bytes are drive sensitive to the sense I/O instruction. Equipment check caused by unsafe, cylinder zero, seek check, seek busy, intervention required, unsafe, head settling, and index are sent with the status bytes only when they apply to the drive addressed by the sense I/O instruction. These bits, if they can be reset by sense I/O, are reset by the same sense I/O instruction that transfers them to storage.

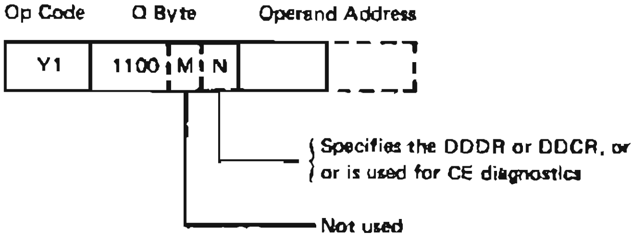
All the status bits not discussed in the preceding paragraph are presented with the status bytes to a sense I/O for either drive. All except no-op are reset by the next start I/O instruction issued to either drive. No-op is reset by the sense I/O instruction to either drive that transfers it to storage.

Example:



Load I/O

Mnemonic: LIO


Op Code
Q Byte
Operand Address
Y1
1100
M
N
Specifies the DDDR or DDCR, or or is used for CE diagnostics
Not used

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

The N code can specify only four values, as follows:

N Code	*Meaning*
100	Specifies the 5445 data address register.
110	Specifies the 5445 disk drive control (address register.
101	Used for CE diagnostics.
111	Used for CE diagnostics.

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

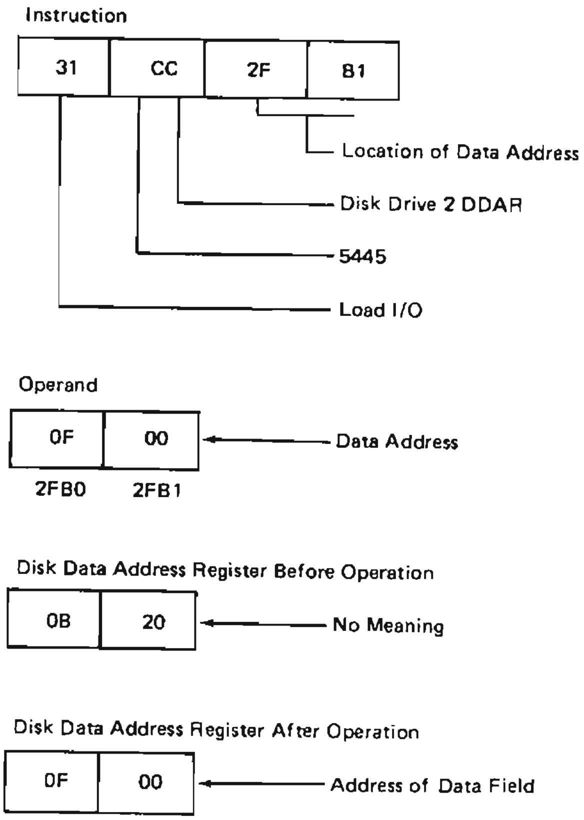
A load I/O instruction issued to a busy control unit causes an automatic program level advance if the system has dual programming feature installed. If the feature is not installed, the program loops on the load I/O instruction until the control unit is no longer busy.

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

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

The load I/O instruction is executed if the addressed drive is not ready, but is rejected if the no-op bit is on.

Example:



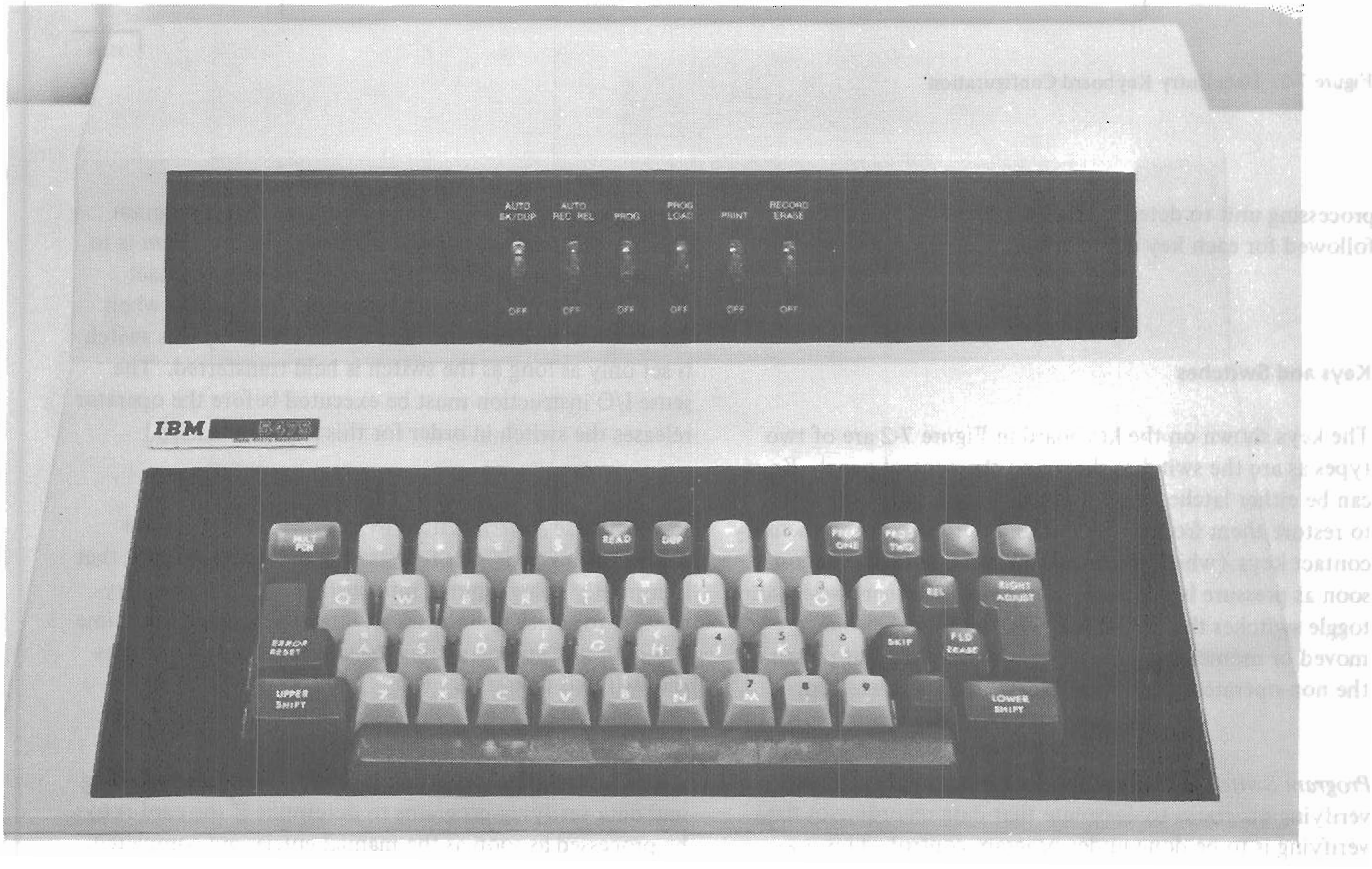
IBM 5475 Data Entry Keyboard

The IBM 5475 Data Entry Keyboard (Figure 7-1) comprises a keyboard, control panel, covers, cables, and a set of attachment circuitry. The keyboard is designed to look and operate as much as possible like the keyboard for the IBM 5496 Data Recorder. Data recording and data verifying can be performed by using the data entry keyboard in conjunction with the card handling capabilities of the MFCU. The functions of data recording and verifying are available when the keyboard and system are used together under control of the data recording and data verifying programs that are available from IBM. The data recording and data verifying functions can be performed when the system is not needed for data processing programs. With the dual program feature installed, data recording and data verifying functions also can be performed while the system is being used for data processing programs.

PRINCIPLES OF OPERATION

Communication between the processing unit and the keyboard is on an interrupt basis. Each time the keyboard needs the processing unit in order to perform its function, it must signal with interrupt. The keyboard is assigned interrupt level 1, which is next to last in interrupt priority.

Pressing the keys on the keyboard (see Figure 7-2 for the keyboard configuration) causes interrupts to occur. Certain switches on the control panel also cause interrupts to occur. Each key or switch also causes some status condition or data byte to appear in status bytes in the attachment. These status conditions and data bytes can be sampled by the



53296

Figure 7-1. IBM 5475 Data Entry Keyboard

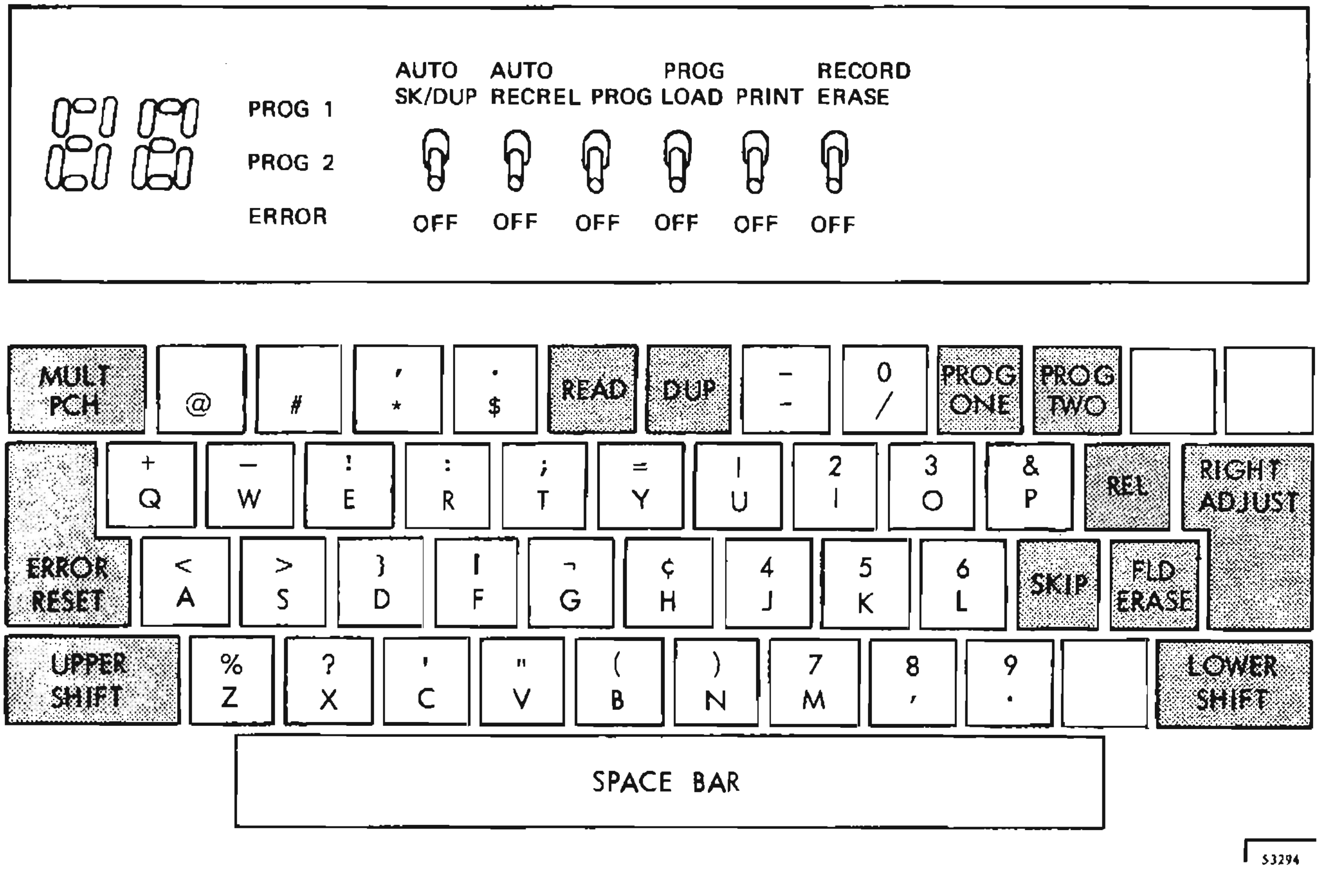


Figure 7-2. Data Entry Keyboard Configuration

processing unit to determine what procedure is to be followed for each key depression.

Keys and Switches

The keys shown on the keyboard in Figure 7-2 are of two types as are the switches shown on the control panel. Keys can be either latched keys (which require specific action to restore them from their operated position) or momentary contact keys (which return to the non-operated position as soon as pressure is released). Switches are either two-position toggle switches that retain the position to which they are moved or momentary-contact toggle switches that return to the non-operated position as soon as they are released.

Program Switch is used by the data recording and data verifying programs to designate that data recording or data verifying is to be done under program control. This is a two-position toggle switch that causes an interrupt request and sets a status bit each time it is transferred from one position to the other.

Program Load Switch is used to indicate that a program card for *the data recording or data verifying program* is to be loaded from the MFCU. It is a momentary-contact toggle switch that causes an interrupt request only when the program switch is on. The status bit set by this switch is set only as long as the switch is held transferred. The sense I/O instruction must be executed before the operator releases the switch in order for this bit to be sensed.

Record Erase Switch is used by the data recording and data verifying programs to erase the data in the record that is currently being entered into storage. This momentary-contact toggle switch causes an interrupt request each time it is operated and maintains the status bit only so long as the switch is operated.

Auto Record Release Switch is used by the data recording and data verifying programs to determine if the card is to be processed as soon as the manual entries are completed. This two-position toggle switch causes an interrupt request each time it is moved from one position to the other and its state is available as a status bit.

Auto Skip/Dup Switch is used by the data recording and data verifying programs to determine if fields coded as automatic operation fields in the program card are to be treated as automatic fields. This two-position toggle switch causes an interrupt request and changes a status bit each time the switch is transferred.

Print Switch determines whether the data being keyed is to be printed on the card after being punched. This two-position toggle switch does not cause an interrupt request but does control a status bit.

Function Keys

The twelve shaded keys in Figure 7-2 are designated as function keys. (The three blank keys do not operate.) Function keys are momentary-contact type and are not interlocked from each other or from the rest of the keys. When an interrupting function key is pressed, the data keys are mechanically locked out and no other function key can generate an interrupt until the first key has been released. If multiple function keys are held down when a sense I/O operation occurs, all the bits will be recorded in the sense byte. If the sense I/O operation is not performed before the function key is released, the function key bit is not recorded in the sense byte. The attachment treats the momentary-contact toggle switches (program load switch and record erase switch) as interrupting function keys and these two rules also apply.

Upper Shift Key conditions the attachment logic to encode upper shift characters. This key does not generate interrupt requests and does not set a status bit.

Lower Shift Key conditions the attachment logic to encode lower shift characters. This key does not generate interrupt requests but sets a status bit if it is held down during a sense I/O operation.

Multi-Punch Key is pressed to place the keyboard in upper shift and, if the processing unit has unlocked the keyboard, causes each data key that is pressed to be restored. The encoded characters associated with the data keys that are pressed are logically ORed in the attachment. When the multi-punch key is released, an interrupt request is generated. If no data key is pressed while the multi-punch key is pressed, no interrupt request is generated by releasing the multi-punch key.

Program 1 Key is used to select the program 1 area as the location of the program control card. This key generates an interrupt request only if the program switch is on. If the key is held down during a sense I/O operation, a sense bit is set.

Program 2 Key is used to select the program 2 control card area. It operates in the same way as the program 1 key.

Release Key signals the end of manual entries on the card. This key generates an interrupt request and sets a status bit when a sense I/O operation is performed while it is pressed.

Field Erase Key is used by the data recording and data verifying programs to signal that the last manually entered field is to be erased. The key causes an interrupt request and sets a status bit if the key is held down until a sense I/O operation is performed to detect it.

Error Reset Key is used to reset program-detected errors. It results in an interrupt request and conditions a status bit if the sense I/O operation is performed while the key is down.

Read Key is used by the data recording and data verifying programs to cause a card to be read into a data area of storage. This key causes an interrupt request and must be held down until the sense I/O operation occurs in order to record the status bit.

Skip Key is used by the data recording and data verifying programs to indicate that the remainder of the field is to be skipped. If the program switch is on, this key generates a single interrupt request each time it is pressed. If the program switch is off, interrupt requests are generated each 1/10 second as long as the key is held down. The down position on the key is recorded in the sense byte if the key is held depressed when the sense I/O operation is performed.

Dup Key is used by the data recording program and the data verifying program to signal that the remainder of the field is to be duplicated from the preceding card. If the program switch is on, this key generates a single interrupt request each time it is pressed. If the program switch is off, interrupt requests are generated each 1/10 second as long as the key is held down. The down position of the key is recorded in the sense byte if the key is held pressed when the sense I/O operation is performed.

Right Adjust Key is used by the data recording and data verifying programs to signal that the data in the field is to be moved to the right end of the field and the remaining left end field positions filled with blanks. Pressing this key generates an interrupt request only if the program switch is on. The right adjust key causes a sense bit if the sense I/O operation is performed while the key is pressed.

Data Keys

The 34 unshaded keys in Figure 7-2 are dual shift keys. These, plus the space bar, are designated as data keys. These keys generate the 63 characters shown on the keyboard plus the code for blank. The data keys are latched type keys and are mechanically interlocked to prevent depressing of more than one key at a time, but a second key can be pressed while the first key is held down if a keyboard restore cycle occurs after the first key is pressed. The attachment generates an interrupt request each time a data key is pressed and the character generated by that key is presented as a sense byte. The data keys must be restored by the processing unit, and a second key cannot be pressed until the processing unit restores the first.

The character generated by pressing a data key depends on the shift of the keyboard. The shift is determined in the following manner:

1. If the program switch is off, the keyboard is in lower shift.
2. If the program switch is on, the keyboard is in upper shift unless the program control card specifies otherwise or the lower-shift key is pressed.
3. With the program switch on, the program control card can specify numeric mode (through a start I/O instruction to the keyboard) for the next entry. In numeric mode, pressing any key other than 0 through 9 or the space bar causes the attachment to turn on the invalid character bit in the sense byte. For the 0 through 9 keys, the attachment operates in upper shift.
4. Any of the preceding shift conditions can be manually overridden by pressing the shift keys or the multipunch key. Manually determined shift states are effective only for as long as the determining key is held down.

Once a data key is pressed, all other data keys are mechanically locked out. The restoring of the data keys is controlled by the program through the use of the start I/O instruction. One bit in the start I/O instruction control code causes the key which has been pressed to be restored, but leaves the keyboard in such a state that all the data keys are locked out. Another control code bit causes the keyboard to be unlocked so that data keys can operate. If the control code contains both bits, a complete restore cycle occurs. However, the following caution should be noted:

> **Caution**
> If a start I/O initiates a complete restore cycle and another start I/O that does not have the unlock-keys control code bit on is issued before the key that was depressed has been restored (about 15 to 20 milliseconds), the data keys will all be locked out.

Indicators

Indicators are provided on the control panel section of the keyboard to indicate the next column in which data will be entered, an error condition has occurred, and the program level control card that is in effect at the moment.

Column Indicators are controlled by the program. The indicators are made up of segments that can be lighted in various combinations to produce the arabic numeral characters. Other characters can be produced, but are not likely to be used for column indication.

Error Indicator is lighted under program control. It is controlled by a bit in the start I/O control code.

Program 1 and Program 2 Indicators are lighted under program control to indicate the program control card level that is in use.

PROGRAMMING CONSIDERATIONS

The following rules must be observed in programming for the data entry keyboard:

1. A start I/O instruction to enable interrupt level 1 must be issued before the keyboard can be used.
2. A start I/O instruction must be issued to unlock the keyboard before the data keys are operable.
3. A sense I/O operation is necessary to obtain data from the keyboard.
4. The processing unit must issue an instruction to restore the data keys.
5. The last instruction in the interrupt routine must be a start I/O instruction to reset the interrupt.

INSTRUCTIONS

Start I/O

Mnemonic: SIO

Op Code	Q Byte	Control Code
F3	0001:0:000	

Operation: This instruction sets the conditions specified in the control code into the attachment.

The Q byte contains only the device address 0001 in the high-order four bits. All other bits of the Q byte are 0.

The control code is bit significant and controls the following conditions:

Bit	*Condition*
0	Programmed Numeric Mode. When this bit is on, the attachment is placed in numeric shift. When this bit is off and the program switch is on, the attachment is placed in upper shift unless the lower shift bit is on. If both the lower shift bit and this bit are on, the attachment is placed in numeric shift. If a data key causes an interrupt, this bit must not be changed before the data is sensed.
1	Programmed Lower Shift. When this bit is on, the attachment is placed in lower shift. When this bit is off and the program switch is on, the attachment is placed in upper shift unless the numeric shift bit is on. If both the numeric shift bit and this bit are on, the attachment is placed in numeric shift. If a data key causes an interrupt, this bit must not be changed before the data is sensed.
2	Error Indicator. When this bit is on, the error indicator is lighted. When this bit is off, the error indicator is turned off.
3	Spare.
4	Restore Data Key. This bit causes the mechanism that restores and locks the data keys to operate. It leaves the keyboard in such condition that all the data keys are prevented from operating.
5	Unlock Data Key. This bit releases the restore and lock mechanism for the data keys. If both the restore and unlock bits are used in the same instruction, the attachment first restores the data keys then unlocks them.
6	Enable/Disable Interrupt. When this bit is on, the attachment is allowed to interrupt the program in progress in the processing unit. When this bit is off, the attachment is blocked from interrupting the processing unit.
7	Reset Interrupt. When this bit is on, it resets the interrupt condition in the attachment and allows the processing unit to return to the program it was processing when the interrupt occurred.

Test I/O and Branch

This instruction is not used with the data entry keyboard. An attempt to execute a test I/O and branch instruction with the data entry keyboard device address (0001) results in a processor check stop with an invalid Q byte indication.

Advance Program Level

This instruction is not used with the data entry keyboard. An attempt to execute an advance program level instruction with the data entry keyboard device address (0001) results in a processor check stop with an invalid Q byte indication.

Load I/O

Mnemonic: LIO

Op Code	Q Byte	Control Code	
Y1	0001:0:000		

Operation: The two-byte field located at the operand address is used to control the segments of the column indicator and to turn on or off the program 1 and program 2 indicators. The operand is addressed by its rightmost byte. The rightmost byte controls the segments of the units position of the column indicator and the program 2 indicator. The high-order byte controls the tens position of the column indicator and the program 1 indicator. The segments of the column indicator are designated by letters as shown in Figure 7-3. Each bit in the bytes controls one segment or a program indicator. When a bit is on, the indicator or segment turns on. When a bit is off, the indicator or segment turns off. The bit assignments for the segments and indicators are:

Bit	*Lights*
0	Segment E.
1	Segment D.
2	Segment F.
3	Segment C.
4	Segment B.
5	Segment G.
6	Segment A.
7	Program Indicator.

The hexadecimal digits to be placed in each byte to obtain the decimal digits are:

Decimal	*Hexadecimal*
0	EE
1	24
2	BA
3	B6
4	74
5	D6
6	DE
7	A4
8	FE
9	F6

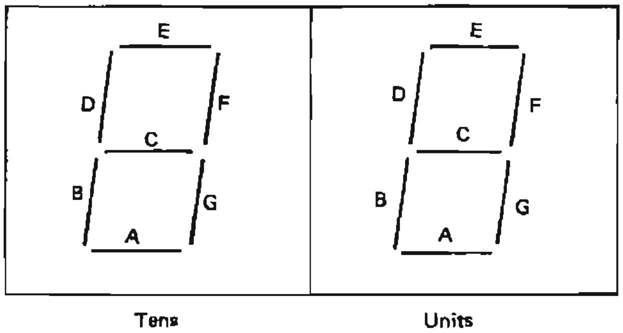


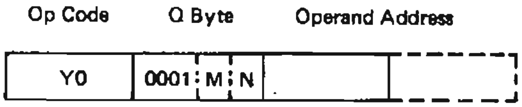
Figure 7-3. Column Indicator Arrangement

If a program indicator is to be controlled, 1 must be added to the low-order hexadecimal digit for the appropriate byte.

The Q byte in this instruction contains the data entry keyboard device address (0001) in the high-order four bits and zeros in the rest of the bits.

Sense I/O

Mnemonic: SNS



Operation: The two sense bytes specified by the Q byte are moved into the two-byte field specified by the operand address. The operand is addressed by its low-order byte.

The Q byte contains the device address (always 0001 for the data entry keyboard), an M bit (always 0), and an N code. The meaning of the sense bytes that are transferred depends upon the value of the N code.

For N code = 010:

Bit	*High-Order Byte*	*Low-Order Byte*
0	Program 1 key pressed.	Auto skip/dup switch on.
1	Program 2 key pressed.	Record erase switch operated.
2	Program load switch operated.	Reserved.
3	Release key pressed.	Program switch on.
4	Field erase key pressed.	Skip key pressed.
5	Error reset key pressed.	Dup key pressed.
6	Read key pressed.	Auto record release switch on.
7	Right adjust key pressed.	Function key interrupt.

For N code = 001 the high-order byte is a data character and the low-order byte is bit significant as follows:

Bit	*Meaning*
0	Print switch on.
1	Reserved.
2	Lower shift key pressed.
3	Invalid character detected.
4	Reserved.
5	Multi-punch interrupt.
6	Reserved.
7	Data key interrupt.

A third pair of sense bytes is provided for use by the CE for diagnosis. The high-order byte of this pair is always all zeros. The pair is obtained by using an N code of 011. The bits in the low-order byte mean:

Bit	*Meaning*
0	Keyboard interrupts enabled.
1	Any function key pressed.
2	Bail forward contacts.
3	Unlock keyboard signal.
4	Bail forward trigger.
5	Toggle switch latch.
6	Any data key.
7	CE sense bit.

The reserved bits are always on (1). The function key interrupt, multi-punch interrupt, and data key interrupt bits indicate the cause of any program interrupt generated by the keyboard attachment. (Function key interrupt is turned on by the interrupting toggle switches as well as by the interrupting function keys.) The following programming requirements exist with regard to these three interrupt bits:

1. One and only one of the three bits should be on any time the keyboard attachment generates a program interrupt request. If none or more than one of these bits is on, a malfunction has occurred. In this case, the keyboard should be locked and the operator forced to try again. This can occur if a data key is pressed and, in servicing the interrupt, the program locks the keyboard by failing to restore that key. If a function key interrupt is generated after this operation, both the function key interrupt (correct) and the data key interrupt (from the unrestored data key) will be on.
2. If an interrupt is generated by changing the state of the program switch, the auto skip/dup switch, or the auto record release switch, the function key interrupt bit is automatically reset 3.3 milliseconds after the interrupt is generated.
3. If the interrupt request is a function key interrupt, the data character should be ignored.

The IBM 5471 Printer-Keyboard (Figure 8-1) comprises a printer-keyboard and a set of attachment circuitry. The printer-keyboard is mounted on the system table top with a forms stand located on the floor behind it. The keyboard and the printer are not physically linked in that key depressions do not automatically cause a character to be printed on the printer. The printer and keyboard are housed together and the printer motor is used to restore the keyboard.

PRINTER CHARACTERISTICS

The printer prints ten characters per inch on a 12.5 inch writing line. The entire 64-character system character set can be printed except for minus zero. The printer signals the attachment when it begins an operation and when it ends the operation. Printing or spacing requires about 64.5 milliseconds per character. Carrier return operates at about 15 inches per second.

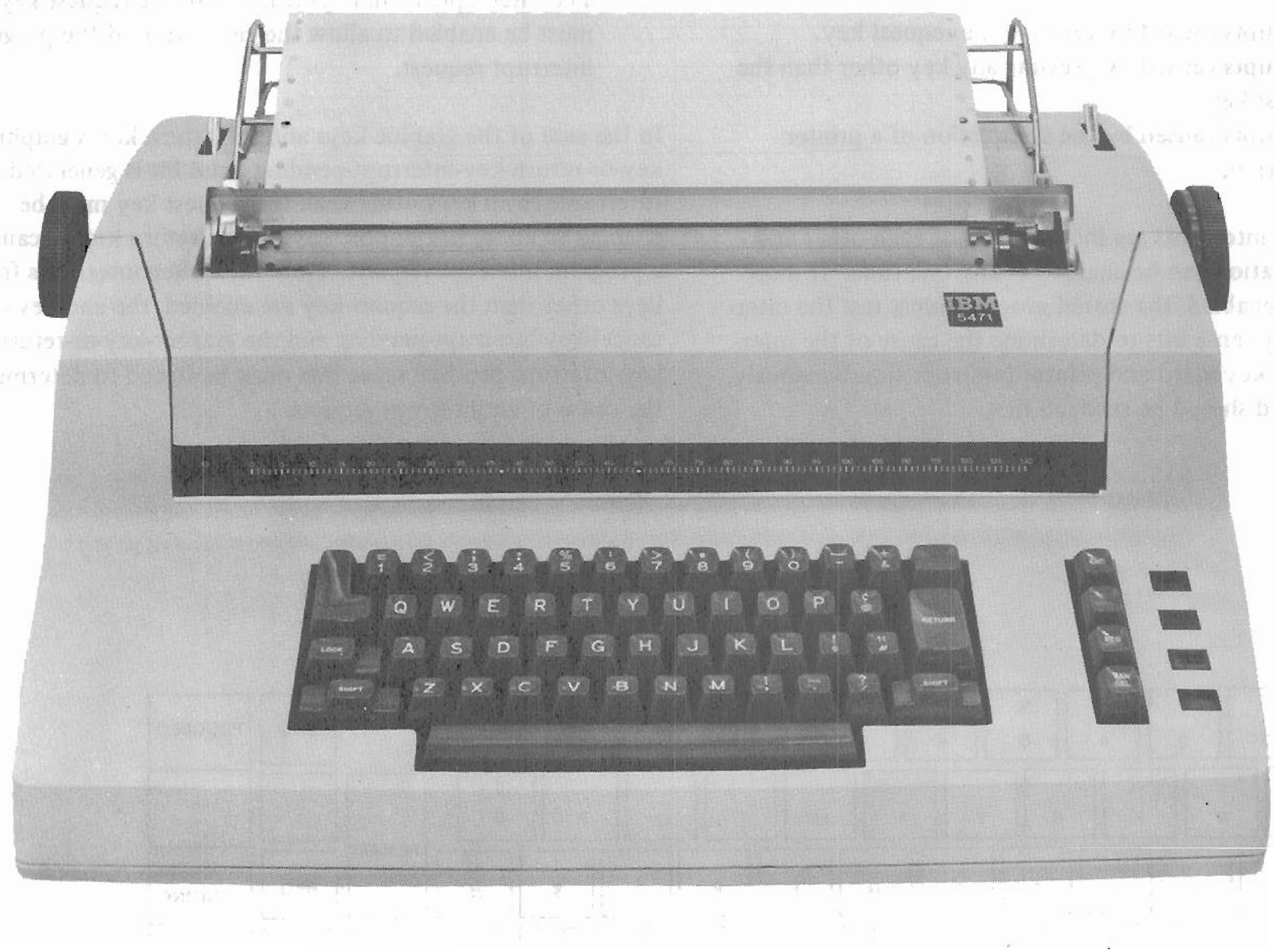


Figure 8-1. IBM 5471 Printer-Keyboard

KEYBOARD CHARACTERISTICS

The keyboard (Figure 8-2) is capable of generating the system character set except for minus zero. The keys are interlocked to prevent pressing two keys simultaneously. In addition to the system graphics set, special codes are generated for shift key depression, shift key release, and return key. Automatic restoration of the keyboard after operation of a graphic, shift, or return key requires about 64.5 milliseconds.

ATTACHMENT CHARACTERISTICS

Keyboard Attachment

Before an operation can be performed on the printer-keyboard, interrupts must be enabled by the processing unit. Three different interrupt conditions can be enabled or disabled.

1. Interrupts caused by pressing the request key.
2. Interrupts caused by pressing any key other than the request key.
3. Interrupts caused by the completion of a printer operation.

All of these interrupts are independent of each other and any combination can be enabled at any one time. If more than one is enabled, the stored program must test the interrupt pending sense bits to determine the cause of the interrupt. If the keyboard and printer interrupt simultaneously, the keyboard should be serviced first.

When the printer-keyboard requires service, the attachment generates an interrupt pending condition. If that interrupt has been enabled by the processing unit, a program interrupt request is generated on interrupt level 1.

Pressing the request key causes the request key interrupt pending. This status remains on until the processing unit issues a reset keyboard interrupt command. If the request key interrupt is enabled or becomes enabled before the interrupt pending is reset, a program interrupt request is generated. If the interrupt pending status is generated while interrupts are disabled or if the enable interrupt and reset interrupt commands are issued simultaneously on the same start I/O instruction, the interrupt is lost.

The end key and the cancel key are treated exactly like the request key with the following exceptions:

1. An end-key-or-cancel-key-interrupt-pending status bit is generated.
2. The interrupts from keys other than the request key must be enabled to allow the generation of the program interrupt request.

In the case of the graphic keys and the return key a graphic-key-or-return-key-interrupt-pending sense bit is generated. Interrupts from keys other than the request key must be enabled to allow the graphic keys or the return key to cause a program interrupt request. Note that when interrupts from keys other than the request key are enabled, the end-key-or-cancel-key-interrupt-pending and the graphic-key-or-return-key-interrupt-pending sense bits must be tested to determine the cause of an interrupt request.

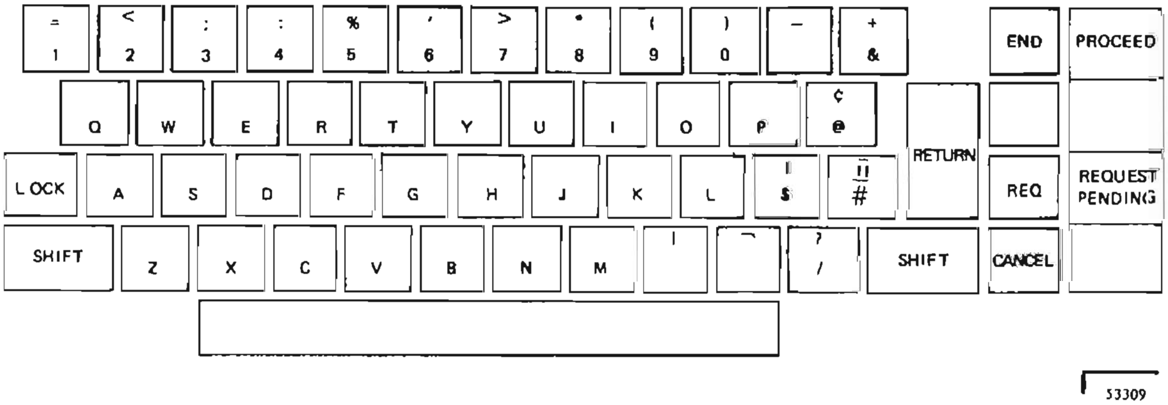


Figure 8-2. Keyboard Format

Graphic characters are handled in the following manner:

1. The graphic keys are encoded to a keyboard code character with parity.
2. The attachment translates the keyboard code character into the appropriate card code character.
3. The card code character is translated to EBCDIC by the translator circuits in the I/O channel when the character is sent to storage.

If a parity error occurs in either the input or the output of the keyboard to card code translator, a corresponding sense bit is also stored.

Printer Attachment

In order to print a character, the character must be loaded into a print character buffer in the attachment by a load I/O instruction. During loading, the I/O channel translates the character from EBCDIC to card code and the attachment then translates the character from card code to a tilt-rotate code used to position the print element. The print mechanism is checked for correct shift at this time. The card code to tilt-rotate translator includes a check bit, and if incorrect parity is detected on the output of the translator, a translator check sense bit is generated. If the character loaded into the print character buffer is outside the printable character set, the tilt-rotate code for a space is established, and a non-printable character sense bit is generated.

After the character has been loaded in the print character buffer, the stored program must issue a start print command through a start I/O instruction. If the print mechanism is in the correct shift, a print cycle starts. If the print mechanism is not in the correct shift, a shift cycle precedes the print cycle.

Carrier return is controlled by the stored program. The carrier is initiated by a start I/O instruction that designates carrier return. Carrier return moves the carrier to the left margin and advances the forms. A carrier return issued when the carrier is at the left margin only advances the forms.

Start print and start carrier return commands cause the printer to become busy. Start I/O and load I/O instructions to the printer will not be accepted while the device is busy. If a start print or start carrier return command is issued while the printer is interrupting without a simultaneous reset interrupt command, it will not be possible to reset the printer interrupt request until after the device becomes not busy. Once the operation is complete, the printer becomes not busy. The transition from busy to not busy generates a printer interrupt pending status. If printer interrupts are enabled, or if they become enabled before a reset printer interrupt command is given, a program interrupt request is generated.

The printer mechanism is checked against the nominal time required for each operation. If the timing is wrong, a printer malfunction sense bit is generated, and a bit specifying the conditions that caused the printer malfunction bit (feedback too late, extra cycle, or cycle too long) is generated. These bits are turned off by a sense I/O instruction that detects them.

The printer contains contacts that detect the approach of the carrier to the end of the print line within 4 to 6 character spaces and the end of the form within 4 to 6 lines. These contacts set sense bits in the attachment.

To aid in servicing the printer-keyboard, the following signals are made available as sense bits:

1. The states of the three enable interrupt latches.
2. The input to the keyboard code to card code translator.
3. The output from the card code to tilt-rotate translator and the printer upper case mode switch.
4. The states and signals from the strobe and feedback contacts.

These bits are provided for servicing and are of no interest to the problem programmer.

INSTRUCTIONS

Start I/O

Mnemonic: SIO

Op Code	Q Byte			Control Code
F3	0001	M	000	

Operation: The printer or the keyboard as specified by the Q byte performs the operation specified by the control code. Spare control code bits should be set to zero.

The Q byte contains the device address (always 0001 for the printer-keyboard) in the first four bits, an M bit that designates either the printer or the keyboard, and an N code of 000. An M bit of 0 specifies that the control code applies to the keyboard. An M bit of 1 specifies that the control code applies to the printer.

The control code specifies the action to be taken. For an M bit of 0 the control code bits cause the following actions:

Bit	*Action*
0	Spare.
1	Spare.
2	Request pending indicator; 1 = on, 0 = off.
3	Proceed indicator; 1 = on, 0 = off.
4	Spare.
5	Request key interrupts; 1 = enable, 0 = disable.
6	Other key interrupts; 1 = enable, 0 = disable.
7	Reset request key or other key interrupts.

For an M bit of 1 the control code bits cause the following actions:

Bit	*Action*
0	Start print.
1	Start carrier return.
2	Spare.
3	Spare.
4	Spare.
5	Printer interrupt; 1 = enable, 0 = disable.
6	Spare;
7	Reset printer interrupt.

Load I/O

Mnemonic: LIO

Op Code	Q Byte	Operand Address
Y1	0001 1 000	

Operation: This operation transfers the high-order byte of the two byte field located at the operand address in storage into the print character buffer in the printer-keyboard attachment. The operand is addressed by its rightmost byte.

The Q byte is fixed with a device address of 0001 in the high-order four bits, an M bit of 1 to designate the printer, and an N code of 000.

This instruction must be used to place the character to be printed in the print character buffer before issuing the start print command. However, if the same character is to be printed several times in succession, it need be loaded only once.

Sense I/O

Mnemonic: SNS

Op Code	Q Byte	Operand Address
Y0	0001 M N	

Operation: Two bytes of sense data selected by the Q byte are transferred from the attachment to the two-byte field addressed by the operand address. The operand is addressed by its low-order byte.

The Q byte comprises the device address (always 0001 for the printer-keyboard), an M bit that determines whether the keyboard or printer sense bytes are to be stored, and an N code that determines which of two pairs of sense bytes for each unit is to be stored. Figure 8-3 shows the M bits and N codes and the bit significance of the sense bytes. Byte 0 or 2 is stored in the high-order byte; byte 1 or 3 is stored in the low-order byte.

The sense bytes (bytes 2 and 3) stored when the N code is 011 are diagnostic bytes and are of little or no interest to the problem programmer.

Test I/O and Branch and Advance Program Level

These instructions are not used by the printer-keyboard. An attempt to use either of these instructions with the printer-keyboard device address results in a processor check stop with an invalid Q byte indication.

M Bit = 0 (Keyboard Sense Bytes)				
N Code = 001			N Code = 011	
Bit	Byte 0	Byte 1	Byte 2	Byte 3
0	Character Keyed	Request Key Interrupt Pending	Keyboard Upper Case Mode Switch	Request Key Enabled
1		End or Cancel Interrupt Pending	Keyboard Data Reed Switch P	Other Key Enabled
2		Cancel Key	Keyboard Data Reed Switch B	Strobe Switch
3		End Key	Keyboard Data Reed Switch A	Strobe Switch Sampled
4		Return or Data Key Interrupt Pending	Keyboard Data Reed Switch 8	Request or End or Cancel Key
5		Return Key	Keyboard Data Reed Switch 4	Request or End or Cancel Key Sampled
6		Keyboard Translator Check	Keyboard Data Reed Switch 2	Keyboard Shifting
7		Keyboard Check	Keyboard Data Reed Switch 1	Reserved

M Bit = 1 (Printer Sense Bytes)				
N Code = 001			N Code = 011	
Bit	Byte 0	Byte 1	Byte 2	Byte 3
0	Printer Enable	Printer Interrupt Pending	Printer Upper Case Mode Switch	Lower Shift Required
1	Reserved	Reserved	No Print	Upper Shift Required
2	Reserved	Non-Printable Character	Tilt-Rotate Code T2	Reserved
3	Reserved	Printer Busy	Tilt-Rotate Code T1	Feedback Switch
4	Reserved	End of Line	Tilt-Rotate Code R5	Feedback Switch Sampled
5	Feedback Too Late	End of Form	Tilt-Rotate Code R2A	Long FN Switch
6	Extra Cycle	Printer Translator Check	Tilt-Rotate Code R2	Long FN Switch Sampled
7	Cycle Too Long	Printer Malfunction	Tilt-Rotate Code R1	CE Sense Bit

Figure 8-3. Printer-Keyboard Sense Bytes

The System/3 Serial Input/Output Channel Adapter (SIOC) provides a means for attaching additional input/output devices for which attachment circuitry is not incorporated in the system. It also provides a means of attaching special units that may be requested by the customer. The control unit of any I/O unit that is to be attached to the SIOC must be designed to be compatible with the SIOC. Only one control unit can be physically attached to the SIOC at any one time, although more than one I/O device can be controlled by that control unit. If the control unit is controlling more than one device, only one device can operate at any time. The SIOC handles data in the form of an 8-bit byte (plus parity). Data is transferred one byte at a time, parallel by bit.

The SIOC provides an intermediate control unit between the system I/O channel and the device control unit. This intermediate control unit produces the necessary signals to control the device control unit from information furnished to the SIOC by instructions from the processing unit, control bytes stored in registers in the SIOC by the processing unit, and information supplied by the device control unit.

SIOC Operational Limitations

Because of the cycle steal priority level and the high data transfer rate, the SIOC can cause overrun conditions to occur when overlapped with other device operations. Therefore, the programmer should exercise caution when overlapping SIOC operations with other devices. Refer to Chapter 2,"Channel Limitations" for allowable overlapped device configurations that will not cause overrun conditions in the system.

SIOC REGISTERS

Data Transfer Register

A nine-bit data transfer register is provided in the SIOC to temporarily store one byte of data (eight bits plus parity) that is to be transferred between the I/O device and core storage. Data transfer is normally on a cycle steal basis, but the contents of this register can be moved between the register and storage with load I/O and sense I/O instructions when this is required by the characteristics of the I/O device involved or for diagnostic purposes. The register is tested for correct parity; a sense bit is set by incorrect parity.

Length Count Register

Because data transfer occurs on a cycle steal basis, the adapter must keep track of the number of bytes transferred. A length count register is provided to perform this function. This counter limits the number of bytes to be transferred to 256 bytes per record. A load I/O instruction is used to place the number of bytes to be transferred in the length count register. The number that is placed in the length count register is the binary representation of a number equal to 256 minus the number of bytes to be transferred. Normally, the I/O device signals when enough bytes have been transferred, but the length count register signals when the correct number of bytes has been transferred, and prevents further data transfer. The contents of this register and the count-exceeded condition can be placed in storage with a sense I/O instruction.

I/O Select Register

This register is used for issuing up to sixteen separate I/O device control signals. It is loaded with a start I/O instruction. The functions that these control signals perform in the I/O device are determined by that device and the data that must be placed in the register for differing conditions will be defined by the I/O device. In general, they will not all be used by any one device.

I/O Transfer Lines

This is not a hardware register but a set of eleven signal lines from the device to the SIOC that can communicate information to the processing unit. These lines can be tested with the sense I/O instruction and used for program decisions based on information received from the I/O unit. The conditions that will be conveyed on these lines are defined by the I/O devices and will be specified in manuals or sections of this manual relating to the I/O device.

Function Register

This register defines the mode of operation of the I/O device. It must be loaded before attempting to execute the program operating the device (it can be loaded by that program before any device operations are attempted). The specific bits that must be stored in this register by a load I/O instruction are defined by the I/O device.

SIOC Data Address Register

This is one of the local storage registers that is used to store the address of the data field that is to be used by the I/O device. The register is loaded by a load I/O instruction and can be sensed by a sense instruction.

SIOC OPERATION

The operation of the SIOC requires that certain I/O instructions be performed to prepare the program and adapter for operation. A means of identifying the individual I/O devices that are attached to the SIOC has been provided. The identification is established at the time the I/O device is designed. These identification lines (four) are stored on a sense I/O operation that specifies the byte that contains their sense bits. The following procedures should be performed to operate the SIOC.

1. Sense the I/O identification byte.
2. Test that an I/O device is attached to the SIOC.
3. Test which I/O device is attached to the SIOC.
4. Load the function register with the appropriate bytes to control the particular I/O device.

I/O operations require that certain instructions be performed before the instruction that transfers data is executed. Before each data transfer operation, the length count register must be loaded with a count equal to 256 minus the number of bytes to be transferred by that operation. The SIOC data address register must be loaded with the address of the first byte of the data field to be operated on. Then the start I/O instruction that actually transfers data can be issued. Testing and sensing operations should be included in the operating program but can be inserted at the discretion of the programmer in accordance with good programming practice.

The SIOC operates in interrupt mode on interrupt level 4. Each time the I/O device requires some special service from the processing unit, such as processing in time for stacker selection, it interrupts the processing unit. Interrupts must be enabled for the I/O device before the SIOC can interrupt the processing unit.

INSTRUCTIONS

The commands for all I/O devices attached to the SIOC are the same; the interpretation given to some of the commands by the I/O devices may be different. The interpretations are discussed in the I/O device sections.

Test I/O and Branch

Mnemonic: TIO

Op Code	Q Byte	Branch-to-Address
Z1	0011 0 N	

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

The Q byte contains the device address (always 0011 for the SIOC), an M bit of 0, and an N code. The N code specifies the condition that is to be tested as follows:

N Code	*Condition Tested*
000	SIOC not ready/check.
001	Invalid.
010	SIOC busy.
011	Invalid.
100	Invalid.
101	Invalid.
110	Invalid.
111	Invalid.

The SIOC not-ready/check-TIOB condition indicates that one or more of these conditions exist:

1. No I/O device is attached to the SIOC.
2. The data transfer register has incorrect parity.
3. A read or write SIO instruction addressed to the SIOC was ignored (no-oped).
4. An attention condition exists at the attached I/O device.

The SIOC busy TIOB condition indicates that the I/O device attached to the SIOC is performing an operation.

Issuing a test I/O and branch instruction with any of the invalid N codes causes a processor check stop with an invalid Q byte indication.

Advance Program Level

Mnemonic: APL

Op Code	Q Byte	
F1	0011:0:N	Not Used

Operation: This instruction tests for the condition specified in the Q byte. In systems with the dual programming feature a change of program level occurs if the condition exists. In systems without the dual programming feature, the processing unit loops on this instruction until the condition no longer exists.

The Q byte contains the device address (always 0011 for the SIOC), an M bit of 0, and an N code. The N code specifies the condition that is to be tested as follows:

N Code	*Condition Tested*
000	SIOC not ready/check.
001	Invalid.
010	SIOC busy.
011	Invalid.
100	Invalid.
101	Invalid.
110	Invalid.
111	Invalid.

Issuing an advance program level instruction with any of the invalid N codes causes a processor check stop with an invalid Q byte indication.

Load I/O

Mnemonic: LIO

Op Code	Q Byte	Operand Address
Y1	0011:0:N	

Operation: This instruction transfers the contents of the two-byte field addressed by the operand address to the register designated by the Q byte. The operand is addressed by the low-order byte. If the SIOC is busy when this instruction is issued, a system with dual programming feature performs an automatic program level advance; a system without dual programming feature loops on the load I/O instruction until the SIOC becomes not busy. If the no-op status bit is on when the LIO instruction is issued, this instruction is ignored and the program advances to the next sequential instruction.

The Q byte contains a device address (always 0011 for the SIOC), an M bit of 0, and an N code. The N code specifies the register to be loaded as follows:

N Code	*Register*
000	Invalid.
001	I/O function register.
010	SIOC length count register.
011	Invalid.
100	SIOC data address register.
101	Data transfer register.
110	Invalid.
111	Invalid.

The bytes loaded into the function register are bit significant as follows:

High-Order Byte

Bit	*Meaning*
0	Write mode set service response.
1	Reset service response after 6 microseconds.
2	Transfer line 2 EOT.
3	Transfer line 1 EOT.
4	Even parity.
5	Decrement DAR.
6	Latch I/O 1 select.
7	Slave (transfer line 6 and 7 latch control).

Low-Order Byte

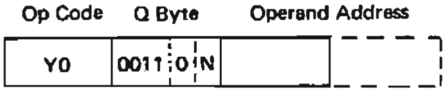
Bit	*Meaning*
0	Diagnostic mode (used only for CE diagnostic testing).
1	Spare.
2	Latch transfer line 4.
3	Latch transfer line 3.
4	Latch transfer line 1.
5	Transfer line 3 reset disconnect latch.
6	Reset disconnect latch after 6 microseconds.
7	Transfer line 5 reset disconnect latch.

The various bits in these two bytes that are set are determined by the I/O device attached to the SIOC at any time, and will be specified by the instructions for programming that device.

Specifying an invalid N code results in a processor check stop with an invalid Q byte indication.

Sense I/O

Mnemonic: SNS



Operation: This instruction causes the two bytes of sense data specified by the Q byte to be transferred to the two-byte field specified by the operand address. The operand is always addressed by the low-order byte. This instruction is executed even though the SIOC is busy or has a not ready/check condition.

The Q byte contains a device address (always 0011 for the SIOC), an M bit of 0, and an N code. The N code specifies the bytes to be sensed as follows:

N Code	*Function*
000	Invalid.
001	I/O function register.
010	Length count register and status byte.
011	I/O transfer lines.
100	Data address register.
101	Data transfer register and diagnostic byte.
110	Invalid.
111	Invalid.

Specification of an invalid N code causes a processor check stop with an invalid Q byte indication.

The status byte and the diagnostic byte are the high-order bytes of their respective sense operations. They are bit significant as follows:

Status Byte

Bit	*Meaning*
0	Spare.
1	End request.
2	Interrupt pending.
3	I/O attention.
4	Data transfer register parity check.
5	No-op.
6	Length count register overflow.
7	I/O ready.

Diagnostic Byte

Bit	*Meaning*
0	SIOC interrupt request latch.
1	Service request.
2	Service response.
3	Interrupt enable.
4	I/O disconnect.
5	Write call.
6	Read call.
7	I/O selected.

The meaning of the I/O attention and I/O ready status bits for an I/O device is described in the chapter about that I/O device.

Bits 0 and 1 of the status byte and all of the bits of the diagnostic byte are for CE diagnostic use and have no meaning to the I/O control program.

The transfer lines are bit significant as follows:

Low-Order Byte

Bit	*Meaning*
0	I/O transfer line 8.
1	I/O transfer line 7.
2	I/O transfer line 6.
3	I/O transfer line 5.
4	I/O transfer line 4.
5	I/O transfer line 3.
6	I/O transfer line 2.
7	I/O transfer line 1.

High-Order Byte

Bit	*Meaning*
0	I/O identifier bit 8.
1	I/O identifier bit 4.
2	I/O identifier bit 2.
3	I/O identifier bit 1.
4	I/O device attached.
5	I/O transfer line 11.
6	I/O transfer line 10.
7	I/O transfer line 9.

The meaning of each of the I/O transfer lines (check condition, device status, etc.) is determined by each individual I/O device control unit and will be specified by manuals discussing that I/O device. Not all the I/O transfer lines will necessarily be used by any one unit.

Start I/O

Mnemonic: SIO

Op Code	Q Byte			Control Code
F3	0011	0	N	

Operation: The start I/O instruction is used to control the mode of operation of the SIOC adapter, and to issue control signals (I/O select lines) to the attached I/O device. A start I/O read or write instruction will electronically attach the adapter to the I/O device by setting it in either the read or write mode respectively. The attachment must be placed in either one of these modes in order for the transfer of data to occur. This instruction is also used to enable or disable the ability of the adapter to request an interrupt priority, if required by the attached I/O device. If an interrupt is requested and the interrupt ability is disabled, the interrupt is kept pending in the SIOC adapter. The interrupt-pending condition can be program-interrogated by a sense instruction. The interrupt request is reset and the SIOC adapter is also removed from the busy state with the SIO instruction.

SIO instructions with N code 000 are accepted and executed by the adapter, regardless of its operating status; SIO instructions with N codes 011 or 100 are accepted and executed by the adapter unless a busy condition exists. A busy condition causes the instruction to be rejected. For systems with the dual program feature, if an SIO instruction is rejected, the program level advances. Without the dual programming feature, the instruction causes the program to loop at the SIO instruction until it can be accepted. When the adapter becomes not busy the instruction is accepted and normal instruction sequencing continues.

If the processor is not executing an SIOC interrupt routine, SIO instructions with N codes 001 or 010 are accepted and executed by the adapter unless an I/O Attention or busy condition exists. In these cases the instruction is rejected as described in the preceding paragraph. When the adapter becomes not busy, or when the cause of the I/O Attention condition is removed, the instruction is accepted and normal instruction sequencing continues.

SIO instructions are no-oped under the three conditions described herein: (1) If an SIO instruction with N code of 001 or 010 is issued when a device is not attached, the instruction cannot be executed. In this situation the instruction is accepted but not executed and the no-op status bit is set. This status bit can be sensed and reset with a SNS instruction. (2) If an SIO instruction with N code 001, 010, 011, or 100 is issued and the no-op status bit is active, the instruction is accepted but is not executed and the no-op status bit remains active. (3) If an SIO instruction with N code 001 or 010 is issued during an SIOC interrupt routine and the I/O attention signal is active, the instruction is accepted but is not executed. The no-op status bit is set and the program advances to the next sequential instruction. This prevents CPU hangup as a result of the I/O attention signal becoming active during the SIOC interrupt routine. The ability to issue and execute SIO instructions with an N code of 000 permits programming to recover from this situation. A reset interrupt request instruction can be used to exit the interrupt routine.

Combinations of the N code not shown in this section are invalid. Figure 9-1 summarizes SIOC operations according to the adapter status.

N Code	*Control Code* *01234567*	*Function*
000	00000001	Reset interrupt request.
000	00000010	Enable interrupt ability.
000	00000100	Disable interrupt ability.
000	00001000	Remove SIOC adapter from busy state.
000	00010000	Set interrupt request.
001	00000000	Read I/O device.
010	00000000	Write I/O device.
011	--------	I/O control 1.
100	--------	I/O control 2.

The I/O control N codes cause the select register to be set with bit significant bytes in the following pattern:

I/O Control Byte 1

Bit	*Meaning*
0	I/O 8 select.
1	I/O 7 select.
2	I/O 6 select.
3	I/O 5 select.
4	I/O 4 select.
5	I/O 3 select.
6	I/O 2 select.
7	I/O 1 select.

I/O Control Byte 2

Bit	*Meaning*
0	I/O 14 select.
1	I/O 13 select.
2	I/O 12 select.
3	I/O 11 select.
4	I/O 10 select.
5	I/O 9 select.
6	I/O unit 2 select.
7	I/O unit 1 select.

CHECKING

The contents of the data transfer register and the I/O channel data are tested for parity errors during data transfer operations and whenever instructions or data is being transmitted over the I/O channel to the SIOC adapter. Detected parity errors on data coming from the processing unit result in a processor check stop with a parity error indication. Parity errors detected in the data transfer register set a data transfer register parity check sense bit that can be tested by a sense I/O instruction.

Instruction (Note 1)	**DBO Parity Error**	**Device Not Attached**	**Busy**	**I/O Attention**		**No-Op Bit On**	**DTR Parity Check (Note 2)**
				Interrupt Routine	**Not Interrupt Routine**		
SIO							
N - Code 0	Processor Check	Execute	Execute	Execute	Execute	Execute	Execute
N - Code 1	Processor Check	No-Op	Reject	No-Op	Reject	No-Op	Execute
N - Code 2	Processor Check	No-Op	Reject	No-Op	Reject	No-Op	Execute
N - Code 3	Processor Check	Execute	Reject	Execute	Execute	No-Op	Execute
N - Code 4	Processor Check	Execute	Reject	Execute	Execute	No-Op	Execute
LIO (all valid N - Codes)	Processor Check	Execute	Reject	Execute		No-Op	Execute
SNS (all valid N - Codes)	Processor Check	Execute	Execute	Execute		Execute	Execute
TIO							
N - Code 0	Processor Check	Branch	Not Applicable	Branch		Branch	Branch
N - Code 2	Processor Check	Not Applicable	Branch	Not Applicable		Not Applicable	Not Applicable

Notes:

1. An invalid instruction causes a CPU check, stopping the system.
2. The data transfer register parity check status bit is reset when the adapter recognizes a valid SIO instruction.
3. When the adapter no-ops an instruction, it accepts the instruction but does not execute it.

Figure 9-1. Summary of Instruction Handling Based Upon Adapter Status

The IBM 1255 Magnetic Character Reader provides the capability of entering data inscribed with magnetic ink characters on paper documents. The 1255 is available in these models:

Model	*Font Read*	*Maximum Throughput***	*Number of Stackers*
1	E–13B	500/min.	6
2	E–13B	750/min.	6
3	E–13B	750/min.	12
21	CMC 7*	500/min.	6
22	CMC 7*	750/min.	6
23	CMC 7*	750/min.	12

* Character font used outside the United States
** Measured with 6-inch documents

A discussion of the capabilities, characteristics, and operations of the magnetic character reader can be found in *IBM 1255 Magnetic Character Reader Components Description,* Form A24-3542.

OPERATION

The 1255 attaches to the system SIOC and operates through the instructions issued to the SIOC. The exact form of these instructions is discussed in the SIOC chapter of this manual.

General Programming Requirements

In addition to the instructions which actually control functions of the reader, the following items must be handled in a specific manner in order for the 1255 to operate with the SIOC:

1. Before executing the instructions that cause the reader to operate, the function register of the SIOC must be loaded by a load I/O instruction. The two bytes loaded must contain a 1 in bits 1 and 5 of the high-order byte and a 1 in bit 6 of the low-order byte. All other bits in these bytes must be 0.
2. The length count register must be loaded by a load I/O instruction issued to the SIOC. The number to be loaded into the register is 256 minus the number of bytes to be read from the 1255. This operation must be performed before each read instruction.
3. The SIOC data address register must be loaded with an address before reading occurs for each read operation. This address designates where in storage the data read from the document is to be stored. The address must be the address of the *low-order* byte of the data field. This register is loaded with a load I/O instruction.
4. The device identification assigned to the 1255 is 0011. The fact that the 1255 is the device attached to the SIOC can be detected by the sense I/O instruction sensing the I/O transfer lines. Bits 0 through 3 of the high-order sense byte stored by this instruction contain the device identification. For the 1255 bits 0 and 1 will be 0 and bits 2 and 3 will be 1.
5. A start I/O instruction must be issued to enable interrupts for the SIOC. The 1255 requires that processing for the documents be performed within specified periods of time to provide correct processing. The 1255 causes an interrupt at the end of every document, and this interrupt must be enabled to allow processing to commence.

Feeding Documents

The 1255 begins to feed documents in the online mode after (1) the 1255 start key has been pressed, and (2) an engage command has been issued. The reader continues to feed documents until:

1. The program issues a disengage instruction,

2. An empty hopper condition occurs,

3. A full stacker condition occurs,

4. The operator presses the 1255 stop key, or

5. A jam, interlock, or late stacker-select condition occurs.

Note: An engage instruction immediately followed by a disengage instruction causes single-document feeding.

A disengage command from the processing unit is required for stopping document feeding under program control. The engage command is issued by executing a start I/O instruction for the SIOC with an N code of 100, and a control code of 00000001. The disengage command is issued by a start I/O instruction with an N code of 100 and a control code of 00000010.

Retrieving Data From Documents

Data is obtained from documents passed through the 1255 by issuing start I/O commands specifying read. A read command must be issued for each document before that document reaches the read head. Failure to issue the necessary read command results in the document's being rejected and an auto reject signal being sent to the processing unit.

For data to be transferred from the 1255, the validity-check-and-readout switches for the desired fields must be pressed.

The 1255 generates an end of transmission (EOT) signal after reading each document and whenever the sorter stops. The EOT signals the SIOC to request an interrupt.

The first character transferred from the 1255 enters storage at the address designated by the SIOC data address register. Subsequent characters enter successively lower storage locations.

Directing the Disposition of Documents

Documents are directed to the stackers in the 1255 by stacker select commands. These commands are generated by start I/O instructions that load the I/O select register. For 500 documents per minute models, the stacker select command must be issued within 24 milliseconds of the time a document leaves the read head (signaled by an interrupt request) if the document is to be stacked in the first (lowest) stacker, or within 50 milliseconds of the document's leaving the read head if it is to be stacked in any other stacker. For 750 documents per minute models, the stacker select command must always be issued within 24 milliseconds after the document leaves the read head. If the stacker select command is not issued within these limits, the document is rejected and the 1255 stops after all documents in the transport are directed to the reject stacker. The fact that the reader is stopped is conveyed to the processing unit.

Obtaining Information about the Condition of the Reader

Indications of the condition of the reader are obtained by issuing a sense I/O command. The sense command is required to determine if the read command was issued in time, if the fields read from the document are valid, where documents are located in the transport, and if the reader is operating.

INSTRUCTIONS

Start I/O

Mnemonic: SIO

Op Code	Q Byte	Control Code
F3	0011:0:N	

Operation: The reader performs the operation specified by the N code and the control code.

The Q byte comprises a device address (always 0011 for the reader) in the first four bits, an M bit of 0, and an N code. The N code in conjunction with the control code, specifies the operation to be performed. The operations performed are:

N Code	*Control Code*	*Operation*
000 or 001	00000001	Reset interrupt request (performed by the SIOC).
000 or 001	00000010	Enable interrupt (performed by the SIOC).
000 or 001	00000100	Disable interrupt (performed by the SIOC).
000 or 001	00001000	Reset SIOC adapter, removing SIOC from busy state (performed by the SIOC).
000 or 001	00010000	Set interrupt request.
001	00000000	Read I/O device.
010	00000000	Invalid for 1255.
011	--------	Control I/O.
100	--------	Control I/O.

The control I/O operations set the I/O select register to produce the desired operation. The following operations can be performed by each N code.

N Code 011

	Models 1, 2, 21, 22	*Models 3, 23*
Control Code Bit	*Operation*	*Operation*
0	Not used.	Select stacker 7.
1	Select stacker 6. **	Select stacker 6.
2	Not used.	Select stacker 5.
3	Select stacker 4.	Select stacker 4.
4	Select stacker 3.*	Select stacker 3.
5	Select stacker 2.	Select stacker 2.
6	Select stacker 1.*	Select stacker 1.
7	Select stacker 0.	Select stacker 0.

N Code 100

	Models 1, 2, 21, 22	*Models 3, 23*
Control Code Bit	*Operation*	*Operation*
0	Not used.	Not used.
1	Select reject stacker.	Select reject stacker.
2	Not used.	Not used.
3	Not used.	Select stacker A.
4	Not used.	Select stacker 9.
5	Select stacker 8.	Select stacker 8.
6	Disengage feed.	Disengage feed.
7	Engage feed.	Engage feed.

Stackers on the 12 stacker readers are arranged in two vertical rows of six stackers each. Stackers on the left bank are numbered, from bottom to top: 0,1,2,3,4, and R. Those on the right bank are numbered 5,6,7,8,9, and A.

Load I/O

Mnemonic: LIO

Op Code	Q Byte	Operand Address
Y1	0011 0 N	

Operation: The two bytes contained in the two-byte field addressed by the operand address are placed in the register designated by the Q byte. The operand is addressed by the low-order byte.

The Q byte comprises a device address (always 0011) in the high-order four bits, an M bit of 0, and an N code. The N code specifies the register into which the contents of the operand field are to be loaded.

N Code	*Destination*
000	Invalid.
001	I/O function register.
010	SIOC length count register.
011	Invalid.
100	SIOC data address register.
101	Data transfer register.
110	Invalid.
111	Invalid.

* Invalid code for standard (even/odd) sort pattern readers

** Invalid code for optional (0-4/5-9) sort pattern readers

Specification of an invalid N code results in a processor check stop with an invalid Q byte indication.

The I/O function register must be loaded with the following:

High-Order Byte	*Low-Order Byte*
01000100	00000010

Sense I/O

Mnemonic: SNS

Op Code	Q Byte	Operand Address
Y0	0011 0 N	

Operation: The two bytes specified by the Q byte are placed in the two-byte field addressed by the operand address. The operand is addressed by the low-order byte.

The Q byte comprises a device address (always 0011) in the high-order four bits, an M bit of 0, and an N code. The N code specifies the sense bytes or registers that are to be sensed.

N Code	*Senses*
000	Invalid.
001	I/O function register.
010	Length count register and status byte.
011	I/O transfer lines and I/O indentification
100	Data address register.
101	Data transfer register and diagnostic byte.
110	Invalid.
111	Invalid.

Specification of an invalid N code causes a processor check stop with an invalid Q byte indication.

The status byte and diagnostic byte are stored as the high-order bytes of their respective sense operations. They are bit-significant as follows:

Status Byte

Bit	*Meaning*
0	Spare.
1	End request.
2	Interrupt pending.
3	I/O attention.
4	Data transfer register parity check.
5	No-op.
6	Length count register overflow.
7	I/O ready.

The I/O transfer lines are bit significant as follows:

High-Order Byte

Bit	*Meaning*	
	Models 1, 2, 3	*Models 21, 22, 23*
0	will be 0. *	will be 0. *
1	will be 0. *	will be 0. *
2	will be 1. *	will be 1. *
3	will be 1. *	will be 1. *
4	1255 attached.	not used.
5	Not used.	Field 7 valid.
6	Not used.	Field 6 valid.
7	Sorter is stopped.	Sorter is stopped.

* If the attached device is an IBM 1255.

Low-Order Byte

Bit	*Meaning*	
	Models 1, 2, 3	*Models 21, 22, 23*
0	Auto reject.	0 Auto Reject
1	Serial number field valid.	Field 5 valid
2	Transit routing field valid.	Field 4 valid
3	Account number field valid.	Field 3 valid
4	Process control field valid.	Field 2 valid
5	Amount field valid.	Field 1 valid
6	Document under read head.	Document under read head
7	Document to be read.	Document to be read

Sorter is stopped is conditioned by the main motor being stopped. A main motor stop is caused by a jam, a late stacker select, an empty feed hopper, or the reader stop key being pressed. This line is deconditioned (bit turned off) by clearing the stop condition and restarting the reader.

All field valid indicators are conditioned when their respective fields including bracketing symbols are read without error, and deconditioned when the leading edge of the next document is sensed at the read head.

The auto reject indication turns on for any document that is rejected automatically by the reader. This occurs if a read command is not issued for a document before the document reaches the read head, a short document, an overly long document, or when a document spacing error occurs. The indicator turns on when the error condition is detected and stays on until the following document arrives at the read head, except that for a document spacing error the indicator stays on through the second document because both documents are rejected. A stacker select command other than reject must not be issued for an auto-reject document to prevent missorting.

The document under read head bit comes on when a document passes under the read head and turns off when the document leaves the read head. It can be used to determine if a document cleared the read head if the read command has been terminated before the end of the document. A stacker select command must not be given for the document until the document leaves the read head.

The document to be read bit is on as soon as the 1255 tries to feed documents. The bit turns off when the document passes under the head after the 1255 stops trying to feed documents. The bit also turns off because of a jam condition between the separator and the read head.

When a hopper runout occurs, the line remains conditioned for about 850 milliseconds after the last document is fed (until the sorter-is-stopped line becomes active).

I/O Ready

This condition indicates that the 1255 is selected for a read operation. When the document reaches the read head, the system must be ready to start receiving data from the 1255.

I/O Attention

The I/O attention condition indicates that normal operator intervention is required on the 1255. Normal operator intervention conditions are:

1. Full stacker.
2. Empty hopper.
3. The 1255 has stopped with the feed light on.
4. Document feeding has been stopped because the stop key has been pressed.
5. No validity-check-and-readout key is in the depressed position.

A jam that occurs in the separator area is indicated as an empty hopper condition. Error conditions (feed jam, transport, interlock, and stacker command) inhibit an I/O attention indication.

Test I/O and Advance Program Level

These instructions operate on the SIOC even though they must be used when operating the 1255. See the SIOC chapter for a discussion of these instructions. The test I/O busy indication means that the 1255 is performing a read operation.

FEATURES

Account Number Checking

For a description of the manner in which account number checking is performed, see the 1255 Components Description manual. If an account number is found incorrect when this feature is installed, the account number field valid indicator bit is turned off. No special programming is involved with the account number checking feature.

51-Column Sort Feature

This feature allows the 1255 to handle documents shorter than the standard documents. These documents lack a transit-routing field. This fact could be used by a program to distinguish 51-column documents from others.

Dash Symbol Transmission

This feature allows the 1255 to transmit the dash symbol from the transit-routing field. Because different nations of the world use the dash symbol in different positions of their transit-routing fields, this fact can be used by programming to distinguish between checks from different countries.

Document Counter

This feature has no effect on programming the 1255 for System/3.

BINARY SYNCHRONOUS COMMUNICATIONS ADAPTER (BSCA)

The binary synchronous communications adapter is a special feature for the IBM System/3 Card and Disk Systems. It provides the system with the ability to function as a point-to-point or multipoint processor terminal. Operation is half duplex, synchronous, and serial by bit, serial by character over either non-switched or switched voice grade or better two-wire, four-wire, or wide band communication facilities.

Operation of the BSCA is fully controlled by a combination of System/3 storedprogram 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.

Two BSCA features can be installed on a single IBM System/3 Model 10 for concurrent operation as independent stations on separate communications networks.

Point-to-Point Communications Networks

The BSCA functions in either a switched or non-switched 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 two-second timeout can be programmed to resolve any contention situations that may occur.

System/3 can be designated, by programming, as either the primary or secondary station.

Multipoint Communications Networks

IBM supports System/3 both as a tributary station and as a control station on a multipoint network.

Data Rates

The first BSCA can operate at various data rates between 600 bits per second (baud) and 50,000 bits per second. The customer selects the data rate to be used, and his BSCA is equipped with an appropriate interface as a no-charge selective feature. Interconnected units must operate at the same data rate. The second BSCA operates at a maximum rate of 7200 bits per second.

Data Set Interface

The data set interface modifies the BSCA for operation on voice grade communications channels. This interface makes possible data rates between 600 and 7200 bits per second, provided the appropriate data set is installed. (For information about acceptable data sets, or their equivalents, consult your IBM sales representative.)

Local Attachment Feature Interface

The BSCA can be equipped with an EIA local attachment feature that allows the BSCA to communicate with an IBM 3270 Information Display System located in the immediate area (without the use of a data set).
With this feature attached, the data rate is either 2400, 4800, or 8000 bits per second, as specified for the installation.

Data Station Interface

The data station interface modifies the first BSCA for operation on wide band communications channels at data rates between 19,200 and 50,000 bits per second. (For information about acceptable data sets, or their equivalents, consult your IBM sales representative.)

Data Sets (Modems)

The data set 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, then makes them available serially by bit, serially by character to the data set. The data set places each bit on the communications line as soon as it receives the bit from the BSCA.

The customer must consider which data set he will be using at the time he orders his BSCA feature.

Transmission Rate Control

A timing device called a clock controls the rate at which data is transmitted and received. For the data set interface, clocking is furnished either as a special feature for the BSCA or else by the data set, depending on which type of data set is selected. For the data station interface, the data set must furnish the clock. Clocking is furnished as part of the feature when the EIA local attachment feature is installed.

Transmission Codes

Data can be transferred in either of two codes, extended binary coded decimal interchange code (EBCDIC) or the IBM version of the American National Standard Code for Information Interchange (American National Standards Institute, 3.4–1968. This code is called ASCII in this publication.) The customer must specify which code he will use at the time he orders the BSCA feature. (Only units using the same code can communicate with each other.)

EBCDIC is the standard, 8-bit plus parity, internal binary code of the IBM System/3. (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 code plus parity. It is illustrated in Appendix B. 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 11-1).

All characters are transmitted over the line low-order bit first. For ASCII, the high-order bit must be a zero bit from core on transmit. If the adapter does not receive a high-order zero from core, 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 core position and so on. For ASCII, the adapter fills a zero into the high-order bit position in core except when the character has a VRC error.

EBCDIC and ASCII have different coding structures to represent characters. When ASCII is used with System/3 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
Auto Call Dial Digit (BCD)	X X X X	8 4 2 1

Figure 11-1. Bit Positions and Significance

SUBFEATURES OF THE BSCA

Two subfeatures of the communications adapter are standard: intermediate block checking and auto answer. The auto answer feature (switched network only) enables the communications adapter to respond to a telephone request for data communications automatically without operator intervention if the data set has unattended answer capability. The intermediate block checking feature allows transmission and reception of checking characters for checking the accuracy of communication without interrupting the steady flow of information from the transmitting station to the receiving station.

In addition to the two standard subfeatures, certain optional subfeatures are offered to enhance the capabilities of the communications adapter.

Station Selection (Special Feature)

This feature allows the system to operate as a tributary station in a multipoint communications network. This feature excludes the auto call feature and is not available with the high-speed interface selective feature.

Internal Clock (Special Feature)

This feature provides an internal clocking system in the communication adapter to allow operation with data sets that do not provide clocking to the adapter. The internal clock feature provides the following transmission rates:

600 bits per second
1200 bits per second
2000 bits per second
2400 bits per second.

Only one of the above transmission rates can be specified for each communication adapter. (Stations can communicate only with other stations using the same transmission rate.) This feature excludes the high-speed interface selective feature.

High-Speed Interface (No-Charge Selective Feature)

This feature (which is used only with the first BSCA) enables the communication adapter to interface with data sets that provide data rates between 19200 bits per second and 50000 bits per second. This feature excludes the internal clock feature, so the data set must furnish data clocking when this feature is installed.

1200 BPS INTEGRATED MODEM SPECIAL FEATURE

This feature eliminates the need for a stand-alone modem between either the first or second BSCA feature and telephone facilities. The 1200 BPS (bits per second) Integrated Modem special feature lets the BSCA operate at 1200 bits per second on either (1) a leased half-duplex or duplex network or (2) a switched network.

The 1200 BPS feature is housed in the BSCA feature inside the CPU. Data interchange with the communications facility is serially by bit and serially by character using frequency shift keying (FSK) modulation. The BSCA internal clock is a prerequisite feature and perforns modem clocking.

The 1200 BPS feature is available in two versions:

- *The leased line (non-switched) version* attaches to a Type 3002 line facility by means of a cable supplied for the 1200 BPS feature.

- *The switched line version* provides automatic answering as a standard function and attaches to a Type CBS or equivalent common carrier arrangement by means of a cable supplied for the 1200 BPS feature.

Neither version can be installed on a BSCA that has the auto call feature.

Note: If the 1200 BPS feature is installed in a BSCA that has the rate selection feature (not available in the United States), the modem can operate at either 600 bits per second or 1200 bits per second under switch control.

Auto Call (Special Feature)

This special feature permits automatic connection with a remote station on a switched network to be established by means of a program instruction. An auto calling unit (ACU), not supplied by IBM, must be used with this feature to enable the automatic connection to occur. This feature excludes the station selection feature.

Full Transparent Text Mode (Special Feature)

This feature allows all the 256 possible bit combinations available in the EBCDIC to be transmitted through the communications adapter as data. This feature is necessary because certain of the EBCDIC characters are designated as line control characters and cause the communications adapter to perform a function. The transparency feature allows these control characters to be handled as data. This feature excludes the ASCII option.

Rate Select Switch (Special Feature)

Systems installed outside the U.S.A. that use data sets capable of operating at two rates are equipped with rate select switches. The rate select switch allows the system to operate at either 600 bits per second or 1200 bits per second, according to the switch setting selected.

EIA Local Attachment (Special Feature)

The EIA local attachment feature allows attachment of an IBM 3270 Information Display System (via an IBM 3271 Control Unit) or an IBM 3275 Display Station in the same local environment without adapting the data signals from either the BSCA or the 3271 for network transmission. The local attachment feature is installed in the 5410; it is equipped with a female connector to which the signal cable from the 3271 is connected. The feature supplies clocking for both the BSCA and the 3271 at data rates of either 2400 or 4800 bits per second, as specified for the installation.

The EIA local attachment feature excludes the internal clock special feature and the attachment of any data set or IBM line adapter to the BSCA housing the EIA feature.

LOCAL COMMUNICATIONS ADAPTER (LCA)

The local communications adapter feature allows direct attachment (no data set/modem) of an IBM 3471 Data Station Model 2, an IBM 3271 Control unit, or an IBM 3275 Display Station to an IBM System/3 Disk System. The LCA is installed in the IBM 5410 Processing Unit. The external (data set/modem) cable furnished with the attached device (3741-2, 3271, or 3275) is plugged directly into a connector provided with the LCA feature.

Only one device may be physically attached to the LCA at a time. The LCA provides clocking at a rate of 2400 bits per second for the attached device and operates in a point-to-point, non-transparent mode using extended binary coded decimal interchange code (EBCDIC).

The LCA cannot be installed on a system with an installed first BSCA, and only one LCA can be installed. However, a system can be equipped with an LCA and a second BSCA feature. None of the BSCA subfeatures can be used with the LCA feature.

Registers and programming required for operation of the first BSCA feature are used for the LCA feature.

LOCAL STORAGE REGISTERS USED BY COMMUNICATIONS ADAPTERS

Three local storage registers (two of which are located in the adapter) are provided for the communications adapter; the current-address register, the transition-address register, and the stop-address register. These registers hold the storage addresses of data or line control characters at which certain actions are to occur, or the address of the next byte to be transmitted or received.

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 plus one under control of the adapter during every I/O cycle steal.

Transition Address Register

The transition-address register stores 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 on cycle steals and begins stealing I/O cycles to store the characters received from the communications line.

Stop Address Register

The stop-address register stores 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.

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 (Figure 11-2)

Note: When transmitting, the adapter turns around 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 (C.O.D.) sequence is at a location one less than the Transition Address. When receiving, any C.O.D. character or sequence causes the adapter to terminate the receive operation and issue an op end interrupt request.

- *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.
- *For recognition of EOT or NAK* as a control character, the adapter requires that four contiguous "1" bits must be received immediately following the EOT or NAK. Also, the EOT character must be the first non-SYN character after establishing character sync. The four "1's" are stored in the four low-order bit positions of the core location following the EOT or NAK. The four high-order bit positions of this byte should be ignored. On Transmit, the adapter automatically generates the four contiguous "1" bits by sending the trailing PAD character.

Name	Mnemonic	EBCDIC	ASCII
Start of Heading	SOH	SOH	SOH
Start of Text	STX	STX	STX
End of Transmission Block *	ETB	ETB	ETB
End of Text *	ETX	ETX	ETX
End of Transmission *	EOT	EOT	EOT
Enquiry *	ENQ	ENQ	ENQ
Negative Acknowledge *	NAK	NAK	NAK
Synchronous Idle	SYN	SYN	SYN
Data Link Escape	DLE	DLE	DLE
Intermediate Block Character	ITB	IUS	US
Even Acknowledge *	ACK 0	DLE (70)	DLE 0
Odd Acknowledge *	ACK 1	DLE/	DLE 1
Wait Before Transmit—Pos. Ack. *	WACK	DLE,	DLE;
Mandatory Disconnect *	DISC	DLE EOT	DLE EOT
Reverse Interrupt *	RVI	DLE@	DLE <
Temporary Text Delay *	TTD	STX ENQ	STX ENQ
Transparent Start of Text	XSTX	DLE STX	
Transparent Intermediate Block	XITB	DLE IUS	
Transparent End of Text *	XETX	DLE ETX	
Transparent End of Trans. Block *	XETB	DLE ETB	
Transparent Synchronous Idle	XSYN	DLE SYN	
Transparent Block Cancel *	XENQ	DLE ENQ	
Transparent TTD *	XTTD	DLE STX DLE ENQ	
Data DLE in Transparent Mode	XDLE	DLE DLE	

* Change of direction character.

Figure 11-2. Control Characters and Sequences

- *ENQ* resets data mode in the adapter.

- *SYN* is generated and transmitted automatically by the adapter to establish and maintain synchronism. SYN does not enter BCC or core. A SYN from core at the transmitting station is transmitted, but does not enter core at the receiving station nor BCC accumulation at either station.

- *SYN SYN* is the sync pattern in non-transparent 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.

- *ITB* is included in the BCC and causes the BCC (s) to be sent or received. Both adapters continue in data mode with the new BCC accumulation starting with the first non-SYN character.

- *DLE* alerts the adapter to test the following character for a defined control sequence. In non-transparent data mode, DLE is treated as data.

- *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 two character DLE sequence does not enter BCC or core. The second character does, if it is not SYN. Also, the transmitting adapter inserts a DLE for each DLE received from core.

- *XSYN* is the sync pattern for maintaining synchronism in transparent mode. It does not enter BCC or core.

- *XENQ* resets data mode and transparent mode in the adapter.

- *XETB or XETX* causes the same adapter action as ETB or ETX and, in addition, resets transparent mode.

- *XITB* causes the same adapter action as ITB and, in addition, resets transparent mode.

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 hexadecimal FF.

The BSCA also generates and transmits a hexadecimal 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 hexadecimal 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 SYN's required for establishing character synchronism at the receiving adapter. The receiving adapter establishes character synchronism by decoding two consecutive SYN's.

An adapter with internal clock feature or EIA local attachment feature establishes and maintains bit synchronism on its own. For this purpose, the BSCA automatically send two additional HEX "55" characters 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 core. In transparent mode, the transparent synchronous idle 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 BCC (or LRC/VRC for ASCII) characters as required for establishing and maintaining synchronism with the remote station and for error checking. When receiving, the BSCA removes all SYN and BCC (or LRC/VRC) characters and some PAD characters received from the data being sent to the storage. The PAD character following a 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.

INTERRUPTS

The BSCA initiates two types of level 2 interrupts: operation end (op end) interrupts and intermediate text block (ITB) interrupts. Whenever an interrupt occurs, the program must determine, by means of TIO ITB interrupt and TIO op end interrupt instructions, the type of interrupt that has occurred and which BSCA is affected. The ITB interrupt latch and the op end interrupt latch are reset by their respective TIO instructions; both latches are reset by disable BSCA.

The interrupt pending condition, which is set by either the op end or ITB interrupt latch, is remembered until it is reset by an SIO reset interrupt request instruction. When interrupts are disabled, the interrupt latches operate as when enabled, except that interrupt pending does not signal an interrupt request to the CPU.

When two BSCAs are installed on System/3, determine which BSCA is originating the interrupt request by issuing a TIO interrupt pending instruction (see Figure 11-3). Interrupt pending indicates that either an ITB interrupt or an Op-End interrupt is needed by the tested BSCA. After determining which BSCA caused the interrupt, the program can issue appropriate TIO Op-End and TIO ITB instructions, using the appropriate M-bit to specify the BSCA requesting the interrupt.

All BSCA interrupt requests should be serviced by routines similar to the one shown in Figure 11-3. Note that both types of interrupt must be tested and the ITB interrupt must be tested first.

Op End Interrupt

If enabled, an op end interrupt occurs at the end of the following BSCA operations:

Auto Call
Transmit & Receive
Receive Initial
Receive
Loop Test
Two Second Timeout (The BSCA need not be enabled to complete the two second timeout operation with an op end interrupt.)

For auto call, an op end interrupt occurs after the connection has been established or the call has been abandoned.

In a receive type operation, an op end interrupt is generated when a C. O. D. character is decoded, when the current address equals the stop address, or when a receive timeout occurs.

In a transmit only operation (see "Start I/O, Transmit and Receive Function"), the interrupt is generated when the current address, transition address, and stop address are all equal. In addition, if an adapter check occurs on transmit, the operation is immediately terminated and an op end interrupt is generated.

In a loop test diagnostic operation, an op end interrupt is generated when the current address is equal to the stop address.

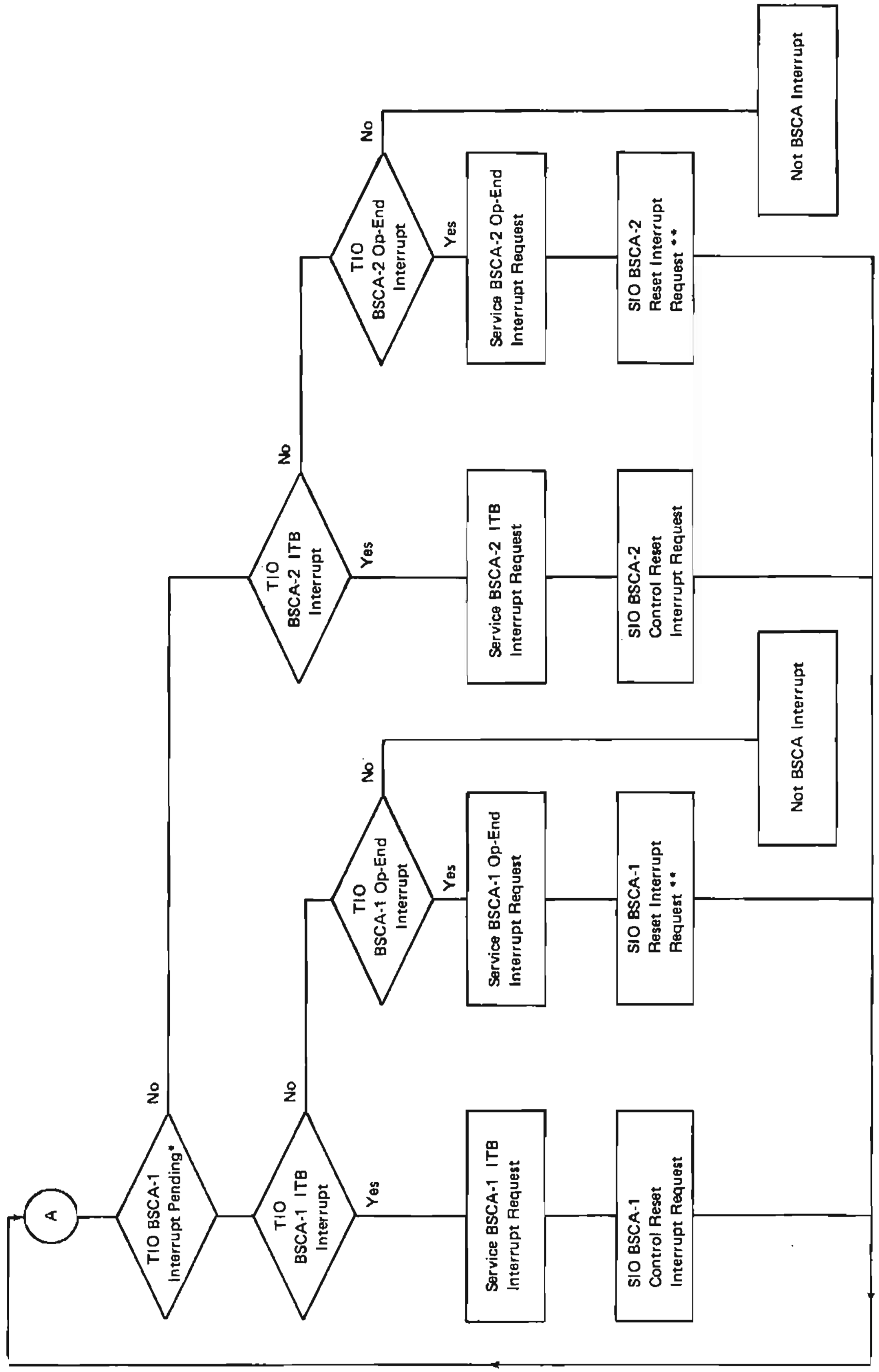
On a start two second timeout operation, an op end interrupt is generated at the end of the two second period.

ITB Interrupt

An ITB interrupt occurs at a slave station whenever interrupt is enabled, an ITB character is received, and no errors have been detected.

The ITB interrupt should be serviced prior to the request for the next succeeding interrupt. (This period of time is a function of baud rate and number of bytes in the next intermediate block.) Allow time for CPU interference caused by I/O cycle steals and by the need to service higher priority interrupts.

If the ITB interrupt is not serviced before the BSCA receives the next ITB character, the next ITB Interrupt request may be lost.


A
TIO BSCA-1 Interrupt Pending*
No
Yes
TIO BSCA-1 ITB Interrupt
Service BSCA-1 ITB Interrupt Request
SIO BSCA-1 Control Reset Interrupt Request
TIO BSCA-1 Op-End Interrupt
Service BSCA-1 Op-End Interrupt Request
SIO BSCA-1 Reset Interrupt Request **
Not BSCA Interrupt
TIO BSCA-2 ITB Interrupt
Service BSCA-2 ITB Interrupt Request
SIO BSCA-2 Control Reset Interrupt Request
TIO BSCA-2 Op-End Interrupt
Service BSCA-2 Op-End Interrupt Request
SIO BSCA-2 Reset Interrupt Request **
Not BSCA Interrupt

* This instruction is not needed if the second BSCA feature is not installed.

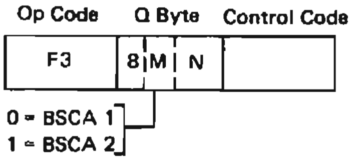
** An Op-End Interrupt is normally reset and the next I/O operation, if any, is started by the last SIO instruction of the interrupt routine.

Figure 11-3. Generalized Communications Adapter Interrupt

COMMUNICATIONS ADAPTER INSTRUCTIONS

Start I/O

Mnemonic: SIO



Operation: The start I/O instruction initiates all communications adapter operations. While the communications adapter is busy or is not ready for any reason except unit check, the program will not accept any start I/O instruction except control. In systems with the dual programming feature, a start I/O instruction issued to a communications adapter that is busy or not ready causes an automatic program level advance. Issuing the start I/O when the communications adapter is in the not ready condition causes the I/O attention light and BSCA attention light on the system control panel to light. Correcting the not ready condition causes the instruction to be executed.

The Q byte specifies the communication adapter as the I/O unit that is to operate and specifies the function to be performed. Bits 0 through 3 of the Q byte are the device address, which is always hexadecimal 8 (1000 binary) for the BSCA. Bit 4 is a modifier bit that is always 0 for the first communications adapter and 1 for the second.

The N code (bits 5, 6, and 7) specifies the operation to be performed as follows:

N Code	*Operation*
000	Control
001	Receive only
010	Transmit and receive
011	Receive initial
100	Auto call
101	Invalid
110	Loop test
111	Invalid

An invalid N code causes the processing unit to stop with the processor-check and invalid-Q indicators lighted.

The third byte of the instruction is a control code. It is used to cause communications adapter control functions as follows:

Control Code	*Function*
Bit 7 = 1	Reset interrupt request
Bit 7 = 0	None
Bit 6 = 1	Enable interrupt request capability
Bit 6 = 0	Disable interrupt request capability
Bit 5 = 1	Start two-second timeout
Bit 5 = 0	Cancel two-second timeout
Bit 4	Not used
Bit 0 = 0	Disregard bits 1, 2, and 3
Bit 0 = 1	
and	
Bit 3 = 1	Enable step mode
Bit 3 = 0	Disable step mode
Bit 2 = 1	Enable test mode
Bit 2 = 0	Disable test mode
Bit 1 = 1	Enable BSCA
Bit 1 = 0	Disable BSCA

Control Function: The N code that specifies the control function provides only the functions specified by the control code. This is the only instruction that can initiate the two-second timeout function.

Receive-Only Function: This operation accepts characters from the line and places them in storage at the location designated by the current-address register. The BSCA updates the current-address register plus one 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 two-second timeout can be initiated by this instruction.

Transmit-and-Receive Function: This function takes characters from storage at the location designated by the current-address register and transmits them on the line to the remote station. The BSCA updates the current-address register plus one as it transmits each character. 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 has been 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 updates the current-address register plus one as it stores each character.

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 two-second timeout can be initiated by this instruction.

The transmit-and-receive instruction can be used as a transmit only instruction (this is mandatory for transmitting transparent ITB blocks) by loading the same address into both the transition address register and the stop-address register. A transmit-and-receive instruction 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 instruction should be used in all transmit sequences that require a response.

Receive-Initial Function: This instruction 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 updates the address register by plus one as each 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 two-second timeout can be combined with this instruction.

Auto Call: This function is provided as a special feature in the communications adapter. In operation, the communications adapter takes the number to be called, one digit at a time, from storage locations specified by the current address register. Each digit to be dialed must be specified in BCD code in the digit portion of a byte. These numbers are sent by BSCA logic to an automatic calling unit (ACU) that dials the number of the remote station. The BSCA updates the current-address register by plus one as each byte is transferred to the ACU. When the current-address register equals the stop-address register, the communications adapter stops sending digits to the auto calling unit and waits for an indication of line connection having been established or of the call's having been aborted. If the connection is established, the adapter is signaled to end the operation. If the call is aborted, the BSCA sets the timeout status bit, ends the operation, and generates an interrupt request. If the timeout status bit is on, the program should retry the operation after disabling the BSCA for two seconds.

Any of the control functions except start two-second timeout or enable BSCA can be combined with this operation.

Loop Test Function: The loop test function is used by the CE to test the functioning of the communications adapter. It is of no use to the problem programmer.

Reset Interrupt Request, Enable Interrupt, and Disable Interrupt Control Functions: These functions control the communications adapter's ability to interrupt the main program. The BSCA operates on interrupt level 2. Two kinds of interruptions can occur from the communications adapter: an ITB interruption and an operation-end (op end) interruption. The interruption routine must determine with a test I/O-and-branch instruction which type of interruption occurred. The ITB interruption should be serviced first.

The ITB interruption occurs during receiving operations when the BSCA receives an ITB character if the block check characters indicate that everything transmitted in that block was received correctly. When the ITB interrupt occurs, the program can store the contents of the transition-address register to indicate the point at which data in the next block begins in storage. All the data up to (but not including) this address is data that is to be processed. The status bytes cannot be sensed during an ITB interrupt because the bits in the status bytes apply to the data being received, rather than to the data that has been received (for ITB operation only).

Op end interruptions occur at the end of all the functions controlled by the N code. In addition, the two-second timeout causes an interruption two seconds after the CPU issues an SIO control instruction with a control code that specifies start two-second timeout. Op end interruption routines usually sense the status byte to determine the status of the last operation. The status bytes are valid for op end interrupts because no data is transferred between the interrupt request and the interrupt routine.

Because the communications adapter continues to receive data from the remote station during ITB interrupt routine servicing, the program should sense the transition address register before the next ITB character is received. The processing time available is a function of the data rate of the data set used and the number of bytes in the next intermediate block. Allow extra time in the interrupt routine to account for time that may be required for CPU interference caused by I/O cycle steals and by the occurrence of higher priority interrupts.

Two-Second Timeout: This SIO control code function is provided to obtain a two second delay before the transmission of TTD or WACK. The start two-second timeout must be given only with the Q code control function. When the timeout is completed, an interrupt is generated. The BSCA is not busy when doing a two-second timeout. It can be aborted by giving any SIO with the control code specifying cancel two-second timeout. A previously issued start two-second timeout must be aborted if an SIO non-control instruction is to be issued. The start two-second timeout instruction must not be issued while the adapter is in the busy state.

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

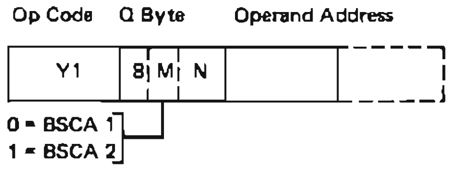
Enable–Disable Step and Test Modes Functions: These are diagnostic functions useful to the customer engineer but of no interest to the problem programmer.

Enable–Disable BSCA Control Functions: The enable BSCA function causes the communications adapter to become operable and allows it to connect to the data set and perform data handling functions. At this point, the program should issue a TIO not ready test instruction. The disable BSCA function deconditions the adapter and disconnects it from the data set.

Instruction Timing: Time in microseconds = 4.56

Load I/O

Mnemonic: LIO



Operation: The contents of the 2-byte field addressed by the operand address are placed in the register specified by the Q byte. The operand is addressed by its rightmost byte.

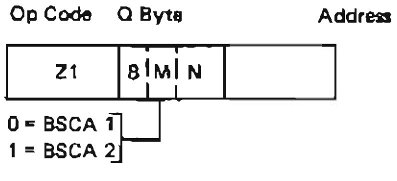
The Q byte contains a device address (always 8 for the communications adapter) in the high-order four bits, an M bit, and an N code (bits 5, 6, and 7). The N code specifies the register to be loaded as follows:

N Code	*Register*
000	Invalid
001	Stop-address register
010	Transition-address register
011	Invalid
100	Current-address register
101	Invalid
110	Current-address buffer (For diagnostic procedures only. Should not be in user's program.)
111	Invalid

If a load I/O instruction is issued to the communications adater when the adapter is busy, the processing unit will not accept the load I/O instruction until the busy condition no longer exists. If the dual programming feature is installed, a load I/O instruction issued to the communications adapter when the adapter is busy causes an automatic program level advance.

Test I/O and Branch

Mnemonic: TIO



Operation: The CPU tests for the conditions specified by the Q byte. If the specified condition exists, the CPU takes the next instruction from the branch-to address and places the next sequential instruction address in the address recall register. If the condition specified does not exist, the CPU issues the next sequential instruction and places the branch-to address in the address recall register. The address recall register is not changed until the next decimal, insert-and-test-characters, or branch instruction is executed.

The Qbyte contains a device address (always 8 for the communications adapter) in the high-order four bits, an M bit, and an N code (bits 5, 6, and 7). The N code specifies the condition to be tested as follows:

N Code	*Condition Tested*
000	Not ready/unit check
001	Op end interrupt
010	Busy
011	ITB interrupt
100	Interrupt pending
101	Invalid
110	New data (diagnostic only)
111	Invalid

Not ready means either (1) data terminal ready off, (2) ACU power off, (3) external test switch on and test mode disabled, or (4) data set ready latch off (non-switched or multipoint network).

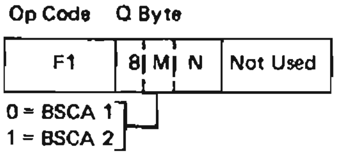
The communications adapter becomes busy under different conditions, depending upon the kind of operation that is being performed. For all operations except receive initial, the adapter becomes busy as soon as the start I/O instruction is accepted; it remains busy until the operation ends. For receive-initial operations, the following conditions cause busy:

1. In a point-to-point non-switched network, the adapter becomes busy as soon as the adapter establishes character synchronization with the remote station.
2. In a point-to-point switched network, the adapter becomes busy as soon as the data set indicates that it has received a call.
3. In a multipoint network, the adapter becomes busy when it recognizes its own address in control mode.

Unit check usually means that one of the status bits in status byte two is on (see "Sense I/O" in this section).

Advance Program Level

Mnemonic: APL



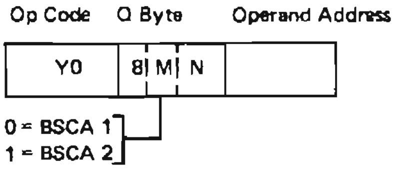
Operation: The CPU tests the conditions specified by the Q byte. If the specified condition exists, systems with the dual programming feature installed advance the program level and continue processing. If the specified condition does not exist, the next sequential instruction is executed. In systems without the dual programming feature installed, the processing unit loops on the advance program level instruction until the condition specified by the Q byte does not exist.

The Q byte contains a device address (always hexadecimal 8–binary 1000–for the communications adapter) in the high-order four bits, an M bit, and an N code (bits 6, 7, and 8). The N code specifies the condition to be tested as follows:

N Code	*Condition Tested*
000	Not ready/unit check
001	Op end interrupt
010	Busy
011	ITB interrupt
100	Interrupt pending
101	Invalid
110	New data (diagnostic only)
111	Invalid

Sense I/O

Mnemonic: SNS



Operation: The contents of the register or the status data specified by the Q byte are stored in the two-byte field addressed by the operand address. The operand is addressed by its rightmost byte.

The Q byte contains a device address (always hexadecimal 8 for the communications adapter) in the high-order four bits, an M bit, and an N code (bits 5, 6, and 7). The N code specifies the register or status data to be stored as follows:

N Code	*Register or Status Data*
000	Diagnostic (only)
001	Stop-address register
010	Transition-address register
011	Status bytes
100	Current-address register
101	Invalid
110	CRC/LRC buffer (diagnostic only)
111	Invalid

The diagnostic and CRC/LRC buffer functions are used by the customer engineer for servicing the adapter. They are of no interest to the problem programmer.

The status bytes are bit-significant as illustrated in Figure 11-4. Byte 1 is stored in the storage location addressed by the operand address; byte 2 is stored in the next lower storage location.

The timeout bit is turned on by either of two conditions:

1. Character synchronization is not established within 3.25 seconds from the start of a receiving operation.
2. An automatic call operation is terminated by an abandon-call-and-retry signal from the automatic calling unit. This indicates that the call was not answered.

Any non-control start I/O instruction resets the timeout bit.

Byte	Bit	Meaning When Set to 1	Reset Off By
1	0	Not assigned.	
1	1		
1	2		
1	3		
1	4		
1	5		
1	6	Data set ready. This indicates that the data set is ready to operate and that the BSCA has been enabled.	Data set losing its ready state or BSCA disabled state.
1	7	Data line occupied. This bit is used on a switched network when the BSCA is equipped with the auto call feature. This bit indicates that the data line is busy and that any SIO auto call or SIO receive initial instruction will be rejected. These instructions should not be issued when in an interrupt routine with the data line occupied.	Data line becoming not busy.
2	0	Timeout status. a. A receive timeout occurred during a receive operation with the adapter in the busy state. b. An auto call operation was terminated by an abandon call and retry signal from the ACU (auto calling unit), indicating that a connection was not established.	Any non-control SIO.
2	1	Data check during receive operation. a. A BCC compare check occurred (EBCDIC). b. A VRC check occurred (ASCII). (*Note:* Characters having VRC checks are distinguished by a high-order bit in core storage. These characters are never recognized as control charactesrs by the BSCA.)	Any non-control SIO.
2	2	Adapter check during transmit operation. a. DBI register parity check. b. I/O cycle steal overrun. c. LSR or shift register parity check. d. Transmit control register check. Adapter check on transmit terminates the operation and causes an immediate op end interrupt.	Any non-control SIO.

Figure 11-4. BSCA Status Indications (Part 1 of 2)

Byte	Bit	Meaning When Set to 1	Reset Off By
2	3	Adapter check during receive operation. a. DBI register parity check. b. I/O cycle steal overrun. c. LSR or shift register parity check. Adapter check on receive does not terminate the operation.	Any non-control SIO.
2	4	Invalid ASCII character. (A byte fetched from core by an adapter using USASCII code contained a 1-bit in the high order bit position.)	Any non-control SIO.
2	5	Abortive disconnect. Indicates BSCA on switched network was enabled, then the data set 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 two-second timeout to ensure this.	SIO disable BSCA.
2	6	Disconnect timeout. Indicates disconnect timeout occurred on a switched network. Disconnect timeout causes data terminal ready to turn off. (May not apply to systems using the IBM remote job entry program.) *Note:* The program must perform a disconnect operation.	SIO disable BSCA
2	7	Not assigned.	
Note: When a SNS transition or SNS stop register instruction is executed, it is possible for an LSR, S register, or DBI register parity check to occur. This can result in a unit check. Under this condition, the byte 2 status bits may all be zero.			

Figure 11-4. BSCA Status Indications (Part 2 of 2)

In a switched network, the disconnect-timeout status bit turns on if no heading, text, response, or control transmission occurs from either station for twenty seconds. A start I/O disable BSCA instruction resets this bit. The 20-second disconnect timeout function can be disabled by the customer engineer at installation time at the customer's request (for example, for those installations using the IBM remote job entry –RJE– control program).

The data-set-ready condition status bit is set on when the data set ready signal is detected and latched on. The bit is turned off if data set ready comes on and then turns off or if the communications adapter is disenabled.

The data-line-occuped status bit turns on when the auto calling unit signals that the data line is occupied. When this bit is on, a start I/O auto call instruction or start I/O receive-initial instruction will not be accepted until the line is unoccupied. No Start I/O auto call or receive initial instructions should be issued in an interrupt routine when this bit is on.

Programming Notes: When the disconnect-timeout bit is on, the BSCA has automatically performed a disconnect operation.

When a sense I/O transition-address register or sense I/O stop-address register instruction is executed, a BSCA detected adapter check condition can occur, causing a unit check indication. If this happens, it is possible that none of the byte 2 status bits will be on.

BSCA OPERATIONS

The BSCA controls all operations on the communication line through a combination of instructions in the System/3 processor and the automatic controls initiated by line control characters and sequences. Figure 11-5 is a basic flowchart of a suggested generalized routine to place the BSCA in operation

11

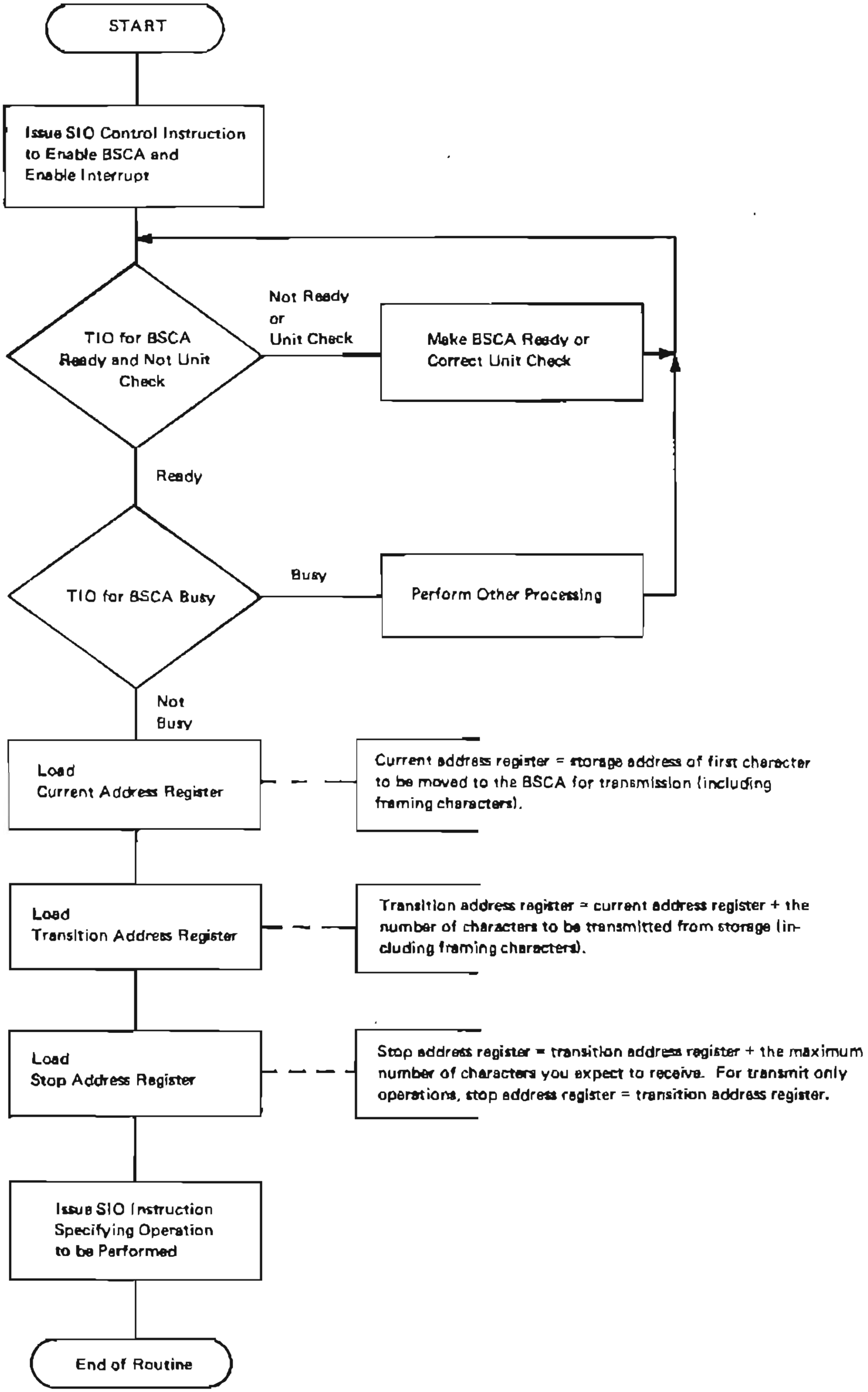

START
Issue SIO Control Instruction to Enable BSCA and Enable Interrupt
TIO for BSCA Ready and Not Unit Check
Not Ready or Unit Check
Make BSCA Ready or Correct Unit Check
Ready
TIO for BSCA Busy
Busy
Perform Other Processing
Not Busy
Load Current Address Register
Current address register = storage address 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 SIO Instruction Specifying Operation to be Performed
End of Routine

Figure 11-5. Initiating Action in BSCA

Enable/Disable BSCA

Enable BSCA sets on the data terminal ready line to the data set; disable BSCA sets off the data terminal ready line and resets the BSCA. Power-On reset or system reset or IPL will also set off the data terminal ready line and reset the BSCA.

Since data terminal ready controls switching the data set 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 data set to disconnect from the switched network before the program again enables BSCA. The two-second timeout may be used to assure this.

Auto Call Operation

At the calling station, data terminal ready must be on when the SIO auto call instruction is issued. Auto call should be issued as soon as possible after enable BSCA to avoid the possibility that another call comes in.

Prior to giving the auto call instruction, the current address register and stop address register must be set up with LIOs to point to the number to be dialed. The stop address register must be set to the initial current address plus the number of digits to be dialed. The auto call instruction is executed by transferring bytes to the ACU at a data rate controlled by the ACU. Only the four low order bits in each byte from core are sent to the ACU. The transfer is on a cycle steal basis from the location specified by the current address register which is updated by plus one each cycle steal. This continues until the current address register is equal to the stop address register. At this point the adapter waits for the ACU to signal that the connection has been established or that the call has been aborted.

An interrupt with no error condition indicates that a connection has been established. If the timeout status bit is on (call aborted due to abandon call and retry signal from ACU), the program should retry the operation after disabling the BSCA for two seconds.

The SIO auto call instruction will be rejected and the I/O attention indicator set if the ACU power is off or data line occupied is on.

When the reject condition is removed by the operator, the SIO auto call will be accepted and the I/O attention indicator will be reset.

Initialization Sequences

Initialization sequences are defined in the BSC GI manual and are transmitted by the transmit and receive instruction. Receive initial instruction is defined for receiving initial sequences. The Receive initial operation is dependent on the data link (Pt-to-Pt Non-Switched, Pt-to-Pt Switched, or Multipoint) selected by the customer.

Receive Initial Operation (Pt-to-Pt Non-Switched)

On a non-switched network, SIO 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 core 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 to be received. 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.

Receive Initial Operation (Pt-to-Pt Switched)

On a switched network, SIO 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 character (starting with the first non-SYN character) is stored in the core 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 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 SIO receive only instruction.

Receive Initial Operation (Multipoint Tributary)

SIO 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 two-character station

11

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 will recognize either BB or SS as the station address. The high order byte in the transition address register is not used.

The basic mode of the BSCA in this operation is monitor mode. 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 passed into core. When a valid EOT sequence is received, control mode will be set.

In control mode the BSCA monitors for its station address. If it is not detected, the BSCA continues monitoring the line. The adapter leaves control mode if no change of direction character is received within the period of the receive timeout. A decoded SOH or STX will drop control mode and put 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 will immediately enter addressed mode, set busy, and transfer the sequence starting with the second station address character into the core 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 when a receive timeout occurs.

Auto Answer Wait Operation

The auto answer wait function requires the following programming support. After BSCA is enabled, an SIO receive initial instruction with interrupt enabled should be issued and then the program can be stopped by a halt instruction. The CPU use meter will then stop. When the call is answered, busy will be set, causing the CPU use meter to commence running. The op end interrupt will take the CPU out of the halt instruction to the BSCA interrupt routine which must take the necessary programming action, e.g., change the halt to a jump on condition, so that the main line program will start when the interrupt routine is exited. The CPU use meter will then continue running until normal job termination.

Transmit and Receive Operation

The SIO transmit and receive instruction is used for any type of transmission, i.e. control sequences or text data. It sets the BSCA to transmit mode where it takes characters from core and transmits them onto the line. BCC accumulation, data mode, and transparent mode are set dependent on the type of line control characters fetched from core. Transmission proceeds until 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 and then stores the characters received into core. As in transmit, the detail function on receive is dependent on the particular line control characters received.

The operation is terminated and an interrupt generated when an adapter check on transmit occurs, 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 unit check condition can be tested, and, if on, 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 11-6.

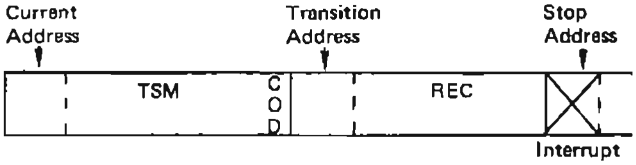


Figure 11-6. I/O Area and Address Register Contents at Start of Transmit and Receive Operation

The transmit and receive instruction is used at both the master and the slave, i.e. 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 and is updated by plus one on each cycle steal. 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 the number of characters to be received.

The current, transition, and stop addresses are unrestricted two byte addresses, except that a zero length transmit field is not permitted. There is no maximum restriction in block length, i.e. current, transition, and stop addresses. Each is a sixteen 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 one hexadecimal "55" character (two additional hexadecimal "55" characters 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 core. BCC compare takes place when an ITB, ETB, or ETX is received.

If the adapter has entered the 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(s) after the ITB, the slave receives and compares it, and both stations continue transferring more data immediately thereafter with no line turn-around.

For non-transparent 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 address plus one of where it is stored in core is loaded in the transition address register. After the BCC comparison has been made, and if no errors have been detected, an ITB interrupt occurs. The adapter remains in a busy state and proceeds to receive the next ITB block. The interrupt program, finding the ITB interrupt latch on, stores the transition address register and processes the ITB block just received. Status bits are not sensed as they will apply to the subsequent block being received. Whenever a BCC error occurs, the BSCA withholds the ITB interrupt for the ITB containing the error and for all the subsequent intermediate blocks, and stops sending data to storage. This continues until a change of direction code is recognized. When the ending sequence—ETB, ETX, or ENQ—is received, it is stored in core 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 core (except DLE STX), which will be stripped by the receiving BSCA. The additional DLE will not enter BCC accumulation.

Either ETB, ETX, ITB, or ENQ ends transparent mode at the master if it is at a location one less than 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 at 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 non-transparent mode.

Each ITB block of transparent data must be transmitted with its own transmit and receive instruction. No turnaround takes place after the ITB and the adapter inserts at least two SYN characters (more, if necessary), until the next transmit and receive is issued or until three seconds elapse. During this period the adapter is not in a busy state. 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 an SIO disable BSCA instruction, which drops the data terminal ready line to the modem. It should previously transmit a DLE EOT sequence with a transmit and receive 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.

If the 20-second disconnect timeout function has not been disabled, (for example because the IBM RJE control program is being used), data terminal ready is also dropped by the disconnect timeout which occurs when there has been no header, text, response, or control transmission on the line for 20 seconds.

Sufficient time must be allowed for the disconnect to occur before the program again enables BSCA. The two-second timeout may be used to assure this.

Receive Operation

The SIO 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 time-out during a receive initial operation on a switched network.

Two Second Timeout

This SIO control code function is provided to obtain a two second delay before the transmission of TTD or WACK. The start two second timeout must be given only with the Q-code function "control". When the timeout is completed, the BSCA generates an interrupt. The BSCA is not busy when doing a two-second timeout. It can be aborted by giving any SIO with the control code specifying cancel two-second timeout. A previously issued start two-second time-out must be aborted if an SIO non-control instruction is to be issued. Start two-second timeout must not be issued if the adapter is in the busy state.

The BSCA does not need to be in the BSCA enabled state to perform the two-second timeout operation.

Testing and Advancing Program Level

The TIO and APL instructions can be given at any time to test the following conditions:

Not ready/unit check
Busy
ITB interrupt
Op end interrupt
Interrupt pending
New data

Not ready means either: (1) data terminal ready off, (2) ACU power off, (3) external test switch on and test mode disabled, or (4) data set ready latch off (non-switched multipoint).

Unit check means that one of the status bits in byte 2 is on. When an SNS transition or SNS stop register instruction is executed, it is possible for an LSR, S register, or DBI register parity check to occur resulting in a unit check condition. Under this condition the byte 2 status bits may all be zero.

Busy means the BSCA is executing a: (1) receive initial, (2) transmit and receive, (3) auto call, (4) receive, or (5) loop test (diagnostic) instruction.

Interrupt pending means that either ITB interrupt latch or op end interrupt latch is on. ITB interrupt and op end interrupt are used to determine the type of interrupt that has occurred and are reset off when tested by TIO/APL.

Loading the Registers

LIO is used to load the current address register, transition address register, and the stop address register.

Sensing

SNS is used to store: (1) the current address register, (2) transition address register, (3) stop address register, (4) diagnostic bits, (5) CRC/LRC buffer, and (6) status bits.

Data Checking

As the remote station transmits messages, it generates block check character(s) from the data bits transmitted. As these bits are received at the local system communications adapter, the communications 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 character(s). The local communications adapter compares these block check character(s) that it receives from the line with the block check character(s) that it has generated from the data bits it has received from the line. If the block check character(s) generated by the local communications adapter do not match the block check character(s) 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 the block check characters match each other.

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 being transmitted in response to even-numbered blocks and ACK1 being transmitted in response to odd-numbered blocks. The program is responsible for transmitting the correct positive acknowledgement. The first block of text transmitted is always considered an odd-numbered block. If the wrong acknowledgement character is returned, the 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 character. If the block check is incorrect, no more data is stored, no more ITB interruptions are generated, and 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 CAR= SAR and a valid ending character have not been received, a lost data error is indicated.

Suggested Error Recovery Procedures

At the end of every transmit and/or receive operation, the program should test the BSCA for a unit check. If a unit check is detected, the program should sense the BSCA for status bytes. Test the status bits and perform the procedures for recovering from the error in the order given in Figure 11-7. The program must check for lost data and analyze the last two characters received to detect an abnormal response error.

System and Error Statistics

The user program should accumulate the following information for each BSCA as a diagnostic aid. These counters should be logged to disk storage at close time (disk systems only).

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 adapter checks on transmit operations.
6. For System/3 multipoint control station applications, a count of transmissions and transmission errors for each terminal on the multipoint network.

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 response to a transmitted WACK should not be included.
4. A count of master station forward aborts (TTD/NAK EOT sequences).
5. A count of adapter checks on receive operations.

Priority	Status		Error Condition	Error Recovery Procedure (Recommended Program Action)
	Byte	Bit		
1	2	4	Invalid ASCII Character	All Cases—Action 1
2	2	5/6	Abortive Disconnect or Disconnect Timeout	All Cases—Action 1
3	2	2	Adapter Check on Transmit	Control Mode—Action 5 Slave—Action 4 Master—Action 3
	2	3	Adapter Check on Receive	Control Mode—Action 5 Slave—Action 4 Master—Action 3
4	2	0	Timeout	Receive Initial (Switched)—Action 8 Auto Call or Control Mode—Action 6 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*		Lost Data (CAR=SAR on Receive)	
6	Program Detected Error*		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

ACTION TABLE:

1. Permanent error....Operator restart.
2. T&R NAK....data N times when a control station.
3. T&R ENQ....last response N times.
4. Issue receive portion of previous operation N times.
5. Polling or selection sequence....retry polling or selection of failing station L times after sending an EOT sequence to ensure control mode at the tributary stations.
   Other than polling or selection sequence....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 revelve, 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 in NAK, do Action 6 N times. If invalid response reoccurs, do Action I.
8. Issue SIO reveive instruction.

The value L should be a minimum of 3.
The value M should be equal to or greater than N.
The value N should be a minimum of 7.

When L, M, or N is reached (permanent error) the program should abort the job and tell the operator the nature of the error condition by some means (such as the halt identifier). 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 a hard stop.

*The program should provide lost data detection.

Figure 11-7. BSCA Error Conditions and Recovery Procedures

IBM 1270 Optical Reader Sorter

(IBM 1270 is not offered in U.S.A.)

The IBM 1270 Optical Reader Sorter reads OCR characters from paper documents and transmits them to the CPU. A discussion of the capabilities, characteristics, and operations of the 1270 optical reader sorter can be found in *IBM System/360 Component Description – IBM 1270 Optical Reader Sorter,*(GA19-0035).

OPERATION

The 1270 attaches to the system SIOC and operates through the instructions issued to the SIOC. The exact form of these instructions is discussed in the SIOC chapter of this manual.

General Programming Requirements

In addition to the instructions which actually control functions of the reader, the following items must be handled in a specific manner in order for the 1270 to operate with the SIOC.

1. Before executing the instructions that cause the reader to operate, the function register of the SIOC must be loaded by a load I/O instruction. The two bytes loaded must contain a 1 in bit 5 of the high-order byte and a 1 in bit 6 of the low-order byte. All other bits in these bytes must be 0.

2. The length count register must be loaded by a load I/O instruction issued to the SIOC. The number to be loaded into the register is 256 minus the number of bytes to be read from the 1270. This operation must be performed before each read instruction.

3. The SIOC data address register must be loaded with an address before reading occurs for each read operation. This address designates where in storage the data read from the document is to be stored. The address must be the address of the low-order (rightmost) byte of the data field. This register is loaded with a load I/O instruction.

4. The device identification assigned to the 1270 is 0011. The fact that the 1270 is the device attached to the SIOC can be detected by the sense I/O instruction sensing the I/O transfer lines. Bits 0 through 3 of the high-order sense byte stored by this instruction contain the device identification. For the 1270 bits 0 and 1 will be 0 and bits 2 and 3 will be 1.

5. A start I/O instruction must be issued to enable interrupts for the SIOC. The 1270 requires that processing for the documents be performed within specified periods of time to provide correct processing. The 1270 causes an interrupt at the end of every document, and this interrupt must be enabled to allow processing to commence.

Feeding Documents

The engage command starts the flow of documents if the 1270 is online and is in a ready-to-feed state.

A disengage command stops document feeding by stopping the separator. Document feeding also stops whenever the CPU is stopped. The following 1255 conditions also stop or inhibit document feeding:

1. The separator stops feeding documents but the transport continues to run whenever:

   a. The start key is being held down.
   b. The stacker is full.
   c. No validity-check-and-readout key has been pressed.

2. The transport mechanism stops because of:

   a. Stop key depression
   b. An empty hopper
   c. A feed jam
   d. A transport jam
   e. A sort check
   f. An interlock

12

When the 1270 is online and not-read-to-feed, the engage command is stored in the 1270 so that the actual document feeding starts as soon as the 1270 becomes ready-to-feed again.

If a disengage instruction is issued when the 1270 is online and not ready-to-feed, no documents are fed when the 1270 is returned to the ready-to-feed state.

The system call light on the 1270 operator panel indicates that the program calls for document feeding.

The engage command is issued by executing a start I/O instruction for the SIOC with an N code of 100, and a control code of 00000001. The disengage command is issued by a start I/O instruction with an N code of 100 and a control code of 00000010.

Note: An engage instruction immediately followed by a disengage instruction causes single-document feeding.

Retrieving Data From Documents

A start I/O instruction specifying read retrieves data from documents passing through the 1270. A read command must be issued for each document before that document reaches the read head. Failure to issue the read command in time results in the document being rejected by the 1270 and a signal (autoselect) being provided for program interrogation.

Validity-check-and-readout keys on the 1270 operator panel select the data to be transferred to system main storage. The first character transferred from the 1270 enters storage at the address designated by the SIOC data address register. Subsequent characters enter successively lower storage locations.

The read operation is terminated either at the end of each document or when the specified number of characters to be read (as initially loaded into the SIOC length count register), have been transferred, whichever occurs first. At the end of a read operation the SIOC requests an interrupt to process the data and select the appropriate stacker (pocket) for that document.

Directing the Disposition of Documents

Stacker select commands direct documents to the stackers in the 1270. These commands are generated by Start I/O instructions that load the I/O select register.

The stacker select command must be issued within 24 milliseconds after the document to be selected leaves the read head. If a stacker select command has not been issued within this time, a sort check occurs, the 1270 rejects the document, and the 1270 stops after all documents in the transport have been directed to the reject stacker.

The sort check light on the 1270 operator panel indicates the error to the operator and signals (Sorter-is-stopped and Autoselect) are provided for program interrogation.

Termination of 1270 Operations

The following rules must be observed in order to prevent non-recoverable 1270 and/or system errors:

- Do not stop the processing unit during 1270 online operations.
- Do not switch from online mode to offline mode or vice versa during 1270 operations.
- The 1270, when attached to the system, should be powered down only when there is no activity on the Serial I/O Channel interface.

While System-is-stopped light (on the 1270 operator panel) is on there is no SIOC interface activity.

INSTRUCTIONS

Start I/O

Mnemonic: SIO

Op Code	Q Byte	Control Code
F3	0011 0 N	

Operation: The reader performs the operation specified by the N code and the control code.

The Q byte comprises a device address (always 0011 for the reader) in the first four bits, an M bit of 0, and an N code. The N code in conjunction with the control code specifies the operation to be performed. The operations performed are:

N Code	*Control Code*	*Operation*
000 or 001	00000001	Reset interrupt request (performed by the SIOC).
000 or 001	00000010	Enable interrupt (performed by the SIOC).
000 or 001	00000100	Disenable interrupt (performed by the SIOC).
000 or 001	00001000	Reset SIOC adapter, removing SIOC from busy state (performed by the SIOC).
000 or 001	00010000	Set interrupt request.
001	00000000	Read I/O device.
010	00000000	Invalid for 1270.
011	-------	Control I/O.
100	------	Control I/O.

The control I/O operations set the I/O select register to produce the desired operation. The following operations can be performed by each N code.

N Code 011:

Control Code Bit	*Operation*
0	Select stacker 7.
1	Select stacker 6.
2	Select stacker 5.
3	Select stacker 4.
4	Select stacker 3.
5	Select stacker 2.
6	Select stacker 1.
7	Select stacker 0.

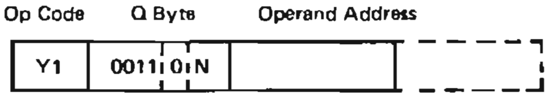
N Code 100:

Control Code Bit	*Operation*
0	Not used.
1	Select reject stacker.
2	Not used.
3	Select stacker A.
4	Select stacker 9.
5	Select stacker 8.
6	Disengage feed.
7	Engage feed.

Not all the stackers indicated in these charts will be available on a 6 pocket (6 stacker) 1270. A 6-pocket 1270 will be ordered with either stacker designations 0 through 4 and R (for reject stacker) or with the even numbered stacker designations 0, 2, 4, 6, 8, and R.

Load I/O

Mnemonic: LIO



Operation: The two bytes contained in the two-byte field addressed by the operand address are placed in the register designated by the Q byte. The operand is addressed by the low-order byte.

The Q byte comprises a device address (always 0011) in the high-order four bits, an M bit of 0, and an N code. The N code specifies the register into which the contents of the operand field are to be loaded.

N Code	*Destination*
000	Invalid.
001	I/O function register.
010	SIOC length count register.
011	Invalid.
100	SIOC data address register.
101	Data transfer register.
110	Invalid.
111	Invalid.

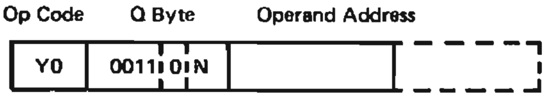
Specification of an invalid N code results in a processor check stop with an invalid Q byte indication.

The I/O function register must be loaded with the following:

High-Order Byte	*Low-Order Byte*
00000100	00000010

Sense I/O

Mnemonic: SNS



Operation: The two bytes specified by the Q byte are placed in the two-byte field addressed by the operand address. The operand is addressed by the low-order byte.

The Q byte comprises a device address (always 0011) in the high-order four bits, an M bit of 0, and an N code. The N code specifies the sense bytes or registers that are to be sensed.

N Code	*Senses*
000	Invalid.
001	I/O function register.
010	Length count register and status byte.
011	I/O transfer lines and I/O identification.
100	Data address register.
101	Data transfer register and diagnostic byte.
110	Invalid.
111	Invalid.

Specification of an invalid N code causes a processor check stop with an invalid Q byte indication.

The status byte is stored as the high-order byte of its sense operation. The status byte is bit-significant, as follows:

Status Byte

Bit	*Meaning*
0	Spare.
1	End request.
2	Interrupt pending.
3	I/O attention.
4	Data transfer register parity check.
5	No-op.
6	Length count register overflow.
7	I/O ready.

The I/O transfer lines are bit significant as follows:

High-Order Byte

Bit	*Meaning*
0	Will be 0*
1	Will be 0*
2	Will be 1*
3	Will be 1*
4	Device attached
5	Field 7 valid.
6	Field 6 valid.
7	Sorter is stopped.

*If the device attached is an IBM 1270.

Low-Order Byte

Bit	*Meaning*
0	Auto select.
1	Field 5 valid.
2	Field 4 valid.
3	Field 3 valid.
4	Field 2 valid.
5	Field 1 valid.
6	Document under read head.
7	Document to be read.

I/O Ready

This condition indicates that the 1270 is selected for a read or write operation and is ready to transfer data.

I/O Attention

The I/O attention condition indicates that normal operator intervention is required on the 1270.

Normal operator intervention conditions are:

1. Stacker full
2. Hopper empty
3. 1270 stopped with the start light on.
4. Interruption of document feeding by keeping the start key pressed down.
5. No validity-check-and-readout key isin the depressed position.

A jam that occurs in the separator area is indicated as an empty-hopper condition.

Error conditions (feed-jam, transport, interlock and sort check) inhibit an I/O attention indication, but still indicate that the sorter is stopped.

Sorter is Stopped

This line is conditioned whenever document feeding has stopped because of a 1270 stop condition (except for disengage). It can be utilized to facilitate online reconciling of errors. This line is deconditioned by clearing and resetting the stop condition.

Field Valid

Field Valid is signalled to the SIOC when the field is selected for transfer to the CPU by the validity check and read out keys and all the characters of the selected field including bracketing symbols are read and transferred to the processing unit without error. The field valid lines are deconditioned when the leading edge of the next document is sensed at the read head.

Auto Select

This line comes on whenever the 1270 cannot allow a document to be stacker selected.

Auto reject occurs because:

1. A read instruction has not been issued by the time a document reaches the read head.
2. One of the following has been detected:
   a. special symbol sequence error
   b. hi/low codeline condition
   c. advance reject condition
   d. short document
   e. overlength document
   f. document spacing error

Auto select comes on when the 1270 detects the error condition, and stays on until the first following document arrives at the read head. (Exception: In the case of a document spacing error, auto select remains on until the second following document arrives at the read head.

The auto select condition is also set by a feed jam, transport condition, interlock condition or sort check condition.

The condition will stay on for all follow-on documents and will be reset by the first document arriving at the read head after error recovery and machine restart. A stacker select instruction, other than reject, must not be issued for a document which is auto selected or a sort check will result.

Document Under Read Head

This line is conditioned while there is a document under the read head. It can be used to determine if a document cleared the read head and if the read instruction has been terminated prior to the end of the document. A stacker select instruction must not be given before the document for which it is intended has left the read head.

Document to be Read

This line is conditioned from the time the separator starts feeding documents until the last document in flight (after a separator stop) has left Document Sensor 2 at the leading edge of the read area.

The "Document to be Read" line is turned off.

- If a feed jam or transport jam has been detected.
- If an interlock has been detected.
- If a sort-check has been detected.
- If there is no validity-check-and-readout key in a depressed state.

Program should test for a document to be read before issuing a read instruction. When the document to be read line is active, further read instructions are necessary.

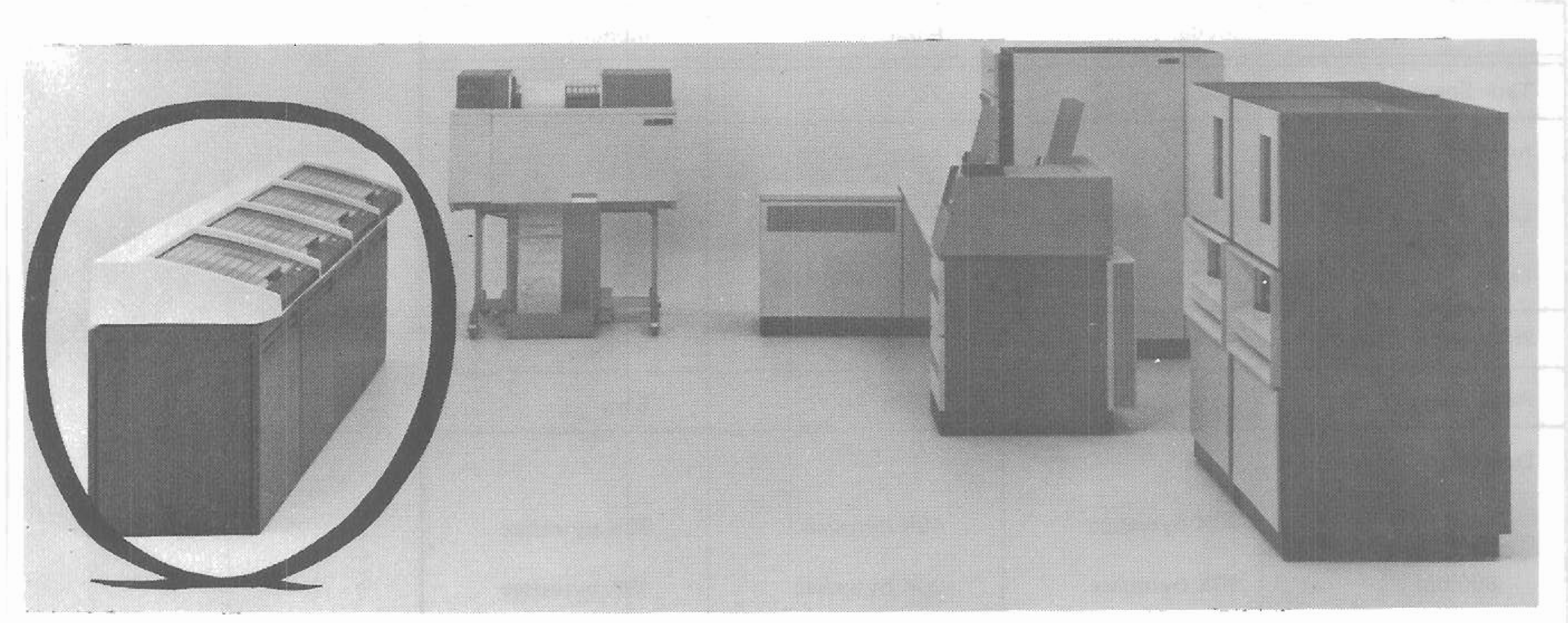
Test I/O and Advance Program Level

These two instructions operate on the SIOC even though they must be used when operating the 1270. See the SIOC chapter for a discussion of these instructions.

FEATURES

Self-Checking Number

For a description of how the self checking number checking is performed, see the 1270 Component Description manual. With this feature, if a self check number is incorrect, a read error has probably occurred and the field valid indicator for that field is not on. No special programming is involved with the self checking number feature. For other features see 1270 component description manual.



The 3410/3411 Models 1, 2, and 3 tape subsystems read and write half-inch magnetic tape. The IBM 3410 Magnetic Tape Unit is a tape unit only; the IBM 3411 Magnetic Tape Unit and Control is a tape unit and a control unit in the same frame.

A 3410/3411 Magnetic Tape Subsystem is available in one of the following configurations for attachment to a System/3 Model 10:

- One 3411 Model 1, 2, or 3
- One 3411 Model 1, 2, or 3 and one 3410*
- One 3411 Model 1, 2, or 3 and two 3410's*
- One 3411 Model 1, 2, or 3 and three 3410's*

* Same model number as the 3411.

13

PERFORMANCE SUMMARY

Figure 13-1 shows the performance information for the 3410/3411 tape subsystem.

	Model 1	Model 2	Model 3
Tape Speed (in./sec)	12.5	25	50
Interblock Gap (IBG)*			
Length/Time (9-track)	0.6 inch/48 ms	0.6 inch/24 ms	0.6 inch/12 ms
Length/Time (7-track)	0.75 inch/60 ms	0.75 inch/30 ms	0.75 inch/15 ms
Write Access Time**	15 ms	12 ms	6 ms
Read Access Time**	15 ms	12 ms	6 ms
Data Rate			
1600 bpi	20K bytes/sec	40K bytes/sec	80K bytes/sec
800 bpi	10K bytes/sec	20K bytes/sec	49K bytes/sec
556 bpi	6.95K bytes/sec	13.9K bytes/sec	27.8K bytes/sec
200 bpi	2.5K bytes/sec	5.0K bytes/sec	10K bytes/sec
Time per Byte			
1600 bpi	50 us	25 us	12.5 us
800 bpi	100 us	50 us	25 us
556 bpi	144 us	72 us	36 us
200 bpi	400 us	200 us	100 us
Rewind Time (±10%)	3 min (2400 ft)	3 min (2400 ft)	2 min (2400 ft)
Reel Sizes (inch)	10.5, 8.5, 7, 6	10.5, 8.5, 7, 6	10.5, 8.5, 7, 6
Tape Threading	Manual	Manual	Manual
Tape Motion	Tape is driven by a single capstan which is directly coupled to a low-inertia, high-torque, dc motor.		
Read/Write Head	The chrome-plated, two-gap head is located in the left vacuum column.		

* An interblock gap is erased tape which separates blocks of data.
** Time given is for a 0.6 inch interblock gap.

Figure 13-1. Performance Information for the 3410/3411 Tape Subsystem

Metric Equivalents:

			12.5 in./sec	=	317.5 mm per second
			0.6 inch	=	15.2 mm
1600 bpi	=	63 bytes per mm	0.75 inch	=	19 mm
800 bpi	=	31.5 bytes per mm	6 inches	=	152.4 mm
556 bpi	=	21.9 bytes per mm	7 inches	=	177.8 mm
200 bpi	=	7.9 bytes per mm	8.5 inches	=	216 mm
50 in./sec	=	1270 mm per second	10.5 inches	=	266.7 mm
25 in./sec	=	635 mm per second	2400 feet	=	732 meters

SPECIAL FEATURES

Each 3410 and 3411 tape unit must be equipped with a special feature that specifies the read/write format desired. The features are:

- Single Density

- Dual Density

- Seven-track

Any tape unit in the subsystem (a 3411 or an attached 3410) can be equipped with either the dual-density feature or the seven-track feature, but not both.

A subsystem can have the following combination of features:

- Each tape unit has the single density feature.

- Each tape unit has the dual density feature.

- Each tape unit has the seven-track feature.

- Some tape units have the single density feature; some have the dual density feature.

- Some tape units have the single density feature; some have the seven-track feature.

Single Density Tape Unit Feature

This feature is installed on tape units to enable nine-track, phase-encoded (PE) operations. Single density control is standard on the 3411.

Dual Density Tape Unit Feature

This feature is installed on tape units to enable nine-track operations in both 1600 bpi phase-encoded mode (PE) and 800 bpi non-return-to-zero mode (NRZI). If any tape unit is equipped with the dual density feature, the 3411 must also be equipped with the dual density control feature.

Seven Track Tape Unit Feature

This feature can be installed on the tape unit portion of the 3411 or on any 3410. It enables the tape unit to read and write data in NRZI mode on seven-track magnetic tape. Reading and writing are done at densities of 200, 556, or 800 bpi. Odd or even parity is provided.

If the seven-track tape unit feature is installed in the 3411 or any attached 3410, the 3411 must also be equipped with a seven-track tape control feature. A 3410 or 3411 tape unit equipped with the seven-track tape unit feature cannot be equipped with either a single density tape unit feature or a dual density tape unit feature.

Dual Density Control Feature

This feature, available for the 3411 control unit, enables the tape units to read and write nine-track tape in either the 800 bpi NRZI or 1600 bpi PE mode.

The program must issue a mode set command to the tape control unit to set the desired writing density. A read operation does not require a mode set operation. When reading, a burst of bits in the parity track at load point identifies, to the tape unit, tape written at 1600 bpi. The lack of this burst identifies tape written at 800 bpi.

Seven-Track Control Feature

This feature, available for the 3411 control unit, enables the 3411 to control any tape unit (including the tape unit portion of the 3411) that is equipped with the seven-track tape unit feature.

A translator and data converter are included with the seven-track control feature. The translator, when set on, translates eight-bit bytes from main storage to six-bit BCD tape characters and vice versa. Each main storage byte becomes a tape character; each tape character becomes one byte in main storage. The data rate is not changed by the translator.

The data converter allows writing and reading of binary data on seven-track tape units. Writing a tape with the data converter on causes four tape characters (24 bits) to be written for every three storage bytes (24 bits). Reading such a tape reverses the process by converting four tape characters into three storage bytes. Data conversion reduces the data transfer rate by 25 percent of that for nine-track NRZI operations. An odd/even count is made during read/write data converter operations to ensure correct transfer of data. An unequal count sets the data converter check bit.

The mode 1 set command bits (Figure 13-2) turn the translator and data converter on and off. The translator and data converter cannot be on at the same time, and the data converter cannot be used with a read backward command.

FUNCTIONAL CHARACTERISTICS

Tape Unit Control

The 3411 houses the clocks, delays, and controls necessary to operate the tape units attached to the system. These circuits receive instructions from the system through a tape attachment feature in the 5410. Upon receipt of the instructions, the control circuits, using a microprogram, direct all required tape unit operations and signal the system when the task is complete.

Operator Controls

Each tape unit has an operator panel that contains all subsystem manual controls. The 3411 also has an enable/disable switch on the operator panel to switch the subsystem online and offline.

File Protection

The 3410/3411 subsystem uses a plastic write-enable ring mounted on the tape reel to permit writing. If a tape is mounted without the ring in position, writing cannot occur; therefore, the file is protected.

Tape Requirements

The following half-inch tapes can be used: IBM Series/500, IBM Heavy Duty, IBM Dynexcel, or competitive formulations which meet the tape and reel criteria in *Tape Specifications,* order number GA31-0006.

Note: IBM tapes other than those named above do not provide adequate reliability and should not be used.

Erase Head

The erase head applies a strong magnetic field that erases the entire tape width during write or erase operations. Full-width erasure eliminates extraneous bits in interblock gaps or skip areas, and destroys previously-written bits.

Parity Checking

During write operations, each byte is parity checked twice: when it is received from the system and when it is written on tape (read back checking). During read operations, each byte is parity checked before it is sent to the system, and single-track errors are corrected. During sense operations, the tape control supplies proper parity for each byte. The tape control parity checks all bytes received from the system.

Tape Subsystem Servicing

The tape subsystem is attached to the system in such a manner that it usually can be serviced offline without impacting other system operations.

Cabling

The 3411 is connected by cable to the system through an opening in the IBM 5203 or 5421; the first attached 3410 is internally attached to the 3411, the second 3410 is internally attached to the first 3410, and the third 3410 is internally attached to the second 3410.

Addressing

Each tape unit has a fixed address.

Inter-System Tape Exchange

Tapes produced on the 3410/3411 subsystem and all other IBM half-inch tape units operating in the same density are interchangeable. Therefore, output data produced on one system, such as the IBM System/360 or System/370, can be used as direct input to another system, such as System/3 using compatible tape.

TAPE OPERATION

The CPU initiates an I/O operation on a tape unit with the start I/O instruction. Figure 13-2 shows the Q-byte bit settings for a read, read backward, and write operation.

Read

A read forward operation is defined by the Q-byte. The tape unit moves tape forward, assembling the data from tape. Whenever a byte of data is available, a data transfer cycle is requested until the byte count reaches zero. The Magnetic Tape Address Register (MTAR) is incremented by 1 after each data transfer cycle (more information about this is given in the note under *Load I/O*). The data is placed in contiguous ascending locations in main storage.

The unit exception condition is set if a tape mark is detected. The end-of-tape (EOT) reflective marker is not recognized during read forward operations.

Note: Seven-track tapes read in the improper mode or on nine-track tape units can result in data checks or tape runaway conditions.

Control Code Formats	0	1	2	3	4	5	6	7	
Read	0	0	0	0	0	0	0	1	
Read Backward	0	0	0	0	0	0	1	1	
Write	0	0	0	0	0	0	1	0	
Control	0	0	C	C	C	1	1	1	Tape Motion
	M	M	M	M	M	0	1	1	Mode 1 Set (7 track)
	1	1	0	0	D	0	1	1	Mode 2 Set (9 track)
	1	0	0	1	0	1	1	1	Data Security Erase
	0	0	0	0	0	1	0	1	Loop-Write-to-Read

C C C	(Control Code)
0 0 0	Rewind
0 0 1	Rewind/Unload
0 1 0	Erase Gap
0 1 1	Write Tape Mark
1 0 0	Backspace Block
1 0 1	Backspace File
1 1 0	Forward Space Block
1 1 1	Forward Space File

D	(Mode 2 Set)
0	1600 bpi
1	800 bpi

M M M M M (Mode 1 Modifiers)	Density bpi			Odd	Even	Data Converter	Data Converter	Translator	Translator
	200	556	800	Parity	Parity	On	Off	On	Off
0 0 0 1 0	X			X		X			X
0 0 1 0 0	X				X		X		X
0 0 1 0 1	X				X		X	X	
0 0 1 1 0	X			X			X		X
0 0 1 1 1	X			X			X	X	
0 1 0 1 0		X		X		X			X
0 1 1 0 0		X			X		X		X
0 1 1 0 1		X			X		X	X	
0 1 1 1 0		X		X			X		X
0 1 1 1 1		X		X			X	X	
1 0 0 1 0			X	X		X			X
1 0 1 0 0			X		X		X		X
1 0 1 0 1			X		X		X	X	
1 0 1 1 0			X	X			X		X
1 0 1 1 1			X	X			X	X	

Figure 13-2. Tape Command Code Formats

Read Backward

A read backward operation is defined by the Q-byte. The tape unit moves tape backward and places data in storage in reverse of the order in which it was written. The MTAR is decremented by 1 after each data transfer cycle. The unit exception condition is set if a tape mark is detected.

Note: Excessive error indications can result if a seven-track read backward operation is attempted using tapes generated by IBM tape models, or others, prior to the IBM 2400 series tape units.

Write

A write operation is defined by the Q-byte. The tape unit moves forward, writing data from main storage. The subsystem remains busy until after read-back checking of the written data.

The EOT reflective marker indicates that about 25 feet of tape remains on the reel. Ignoring this indication can unwind tape off the reel. When repositioning tape past the EOT marker, the only indication guaranteed is when the reflective is first sensed.

Note: The recommended minimum block length to facilitate noise recognition is 18 bytes.

Control

A control command is to be performed when the Q-byte bits 5, 6, and 7 are zero. Then, the R-byte defines the control command. Figure 13-2 shows the command code bit settings. A control command is initiated at the tape control and tape unit. No transfer of data is involved.

Tape Motion Commands

Rewind: This command rewinds the tape. The tape unit remains loaded when the tape reaches load point. The tape unit is busy until the tape unit reaches the load point.

Rewind/Unload: This command rewinds the tape to load point and automatically unloads it. If the tape unit is at load point when the command is issued, tape immediately unloads because no rewind is required. The tape unit becomes not ready after accepting a rewind/unload command. It remains not ready until made ready by the operator. The subsystem is busy only until the tape unit accepts and begins executing the command.

Erase Gap: The tape unit moves forward, erasing tape for a distance of approximately 3-1/2 inches. When this operation is performed in the end-of-tape area, it sets the unit exception condition. The subsystem is busy during an erase gap operation.

Write Tape Mark: This command causes the subsystem to write a tape mark. A tape mark is a block of special non-data bytes separating files. Tape mark formats are predetermined in the subsystem and require no communication with the system while writing the tape mark. When this operation is performed in the end-of-tape area, it sets the unit exception condition. The subsystem is busy while the tape mark is being written.

Forward Space Block: This command moves tape forward to the next interblock gap. Data is not transferred, and errors associated with that block of data are not detected. When a tape mark is sensed, it sets the unit exception condition. The EOT reflective marker is not recognized during this operation. The subsystem is busy during a forward space block operation.

Backspace Block: This command moves tape backward to the nearest interblock gap or to load point, whichever comes first. No data is transferred. When a tape mark is sensed, it sets the unit exception condition.

Forward Space File: This command moves tape forward to the interblock gap beyond the next tape mark. Data is not transferred and data errors are not detected. The end-of-tape reflective marker is not recognized during this operation. The subsystem remains busy until a tape mark has been detected.

Backspace File: This command moves tape backward to the interblock gap beyond the next tape mark or to load point, whichever comes first. Data is not transferred and data errors are not detected. The subsystem remains busy until a tape mark or the load point has been detected. If the load point marker is detected, the not ready/unit check and backward-at-load-point conditions are set.

Data Security Erase: This command erases tape from the tape's present position to the end-of-tape marker. The subsystem remains busy until the tape unit accepts and begins executing the command. The tape unit remains busy until the erase is completed. This command must be issued only after the tape unit has been put in write status. If the tape unit is in read status or is file protected when a data security erase command is issued, the command reject and not ready/unit check conditions are set.

Erasing data beyond the end-of-tape marker is the responsibility of the user. Fifteen erase gap commands will erase about 4-1/2 feet of tape.

This command is accepted and terminated without error if it is issued when the tape is at end-of-tape and the tape unit is in write status. However, if the tape unit is in read status, a command reject condition is set. The subsystem does not present busy status in either case.

Loop-Write-to-Read: This command is used for diagnostic purposes. It is defined by the Q-byte setting.

Mode Set Commands

Mode set commands are used to select density, parity, data converter, and code translator for seven-track operation. Figure 13-2 shows the mode modifier bit settings that set these conditions. Figure 13-3 gives the subsystem response to mode set commands for write operations.

Mode 1 Set: This sets the control unit to seven-track NRZI operation. It affects operation of all seven-track tape units attached to the tape control. Unless reset, the tape control retains its mode setting until it receives another mode 1 set command. A system reset forces a default condition of X'93'.

Feature Installed on Control Portion of 3411 and Selected 3410/3411 Tape Units	Action Taken by Subsystem (Write)		
	1600 bpi	800 bpi	800/556/200 bpi
Dual Density Control Feature and Dual Density Tape Unit Feature			
1600 bpi mode set*	X		
800 bpi mode set*		X	
Seven-track mode set	Previous Setting		
No mode set	Previous Setting		
Dual Density Control Feature either installed or uninstalled and Dual Density Tape Unit Feature not installed			
1600 bpi mode set	X		
800 bpi mode set	X		
Seven-track mode set	X		
No mode set	X		
Seven-track Control Feature installed and Seven-track Tape Unit Feature installed			
1600 bpi mode set			Previous Setting
800 bpi mode set			Previous Setting
Seven-track mode set			Density Specified
No mode set			Previous Setting
Seven-track Control Feature installed and Seven-track Tape Unit Feature not installed			
1600 bpi mode set	X		
800 bpi mode set	X		
Seven-track mode set	X		
No mode set	X		

*Effective only at load point. If at other than load point, the previous setting is used.

Figure 13-3. Subsystem Response to Mode Set Commands for Write Operations

Mode 2 Set: This sets dual density tape controls to either 1600 bpi (PE) or 800 bpi (NRZI) mode for succeeding write operations. It is effective only when tape is positioned at load point and the tape unit is in ready status. The tape unit retains this mode setting until the tape again reaches load point, at which time the tape unit is automatically set to PE mode (this also applies to rewind operations). The control unit retains the last mode set, and successive operations are performed in that mode unless there is a system reset. The tape control is set to PE mode after a reset.

TAPE INSTRUCTIONS

Start I/O

Mnemonic: SIO

Op Code	Q Byte			Control Code
F3	DA	M	N	Control Code

Operation: This instruction initiates the subsystem's operation.

The Device Address (DA) and M-bit of the Q-byte specify the tape unit that is to be used.

Bits	*Tape Unit*
01100	0
01101	1
01110	2
01111	3

The N-code of the Q-byte specifies the operation to be performed. The operations that can be specified are:

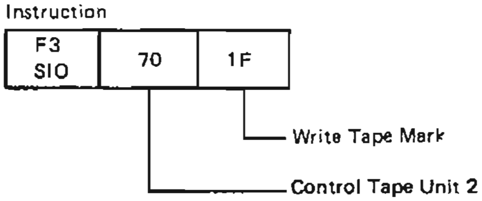
N Bits	*Operation*
000	Control
001	Read Forward
010	Write Data
011	Read Backward
100	Subsystem Diagnostic (write)
101	Subsystem Diagnostic (read)
110	Adapter Diagnostic (write)
111	Adapter Diagnostic (read)

The third byte contains the control code. It is used only when bits 5 through 7 of the Q-byte are equal to 000, 100, or 101. Q-byte 100 or 101 is used for diagnostic purposes. For control coding, see Figure 13-2.

A starting address and a record length must be provided prior to a read, read backward, or write operation. Not ready/unit check and no-op are set if they are not given.

The condition (I/O working) becomes active when the start I/O instruction is accepted. The instruction is not performed if the no-op bit is on, the I/O attention or busy condition exists, or parity of the Q-byte is incorrect. Any start I/O resets all sense information except no-op, which is reset only with a sense to the byte in which it exists.

Example:



Load I/O

Mnemonic: LIO

Op Code	Q Byte			Operand Address	
Y1	DA	M	N		

Operation: The operand address portion of the instruction addresses a two-byte operand. The operand is addressed by its rightmost byte and remains unchanged. As specified by the N-field, the operand is loaded into either the MTAR in the CPU or the byte count register in the tape subsystem. Each register contains two bytes, which must be loaded prior to each SIO instruction (write, read, or read backward) following an LIO. Any SIO other than write, read, or read backward resets the byte count to zero. The program must provide the address of the tape unit that uses this register information.

Note: An LIO must load the starting address and byte count for a read, read backward, or write command. For a read forward or write operation, the LIO must load the data starting address in the MTAR. This address is incremented by 1 after each data transfer. For a read backward operation, the LIO must load the MTAR with the ending address of the location where data is to be stored. This address is decremented by 1 after each data transfer.

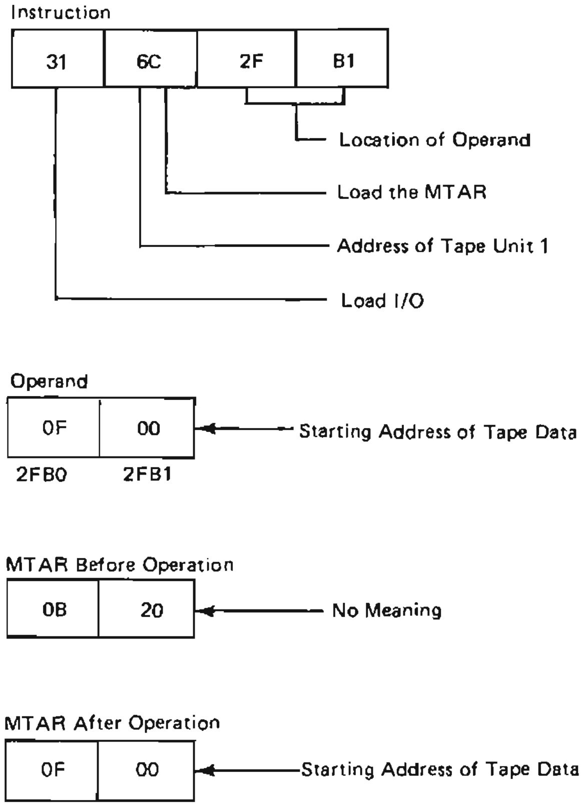
The N-code of the Q-byte specifies two values:

N-Bits	*Condition*
000	Load byte count register. The byte count register is loaded with a value equal to the number of bytes to be transferred between the CPU and the tape subsystem.
100	Load the MTAR. The MTAR is loaded with the main storage starting address while data is written in or read out during a tape operation.

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

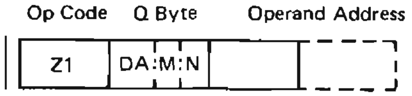
The LIO instruction is accepted if the subsystem and addressed device are not busy. The instruction is rejected if the subsystem or device are busy.

Example:



Test I/O and Branch

Mnemonic: TIO



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

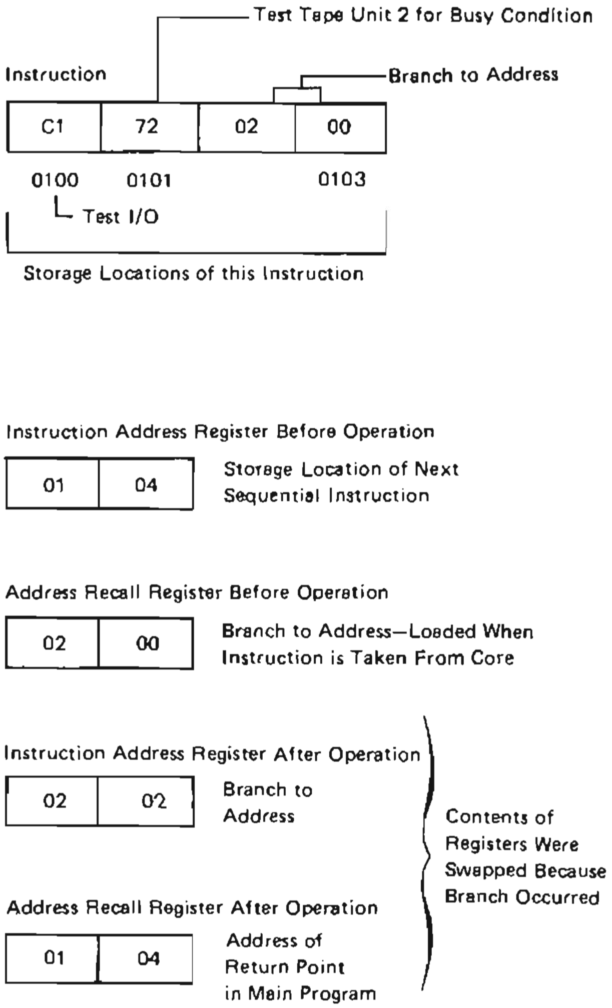
The Q-byte specifies the tape unit and the condition to be tested. The DA and M-bit portion of the Q-byte specify the tape unit.

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

1. N-code 000—not ready/check. This condition occurs any time the addressed tape unit becomes not ready. This condition is removed when the reason for the not ready condition is corrected.
2. N-code 010—busy. This condition occurs for all addresses when the subsystem is executing a command. It also occurs for a specific addressed tape unit when that tape unit is executing a rewind or a data security erase, or when it accepts a rewind/unload command.

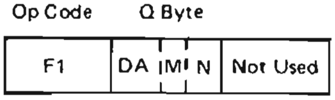
Any N-code other than those listed causes a processor check stop with an invalid Q-byte indication.

Example:


Test Tape Unit 2 for Busy Condition
Instruction
Branch to Address
C1
72
02
00
0100
0101
0103
Test I/O
Storage Locations of this Instruction
Instruction Address Register Before Operation
01
04
Storage Location of Next Sequential Instruction
Address Recall Register Before Operation
02
00
Branch to Address—Loaded When Instruction is Taken From Core
Instruction Address Register After Operation
02
02
Branch to Address
Contents of Registers Were Swapped Because Branch Occurred
Address Recall Register After Operation
01
04
Address of Return Point in Main Program

Advance Program Level

Mnemonic: APL



Operation: This instruction tests for the condition specified in the N-code. If the tested condition is present, a system with the dual programming feature would activate the inactive program level; a system without the dual programming feature would loop on the advance program level instruction until the condition no longer existed. If the condition is not present, systems with or without the dual programming feature would take the next sequential instruction in the active program level.

The Q-byte specifies the tape unit and the condition to be tested. The DA and M-bit of the Q-byte specify the tape unit to be used:

Bits	*Tape Unit*
01100	0
01101	1
01110	2
01111	3

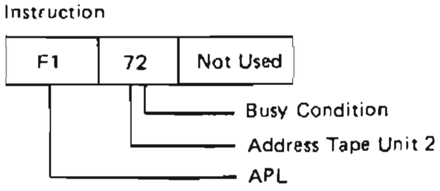
The N-code of the Q-byte can specify testing for any of these conditions.

1. N-code 000—not ready/check. This condition occurs any time the addressed tape unit becomes not ready. The condition is removed when the reason for the not ready condition is corrected.

2. N-code 010—busy. This condition occurs for all addresses when the subsystem is executing a command. It also occurs for a specific addressed tape unit when that tape unit is executing a rewind or a data security erase operation, or when it accepts a rewind/unload command.

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

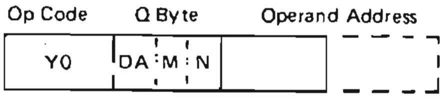
Example:



The next instruction address is taken from the instruction address register of the program level that did not execute this instruction.

Sense I/O

Mnemonic: SNS



Operation: This instruction causes two sense bytes, from the subsystem attachment or the MTAR, to be transferred to the two-byte operand in main storage, addressed by the operand address. The operand is addressed by its low-order byte. The low-order sense byte is stored at the operand address; the high-order sense byte is stored at the operand address minus 1.

The DA and M-bit of the Q-byte specify the tape unit to be used.

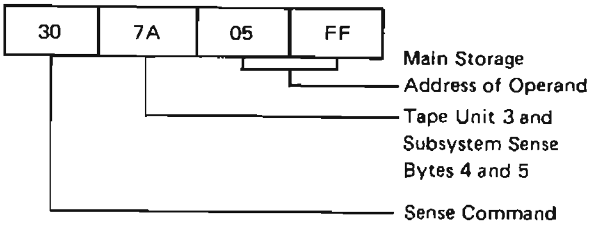
Bits	*Tape Unit*
01100	0
01101	1
01110	2
01111	3

The Q-byte specifies the subsystem, attachment, or MTAR bytes to be transferred. The device address must be the address of the tape unit that caused the error being sensed (except when sensing the CPU MTAR). The N-code of the Q-byte specifies a certain subsystem, attachment, or the MTAR bytes transferred as follows:

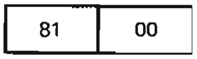
1. N-code 000–subsystem bytes 0 and 1
2. N-code 001–subsystem bytes 2 and 3
3. N-code 010–subsystem bytes 4 and 5
4. N-code 011–subsystem bytes 6 and 7
5. N-code 100–MTAR
6. N-code 101–attachment sense
7. N-code 110–subsystem hardware error sense
8. N-code 111–invalid (causes processor check)

Example:

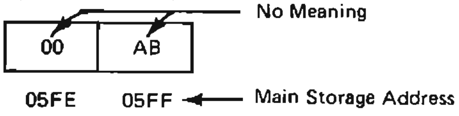
Instruction


30
7A
05
FF
Main Storage Address of Operand
Tape Unit 3 and Subsystem Sense Bytes 4 and 5
Sense Command

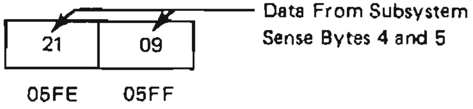
Subsystem Sense Bytes 4 and 5



Operand Before Operation



Operand After Operation



SUBSYSTEM SENSE INFORMATION

Tape error conditions and general status information about the tape are conveyed to the processing unit as bits in sense bytes. Figure 13-4 shows significant sense-byte information; Figure 13-5 shows conditions that set bits in sense bytes 1, 2, 3, and 5. The following information describes the various conditions.

A valid sense instruction is always executed, either by the attachment or the subsystem. A complete sense must be performed every time the not ready/unit check indication is on. A complete sense includes the two attachment sense bytes and all eight subsystem sense bytes. To perform a complete sense, an SNS instruction must be issued to each of the four Q-byte configurations that define subsystem sense bytes. An SNS instruction does not reset any sense or status information (except for no-op when sense byte 0 is sensed).

13

Bit	Byte 0	Byte 1	Byte 2	Byte 3	Byte 5
0	Noise	Data Converter Check	Reserved	Tape Mark Check	Attachment Bus Out
1	Wrong Length Record	Command Reject	Reserved	End Velocity Check	Multitrack/LRC
2	Unit Exception	Backward at Load Point	Tape Indicate	Tape Unit Position	Data Timing Check
3	Data Check	Start Velocity Check	Reserved	Reject Tape Unit	End Data/CRC
4	Diagnostic Track Error	Illegal Command	Reserved	Reserved	Envelope Check
5	No-op (NOP)	Tape Unit Status Changed	Reserved	No Readback Data	False End Marker
6	Equipment Check	Word Count Zero	Tape Unit Check	Tachometer Failure	PE ID Burst Check
7	Sense Valid	Not Capable	Reserved	Overrun	VRC

Figure 13-4. Tape Sense Byte Information

			Sense Byte 0 Bits			
Sense Byte	Bit	Condition*	Unit Exception Bit 2	Data Check Bit 3	No-op Bit 5	Equipment Check Bit 6
1	0	Data Converter Check		X		
	1	Command Reject			X	
	2	Backward at Load Point			X	
	3	Start Velocity Check			X	
	4	Illegal Command			X	
	5	Tape Unit Status Changed			X	
	6	Word Count Zero			X	
	7	Not Capable			X	
2	2	Tape Indicate	X			
	6	Tape Unit Check				X
3	0	Tape Mark Check				X
	1	End Velocity Check		X		
	2	Tape Unit Position				X
	3	Reject Tape Unit				X
	5	No Readback Data				X
	6	Tachometer Failure				X
	7	Overrun		X		
5	0	Attachment Bus Out Check		X		
	1	Multitrack/LRC		X		
	2	Data Timing Check		X		
	3	End Data/CRC		X		
	4	Envelope Check		X		
	5	False End Marker		X		
	6	PE ID Burst Check		X		
	7	Vertical Redundancy Check		X		
			Bit 2	Bit 3	Bit 5	Bit 6

*These conditions set sense byte 0 bits designated by an X.

Figure 13-5. Tape Subsystem Sense Information

An SNS instruction causes the subsystem to request an I/O cycle steal after the Q-byte. This provides time for the subsystem to assemble the requested sense information. No data is transferred during the I/O cycle, and the MTAR does not change.

Before the instruction is performed, the MTAR must have proper parity and the address must be less than the system's memory size. Improper parity or an address equal to or larger than the memory size causes a processor check stop when the MTAR is used during this I/O cycle.

Sense information is defined by byte 0, bit 7 of the attachment and subsystem sense bytes. Sensing a tape subsystem that is busy (performing a command) or incapable (hardware error) forces attachment sense bytes (CE sense bits) to the CPU in place of the subsystem sense bytes requested. This is indicated by bit 7 being off. The subsystem sense was performed properly if bit 7 is on.

Sense Byte 0

Bit 0—Noise

This bit indicates that a block of information read in NRZI mode was 12 or fewer bytes long, and that a data check occurred. It also indicates that signals were detected during the read-back check of an erase-gap operation (PE or NRZI).

Bit 1—Wrong Length Record

This bit indicates that the number of bytes in a block is different from the CPU byte count.

Bit 2—Unit Exception

This bit indicates that an end-of-tape (EOT) marker was detected during a write operation. It also indicates that a tape mark was detected during a read forward or space block operation.

Bit 3—Data Check

This bit indicates an error occurred that can be retried after proper tape positioning.

Bit 4—Diagnostic Track Error

This bit indicates that a dead track was found during a Loop-Write-to-Read (LWR) operation. Normally, this condition is corrected during a read operation. This bit is used as a maintenance aid.

Bit 5—NOP

This bit indicates that a command was accepted but could not be executed. The command can be retried.

Bit 6—Equipment Check

This bit indicates that a command could not be sent to the addressed tape unit, the tape position was unknown, or a tape mark could not be properly written.

Bit 7—Sense Valid

This bit is used to differentiate between attachment and subsystem sense bytes. If this bit is active (1), the subsystem sense was performed properly; otherwise, subsystem sense bytes 0 and 1 were replaced with attachment sense bytes 0 and 1.

Sense Byte 1

Bit 0—Data Converter Check

This bit indicates that the data block read during a seven-track read forward operation did not contain a multiple of four bytes (see Special Features). The data check bit is also set.

Bit 1—Command Reject

This bit indicates that a file-protected tape unit was issued a write, write tape mark, or erase gap command. It also indicates that a tape unit in read status was issued a data security erase command. The no-op check bit is also set.

Bit 2—Backward at Load Point

This bit indicates that the tape was moved into the load point or a command was issued at load point during any backward operation. The no-op bit is also set.

Bit 3—Start Velocity Check

This bit indicates that the selected tape unit had not attained the proper speed when the data was ready to be written. The no-op bit is also set.

Bit 4—Illegal Command

This bit indicates that a command other than one of the start I/O commands was received in the R-byte by the tape subsystem. The no-op check bit is also set.

Bit 5—Tape Unit Status Changed

This bit indicates that a start I/O instruction was accepted but could not be executed because the tape unit was not ready. The no-op bit is also set.

Bit 6—Word Count Zero

This bit indicates that the byte count was zero at the start of a read, read backward, or write data operation. The no-op bit is also set.

Bit 7—Not Capable

This bit indicates that, during a read from load point, a PE identification was not detected on a subsystem with (1) a PE-only control unit or (2) a dual-density control unit with a PE-only tape unit addressed. The no-op bit is also set.

Sense Byte 2

Bit 2—Tape Indicate

This bit indicates that a write, erase gap, data security erase, or write tape mark operation was performed when the tape was positioned at, or past, the end-of-tape reflective marker. This bit remains set until any backward operation over the reflective marker is completed. The unit exception bit is also set.

Bit 6—Tape Unit Check

This bit indicates that an error occurred in the tape unit. Subsystem sense byte 6 (bits 0, 1, 2, or 3) indicates the condition that caused the tape unit check. The equipment check bit is also set.

Sense Byte 3

Bit 0—Tape Mark Check

This bit indicates that a tape mark was improperly written. The subsystem automatically repositions the tape and retries the write tape mark operation up to 15 times. If the tape mark cannot be properly written during a retry operation, the equipment check bit is set; otherwise, no error condition is set.

Bit 1—End Velocity Check

This bit indicates that the tape, at the end of the read back check of a write operation, was not moving at proper speed. The data check bit is also set.

Bit 2—Tape Unit Position

This bit indicates that, during selection of a tape unit, the tape was positioning within the IBG when it was expected to be stopped. The equipment check bit is also set.

Bit 3—Reject Tape Unit

This bit indicates that a selected tape unit failed to respond to set read or write status when instructed, or became not ready during execution of a tape motion operation. The equipment check bit is also set.

Bit 5—No Readback Data

This bit indicates that the data was not sensed at the read head during a write operation. The equipment check bit is also set.

Bit 6—Tachometer Failure

This bit indicates the absence of tachometer pulses when the tape should be in motion. The equipment check bit is also set.

Bit 7—Overrun

This bit indicates that data cannot be transferred during read, write, or read backward operations. Data transfer stops as soon as the condition is detected. The data check bit is also set.

Sense Byte 5

Bit 0—Attachment Bus Out Check

This bit indicates that, during a write operation, the data from the CPU to the tape subsystem had even parity. The data check bit is also set.

Bit 1—Multitrack/Longitudinal Redundancy Check (LRC)

This bit indicates that a tape unit in PE mode had envelope dropout in two or more tracks and/or a phase error after a read, read backward, or write operation. It also indicates that the LRC register is not zero or has incorrect parity after a read, read backward, or write operation in NRZI mode. The data check bit is also set.

Bit 2—Data Timing Check

This bit indicates that the bits within a data byte are excessively misaligned during a read or read backward operation in PE mode or during a write operation in NRZI mode. The data check bit is also set.

Bit 3—End Data/Cyclic Redundancy Check (CRC)

This bit indicates that the ending burst of bits following a data block during a read or read backward operation in PE mode was not detected. In NRZI mode, it indicates that the CRC byte read from tape did not match the CRC pattern regenerated while the data block was being read. It also indicates, during NRZI write operations, that the CRC byte parity during read back check has improper parity, or that the CRC pattern read back did not match the pattern that was written. The data check bit is also set.

Bit 4—Envelope Check

This bit indicates that an envelope check or phase error occurred on a read, read backward, or write operation in PE mode. The data check bit is also set during write operations. In read operations, data check is set only if an uncorrectable error occurred with an envelope check or phase error.

Bit 5—False End Marker

This bit indicates that an incorrect end marker was detected during a read or write operation. The data check bit is also set.

Bit 6—PE ID Burst Check

This bit indicates that the PE ID (identification) burst was improperly written, or a start velocity error occurred during PE write operations. It also indicates that a start velocity error occurred during NRZI write operations from load point. The data check bit is also set.

Bit 7—Vertical Redundancy Check (VRC)

This bit indicates that a parity error occurred on read data, which was not corrected during a read or read back operation. It also indicates that a parity error occurred during the read back check of a write operation. The data check bit is also set.

SUGGESTED ERROR RECOVERY PROCEDURES

The following minimum error recovery procedures are defined for the 3410/3411 tape subsystem to achieve acceptable performance and read/write reliability.

General Actions

The system logs the number, severity, and type of I/O errors that occur while processing each reel of tape. This log helps the CE determine whether a problem is tape or machine oriented.

An operating system provides the following facilities:

- Operator control interface
- Additional programmed recovery interface

First, exit to the operator control interface if both of the above items are defined.

Some of the operator control interface options that can be defined are:

- Retry the recovery procedure
- Continue to additional programmed recovery interface
- Dump the failing record
- Cancel the job

13

Messages

Any operator message (printed or message display unit) issued for error recovery procedures should contain the following information:

- Message code
- Error condition that caused the message

Sense Procedures

When an error occurs, sense information must be taken as follows:

1. Obtain and analyze attachment sense bytes 0 and 1 before attempting any subsystem sense byte actions.
2. Perform attachment sense byte actions.
3. Obtain subsystem sense bytes 0-1 and verify valid sense.
4. Obtain sense bytes 2-3, 4-5, and 6-7, (in that order) when the sense valid bit is set.

When sense is (or has become) valid, successive sense instructions must be executed within 30 ms of each other. Otherwise, the sense information in bytes 2-7 becomes invalid due to normal subsystem activity.

Sense Instructions

Attachment sense (2 bytes) and subsystem sense (8 bytes) must be executed to the failing unit, without any intervening instructions to the subsystem when an error is detected. The subsystem hardware error sense byte (an additional sense byte) is available for hardware error information. This byte is sensed only when specified by the error recovery procedure. Update the tape error statistics with this sense information.

The sense bytes and bits must be tested in the order (priority) shown in Figure 13-6, and the actions must be performed as described in Figure 13-7.

				Applicable for			Perform
Priority	Byte	Bit	Condition	Read	Write	Control	Action
Attachment Sense							
1	0	3	Tape Control Disabled	X	X	X	A
2	0	5	Subsystem Busy	X	X	X	B
3	0	1	ABI Parity Error	X	X	X	C
4	0	2	ABO Parity Error	X	X	X	C
5	0	4	Two Tag Error	X	X	X	D
6	0	6	Sequence Error	X	X	X	D
7	–	–	No Error Found	–	–	–	E
Subsystem Sense							
8	0	7	Sense Valid	X	X	X	I
9	0	6	Equipment Check	X	X	X	II
10	5	6	PE ID Burst Check		X	X	IV
11	0	5	No-op	X	X	X	III
12	0	0	Noise	X		X	V
13	0	3	Data Check	X			VI
13	0	3	Data Check		X		VII
13	0	3	Data Check			X	VIII
14	0	2	Unit Exception	X	X	X	IX
15	0	1	Wrong Length Record	X			X
16	–	–	No Error Indicated	X	X	X	XI

Figure 13-6. Attachment and Subsystem Sense Information

Action A

1 If the subsystem is busy, issue a message to tell the operator to enable the tape unit, then stop, Upon operator restart, proceed with the job.

2 If the subsystem Is not busy, perform a subsystem hardware error sense operation, perform an operator message, and stop. Attempt at least one job restart if a hardware error occurred.

Action B

1 Repeat the attachment sense operation for at least 15 times.

2 If the busy condition persists, perform a subsystem hardware error sense operation. Log the error, perform an operator message, and stop.

3 If the busy condition ends, continue checking the sense information.

Action C

1 Log the error and retry the operation up to 15 times.

2 If unsuccessful, the condition becomes a permanant error.

Action D

1 Perform a subsystem hardware error sense operation.

2 Log the error, perform an operator message, and await operator action.

3 One job restart is recommended. (These errors are not recoverable and can be reset only with a system reset.)

Action E

1 Continue checking the subsystem sense bytes.

Action I

1 This bit on indicates a vaild sense; continue checking the subsystem sense bits.

2 This bit off indicates an invalid sense; repeat the entire sense operation.

Action II

1 Perform a complete sense operation and make the information available to the CE.

2 Total all current statistical counter and make the total available to the CE.

3 Issue an operator message, and await operator action.

4 One job restart is recommended.

Action III (This condition is set by subsystem sense byte 1.)

1 If bit 1 is on, issue a message to tell the operator to install the write-enable ring, and await operator action. Reissue the command sequence.

2. If bit 2 is on, the error can be handled by the operating program. If so, return to the operating program, otherwise, log the error, issue an operator message, and await operator action.

3 If bits 3 or 5 are on, reissue the instruction sequence up to 15 times. If the error persists, issue an operator message and await operator action.

4 If bits 4, 6, or 7 are on, log the error, issue an operator message, and await operator action.

Figure 13-7 (Part 1 of 3). Tape Error Recovery Procedures

Action IV

1 Rewind tape and reissue the command sequence up to 15 times.

2 If the error is correctable, log the error and return to the operating program.

3 If the error is not correctable in 15 retries, log the error, issue an operator message, and await operator action. Suggested operator action: Move the load point marker one or two centimeters toward the center of the tape reel and restart.

Action V

1 Read operation: Tell the operating program about the condition and let it decide whether this is a noise block on tape. If this condition is treated as a noise block, log the error, and return to the operating program. If a block of 12 or fewer bytes was expected, retry the read as outlined in *Action VI.* If a block of 12 or fewer bytes was not expected, discard the data as a noise block and reissue the same instruction sequence to read the expected block.

2 Erase gap operation: Perform *Action VII.* If successful, tell the operator an erase gap check occurred. If unsuccessful, a permanent write error results.

3 This condition may also result from generating a tape that cannot be read properly. There is no reliable error recovery for this failure except a Job restart. The error may be due to faulty tape or a temporary erase error. Give the operator the option of proceding with the job, after the message is given, or cancelling the job.

Action VI The data converter is invalid during read backward operations in seven-track mode. A mode-1 set command can be issued at any time, whether required or not. Retry the read 40 times in the same direction as the original read, then 40 times in the opposite direction.

1 Space the block in the opposite direction of the read in which the error occurred.

2 Set correct seven-track mode (if seven-track tape) and reissue the instruction sequence to reread in the same direction in which the error was originally detected. If the error does not exist, proceed to step 8. If the error remains, repeat steps 1 and 2 three more times. If the error still persists after a total of four rereads, issue a cleaner-blade operation as described in step 7, and return to step 3.

3 Read in the direction opposite that performed in step 2. If the error does not exist, proceed to step 8. If the error persists, proceed to step 4.

4 Space block in the direction opposite that performed in step 3.

5 Set correct seven-track mode and reissue the instruction sequence to read in the same direction as in step 3.

6 If the error persists, repeat steps 4 and 5 three more times. If the error still persists after a total of four rereads, issue a cleaner-blade operation as described in step 7. If this corrects the error, proceed to step 8. If the error persists, repeat steps 1 through 6 ten times. If the error still persists after all this, proceed to step 9.

7 Cleaner-blade operation: Perform this operation by issuing five backspace block commands followed by five forward space block commands. If, during a tape cleaner operation, load point is reached in n backspaces, reposition the tape with n-1 forward spaces. If a tape mark is encountered in n space block operations, reposition the tape with n space block commands in the opposite direction. Repeat steps 1 through 6 until the record is read successfully or until a minimum of 80 retries (as described in steps 1 through 6) have been performed.

8 Log the error and return to the operating program.

9 The error is permanent if it still persists. Perform a complete SNS by issuing a loop-write-to-read operation, using any available data, then testing for not ready/unit check. If the condition is satisfied, perform a complete sense operation and log the sense information. If the condition is not met, log the previous sense information. In either case, make the total of all statistical counters available to the CE, then issue an operator message and await operator action.

Figure 13-7 (Part 2 of 3). Tape Error Recovery Procedures

Action VII

1 Check for unit exception. If unit exception (end-of-tape) is not properly handled, it can be lost and writing off the end of the tape can result.

2 Reposition the tape, issue an erase gap command and reissue the instruction sequence. Repeat the procedure 15 times. If the error does not recur during the retry, log the error and return to the operating program. If the error persists, follow the procedures in *Action VI,* step 9, but use the previous data for the loop-write-to-read operation.

Action VIII

1 Erase gap command. This command is performed by issuing a rewind command and reissuing the original command. Repeat the procedure at least 15 times. If the error does not recur during the retry, log the error and return to the operating program. If the error persists, follow the procedure described in *Action VI,* step 9.

2 Commands other than erase gap. Log the error, issue an operator message, and await operator action.

Action IX

1 Log the unit exception condition.

2 Return to the operating program.

Action X

1 Log the wrong length record condition.

2 Return to the operating program.

Action XI This condition indicates an unexpected tape unit status. If none of the following conditions exist, log the error and await operator action. One job restart is recommended. If any of the following conditions exist, log the error, issue an operator message, and await operator action.

1 Subsystem sense byte 2, bit 6 indicates that the tape unit had a failure. Subsystem sense byte 6, bits 0-3 define the failure.

2 Subsystem sense byte 2, bits 5 and 7 off, indicates a power-off condition on the tape unit or a disconnection to the tape unit.

3 Subsystem sense byte 2, bits 5 and 7 on, indicatas a tape unit not ready. Expect one of these conditions: dropped ready, manually reset once it was in reedy status, or a rewind/unload was issued.

Figure 13-7 (Part 3 of 3). Tape Error Recovery Procedures

ERROR RECORDING

Error Statistic Counter Assignments

Figure 13-8 shows the format of the combined volume error and statistical data recording counters. These counters should be updated immediately before stopping due to an uncorrectable error and also before returning to the operating program when an error has been corrected. The attachment and subsystem sense bytes, logged because of an error, should be printed out daily or after each job.

Counter	Bytes	By Unit	By Volume
Noise Blocks	1	X	X
Write Skips	1	X	X
Start I/O	2	X	X
Temporary Read Forward	1	X	X
Temporary Read Backward	1	X	X
Temporary Write	1	X	X
Diagnostic Track Error	1	X	
Short Gap Mode	1	X	
Multitrack Error	2	X	
End Data Check	1	X	
Envelope Check	1	X	
End Velocity	1	X	
TIE Byte	4	X	
Volume Identification	1	X	
Device Type	1/2	X	
Overrun	1	X	
Tape Mark Check	1	X	
PE ID Burst Error	1	X	
Start Velocity	1	X	
Write Feedthrough	1	X	
False End	1	X	
No Readback Data	1	X	
VRC	1	X	
First and Last Volume Serial Number	6	X	
Q-byte	1	X	
Tape Density	1/2	X	
Block Length	2	X	

Figure 13-8. Combined Volume Error and Statistical Data Recording Counters

Error Card Formatting

The CE log analysis programs require cards containing temporary and permanent error information. Figure 13-9 shows the format of the temporary error card; Figure 13-10 shows the format of the permanent error card.

Description	Column
V-record type	1
System Date	2-7
Volume ID for First Volume on This Unit	8-13
Volume ID for Last Volume on This Unit	14-19
Q-byte	20
Density 8=800; 1=1600	21
Device Type (decimal equivalent of bits 5, 6, and 7 of sense byte 4)	22
Block Length	23-26
Number of Volumes	27-28
Start I/O Counter (last volume)	29-32
Noise Block Counter (last volume)	33-34
Write Skip Counter (last volume)	35-36
Temporary Read Forward Errors (last volume)	37-38
Temporary Read Backward Errors (last volume)	39-40
Temporary Write Errors (last volume)	41-42
Start I/O Counter (all volumes)	43-46
Noise Block Counter (all volumes)	47-48
Write Skip Counter (all volumes)	49-50
Temporary Read Forward Errors (all volumes)	51-62
Temporary Read Backward Errors (all volumes)	63-64
Temporary Write Errors (all volumes)	65-56
Diagnostic Track Errors	57-58
Start Velocity Check	59-60
Tape Mark Check	61-62
End Velocity Check	63-64
Write Feed Through	65-66
No Read Back Data	67-68
Overrun	69-70
Short Gap Mode	71-72
Multitrack Error	73-76
End Data Check	77-78
Envelope Check	79-80
False End Marker	81-82
PE ID Burst Error	83-84
VRC	85-86
Track in Error Data	87-94
Blank	95-96

Figure 13-9. Format of the Temporary Error Card

Description	Column
O-Record Type	1
System Date - MMDDYY	2-7
Volume ID for Volume Presently Mounted (commas if volume ID cannot be determined) (blanks for non-labeled tapes)	8-13
Q-byte	14-15
R-byte	16-17
Adapter Sense Bytes	18-21
Subsystem Sense Bytes	22-37
Blank	38-96

Figure 13-10. Format of the Permanent Error Card

An IBM 1442 Card Read Punch Model 6 or 7 can be attached to an IBM System/3 to provide 80-column card reading and punching. See Figures 14-1 and 14-2.

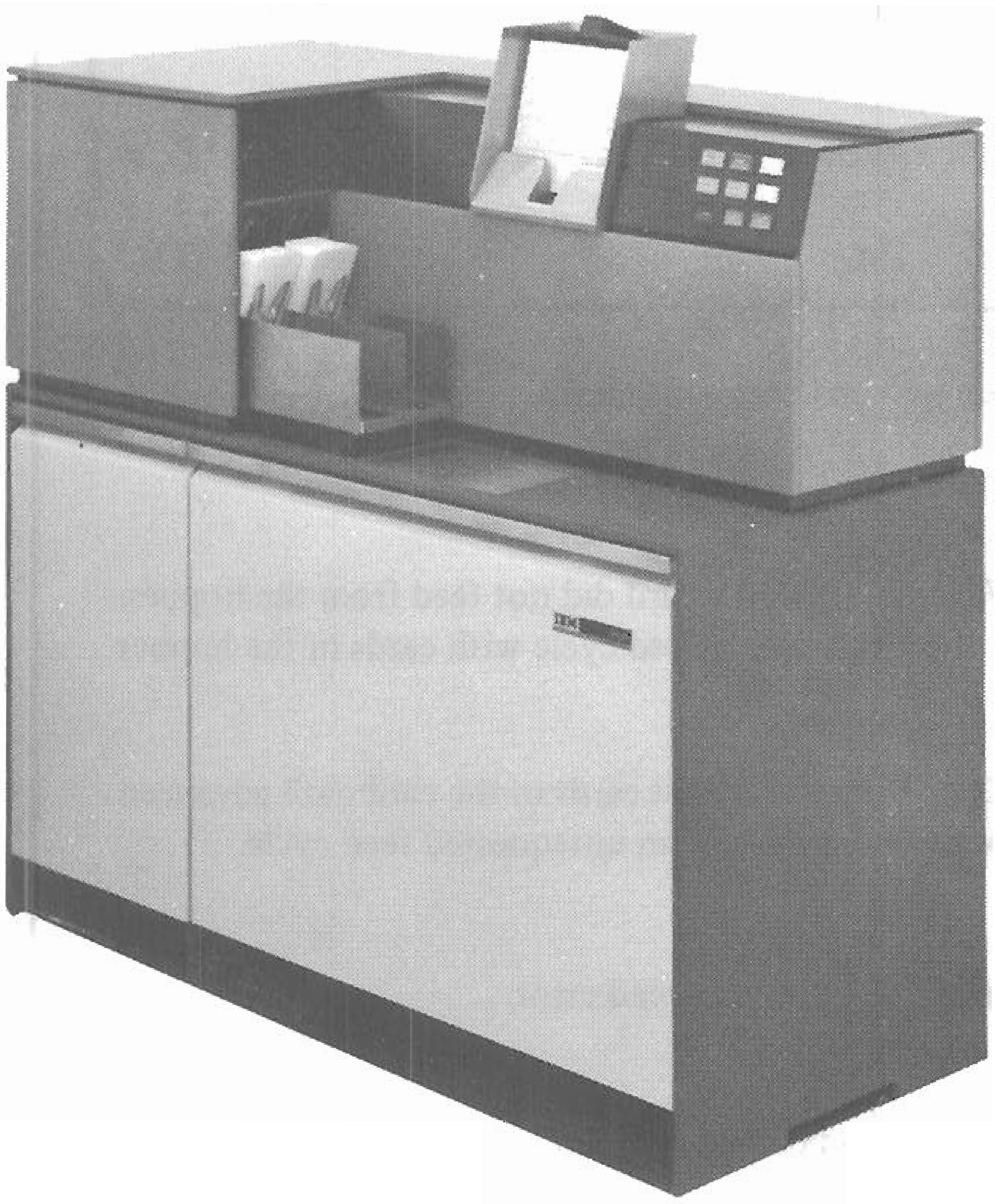


Figure 14-1. IBM 1442 Card Read Punch

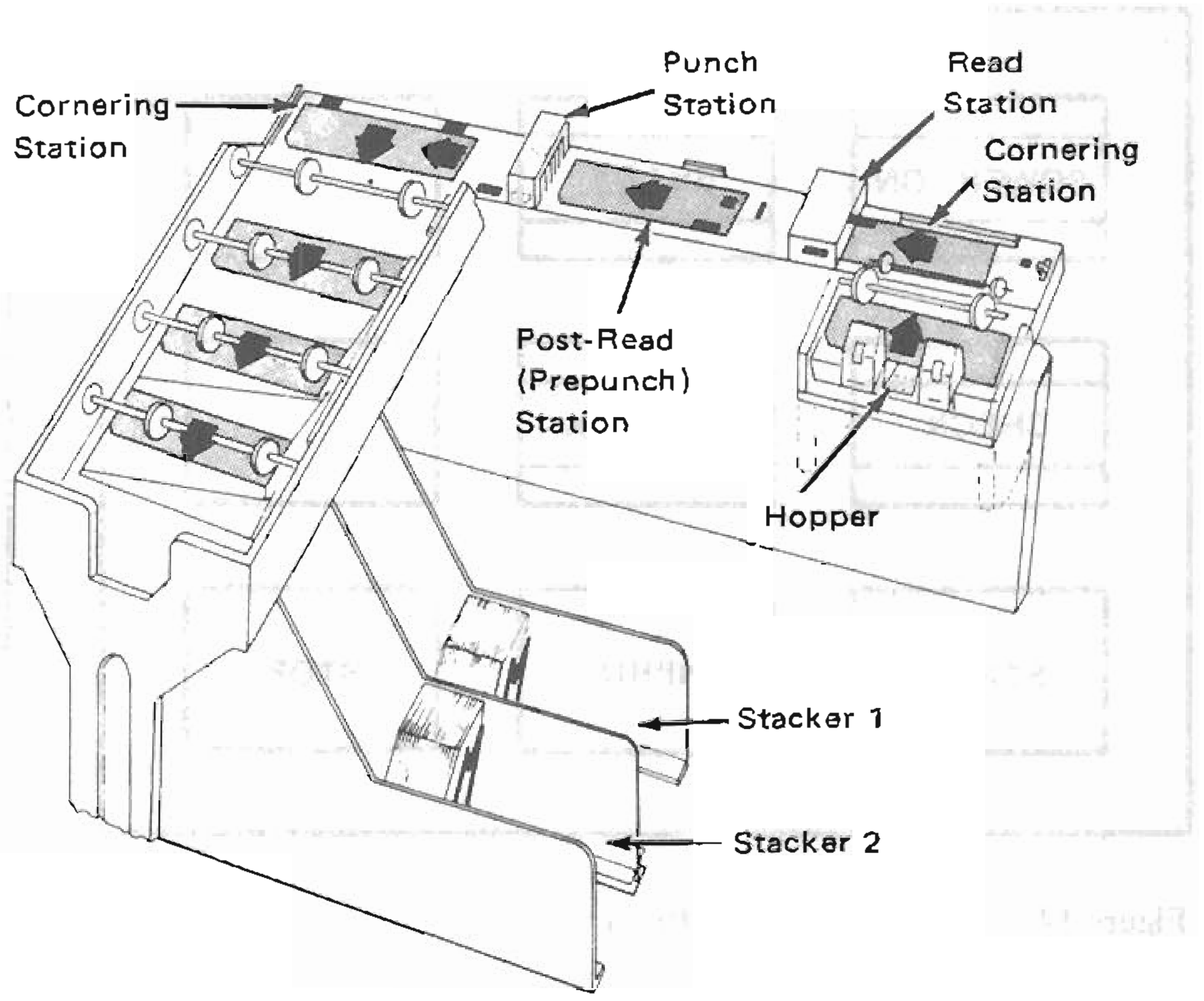


Figure 14-2. 1442 Card Path

The following operations can be performed on the 1442:

1. Read and punch with no feed
2. Read column binary
3. Read
4. Punch and feed

Each of these operations is initiated by a start I/O instruction. The operation is specified by the Q byte and the third byte of the instruction, called a control code.

1442 Operator Panel

Figure 14-3 shows the operator's panel.

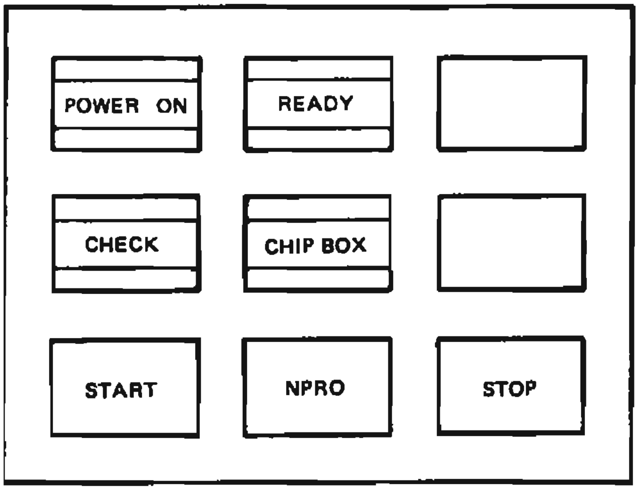


Figure 14-3. 1442 Operator's Panel

Power On Light

This light indicates that system power is on.

Ready Light

This light indicates that the 1442 is ready for processing. The ready light turns on when the start key is pressed and all the following conditions apply:

1. System power is on.
2. Cards are in the hopper.
3. Stacker is not full.
4. The check light and the chip box light are off.
5. The 1442 covers are closed.

Check Light

This light turns on when any of the error indicators (Figure 14-4) turn on.

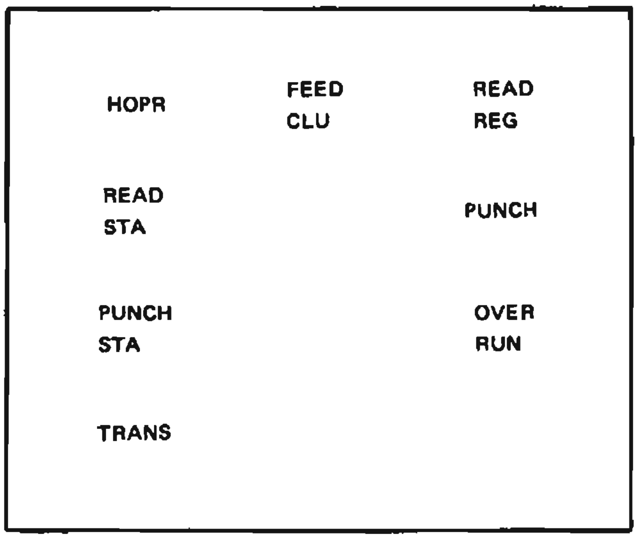


Figure 14-4. 1442 Error Indicators

HOPR indicates that a card did not feed from the hopper when the 1442 took a feed cycle with cards in the hopper.

FEED CLU indicates that cards in the card path advanced one position because of an unrequested feed cycle.

READ REG indicates a read error.

READ STA indicates a card jam at the read station.

PUNCH indicates a punch error.

PUNCH STA indicates a card jam at the punch station.

OVER RUN indicates that data was lost because the processing unit was unable to accept data from the 1442 or send data to the 1442 fast enough.

TRANS indicates a card jam in the stacker area.

Chip Box Light

This light indicates that the chip box is full or out of place.

Start Key

This key places the 1442 in ready status if the following conditions apply:

1. System power is on.
2. Cards are in the hopper.
3. Stacker is not full.
4. The check light and the chip box light are off.
5. The 1442 covers are closed.

The start key is also used to return the 1442 to a ready status after the 1442 stop key has been pressed.

NPRO Key

This key clears all cards from the card feed path. The hopper should be empty before this key is pressed. The first card that enters the stacker after this key is pressed is read but not punched. The second card that enters the stacker is neither read nor punched.

If the hopper is not empty when this key is pressed, the cards will not be cleared from the feed path.

Stop Key

This key causes the 1442 to stop after the operation in process has been completed.

INSTRUCTIONS

Test I/O and Branch

Mnemonic: TIO

Op Code	Q Byte			Branch-to-Address
Z1	0101	0	N	

Operation: This instruction tests for the conditions specified in the Q byte. If the condition tested for is present, the next instruction is taken from the storage location specified by the operand address. If the condition does not exist, the next sequential instruction is executed.

The Q byte contains the device address (always 0101 for the 1442), an M bit of 0, and an N code. The N code can specify testing for two conditions:

1. N code 000 – Not ready/unit check. This condition indicates that one of the following conditions has occurred:

   Feed check (not ready)
   Read check (unit check)
   Punch check (unit check)
   No-op (unit check or not ready)
   I/O attention (not ready)

2. N code 010 – The 1442 is feeding, reading, or punching a card.

Any N code other than 000 or 010 causes a processor check.

Advance Program Level

Mnemonic: APL

Op Code	Q Byte			
F1	0101	0	N	Not Used

Operation: If the dual programming feature is used, the condition specified by the N portion of the Q byte is tested. If the condition exists, the address of the next instruction is taken from the instruction address register of the program level that is not active at the time the APL instruction is encountered. The program on this level now becomes the active program level; the program level from which the advance occurred becomes the inactive program level. If the condition is not present, the next sequential instruction is taken and no program level advance occurs. If a conditional program level advance occurs, the return point to the program level advanced from is the address of the start of the advance program level instruction. The return point for an unconditional program level advance is the next sequential instruction.

If the dual programming feature is not used, the program loops on the advance program level instruction until the specified condition is no longer present, then executes the next sequential instruction. An unconditional advance program level instruction results in execution of the next sequential instruction.

14

The Q byte contains the device address (always 0101 for the 1442), an M bit of 0, and an N code. The N code can specify testing for two conditions:

1. N code 000 – Not ready/unit check. This condition indicates that one of the following conditions has occurred:

   Feed check (not ready)
   Read check (unit check)
   Punch check (unit check)
   No-op (unit check or not ready)
   I/O attention (not ready)

2. N code 010 – The 1442 is feeding, reading, or punching a card.

Any N code other than 000 or 010 causes a processor check.

Load I/O

Mnemonic: LIO

Op Code	Q Byte			Operand Address
Y1	0101	0	N	

Operation: The contents of the two-byte field addressed by the operand address is moved to the local storage register designated by the Q byte. If the dual programming feature is used and the selected register is busy, an unconditional program advance occurs. If the dual programming feature is not used and the selected register is busy, the program loops on the load I/O instruction until the register is not busy.

The Q byte contains the device address (always 0101 for the 1442), an M bit of 0, and an N code. The N code spcifies the register to be loaded as follows:

1. N code 000 – Length count register.
2. N code 100 – 1442 data address register.

Any N code other than 000 or 100 causes a processor check.

Sense I/O

Mnemonic: SNS

Op Code	Q Byte			Operand Address
Y0	0101	0	N	

Operation: Two bytes of 1442 status information are stored in the field specified by the operand address. The field is addressed by its rightmost byte.

The Q byte contains the device address (always 0101 for the 1442), an M bit of 0, and an N code. The N code specifies the information to be stored as follows:

1. N code 001 – Special indicators for CE use.
2. N code 010 – Special indicators for CE use.
3. N code 011 – Status indicators.
4. N code 100 – 1442 data address register.

Any N code other than 001, 010, 011, or 100 causes a processor check.

Figure 14-5 gives the meaning of the status bits in the status bytes. The conditions that set the 1442 status bits are:

1. Read station jam: This bit is set when the read station operates improperly or when a card jams in the read station. A read station jam makes the 1442 not ready and lights the READ STA light and the check light. The read station jam bit is reset by the next start I/O instruction, system reset, nonprocess runout, or check reset.
2. Hopper misfeed: This bit is set when a card fails to feed properly from the hopper. Hopper misfeed makes the 1442 not ready and lights the HOPR light and the check light. The hopper misfeed bit is reset by the next start I/O instruction, system reset, nonprocess runout, or check reset.
3. Extra feed cycle: This bit is set when the 1442 takes an unrequested feed cycle. The extra feed cycle makes the 1442 not ready and lights the FEED CLU light and the check light. The extra feed cycle bit is reset by the next start I/O instruction, system reset, nonprocess runout, or check reset.
4. Punch station jam: This bit is set when a card jams in the punch station. A punch station jam makes the 1442 not ready and lights the PUNCH STA light and the check light. The punch station jam bit is reset by the next start I/O instruction, system reset, nonprocess runout, or check reset.
5. Transport jam: This bit is set when a card jams in the stacker transport area. A transport jam makes the 1442 not ready and lights the TRANS light and the check light. The transport jam bit is reset by the

next start I/O instruction, system reset, nonprocess runout, or check reset.

6. Read check: This bit is set if data is read from the card incorrectly. This check also sets the check condition that can be tested by the test I/O and branch instruction. A read check lights the READ REG light and the check light. The read check bit is reset by the next start I/O instruction, system reset, nonprocess runout, or check reset.
7. Last card: This bit is set if the 1442 is restarted with no cards in the hopper. If the bit is on, the 1442 becomes not ready. Only a feed, punch and feed, or punch with no feed command should be issued when this bit is on. A punch with no feed command will not set the last card bit off, so that command should be followed by a feed command.
8. Punch check: This bit is set if the correct punches for the specified data are not selected. This check also sets the check condition that can be tested by the test I/O and branch instruction. A punch check lights the PUNCH light and the check light. The punch check bit is reset by the next start I/O instruction, system reset, nonprocess runout, or check reset.
9. Data overrun: This bit is set if data was lost because the processing unit was unable to accept data from the 1442 or send data to the 1442 fast enough. Data overrun lights the OVER RUN light and the check light. The data overrun bit is reset by the next start I/O instruction, system reset, nonprocess runout, or check reset.
10. Not ready: This bit is set if the 1442 is not ready because the hopper is empty, the stacker is full, the chip box is full or missing, the covers are open, or the 1442 stop key has been pressed. This check also sets the check condition that can be tested by the test I/O and branch instruction. A not ready condition lights the I/O attention light. If the chip box is full or missing, the chip box light is also on. The not ready condition is corrected and the start key is pressed.
11. No-op: This bit is set when the 1442 is issued a command it is unable to execute. This check also sets the check condition that can be tested by the test I/O and branch instruction. The no-op bit is set off by a sense I/O instruction.
12. Feed check: This bit is set by any improper card movement in the feed and transport sections of the 1442. This check also sets the check condition that can be tested by the test I/O and branch instruction. A feed check makes the 1442 not ready and lights the check light and the appropriate error indicator. The feed check bit is reset by correcting the error indicated by the error indicator lights.
13. Read invalid: This bit is set if multiple punches were found in rows 1-7 in any card column. This could be caused if the card was upside down or backwards. A read invalid lights the READ REG light and the check light. The read invalid bit is reset by the next start I/O instruction, system reset, nonprocess runout, or check reset.

Bit	Status Byte 2	Status Byte 1
0	Not Used	Read Check
1	Not Used	Last Card
2	Not Used	Punch Check
3	Read Station Jam	Data Overrun
4	Hopper Misfeed	Not Ready
5	Extra Feed Cycle	No-Op
6	Punch Station Jam	Feed Check
7	Transport Jam	Read Invalid

Figure 14-5. 1442 Status Bytes

Start I/O

Mnemonic: SIO

Op Code	Q Byte			Control Code
F3	0101	0	N	

Operation: The start I/O instruction is used to initiate each 1442 operation. If the 1442 is busy for that instruction or is not ready for any reason except unit check, the program loops on the start I/O instruction until the 1442 is not busy or is made ready. If the start I/O instruction is issued when the 1442 is not ready, the I/O attention light on the system control panel lights. When the not ready condition is corrected, the instruction is executed. If the 1442 has a feed check when the start I/O instruction is issued, the instruction is ignored and the no-op status bit is set. (Status bits are discussed under *Sense I/O*.) If the dual programming feature is used, a start I/O instruction issued to a 1442 that is busy or not ready causes an automatic program level advance.

The Q byte contains the device address (always 0101 for the 1442), an M bit of 0, and an N code. The N code specifies the operations the 1442 can perform as follows:

1. N code 000 – Feed
2. N code 001 – Read only
3. N code 010 – Punch and feed
4. N code 011 – Read column binary
5. N code 100 – Punch with no feed

Any N code other than 000, 001, 010, 011, or 100 causes a processor check.

The third byte in the instruction is a control code that controls stacker selection. Bits 0-4 in this byte are not used and should be set to zero. Bits 5-7 indicate the stacker selected:

1. 000 – Stacker 1
2. 001 – Stacker 2

Program Note: If a 1442 check prevents execution of the start I/O instruction, the instruction is ignored and a no-op status bit is set in the device attachment. If a check does not prevent execution of the instruction, the instruction is executed and the check is reset. Feed checks or not ready conditions cause no-ops.

READ OPERATIONS

A load I/O instruction must be executed before each start I/O instruction that specifies card reading. This load I/O instruction must load the address of the high-order byte of the read data field into the 1442 data address register. To meet performance specifications, the address for a normal read must be on a 128-byte boundary; the address for a read column binary must be on a 256-byte boundary.

The read/feed functions of start I/O instructions move cards from the hopper through the read station. If read is specified, the data contained in all 80 columns of the card is transferred to a storage field (1442 data field) specified by a load I/O instruction. The data read is checked to ensure that it is read correctly. An error in reading causes a read check.

The card feeding and reading rate is determined by the operations being performed. The rated reading speeds (300 cards per minute for Model 6 and 400 cards per minute for Model 7) are for read operations only. If punching is performed at the same time, the reading rate is reduced to the rate at which punching is performed. To maintain the rated reading rate, successive start I/O instructions specifying reading must be issued within 40 milliseconds (Model 6) or 30 milliseconds (Model 7) after the preceding card is read.

To test for a busy condition, use a test I/O and branch instruction.

PUNCH OPERATIONS

A load I/O instruction must be executed before each start I/O instruction that specifies a punch operation. This load I/O instruction places the address of the high-order byte of the punch data field in the 1442 data address register. Column 1 of the card is punched with the data contained in storage at this address; column 2 is punched with the data contained in storage at the next higher address. The punch data fields must be on 128-byte boundaries.

In addition to loading the 1442 data address register, a load I/O instruction must be issued to load the length count register with 128 minus the number of columns to be punched.

Start I/O instructions that specify punching move a card from the read station to the punch station. If a punch and feed command is issued, the card is punched and ejected into one of the stackers. If a punch with no feed command is issued, the card is punched but is not ejected.

As the cards pass through the punch station, data from storage is recorded in the cards in the form of punched holes. The punching is checked to ensure that the data is punched correctly. An error causes a punch check.

Card punching is performed at a rate of 80 columns per second (Model 6) or 160 columns per second (Model 7). To maintain the best card throughput, confine punching to the beginning card columns.

COMBINED OPERATIONS

Through proper sequencing of start I/O instructions, a card can be read and punched during one pass through the 1442.

STACKER SELECTION

Stacker selection is done by including the stacker select information in the start I/O instruction control code. Stacker selection is performed on the card that is in the punch station when the start I/O instruction is executed. If no stacker select information is given, the cards are automatically routed to stacker 1.

An IBM 3881 Optical Mark Reader Model 1 can be attached to the serial input/output channel to provide direct data entry from mark read data forms to System/3 Model 10. The *IBM 3881 Optical Mark Reader Models 1 and 2 Reference Manual and Operator's Guide,* GA21-9143, describes the 3881, its internal microprocessor, and some of its applications. It also explains the theory and use of mark read data input forms, describing how to adapt the microprocessor's built-in error detection logic and mark translation ability to the exact requirements of each data field being read. It gives complete details about keys, lights, error indications, and operating procedures, and contains forms specifications and marking recommendations.

OUTPUT RECORD

The 3881 assembles an output record that contains a segment descriptor word, record descriptor word, normal mark data bytes, BCD data bytes (if any), and serial number data (if any) in the format shown in Figures 15-1 and 15-2. The output record moves to the system under system control.

Segment Descriptor Word: The first two bytes indicate (in binary code) the number of bytes in the record. The last two bytes contain zeros.

Record Descriptor Word: Contains two bytes of status information that indicate whether the form was selected by the 3881, whether the record contains a serial number, etc. (see Figure 15-2).

Normal Mark Data Field: Contains all the bytes of data from regular mark data fields on the form. The type of data in each byte (digit, letter, or image format bit), the sequence in which each field is loaded into the normal data field of the record, and the number of data bytes in each field is controlled by a microprocessor in the 3881.

BCD Data Field: This field is in the output record only if the BCD special feature is installed and the 3881 has been controlled (via a format control sheet) to read BCD data from that form. The field contains one EBCDIC byte for each BCD-encoded digit read. BCD fields are loaded into the output record BCD area (at the end of the normal mark data field, as shown in Figure 15-1) in the sequence they were read from the form.

Serial Number Field: This field contains a seven digit decimal number for batch and/or serial numbering identical to the number printed on the input form while it passed through the 3881 transport. If the serial numbering device (special feature) is not installed on the 3881, or if it is installed and is not active when the form is read, the 3881 does not place a serial number field in the output record. If the device is installed and active for one type of form only (those directed by the 3881 to the select stacker, for example), the 3881 loads blanks into the serial number field for all unnumbered forms. The decision to number or not to number a particular form is a function of 3881 logic only, and is not affected by CPU programming control over form selection.

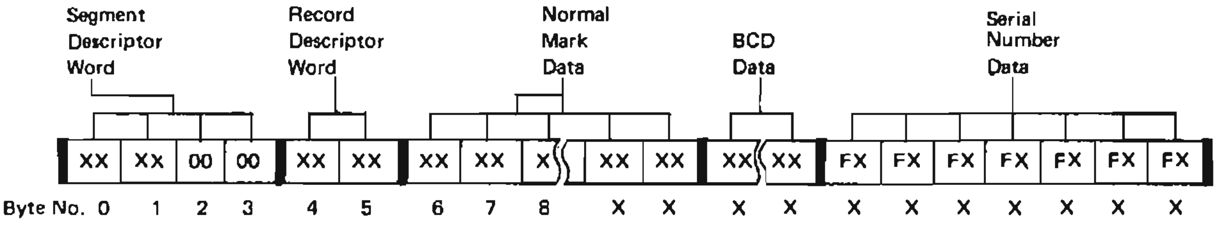

Segment Descriptor Word
Record Descriptor Word
Normal Mark Data
BCD Data
Serial Number Data
XX XX 00 00 XX XX XX XX X XX XX XX XX FX FX FX FX FX FX FX
Byte No. 0 1 2 3 4 5 6 7 8 X X X X X X X X X X X

Figure 15-1. Format of Output Record

BYTE 4 of Output Record (First Byte of Descriptor Word)

Value in Byte 4		Conditions Indicated by Hexadecimal Value			
Hex	EBCDIC	Serial Numbering Feature Used	3881 Sent Form to Select Stacker	Reject Characters on Form	BCD Data Length Error
C8	H	No	No	No	No
C1	A	No	No	No	Yes
C2	B	No	No	Yes	No
C3	C	No	No	Yes	Yes
C4	D	No	Yes	No	No
C5	E	No	Yes	No	Yes
C6	F	No	Yes	Yes	No
C7	G	No	Yes	Yes	Yes
F0	0	Yes	No	No	No
F1	1	Yes	No	No	Yes
F2	2	Yes	No	Yes	No
F3	3	Yes	No	Yes	Yes
F4	4	Yes	Yes	No	No
F5	5	Yes	Yes	No	Yes
F6	6	Yes	Yes	Yes	No
F7	7	Yes	Yes	Yes	Yes

BYTE 5 of Output Record (Second Byte of Descriptor Word)

Value in Byte 5		Format Type of Form
Hex	EBCDIC	
F0	0	Basic
F1	1	Alternate 1
F2	2	Alternate 2
F3	3	Alternate 3
F4	4	Alternate 4
F5	5	Alternate 5

Figure 15-2. Record Descriptor Word

CODE REPRESENTING INVALID COMBINATION OF MARKS

Whenever the 3881 recognizes an invalid combination of marks on a form, it inserts a special code (instead of the invalid data) in the output record. For this code, the customer can specify either the standard hexadecimal 3F or (as a no-cost option) hexadecimal 7C, which is a printable graphic. Hexadecimal 3F is not a printable EBCDIC graphic character, whereas hexadecimal 7C causes the @ character to print from most IBM print chains.

I/O ATTENTION LIGHT ON 5410

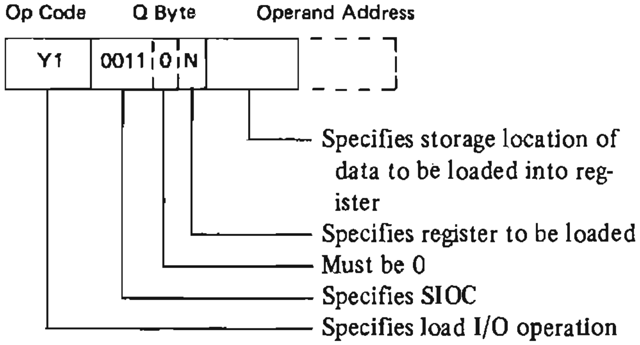
The 3881 does not make use of the I/O attention light on the 5410.

INSTRUCTIONS

The IBM 3881 operates through the instructions issued to the SIOC. The exact form of these instructions is discussed in the Chapter 9 of this manual.

Load I/O

Mnemonic: LIO


Op Code
Q Byte
Operand Address
Y1
0011
0
N
Specifies storage location of data to be loaded into register
Specifies register to be loaded
Must be 0
Specifies SIOC
Specifies load I/O operation

Operation: The two bytes contained in the two-byte field addressed by the operand address are placed in the register designated by the Q byte. The operand is addressed by the low-order byte.

N Code	*Desitination*
000	Invalid.
001	I/O function register.
010	SIOC length count register.
011	Invalid.
100	SIOC data address register.
101	Data transfer register.
110	Invalid.
111	Invalid.

Specification of an invalid N code results in a processor check stop with an invalid Q byte indication.

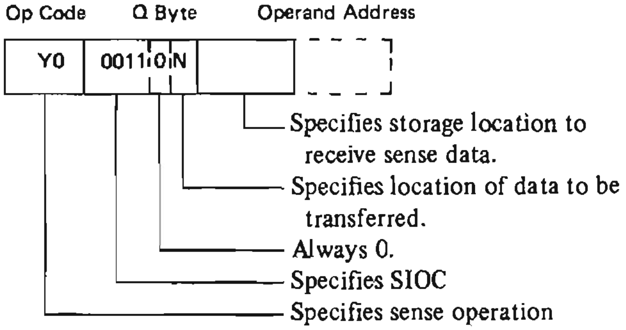
The I/O function register must be loaded with the following:

High-Order Byte	*Low-Order Byte*
01000000	00000010

The number that is placed in the length count register is the binary representation of a number equal to 256 minus the number of bytes to be transferred, unless the program is to test for the output record ready status bit off at the end of each read operation. In that case, the number placed in the register is hexadecimal 00.

Sense I/O

Mnemonic: SNS


Op Code
Q Byte
Operand Address
Y0
0011
0
N
Specifies storage location to receive sense data.
Specifies location of data to be transferred.
Always 0.
Specifies SIOC
Specifies sense operation

Operation: The two bytes specified by the Q byte are placed in the two-byte field addressed by the operand address. The operand is addressed by the low-order-byte.

The Q byte comprises a device address (always 0011) in the high-order four bits, an M bit of 0, and an N code. The N code specifies the sense bytes or registers that are to be sensed.

N Code	*Senses*
000	Invalid.
001	I/O function register.
010	Length count register and status byte.
011	I/O transfer lines and I/O indentification
100	Data address register.
101	Data transfer register and diagnostic byte.
110	Invalid.
111	Invalid.

Specification of an invalid N code causes a processor check stop with an invalid Q byte indication.

The status byte and diagnostic byte are stored as the high-order bytes of their respective sense operations. They are bit-significant as follows:

Status Byte

Bit	*Meaning*
0	Spare.
1	End request.
2	Interrupt pending.
3	Not used by 3881.
4	Data transfer register parity check.
5	No-op.
6	Length count register overflow.
7	I/O ready.

The I/O transfer lines are bit significant as follows:

High-Order Byte

Bit	*Meaning*
0	Will be 0*
1	Will be 0*
2	Will be 1*
3	Will be 0*
4	Device is attached
5	Not used
6	Not used
7	Not used

*If device attached is an IBM 3881.

Low-Order Byte

Bit	*Meaning*
0	Not used
1	3881 ready
2	Equipment check
3	Output record ready
4	Diagnostic not ready
5	Not used
6	Not used
7	End of file

Equipment Check: Indicates that a read head or microprocessor failure has been detected in the 3881. If equipment check is presented as a result of a read command, any data transferred during that command execution should be ignored.

Output Record Ready: The 3881 has read all the marks from a form passing under the read heads, has translated the marks into EBCDIC data bytes or mark image bytes, and has stored them (along with special status bytes that relate to the data just read) in the 3881 read (output) buffer. This bit is reset when the last byte has been transmitted to the system by a read command. The program should test this bit after each read command has been executed to determine whether or not another read command must be issued to read more data from the buffer to the system.

Additional Status Information: The record descriptor word (which is part of each 3881 output record) contains two bytes of status information that indicate whether the form was selected by the 3881, whether the record contains a serial number, etc. For details about the 3881 output record and the record descriptor word, see *IBM 3881 Optical Mark Reader Models 1 and 2.*

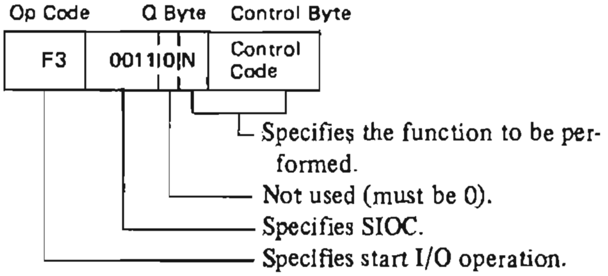
Reference Manual and Operator's Guide GA21-9143.

Test I/O and Advance Program Level

These instructions operate on the SIOC even though they may be used when operating the 3881, see Chapter 9 for a discussion of these instructions.

Start I/O

Mnemonic: SIO



Operation: The reader performs the operation specified by the N code and the control code:

N Code	*Control Code*	*Operation*
000 or 001	00000001	Reset interrupt request (performed by the SIOC).
000 or 001	00000010	Enable interrupt (performed by the SIOC).
000 or 001	00000100	Disable interrupt (performed by the SIOC).
000 or 001	00001000	Reset SIOC adapter, removing SIOC from busy state (performed by the SIOC).
000 or 001	00010000	Set interrupt request.
001	00000000	Read only.
010	00000000	Invalid for 3881.
011	00000001	Feed and select normal stacker.*
011	00000010	Feed and select select stacker.
011	00000100	Enable reread.

*The 3881 microprocessor can override this selection, sending the form to the select stacker because it detected a 3881 format-control-sheet-controlled forms selection condition. The stored program can examine byte 4 of the output record to determine whether or not the 3881 selected the form.

3881 OPERATIONS

Operator action during initial 3881 setup procedures causes a forms feed cycle. The first data form moves past the read heads, is read by the 3881, and stops at the wait station awaiting a stacker selection decision. As the form passes the read head, the 3881 microprocessor converts the marks into meaningful data bytes and formats them into an output record in the 3881 read buffer. At the same time, the 3881 error detection and mark discrimination logic (using parameters stored in the 3881 microprocessor for the type of form being read) selects the select stacker for the form if it contains an error or questionable mark.

The microprogram builds record descriptor bytes for each form read. These are bytes 4 and 5 of the output record, and they indicate whether or not the 3881 logic directed the form to the select stacker. Record descriptor bytes never reflect programmed stacker selection.

After the first form has been fed and read by the 3881, the 3881 enters a ready status. The program must now issue one or more read commands to move the data from the 3881 read buffer to the CPU. (Each System/3 read command moves a maximum of 256 bytes from the 3881 buffer.)

An SIO specifying a read operation places the SIOC in the read mode; this permits the serially-by-byte transfer of data from the 3881 read buffer to the CPU. Data transfer starts immediately if the entire output record has been formatted in the 3881 read buffer (signalled by the 3881 setting the output record ready bit). If the command is issued after a feed command has been issued but before the output record has been completely loaded into the read buffer, the 3881 accepts the command, and data transfer starts when the output record ready bit comes on.

If the command is issued after the end of data transfer has occurred for one record and before a feed command has been issued to load the buffer with data from the next form, the SIOC returns a busy status indication, and the 3881 ignores the command. To recover, issue an SIO with an N code of 000 and a control code with bit 3 on; this will turn the busy bit off.

The CPU stores the first byte of data transferred in the storage location specified by the SIOC data address register, storing subsequent bytes in successively higher storage locations. Data transfer continues until the read operation is stopped by (1) a length count register overflow or (2) an end of data transfer (EOT) condition, which indicates that the last byte has been moved from the 3881 buffer to the system.

If more than 256 bytes are to be read from the buffer, the program must issue multiple read commands to move the entire output record from the buffer to the CPU.

If the programmer knows the exact number of bytes to be transferred for each record, he can specify a certain number of bytes to be moved per read command, and issue the number of read commands necessary to move all the data to the CPU. If the sum of the bytes specified by all the read commands for that output record equals the exact number of bytes in the output record, all the data will be transferred to the system.

Often, however, the programmer will not know the exact number of bytes in the output record to be transmitted to the system. In this case, he can load hexadecimal 00 into the length count register, and then test for an output-record-ready status bit at the end of the read operation. If the bit is on, there is still data to be transferred to the CPU, so he must issue another read command with 00 in the length count register. All the data will have been transmitted to the system when the program detects the output-record-ready status bit off.

The program may need to retransmit data from the 3881 read buffer to the CPU before that data is destroyed by reading new data into the buffer with a feed instruction. An SIO command specifying the reread enable function can be issued any time after the output record ready bit is turned on and before the next feed command is issued. The reread enable command allows the program to retransmit data from the buffer, starting with the first byte of the output record and performing the identical read functions described earlier.

The feed and select stacker command must now be issued to move a new form past the 3881 read station and place a new output record in the 3881 buffer. This feed command forces the form whose data was last transferred to the CPU to move to a stacker. Unless the 3881 has specified the select stacker when the program has specified the normal stacker for the same form, the form enters the stacker specified by the System/3 feed and select stacker command. Each form stops at the wait station awaiting stacker selection decision if the feed and select stacker command is not issued before the form reaches the wait station. Any subsequent feed and select stacker command issued without an intervening read command is ignored by the 3881. This ensures that no data from a form can be destroyed by data from the next form until it has been transferred to the system.

IBM 3741 Data Station Models 1 and 2
IBM 3741 Programmable Work Station Models 3 and 4

Either an IBM 3741 Data Station Model 1 or 2 or an IBM 3741 Programmable Work Station Model 3 or 4 can be directly attached to the IBM System/3. The attached 3741 can be used online to System/3 as a diskette input/output device or offline to perform 3741 functions such as data entry and communications, and can be used (Models 3 and 4 only) as a programmable work station. An IBM System/3, using IBM programs, supports the 3741 Models 3 and 4 in data station mode only; the System/3 does not support the Application Control Language for Models 3 and 4.

Data Transfer Rate

The 3741 directly attached to System/3 provides input/output rates of about 1500 records per minute when reading from the diskette, and about 1000 records per minute when writing to the diskette. These rates depend on the complexity of the application and the following assumptions:

- Records are transferred between the 3741 and a disk device (5444 or 5445).
- Diskette records contain 128 bytes, and the 3741 data is double buffered.
- The disk uses 1028-byte blocks (eight diskette records per disk block), and the data is double buffered.
- The system is dedicated (without spooling, multiprogramming, or dual programming).
- The operation is error-free, with no alternate tracks assigned on the disk or diskette.

Maximum Diskette Record Size

A diskette record may not contain more than 128 bytes.

Attachment to System

The 3741 is attached, via a signal cable, to an attachment feature in the processing unit. I/O control is similar to that of an I/O device attached to the serial input/output channel on the system.

Power

The 3741 has a power cord that receives power from a receptacle in the room. (The 3741 attachment feature is powered from the System/3.) Therefore, supplying power to the system does not supply power to the 3741. Also, there is no emergency power-off between the processing unit and the 3741.

Online Selection

A 3741 attached to the system can operate in either online mode or offline mode. In online mode, the 3741 keyboard is inoperative, except to take the 3741 offline. Data transfer between the 3741 and the system is always between the system and the diskette, never between the system and the keyboard/display screen.

Before the 3741 can be used as a system I/O device, the operator must bring up 3741 power, load a diskette into the 3741, and place the 3741 in online mode.

The system must check to determine that the 3741 is online before performing any 3741 I/O operations. If the operator has placed the 3741 offline, the program must detect this condition and halt.

REGISTERS AND PROGRAM-TESTABLE LINES

The following registers and testable lines are used to program 3741 I/O operations.

Data Transfer Register

A 9-bit data transfer register temporarily stores 1 byte of data (8 bits plus a parity bit) that is to be moved in either direction between the diskette and main storage. Data transfer usually occurs on a cycle steal basis, but the contents of this register can be moved between the register and main storage with load I/O and sense I/O instructions. The system tests this register for correct parity, setting a sense bit if incorrect parity is encountered.

Length Count Register

Because data transfer occurs on a cycle steal basis, the 3741 attachment must keep track of the number of bytes transferred. A length count register performs this function. The 3741 length count register limits the number of bytes to be transferred to the number loaded by the program.

This register must be loaded before each I/O operation, using a load I/O instruction. The number to be loaded into the register depends on the number of bytes to be transferred. The number should be 255 minus the number of bytes to be transferred. For example, if you wish to transfer 128 bytes (the maximum length 3741 record), load hexadecimal 7F into the length count register. The 3741 signals when the last byte has been transferred from the diskette to the system.

Upon successful completion of a record transfer operation, the length count register should contain hexadecimal FF or a lower number. If the program loads a record length less than the physical record length, the system indicates an overflow and places the 3741 in offline mode.

If the program loads a record length greater than the physical record length, the length count register does not read hexadecimal FF at the end of data transfer.

Note: IBM programming support loads 255 minus the length of the record into the length count register. At the end of data transfer, if overflow did not occur and if the register contains hexadecimal FF, the program assumes that the correct number of bytes has been transferred.

The contents of the length count register and the overflow condition can be placed in storage for testing with a sense I/O instruction.

I/O Transfer Lines

Although this set of signal lines from the 3741 to the attachment is not a register, the lines can be tested by a sense I/O instruction. The lines, which provide information about 3741 operations, status, and identification, can be used by the System/3 program for program decisions.

I/O transfer lines, their associated testable bits, and their meanings, are:

I/O Transfer Line	*Associated Byte and Bit**	*Meaning*
1	Byte 1, bit 7	Not used
2	Byte 1, bit 6	Not used
3	Byte 1, bit 5	3741 attention required
4	Byte 1, bit 4	End-of-job out (3741 indicates end of job)
5	Byte 1, bit 3	End-of-record out (3741 indicates end of data transfer for a System/3 read instruction)
6	Byte 1, bit 2	Bus-in parity error (attachment detected a parity error in data received from System/3)
7	Byte 1, bit 1	End of data set out (3741 indicates end of data set)
8	Byte 1, bit 0	Not used
9	Byte 2, bit 7	Read from attachment (3741 requests a System/3 write operation)
10	Byte 2, bit 6	Write to attachment (3741 requests a System/3 read operation)
11	Byte 2, bit 5	3741 online
12	Byte 2, bit 4	I/O cable attached
13	Byte 2, bit 3	I/O identification 1-bit (will be 0)
14	Byte 2, bit 2	I/O identification 2-bit (will be 0)
15	Byte 2, bit 1	I/O identification 4-bit (will be 1)
16	Byte 2, bit 0	I/O identification 8-bit (will be 0)

*Byte 1 is the low-order (rightmost, high address) byte; byte 2 is the high-order (leftmost, low address) byte.

Function Register

The program must load hexadecimal 4000 into the function register at the start of each job that uses the 3741. To load the function register, use the load I/O instruction.

Data Station Address Register (DSAR)

This local storage register stores the address of the data field that is to be used for the next 3741 read or write operation. The register must be loaded before each read or write operation. The register can be sensed with a sense instruction.

3741 OPERATIONS

The 3741 reads data to the system from one sector of the diskette or writes data into one sector of the diskette under control of start I/O read or write instructions. Data is transferred sequentially, one byte at a time, in EBCDIC.

At the start of the job, the operator must manually place the 3741 in proper mode and place the 3741 online. Thereafter, the 3741 operates under System/3 program control.

Establishing Synchronization

The program must ensure that the 3741 is synchronized with the system before issuing each control SIO instruction. The following procedure can be used:

1. Sense the 3741 status byte for a processor program interrupt pending condition.

2. Reset the interrupt pending latch by issuing an SIO instruction.

Initial Adapter Setup

The adapter must be set up once per job. Use the following procedure:

1. Reset the adapter.

2. Load the attachment function register with hexadecimal 4000. (This sets the attachment service response signal to drop six microseconds after the drop of the service request signal.)

Data Transfer

Use the following procedure to transfer data to and from the 3741:

1. Test with a SNS instruction to ensure that the 3741 is attached and online.

2. Test with a TIO instruction to ensure that the 3741 is ready and not busy.

3. Wait for the 3741 to bring up either the read or write command line.

4. Respond to the 3741 request with an appropriate SIO instruction.

5. Load the address of the first byte of data into the attachment data address register.

6. Load 255 minus the number of bytes to be transferred into the length count register.

7. Issue a read or write SIO instruction.

8. Proceed to the error-checking operation when a test for attachment busy fails.

Error Checking

The program should check for errors after each read or write SIO instruction has been executed. You can use the following routine:

1. Sense the I/O transfer lines, the length count register, and the attachment status byte before issuing any subsequent SIO instruction.

2. Check for the following conditions:
   a. 3741 attached and online. (If the 3741 goes offline, no op-end interrupt is generated.)
   b. Bus-in parity error (3741 detected).
   c. Bus-out parity error (attachment detected).
   d. 3741 attention required.
   e. Length error. To determine this, check for length count register residual of hexadecimal FF. If the residual is not FF, a length error has occurred.
   f. No-op. If the last SIO instruction set the no-op bit, re-issue the instruction, if possible.

End-of-Record Processing

After each System/3 read operation, the program should perform the following routine:

1. Check for end-of-job indication from 3741 and halt if it is the end of the job.

2. Check for end-of-data indication from 3741. If there is an end-of-data indication, the system should process appropriately, then return to the data transfer operation.

At the end of data or end of job during a System/3 write operation, the program should:

1. Send end of data to the 3741 for end-of-data condition, then return to the data transfer operation, or

2. Send end of job to the 3741 for end-of-job condition.

INSTRUCTIONS

Start I/O

Mnemonic: SIO

Op Code	Q Byte			Control Byte
F3	0100	0	N	Control Code

Operation: The 3741 performs the function specified by the N code and the control code.

The DA is always 0100 (hex 4), and the M bit is always binary 0 for the 3741.

The N code of the Q byte and the control byte specify the operation to be performed:

N Code	*Control Code*	*Operation*
000*	0000 0001	Reset interrupt request (performed by 3741 adapter)
	0000 0010	Enable interrupt request (performed by 3741 adapter)
	0000 0100	Disable interrupt request (performed by 3741 adapter)
	0000 1000	Reset 3741 adapter, removing 3741 from busy state (performed by 3741 adapter)
	0001 0000	Set interrupt request
001	0000 0000	Read
010	0000 0000	Write
011	0000 1000	Indicate normal response to 3741
	0001 0000	Indicate record-length error to 3741
	0001 0100	Indicate attachment (mode) error to 3741
	0011 0000	Indicate end of data set to 3741
	0101 0000	Indicate end of job to 3741
	1001 0000	Indicate bus-out parity error to 3741

*N code 000 specifies processor program interrupt control only. The interrupt control function can also be programmed with read and write instructions (N codes of 001 and 010).

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

Test I/O and Branch

Mnemonic: TIO

Op Code	Q Byte			Branch-to Address
Z1	0100	0	N	

Operation: The CPU tests for the condition specified by the N code. If the condition is present, the CPU places the address from the instruction address register into the address recall register, then places the operand address from the instruction into the instruction address register, and then accesses that instruction. If the condition tested is not present, the CPU places the operand address from the instruction in the address recall register and executes the instruction at the address specified by the instruction address register (the next sequential address).

The DA portion of the Q byte is always hex 4; the M bit is always 0. The N code conditions that can be tested are:

N Code	*Condition*
000	Attachment not ready/check
010	Attachment busy

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

Program Notes:

1. The attachment-not-ready/check condition indicates that one or more of these conditions exist:
   a. No I/O device is attached to the 3741 attachment.
   b. The data transfer register has incorrect parity.
   c. A read or write SIO instruction addressed to the 3741 has been ignored (no-oped).
   d. The 3741 is busy or not ready.

2. The attachment-busy condition indicates that the 3741 attachment is performing an operation.

Advance Program Level

Mnemonic: APL

Op Code	Q Byte			
F1	0100	0	N	Not Used

Operation: This instruction tests the 3741 attachment for the condition specified by the N code. If a tested condition exists and the system is not equipped with the dual programming feature, the program loops on the APL instruction until the condition no longer exists, then advances to the next sequential instruction. If a tested condition exists and the system is equipped with the dual programming feature, the sytem activates the inactive program level. If a tested condition does not exist, systems with and without the dual programming feature take the next sequential instruction in the active program level.

The device address for the 3741 is always hex 4, and the M bit is always 0.

The N codes used to specify conditions to be tested are:

N Code	*Condition*
000	3741 attachment not ready/check
010	3741 attachment busy

Any N code not shown causes the processing unit to stop with a processor check and an invalid Q byte indication.

Program Notes:

1. The attachment not ready/check condition indicates that one or more of the following conditions exist:
   a. No I/O device is attached to the 3741 attachment.
   b. The data transfer register has incorrect parity.
   c. A read or write SIO instruction addressed to the 3741 has been ignored (no-oped).
   d. The 3741 is busy or not ready.

2. The 3741 attachment busy condition indicates that the 3741 attachment is performing an operation.

Load I/O

Mnemonic: LIO

Op Code	Q Byte			Operand Address	
Y1	0100	0	N		

Operation: The processing unit loads the two bytes of data from the field specified by the operand address into the register specified by the N code. The operand is addressed by its low-order (higher numbered) storage position.

The device address (bits 0 through 3 of the Q byte) of binary 0100 (hex 4) specifies that 3741 registers will be loaded. The M bit is always 0. The register to be loaded is specified by one of the N codes shown below:

N Code	*Register*
001	I/O function register (you must load hex 4000 into the I/O function register)
010	3741 length count register (load 255 minus the number of bytes to be transferred; load hex value)
100	3741 data address register
101	Data transfer register

Any N code not shown causes the processing unit to stop with a processor check and an invalid Q byte indication.

Sense I/O

Mnemonic: SNS

Op Code	Q Byte			Operand Address	
Y0	0100	0	N		

Operation: The CPU moves data from the source specified by the N code to the two-byte field specified by the operand address.

The device address for this instruction is always 0100 (hex 4), and the M bit is always 0. The N codes for the instruction and their meanings are shown below:

N Code	*Data Source*
001	I/O function register
010	Length count register and status byte
011	I/O transfer lines and I/O identification. These lines from the 3741 are bit significant as follows:

Low-Order Byte (operand address): Byte 1

Bit	*Meaning*
0	Not used
1	End of data
2	Bus-in parity
3	End of record
4	End of job
5	3741 attention required
6	Not used
7	Not used

High-Order Byte (operand address minus 1): Byte 2

Bit	*Meaning*	
0	Must be 0	hex 4
1	Must be 1	
2	Must be 0	
3	Must be 0	
4	3741 attached	
5	3741 online	
6	Write to attachment	
7	Read from attachment	

100	Data address register
101	Data transfer register* and diagnostic byte

*This byte is stored at the operand address. The associated byte is stored at the operand address minus 1.

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

Program Notes:

1. The operand is always addressed by the low-order (higher storage number) byte.

2. This instruction is executed even though the 3741 attachment is busy or has a not-ready/check condition.

3. The diagnostic byte is for CE diagnostics and has no meaning to the I/O control program.

4. Figure 16-1 shows the 3741 status bits and their meanings.

Byte	Bit	Name	Indicates	Reset By
2	0	CE diagnostic	Used for CE diagnostic program.	CE action
	1	CE diagnostic	Used for CE diagnostic program.	CE action
	2	Interrupt pending	The 3741 requires program action.	Next SIO issued by program
	3	Not used	Not used.	Not used
	4	Data transfer register parity error	The 3741 attachment detected a parity error in at least one byte of data passing through the data transfer register for transmission to the read data field in main storage.	Next system reset, check reset, or SIO operation
	5	No-op	The last instruction issued to the 3741 was rejected because the 3741 is not capable of performing the operation specified by the instruction.	Sense instruction
	6	Length count register overflow	The operation tried to transfer more data than the number of bytes specified by the length count register.	Next LIO that reloads the length count register
	7	I/O ready	The 3741 is ready and is not busy.	The 3741 becoming busy or not ready
1	All	This byte contains the current contents of the length count register when the register was sensed.		

Figure 16-1. 3741 Status Byte

Appendix

Instruction Formats

Zero and Add (ZAZ)

Op Code	Q Byte		Operand Addresses (2 to 4 Bytes)			
X4	L1	L2				

Add Zoned Decimal (AZ)

Op Code	Q Byte		Operand Addresses (2 to 4 Bytes)			
X6	L1	L2				

Subtract Zoned Decimal (SZ)

Op Code	Q Byte		Operand Addresses (2 to 4 Bytes)			
X7	L1	L2				

Add Logical Characters (ALC)

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

Subtract Logical Characters (SLC)

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

Add to Register (A)

Op Code	Q Byte	Operand Address	
Y6	Q		

Move Hex Character (MVX)

Op Code	Q Byte	Operand Addresses (2 to 4 Bytes)			
X8	Q				

Move Characters (MVC)

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

Edit (ED)

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

Insert and Test Characters (ITC)

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

Move Logical Immediate (MVI)

Op Code	Q Byte	Operand Address	
YC	IO		

Set Bits On Masked (SBN)

Op Code	Q Byte	Operand Address	
YA	M		

Set Bits Off Masked (SBF)

Op Code	Q Byte	Operand Address	
YB	M		

Store Register (ST)

Op Code	Q Byte	Operand Address	
Y4	Q		

Load Register (L)

Op Code	Q Byte	Operand Address	
Y5	Q		

Load Address (LA)

Op Code	Q Byte	Operand	
Z2	Q		

Compare Logical Characters (CLC)

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

Compare Logical Immediate (CLI)

Op Code	Q Byte	Operand Address	
YD	IO		

Test Bits On Masked (TBN)

Op Code	Q Byte	Operand Address	
Y8	Mask		

Test Bits Off Masked (TBF)

Op Code	Q Byte	Operand Address	
Y9	Mask		

Branch On Condition (BC)

Op Code	Q Byte	Branch Address	
Z0	Q		

Jump On Condition (JC)

Op Code	Q Byte	Control Code
F2	Q	

Halt Program Level (HPL)

Op Code	Halt Identifier	
F0	Tens Code	Units Code

Start I/O (SIO)

Op Code	Q Byte	Control Code
F3	DA M N	

Sense I/O (SNS)

Op Code	Q Byte	Operand Address	
Y0	DA M N		

Load I/O (LIO)

Op Code	Q Byte	Operand Address	
Y1	DA M N		

Test I/O and Branch (TIO)

Op Code	Q Byte	Branch-to Address	
Z1	DA M N		

Advance Program Level (APL)

Op Code	Q Byte	
F1	DA M N	Not Used

Code Conversions

Dec	Hex	Card Code	Mnem	IPL*		EBCDIC Code	EBCDIC	ASCII	ASCII	System/3
Val	Val	DCBA8421		T1T3	T2T3	01234567	Character	Code	Character	Symbol
000	00	C		4	1	00000000	NUL	0000000	NUL	
001	01	DCBA 1		A @	A 3	00000001	SOH	0000001	SOH	
002	02	DCBA 2		B @	B 3	00000010	STX	0000010	STX	
003	03	DCBA 21		C @	C 3	00000011	ETX	0000011	ETX	
004	04	DCBA 4	ZAZ	D @	D 3	00000100	PF	0000100	EOT	
005	05	DCBA 4 1		E @	E 3	00000101	HT	0000101	ENQ	
006	06	DCBA 42	AZ	F @	F 3	00000110	LC	0000110	ACK	
007	07	DCBA 421	SZ	G @	G 3	00000111	DEL	0000111	BEL	
008	08	DCBA8	MVX	H @	H 3	00001000		0001000	BS	
009	09	DCBA8 1		I @	I 3	00001001	RLF	0001001	HT	
010	0A	CBA8 2	ED	¢ 4	¢ 1	00001010	SMM	0001010	LF	
011	0B	CBA8 21	ITC	. 4	. 1	00001011	VT	0001011	VT	
012	0C	CBA84	MVC	< 4	< 1	00001100	FF	0001100	FF	
013	0D	CBA84 1	CLC	(4	(1	00001101	CR	0001101	CR	
014	0E	CBA842	ALC	+ 4	+ 1	00001110	SO	0001110	SO	
015	0F	CBA8421	SLC	| 4	| 1	00001111	SI	0001111	SI	
016	10	C A8 2		& 4	& 1	00010000	DLE	0010000	DLE	
017	11	DCB 1		J @	J 3	00010001	DC1	0010001	DC1	
018	12	DCB 2		K @	K 3	00010010	DC2	0010010	DC2	
019	13	DCB 21		L @	L 3	00010011	DC3(TM)	0010011	DC3	
020	14	DCB 4	ZAZ	M @	M 3	00010100	RES	0010100	DC4	
021	15	DCB 4 1		N @	N 3	00010101	NL	0010101	NAK	
022	16	DCB 42	AZ	O @	O 3	00010110	BS	0010110	SYN	
023	17	DCB 421	SZ	P @	P 3	00010111	IL	0010111	ETB	
024	18	DCB 8	MVX	Q @	Q 3	00011000	CAN	0011000	CAN	
025	19	DCB 8 1		R @	R 3	00011001	EM	0011001	EM	
026	1A	CB 8 2	ED	! 4	! 1	00011010	CC	0011010	SUB	
027	1B	CB 8 21	ITC	$ 4	$ 1	00011011	CU1	0011011	ESC	
028	1C	CB 84	MVC	* 4	* 1	00011100	IFS	0011100	FS	
029	1D	CB 84 1	CLC	) 4	) 1	00011101	IGS	0011101	GS	
030	1E	CB 842	ALC	; 4	; 1	00011110	IRS	0011110	RS	
031	1F	CB 8421	SLC	¬ 4	¬ 1	00011111	IUS	0011111	US	
032	20	CB		– 4	– 1	00100000	DS	0100000	SPACE	
033	21	C A 1		/ 4	/ 1	00100001	SOS	0100001	!	
034	22	DC A 2		S @	S 3	00100010	FS	0100010	"	
035	23	DC A 21		T @	T 3	00100011		0100011	#	
036	24	DC A 4	ZAZ	U @	U 3	00100100	BYP	0100100	$	
037	25	DC A 4 1		V @	V 3	00100101	LF	0100101	%	
038	26	DC A 42	AZ	W @	W 3	00100110	ETB(EOB)	0100110	&	
039	27	DC A 421	SZ	X @	X 3	00100111	ESC(PRE)	0100111	'	
040	28	DC A8	MVX	Y @	Y 3	00101000		0101000	(	
041	29	DC A8 1		Z @	Z 3	00101001		0101001	)	
042	2A	DCBA	ED	} @	} 3	00101010	SM	0101010	*	
043	2B	C A8 21	ITC	, 4	, 1	00101011	CU2	0101011	+	
044	2C	C A84	MVC	% 4	% 1	00101100		0101100	,	
045	2D	C A84 1	CLC	_ 4	_ 1	00101101	ENQ	0101101	-	
046	2E	C A842	ALC	> 4	> 1	00101110	ACK	0101110	.	
047	2F	C A8421	SLC	? 4	? 1	00101111	BEL	0101111	/	

Dec	Hex	Card Code	Mnem	IPL*		EBCDIC Code	EBCDIC	ASCII Code	ASCII	System/3
Val	Val	DCBA8421		T1T3	T2T3	01234567	Character	7654321	Character	Symbol
048	30	DC A	SNS	0 @	0 3	00110000		0110000	0	
049	31	DC 1	LIO	1 @	1 3	00110001		0110001	1	
050	32	DC 2		2 @	2 3	00110010	SYN	0110010	2	
051	33	DC 21		3 @	3 3	00110011		0110011	3	
052	34	DC 4	ST	4 @	4 3	00110100	PN	0110100	4	
053	35	DC 4 1	L	5 @	5 3	00110101	RS	0110101	5	
054	36	DC 42	A	6 @	6 3	00110110	UC	0110110	6	
055	37	DC 421		7 @	7 3	00110111	EOT	0110111	7	
056	38	DC 8	TBN	8 @	8 3	00111000		0111000	8	
057	39	DC 8 1	TBF	9 @	9 3	00111001		0111001	9	
058	3A	C 8 2	SBN	: 4	: 1	00111010		0111010	:	
059	3B	C 8 21	SBF	# 4	# 1	00111011	CU3	0111011	;	
060	3C	C 84	MVI	@ 4	@ 1	00111100	DC4	0111100	<	
061	3D	C 84 1	CLI	' 4	' 1	00111101	NAK	0111101	=	
062	3E	C 842		= 4	= 1	00111110		0111110	>	
063	3F	C 8421		" 4	" 1	00111111	SUB	0111111	?	
064	40	None				01000000	SPACE	1000000	@	SPACE
065	41	D BA 1		A 8	A 2	01000001		1000001	A	
066	42	D BA 2		B 8	B 2	01000010		1000010	B	
067	43	D BA 21		C 8	C 2	01000011		1000011	C	
068	44	D BA 4	ZAZ	D 8	D 2	01000100		1000100	D	
069	45	D BA 4 1		E 8	E 2	01000101		1000101	E	
070	46	D BA 42	AZ	F 8	F 2	01000110		1000110	F	
071	47	D BA 421	SZ	G 8	G 2	01000111		1000111	G	
072	48	D BA8	MVX	H 8	H 2	01001000		1001000	H	
073	49	D BA8 1		I 8	I 2	01001001		1001001	I	
074	4A	BA8 2	ED	¢	¢	01001010	¢	1001010	J	¢
075	4B	BA8 21	ITC	.	.	01001011	.	1001011	K	.
076	4C	BA84	MVC	<	<	01001100	<	1001100	L	<
077	4D	BA84 1	CLC	(	(	01001101	(	1001101	M	(
078	4E	BA842	ALC	+	+	01001110	+	1001110	N	+
079	4F	BA8421	SLC	|	|	01001111	|	1001111	O	|
080	50	A8 2		&	&	01010000	&	1010000	P	&
081	51	D B 1		J 8	J 2	01010001		1010001	Q	
082	52	D B 2		K 8	K 2	01010010		1010010	R	
083	53	D B 21		L 8	L 2	01010011		1010011	S	
084	54	D B 4	ZAZ	M 8	M 2	01010100		1010100	T	
085	55	D B 4 1		N 8	N 2	01010101		1010101	U	
086	56	D B 42	AZ	O 8	O 2	01010110		1010110	V	
087	57	D B 421	SZ	P 8	P 2	01010111		1010111	W	
088	58	D B 8	MVX	Q 8	Q 2	01011000		1011000	X	
089	59	D B 8 1		R 8	R 2	01011001		1011001	Y	
090	5A	B 8 2	ED	!	!	01011010	!	1011010	Z	!
091	5B	B 8 21	ITC	$	$	01011011	$	1011011	[	$
092	5C	B 84	MVC	*	*	01011100	*	1011100	\	*
093	5D	B 84 1	CLC	)	)	01011101	)	1011101	]	)
094	5E	B 842	ALC	;	;	01011110	;	1011110	¬	;
095	5F	B 8421	SLC	¬	¬	01011111	¬	1011111	-	¬

Dec	Hex	Card Code	Mnem	IPL*		EBCDIC Code	EBCDIC	ASCII Code	ASCII	System/3
Val	Val	DCBA8421		T1T3	T2T3	01234567	Character	7654321	Character	Symbol**
096	60	B		-	-	01100000	-	1100000	'	-
097	61	A 1		/	/	01100001	/	1100001	a	/
098	62	D A 2		S 8	S 2	01100010		1100010	b	
099	63	D A 21		T 8	T 2	01100011		1100011	c	
100	64	D A 4	ZAZ	U 8	U 2	01100100		1100100	d	
101	65	D A 4 1		V 8	V 2	01100101		1100101	e	
102	66	D A 42	AZ	W 8	W 2	01100110		1100110	f	
103	67	D A 421	SZ	X 8	X 2	01100111		1100111	g	
104	68	D A8	MVX	Y 8	Y 2	01101000		1101000	h	
105	69	D A8 1		Z 8	Z 2	01101001		1101001	i	
106	6A	D BA	ED	} 8	} 2	01101010	¦	1101010	j	
107	6B	A8 21	ITC	,	.	01101011	,	1101011	k	,
108	6C	A84	MVC	%	%	01101100	%	1101100	l	%
109	6D	A84 1	CLC	_	_	01101101	_	1101101	m	_
110	6E	A842	ALC	>	>	01101110	>	1101110	n	>
111	6F	A8421	SLC	?	?	01101111	?	1101111	o	?
112	70	D A	SNS	0 8	0 2	01110000		1110000	p	
113	71	D 1	LIO	1 8	1 2	01110001		1110001	q	
114	72	D 2		2 8	2 2	01110010		1110010	r	
115	73	D 21		3 8	3 2	01110011		1110011	s	
116	74	D 4	ST	4 8	4 2	01110100		1110100	t	
117	75	D 4 1	L	5 8	5 2	01110101		1110101	u	
118	76	D 42	A	6 8	6 2	01110110		1110110	v	
119	77	D 421		7 8	7 2	01110111		1110111	w	
120	78	D 8	TBN	8 8	8 2	01111000		1111000	x	
121	79	D 8 1	TBF	9 8	9 2	01111001	`	1111001	y	
122	7A	8 2	SBN	:	:	01111010	:	1111010	z	:
123	7B	8 21	SBF	#	#	01111011	#	1111011	{	#
124	7C	84	MVI	@	@	01111100	@	1111100	¦	@
125	7D	84 1	CLI	'	'	01111101	'	1111101	}	'
126	7E	842		=	=	01111110	=	1111110	~	=
127	7F	8421		"	"	01111111	"	1111111	DEL	"
128	80	DC		@	3	10000000				
129	81	CBA 1		A 4	A 1	10000001	a			a
130	82	CBA 2		B 4	B 1	10000010	b			b
131	83	CBA 21		C 4	C 1	10000011	c			c
132	84	CBA 4	ZAZ	D 4	D 1	10000100	d			d
133	85	CBA 4 1		E 4	E 1	10000101	e			e
134	86	CBA 42	AZ	F 4	F 1	10000110	f			f
135	87	CBA 421	SZ	G 4	G 1	10000111	g			g
136	88	CBA8	MVX	H 4	H 1	10001000	h			h
137	89	CBA8 1		I 4	I 1	10001001	i			i
138	8A	DCBA8 2	ED	c @	c 3	10001010				
139	8B	DCBA8 21	ITC	. @	. 3	10001011				
140	8C	DCBA84	MVC	< @	< 3	10001100				Note 1 ≤
141	8D	DCBA84 1	CLC	(@	(3	10001101				(
142	8E	DCBA842	ALC	+ @	+ 3	10001110				+
143	8F	DCBA8421	SLC	| @	| 3	10001111				+

**Characters on right side of column are not handled by six-bit devices.

Note 1. Symbols printed by System/3 devices equipped with TN character sets.
8D and 8E are superscript characters.

Dec	Hex	Card Code	Mnem	IPL*		EBCDIC Code	EBCDIC	ASCII Code	ASCII	System/3
Val	Val	DCBA8421		T1T3	T2T3	01234567	Character	7654321	Character	Symbol**
144	90	CBA		} 4	} 1	10010000				
145	91	CB 1		J 4	J 1	10010001	j			j
146	92	CB 2		K 4	K 1	10010010	k			k
147	93	CB 21		L 4	L 1	10010011	l			l
148	94	CB 4	ZAZ	M 4	M 1	10010100	m			m
149	95	CB 4 1		N 4	N 1	10010101	n			n
150	96	CB 42	AZ	O 4	O 1	10010110	o			o
151	97	CB 421	SZ	P 4	P 1	10010111	p			p
152	98	CB 8	MVX	Q 4	Q 1	10011000	q			q
153	99	CB 8 1		R 4	R 1	10011001	r			r
154	9A	DCB 8 2	ED	! @	! 3	10011010				
155	9B	DCB 8 21	ITC	$ @	$ 3	10011011				
156	9C	DCB 84	MVC	* @	* 3	10011100				Note 1 ¤
157	9D	DCB 84 1	CLC	) @	) 3	10011101				)
158	9E	DCB 842	ALC	; @	; 3	10011110				±
159	9F	DCB 8421	SLC	¬ @	¬ 3	10011111				□
160	A0	DCB		– @	– 3	10100000				–
161	A1	DC A 1		/ @	/ 3	10100001	~			~
162	A2	C A 2		S 4	S 1	10100010	s			s
163	A3	C A 21		T 4	T 1	10100011	t			t
164	A4	C A 4	ZAZ	U 4	U 1	10100100	u			u
165	A5	C A 4 1		V 4	V 1	10100101	v			v
166	A6	C A 42	AZ	W 4	W 1	10100110	w			w
167	A7	C A 421	SZ	X 4	X 1	10100111	x			x
168	A8	C A8	MVX	Y 4	Y 1	10101000	y			y
169	A9	C A8 1		Z 4	Z 1	10101001	z			z
170	AA	DC A8 2	ED	& @	& 3	10101010				
171	AB	DC A8 21	ITC	, @	, 3	10101011				└
172	AC	DC A84	MVC	% @	% 3	10101100				┌
173	AD	DC A84 1	CLC	_ @	_ 3	10101101				[
174	AE	DC A842	ALC	> @	> 3	10101110				>
175	AF	DC A8421	SLC	? @	? 3	10101111				●
176	B0	C A	SNS	0 4	0 1	10110000				^{0}
177	B1	C 1	LIO	1 4	1 1	10110001				^{1}
178	B2	C 2		2 4	2 1	10110010				^{2}
179	B3	C 21		3 4	3 1	10110011				^{3}
180	B4	C 4	ST	4 4	4 1	10110100				^{4}
181	B5	C 4 1	L	5 4	5 1	10110101				^{5}
182	B6	C 42	A	6 4	6 1	10110110				^{6}
183	B7	C 421		7 4	7 1	10110111				^{7}
184	B8	C 8	TBN	8 4	8 1	10111000				^{8}
185	B9	C 8 1	TBF	9 4	9 1	10111001				^{9}
186	BA	DC 8 2	SBN	: @	: 3	10111010				
187	BB	DC 8 21	SBF	# @	# 3	10111011				┘
188	BC	DC 84	MVI	@ @	@ 3	10111100				┐
189	BD	DC 84 1	CLI	' @	' 3	10111101				]
190	BE	DC 842		= @	= 3	10111110				≠
191	BF	DC 8421		" @	" 3	10111111				–

**Characters on right side of column are not handled by six-bit devices.

Note 1. Symbols printed by System/3 devices equipped with TN character sets. 9D, A0, and B0 through B9 are superscript characters.

Dec	Hex	Card Code	Mnem	IPL*		EBCDIC Code	EBCDIC	ASCII Code	ASCII	System/3
Val	Val	DCBA8421		T1T3	T2T3	01234567	Character	7654321	Character	Symbol**
192	C0	D	BC	8	2	11000000	{			
193	C1	BA 1	TIO	A	A	11000001	A			A
194	C2	BA 2	LA	B	B	11000010	B			B
195	C3	BA 21		C	C	11000011	C			C
196	C4	BA 4		D	D	11000100	D			D
197	C5	BA 4 1		E	E	11000101	E			E
198	C6	BA 42		F	F	11000110	F			F
199	C7	BA 421		G	G	11000111	G			G
200	C8	BA8		H	H	11001000	H			H
201	C9	BA8 1		I	I	11001001	I			I
202	CA	D BA8 2		¢ 8	¢ 2	11001010				Note 1 } °
203	CB	D BA8 21		. 8	. 2	11001011				
204	CC	D BA84		< 8	< 2	11001100	ʃ Note 2			
205	CD	D BA84 1		(8	(2	11001101				
206	CE	D BA842		+ 8	+ 2	11001110	Ч			
207	CF	D BA8421		| 8	| 2	11001111				
208	D0	BA	BC	}	}	11010000	}			}
209	D1	B 1	TIO	J	J	11010001	J			J
210	D2	B 2	LA	K	K	11010010	K			K
211	D3	B 21		L	L	11010011	L			L
212	D4	B 4		M	M	11010100	M			M
213	D5	B 4 1		N	N	11010101	N			N
214	D6	B 42		O	O	11010110	O			O
215	D7	B 421		P	P	11010111	P			P
216	D8	B 8		Q	Q	11011000	Q			Q
217	D9	B 8 1		R	R	11011001	R			R
218	DA	D B 8 2		! 8	! 2	11011010				
219	DB	D B 8 21		$ 8	$ 2	11011011				
220	DC	D B 84		* 8	* 2	11011100				
221	DD	D B 84 1		) 8	) 2	11011101				
222	DE	D B 842		; 8	; 2	11011110				
223	DF	D B 8421		¬ 8	¬ 2	11011111				
224	E0	D B	BC	– 8	– 2	11100000	\			\
225	E1	D A 1	TIO	/ 8	/ 2	11100001				
226	E2	A 2	LA	S	S	11100010	S			S
227	E3	A 21		T	T	11100011	T			T
228	E4	A 4		U	U	11100100	U			U
229	E5	A 4 1		V	V	11100101	V			V
230	E6	A 42		W	W	11100110	W			W
231	E7	A 421		X	X	11100111	X			X
232	E8	A8		Y	Y	11101000	Y			Y
233	E9	A8 1		Z	Z	11101001	Z			Z
234	EA	D A8 2		& 8	& 2	11101010				
235	EB	D A8 21		, 8	, 2	11101011				

**Characters on right side of column are not handled by six-bit devices.

Note 1. Symbols printed by System/3 devices equipped with TN character sets.

Note 2. Special graphics.

Dec	Hex	Card Code	Mnem	IPL*		EBCDIC Code	EBCDIC	ASCII Code	ASCII	System/3
Val	Val	DCBA8421		T1T3	T2T3	01234567	Character	7654321	Character	Symbol**
236	EC	D A84		% 8	% 2	11101100	⑁			
237	ED	D A84 1		_8	_2	11101101				
238	EE	D A842		> 8	> 2	11101110				
239	EF	D A8421		? 8	? 2	11101111				
240	F0	A	HPL	0	0	11110000	0			0
241	F1	1	APL	1	1	11110001	1			1
242	F2	2	JC	2	2	11110010	2			2
243	F3	21	SIO	3	3	11110011	3			3
244	F4	4		4	4	11110100	4			4
245	F5	4 1		5	5	11110101	5			5
246	F6	42		6	6	11110110	6			6
247	F7	421		7	7	11110111	7			7
248	F8	8		8	8	11111000	8			8
249	F9	8 1		9	9	11111001	9			9
250	FA	D 8 2		: 8	: 2	11111010	|			|
251	FB	D 8 21		# 8	# 2	11111011				
252	FC	D 84		@ 8	@ 2	11111100				
253	FD	D 84 1		' 8	' 2	11111101				
254	FE	D 842		= 8	= 2	11111110				
255	FF	D 8421		" 8	" 2	11111111				

* If both tier 1 and tier 2 are being used, the tier 3 punches are added together as shown in the table at the end of this chart.

** Characters on right side of column are not handled by six-bit devices.

*Tier 3 character addition table

Tier 3 Character Required by Tier 1	Tier 3 Character Required by Tier 2		
	1	2	3
4	5	6	7
8	9	:	#
@	'	=	"

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

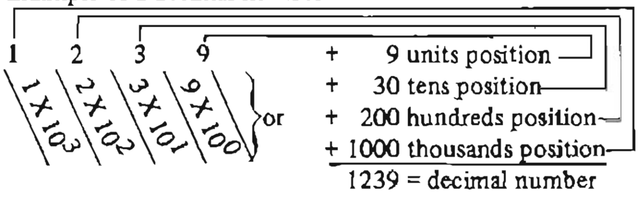
Binary and Hexadecimal Number Notation

Binary Number Notation

A binary number system, such as is used in System/3 uses a base of two. The concept of using a base of two can be compared with the base of ten (decimal) number system.

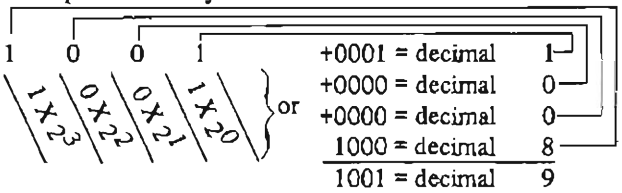
decimal number		binary number
0	=	0
1	=	1
2	=	10
3	=	11
4	=	100
5	=	101
6	=	110
7	=	111
8	=	1000
9	=	1001

Example of a decimal number



As shown above, the decimal number system allows counting to ten in each position—from units to tens to hundreds to thousands etc. The binary system allows counting to two in each position. Register displays in the System/3 are in binary forms: a bit light on is a "one"; a bit light off is a "zero".

Example of a binary number



Hexadecimal Number System

It has been noted that binary numbers require about three times as many positions as decimal numbers to express the equivalent number. This is not much of a problem to the computer; however, in talking and writing or in communicating with the computer, these binary numbers are bulky. A long string of 1's and 0's cannot be effectively transmitted from one individual to another. Some shorthand method is necessary.

The hexadecimal number system fills this need. Because of the simple relationship of hexadecimal to binary, numbers can be converted from one system to another by inspection. The base or radix of the hexadecimal system is 16. This means there are 16 symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. The letters A, B, C, D, E, and F represent the 10-base system values of 10, 11, 12, 13, 14, and 15, respectively.

Four binary positions are equivalent to one hexadecimal position. The following table shows the comparable values of the three number systems.

Decimal	*Binary*	*Hexadecimal*
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

At this point all 16 symbols have been used, and a carry to the next higher position of the number is necessary. For example:

Decimal	*Binary*	*Hexadecimal*
16	0001 0000	10
17	0001 0001	11
18	0001 0010	12
19	0001 0011	13
20	0001 0100	14
21	0001 0101	15
-- and so on --		

Remember that as far as the internal circuitry of the computer is concerned, it only understands binary. But an operator can look at a series of lights on the computer console showing binary 1's and 0's, for example: 0001 1110 0001 0011, and say that the lights represent the hexadecimal value 1E13 which is easier to state than the string of 1's and 0's.

Hexadecimal – Decimal Conversion Tables

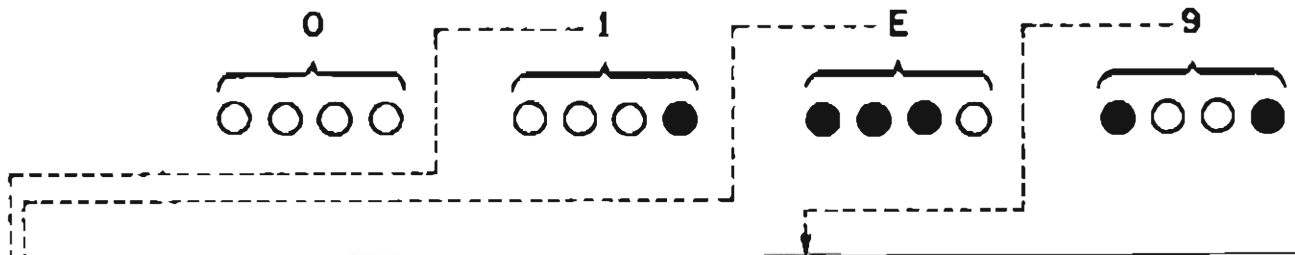
The table in this appendix provides for direct conversion of decimal and hexadecimal number in these ranges:

Hexadecimal	Decimal
000 to FFF	0000 to 4095

For numbers outside the range of the table, add the following values to the table figures:

Hexadecimal	Decimal
1000	4096
2000	8192
3000	12288

Hexadecimal	Decimal
4000	16384
5000	20480
6000	24576
7000	28672
8000	32768



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 _	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015
01 _	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031
02 _	0032	0033	0034	0035	0036	0037	0038	0039	0040	0041	0042	0043	0044	0045	0046	0047
03 _	0048	0049	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	0060	0061	0062	0063
04 _	0064	0065	0066	0067	0068	0069	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079
05 _	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	0090	0091	0092	0093	0094	0095
06 _	0096	0097	0098	0099	0100	0101	0102	0103	0104	0105	0106	0107	0108	0109	0110	0111
07 _	0112	0113	0114	0115	0116	0117	0118	0119	0120	0121	0122	0123	0124	0125	0126	0127
08 _	0128	0129	0130	0131	0132	0133	0134	0135	0136	0137	0138	0139	0140	0141	0142	0143
09 _	0144	0145	0146	0147	0148	0149	0150	0151	0152	0153	0154	0155	0156	0157	0158	0159
0A _	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	0173	0174	0175
0B _	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0186	0187	0188	0189	0190	0191
0C _	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0202	0203	0204	0205	0206	0207
0D _	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0218	0219	0220	0221	0222	0223
0E _	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0234	0235	0236	0237	0238	0239
0F _	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0250	0251	0252	0253	0254	0255
10 _	0256	0257	0258	0259	0260	0261	0262	0263	0264	0265	0266	0267	0268	0269	0270	0271
11 _	0272	0273	0274	0275	0276	0277	0278	0279	0280	0281	0282	0283	0284	0285	0286	0287
12 _	0288	0289	0290	0291	0292	0293	0294	0295	0296	0297	0298	0299	0300	0301	0302	0303
13 _	0304	0305	0306	0307	0308	0309	0310	0311	0312	0313	0314	0315	0316	0317	0318	0319
14 _	0320	0321	0322	0323	0324	0325	0326	0327	0328	0329	0330	0331	0332	0333	0334	0335
15 _	0336	0337	0338	0339	0340	0341	0342	0343	0344	0345	0346	0347	0348	0349	0350	0351
16 _	0352	0353	0354	0355	0356	0357	0358	0359	0360	0361	0362	0363	0364	0365	0366	0367
17 _	0368	0369	0370	0371	0372	0373	0374	0375	0376	0377	0378	0379	0380	0381	0382	0383
18 _	0384	0385	0386	0387	0388	0389	0390	0391	0392	0393	0394	0395	0396	0397	0398	0399
19 _	0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	0410	0411	0412	0413	0414	0415
1A _	0416	0417	0418	0419	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	0430	0431
1B _	0432	0433	0434	0435	0436	0437	0438	0439	0440	0441	0442	0443	0444	0445	0446	0447
1C _	0448	0449	0450	0451	0452	0453	0454	0455	0456	0457	0458	0459	0460	0461	0462	0463
1D _	0464	0465	0466	0467	0468	0469	0470	0471	0472	0473	0474	0475	0476	0477	0478	0479
1E _	0480	0481	0482	0483	0484	0485	0486	0487	0488	0489	0490	0491	0492	0493	0494	0495
1F _	0496	0497	0498	0499	0500	0501	0502	0503	0504	0505	0506	0507	0508	0509	0510	0511

53304

	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

53305

	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

53306

	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

53307

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E0_	3584	3585	3586	3587	3588	3589	3590	3591	3592	3593	3594	3595	3596	3597	3598	3599
E1_	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612	3613	3614	3615
E2_	3616	3617	3618	3619	3620	3621	3622	3623	3624	3625	3626	3627	3628	3629	3630	3631
E3_	3632	3633	3634	3635	3636	3637	3638	3639	3640	3641	3642	3643	3644	3645	3646	3647
E4_	3648	3649	3650	3651	3652	3653	3654	3655	3656	3657	3658	3659	3660	3661	3662	3663
E5_	3664	3665	3666	3667	3668	3669	3670	3671	3672	3673	3674	3675	3676	3677	3678	3679
E6_	3680	3681	3682	3683	3684	3685	3686	3687	3688	3689	3690	3691	3692	3693	3694	3695
E7_	3696	3697	3698	3699	3700	3701	3702	3703	3704	3705	3706	3707	3708	3709	3710	3711
E8_	3712	3713	3714	3715	3716	3717	3718	3719	3720	3721	3722	3723	3724	3725	3726	3727
E9_	3728	3729	3730	3731	3732	3733	3734	3735	3736	3737	3738	3739	3740	3741	3742	3743
EA_	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755	3756	3757	3758	3759
EB_	3760	3761	3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774	3775
EC_	3776	3777	3778	3779	3780	3781	3782	3783	3784	3785	3786	3787	3788	3789	3790	3791
ED_	3792	3793	3794	3795	3796	3797	3798	3799	3800	3801	3802	3803	3804	3805	3806	3807
EE_	3808	3809	3810	3811	3812	3813	3814	3815	3816	3817	3818	3819	3820	3821	3822	3823
EF_	3824	3825	3826	3827	3828	3829	3830	3831	3832	3833	3834	3835	3836	3837	3838	3839
F0_	3840	3841	3842	3843	3844	3845	3846	3847	3848	3849	3850	3851	3852	3853	3854	3855
F1_	3856	3857	3858	3859	3860	3861	3862	3863	3864	3865	3866	3867	3868	3869	3870	3871
F2_	3872	3873	3874	3875	3876	3877	3878	3879	3880	3881	3882	3883	3884	3885	3886	3887
F3_	3888	3889	3890	3891	3892	3893	3894	3895	3896	3897	3898	3899	3900	3901	3902	3903
F4_	3904	3905	3906	3907	3908	3909	3910	3911	3912	3913	3914	3915	3916	3917	3918	3919
F5_	3920	3921	3922	3923	3924	3925	3926	3927	3928	3929	3930	3931	3932	3933	3934	3935
F6_	3936	3937	3938	3939	3940	3941	3942	3943	3944	3945	3946	3947	3948	3949	3950	3951
F7_	3952	3953	3954	3955	3956	3957	3958	3959	3960	3961	3962	3963	3964	3965	3966	3967
F8_	3968	3969	3970	3971	3972	3973	3974	3975	3976	3977	3978	3979	3980	3981	3982	3983
F9_	3984	3985	3986	3987	3988	3989	3990	3991	3992	3993	3994	3995	3996	3997	3998	3999
FA_	4000	4001	4002	4003	4004	4005	4006	4007	4008	4009	4010	4011	4012	4013	4014	4015
FB_	4016	4017	4018	4019	4020	4021	4022	4023	4024	4025	4026	4027	4028	4029	4030	4031
FC_	4032	4033	4034	4035	4036	4037	4038	4039	4040	4041	4042	4043	4044	4045	4046	4047
FD_	4048	4049	4050	4051	4052	4053	4054	4055	4056	4057	4058	4059	4060	4061	4062	4063
FE_	4064	4065	4066	4067	4068	4069	4070	4071	4072	4073	4074	4075	4076	4077	4078	4079
FF_	4080	4081	4082	4083	4084	4085	4086	4087	4088	4089	4090	4091	4092	4093	4094	4095

53308

Timing Summary

INSTRUCTION TIMING

Instruction	*Mnemonic*	*Time (in microseconds)*
Zero and Add Zoned	ZAZ	1.52 (N+L1+L2) + 1.52 (R)
Add Zoned Decimal	AZ	1.52 (N+L1+L2) + 1.52 (R)
Subtract Zoned Decimal	SZ	1.52 (N+L1+L2) + 1.52 (R)
Add Logical Characters	ALC	1.52 (N+2L)
Subtract Logical Characters	SLC	1.52 (N+2L)
Add to Register	A	1.52 (N+2)
Move Hex Character	MVX	1.52 (N+2)
Move Characters	MVC	1.52 (N+2L)
Edit	ED	1.52 (N+L1+L2)
Insert and Test Characters	ITC	1.52 (N+1+L1)
Move Logical Immediate	MVI	1.52 (N+1)
Set Bits On Masked	SBN	1.52 (N+1)
Set Bits Off Masked	SBF	1.52 (N+1)
Store Register	ST	1.52 (N+2)
Load Register	L	1.52 (N+2)
Load Address	LA	1.52 (N)
Compare Logical Characters	CLC	1.52 (N+2L)
Compare Logical Immediate	CLI	1.52 (N+1)
Test Bits On Masked	TBN	1.52 (N+1)
Test Bits Off Masked	TBF	1.52 (N+1)
Branch on Condition	BC	1.52 (N)
Jump on Condition	JC	4.56
Halt Program Level	HPL	4.56
Start I/O	SIO	4.56
Sense I/O	SNS	1.52 (N+2)
Load I/O	LIO	1.52 (N+2)
Test I/O and Branch	TIO	1.52 (N)
Advance Program Level	APL	4.56

Note:

In the timing formulas,

N= Instruction length in bytes.

L1=Length of destination field (two address instruction) in bytes. Destination field is that field addressed by operand 1.

L2=Length of source field (two address instruction) in bytes. Source field is that field addressed by operand 2.

L= Length of the operands when the length of operand 1 must equal the length of operand 2.

R= Length of operand 1 when recomplementing is necessary.

DISK TIMING, 5444

Seek time for 1 track is 39 ms. Seek time for 2 or more tracks: 47 + 3.42 (N-2) = time in milliseconds, where N = the number of tracks to be crossed. (The factor 3.42 represents the average maximum track crossing time after two tracks have been crossed.)

Glossary

The following terms are defined as they are used in this manual. If you do not find the term you are looking for, refer to the *IBM Data Processing Glossary*, GC20-1699.

Access Arm–A part of a disk storage unit that is used to hold one or more reading and writing heads.

ALU–Arithmetic and logical unit.

Base Address–A given address from which a storage address is derived by combination with a relative address.

Binary–(1) Pertaining to a characteristic or property involving a selection, choice, or condition in which there are two possibilities. (2) Pertaining to the numeration system with a radix of two.

Binary Digit–In binary notation, either of the characters 0 or 1.

Binary Notation–A fixed radix notation where the radix is two. For example, in binary notation the numeral 110.01 represents the number 1 x 2 squared plus 1 x 2 to the first power plus 1 x 2 to the minus 2 power, that is, six and a quarter.

Binary Number–Loosely, a binary numeral.

Binary Numeral–A binary representation of a number. For example, 101 is the binary numeral and V is the equivalent Roman numeral.

Bit–(1) A binary digit. (2) Contraction of 'binary digit', the smallest unit of information in a binary system. A bit may be either a 1 (on) or a zero (off).

Blank–A code character to denote the presence of no information rather than the absence of information.

Blank Character–See Space Character.

Block – A collection of contiguous records recorded as a unit. Blocks are separated by interrecord gaps on tape, and each block may contain one or more records.

Bpi – The number of bits per inch per row (track) or the number of bytes per inch on tape. This term is used when referring to the recording density of the tape unit.

Byte–A sequence of adjacent binary digits (8) operated upon as a unit.

Card Code–The combinations of punched holes that represent characters (letters, digits, etc.) in a punched card.

Card Column–A single line of punching positions parallel to the short edge of a punched card.

Card Feed–A mechanism which moves cards into a machine one at a time.

Card Hopper–A device that holds cards and makes them available to a card feed mechanism.

Card Stacker–An output device that accumulates punched cards in a deck.

Carry–One or more characters, produced in connection with an arithmetic operation on one digit place of two or more numerals in positional notation, that are forwarded to another digit place for processing there.

Character–A letter, digit, or other symbol that is used as part of the organization, control, or representation of data.

Character Printer–A device that prints a single character at a time.

Character Set–An ordered set of unique representation called characters, for example, the 26 letters of the English alphabet, 0 and 1 of the Boolean alphabet, the set of signals in the Morse code alphabet.

Check–A process for determining accuracy.

Check Bit–A binary check digit, for example, a parity bit.

Check Character–A character used for the purpose of performing a check.

Code–A set of unambiguous rules specifying the way in which data may be represented.

Code Conversion–A process for changing the bit grouping for a character in one code into the corresponding bit grouping for a character in a second code.

Collate–To compare and merge two or more similarly ordered sets of items into one ordered set.

Collator–A device to collate sets of punched cards or other documents into a sequence.

Column–A vertical arrangement of characters or other expressions. Loosely, a digit place.

Command–An instruction.

Comparison–The examination of the relationship between two similar items of data.

Complement–A number that can be derived from a specified number by subtracting the specified number from another specified number.

Conditional Jump–A jump that occurs if specified criteria are met.

Data–A representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by humans or automatic means.

Data Processing System–A network of machine component capable of accepting information, processing it according to a plan, and producing the desired results.

Direct Address–An address that specifies the location of an operand.

Disk–A physical element of disk storage.

Disk Storage–A storage device which uses magnetic recording on flat rotating disks.

Display–A visual presentation of data.

Document–(1) A medium and the data recorded on it for human use, for example, a report sheet. (2) By extension, any record that has permanence and that can be read by man or machine.

Edit–To modify the form or format of data, for example, to insert or delete characters such as page numbers or decimal points.

Effective Address–The address that is derived by applying any specified indexing or indirect addressing rules to the specified address and that is actually used to identify the current operand.

End-Of-Tape Marker – A marker on a magnetic tape used to indicate the end of the permissible recording area, for example, a photo-reflective strip, a transparent section of tape, or a particular bit pattern.

End of Transmission (EOT)–The specific character or sequence of characters which indicates termination of sending.

EOT–End of transmission.

Erase–To obliterate information from a storage medium.

Erase Head – A device on a magnetic tape unit whose sole function is to erase previous information before writing new information.

Execute–To carry out an instruction or perform a routine.

Fetch–To locate and load a quantity of data from storage.

Field–In a record, a specified area used for a particular category of data, for example, a group of card columns used to represent a wage rate or a set of byte locations in a computer storage used to express another storage address.

File Protection – Prevention of the destruction of data recorded on a volume by disabling the write head of a unit.

Font–A family or assortment of characters of a given size and style.

Format–A specific arrangement of data.

Graphic–A symbol produced by a process such as handwriting, drawing, or printing.

Graphic Character–A character normally represented by a graphic.

Halt Instruction–A machine instruction which stops the execution of the program.

Hard Copy–A printed copy of machine output in a visually readable form, such as printed reports.

Head–A device that reads, records, or erases data on a storage medium; for example, a small electromagnet used to read, write, or erase data on magnetic disk or tape.

Hexadecimal–Pertaining to the numeration system with a radix of sixteen.

Hit–A successful comparison of two items of data.

Hopper–A card hopper.

I/O–Input/output. Input or output, or both.

Indexed Address–An address which is modified by the content of an index register prior to or during the execution of a computer instruction.

Indexing—A technique of address modification often implemented by means of index registers.

Indicator—A device which registers a condition in the computer.

Indirect Address—An address that specifies a storage location derived by adding an indexing factor to an index register.

Initialize—To set counter, switches, and addresses to zero or other starting values at the beginning of, or at a prescribed point in a computer routine.

Initial Program Load (IPL)—The procedure that causes the initial part of an operating system or other program to be loaded so that the program can then proceed under its own control.

Input Data—Data to be processed.

Input Device—A device used for conveying data to the processing unit.

Input/Output—(1) Commonly called I/O. (2) A general term for the equipment used to communicate with the processing unit.

Instruction—A statement that specifies an operation and the values or locations of its operands.

Instruction Address—The address of the location where an instruction word is stored.

Instruction Format—The allocation of bits or bytes of a machine instruction to specific functions.

Interblock Gap — A blank space on magnetic tape that separates physical records.

Interpreter—A device that prints on a punched card the data already punched in the card.

Interrupt—To stop a process in such a way that it can be resumed.

Jump—A departure from the normal sequence of executing instructions in a computer.

Justification—The act of adjusting or arranging characters or digits to the left or right to fit a prescribed pattern.

Justify—To align data about a specified reference.

Line Printer—A device that prints all characters of a line as a unit.

Load—In programming, to enter data into storage or working registers.

Loadpoint – The beginning of the usable portion of a reel of tape, indicated by a load point marker, where reading or writing is to begin.

Location—Loosely, any place in which data can be stored.

Loop—(n) A sequence of instructions that is repeated until a terminal condition exists. (v) To repeat an instruction or series of instructions until a terminal condition exists.

Magnetic Disk—A flat circular plate with a magnetic surface on which data can be stored by selective magnetization of portions of the flat surface.

Magnetic Tape — A tape with a magnetic surface on which data can be stored by selective polorization of portions of the surface.

Main Storage—The general purpose internal storage of a computer.

Mask—A pattern of bits that is used to control the retention or elimination of portions of another pattern of bits.

Mnemonic—Same as mnemonic symbol.

Mnemonic Symbol—A symbol chosen to assist the human memory, for example, the abbreviation MPY for multiply.

Multivolume Tape File — A file stored on more than one tape reel.

Nines Complement—The radix-minus-one complement in decimal notation.

No Op—An instruction that performs no function except to proceed to the next instruction in sequence.

Notation—A representational system which utilizes characters and symbols in positional relationships to express information.

NRZI (Non-Return-To-Zero IBM) – A method of recording on tape where only the one-bits are written as magnetized spots on tape.

Number—A mathematical entity that may indicate quantity or amount of units.

One-Address—Pertaining to an instruction format containing one address part.

Operand—That which is operated upon. An operand is usually identified by an address part of an instruction.

Operation—(1) A defined action, namely the act of obtaining a result from one or more operands in accordance with a rule that completely specifies the result for any permissible combination of operands. (2) The act specified by a single computer instruction.

Operation Code—A code that represents specific operations.

Operator—A person who operates a machine.

Output—The data that has been processed.

Overflow—That portion of the result of an operation that exceeds the capacity of the intended unit of storage.

Parity Bit—A binary digit appended to an array of bits to make the sum of all the bits always odd or always even.

Parity Check—A check that tests whether the number of ones or zeros in an array of binary digits is odd or even.

Pass—One cycle of processing a body of data.

PE (Phase Encoding) – A method of recording on tape where both zero and one-bits are written as magnetized spots. The zero and one-bit are opposite in polarity. This method allows distinction between zero-bits and no recording.

Printer—A device which expresses coded characters as hard copy.

Program—A series of actions proposed in order to achieve a certain result.

Programmer—A person mainly involved in designing, writing, and testing programs.

Programming—The design, the writing, and the testing of a program.

Punched Card—A card punched with a pattern of holes to represent data.

Radix—In positional representation, the integral ration of the significances of any two specified adjacent digit positions.

Radix-Minus-One-Complement—A complement obtained by subtracting each digit from one less than the radix.

Read—To acquire or interpret data from a storage device, a data medium, or any other source.

Read Access Time – The interval from issuance of a read forward read command given to the tape control when tape is not at load point, until the first data byte is read when tape is brought up to speed from stopped status.

Reel – A mounting for a roll of tape.

Register—A device capable of storing a specified amount of data, such as two bytes.

Seek—To position the access mechanism of a disk drive at a specified track.

Serdes—Serializer-deserializer. A device that changes data flow from parallel-by-bit to serial-by-bit or from serial-by-bit to parallel-by-bit.

Space Character—A normally non-printing graphic character used to separate words.

Storage—Pertaining to a device into which data can be entered, in which it can be held, and from which it can be retrieved at a later time.

Storage Capacity—The amount of data that can be contained in a storage device.

Storage Device—A device into which data can be inserted, in which it can be retained, and from which it can be retrieved.

Subsystem – A secondary or subordinate system, usually capable of operating independenly of or asynchronously with a controlling system.

Tape Labels – Special records at the beginning and ending of tape files. There are volume labels, and trailer labels. They are used to identify the reel and the recorded data file. They also contain certain housekeeping information.

Tape Mark – A special symbol that can be read from, or written on, magnetic tape. It is used to indicate the end of a file or a file segment, and to segregate the labels from data. Tape marks must be read in the same density in which they were written.

Tape Unit – A device containing a tape drive to move tape past the reading and writing heads, and containing the associated controls.

Time-Share—To use a device for two or more interleaved purposes.

Time-Sharing—Pertaining to the interleaved use of the time of a device.

Track—The portion of a moving storage medium, such as a drum, tape, or disk, that is accessible to a given reading head position.

Two-Address—Pertaining to an instruction format containing two address parts.

Verify—To determine whether a transcription of data or other operation has been accomplished correctly.

Write—To record data in a storage device or a data medium.

Write Access Time – The interval from the issuance of a write command given to the tape control, when the tape is not at load point, until the first data byte is written on tape when tape is brought up to speed from stopped status.

Write—Enable Ring – A plastic ring that fits in a circular groove molded in the back (machine side) of the tape reel. This ring must be in place to enable the machine to write on the tape. When the ring is removed, only reading can take place; the file is protected from accidental writing, which could erase valuable information.

INDEX

IBM System/3 Model 10 Components Reference (File No. S3-01) Printed in U.S.A. GA21-9103-5

IBM

International Business Machines Corporation
General Systems Division
875 Johnson Ferry Road N. E.
Atlanta, Ga. 30342

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

IBM Technical Newsletter

This Newsletter No.	GN21-0205
Date	17 April 1975
Base Publication No.	GA21-9103-4 and -5
File No.	S3-01
Previous Newsletters	GN21-0169 (for -4 only)

IBM System/3
Model 10
Components
Reference Manual

This Technical Newsletter provides new and replacement pages for the subject publication. These new and replacement pages remain in effect for subsequent editions unless specifically altered. Pages to be inserted and/or removed are:

Title Page, ii
v, vi
3-13, 3-14
4-9, 4-10
10-5, 10-6

11-1, 11-2
13-9, 13-10
16-1 through 16-8 (added)
A-3 through A-8

Changes to text and illustrations are indicated by a vertical line at the left of the change; new or extensively revised illustrations are denoted by the symbol ● at the left of the caption.

Summary of Amendments

- Information on IBM 3741 directly attached to the System/3
- Minor changes

Note: Please file this cover letter at the back of the manual to provide a record of changes.

IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901