File No. S360-01
Order No. GA24-3231-7

IBM Systems Reference Library

IBM System/360 Model 30 Functional Characteristics

This reference publication describes the relationship of the IBM System/360 Model 30 to the entire System/360. The system's capabilities, features, I/O channels, and operations are also discussed.

The reader can find a more detailed description of System/360 operations in *IBM System/360 Principles of Operation,* GA22-6821. The interrelationships of the models and units available with System/360 are broadly described in *IBM System/360 System Summary,* GA22-6810. Other related literature is referenced by order number and briefly described in *IBM System/360 Bibliography,* GA22-6822. Information on channel load limits for IBM System/360 Model 30 is given in *IBM System/360 Model 30 Channel Characteristics and Functional Evaluation,* GA24-3411. System-console operating procedures are contained in *IBM System/360 Model 30 Operating Guide,* GA24-3373.

Preface

The first section of this manual—*IBM System/360 Model 30, General Concepts*—presents a general introduction to some important IBM System/360 concepts and terminology. Throughout most of this section, concepts such as program status words, supervisor program, and interrupts are related to input/output operations. The object of this section is to present certain system operations, referenced to IBM System/360 Model 30, without taking into account all possible variations and exceptions that may apply. Subsequent sections of this publication provide more detailed information.

IBM System/360 Model 30 has a storage cycle time of 1.5 microseconds. Early systems, however, have a 2-microsecond storage cycle time. In this publication, timings that pertain to the 2-microsecond Central Processing Unit (CPU) are shown in parentheses following the timings given for the 1.5-microsecond CPU. For example:

> The RCW (read/compute/write) cycle requires 2.25 microseconds (for 2-microsecond CPU: 3 microseconds).

Eighth Edition (August 1971)

This is a reprint of GA24-3231-6 incorporating changes released in Technical Newsletter, GN24-0475, dated July 19, 1971. Changes are periodically made to the specifications herein: before using this publication in connection with the operation of IBM systems, refer to the latest SRL Newsletter, GN20-0360, for the 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.

This manual has been prepared by the IBM Systems Development Division, Product Publications, Dept. K10, P.O. Box 6, Endicott, N.Y. 13760. A form has been provided at the back of this publication for reader's comments.

Contents

Pages 35 through 43 in this publication are obsolete and have been replaced by *IBM System/360 Model 30 Operating Guide*, GA24-3373.

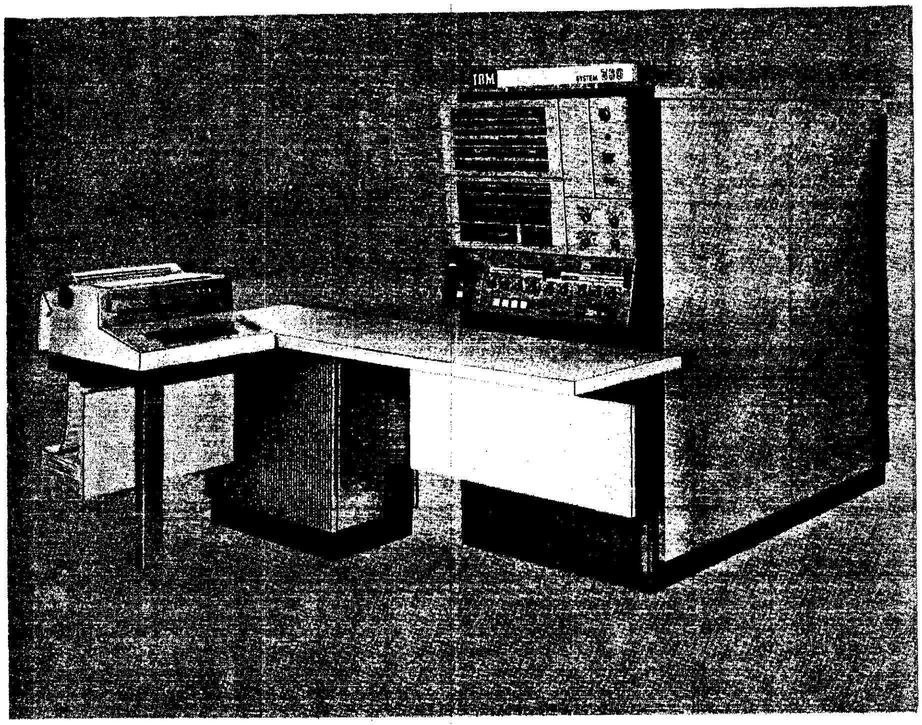


Figure 1. IBM 2030 Processing Unit with IBM 1050 Documentary Console Control

IBM System/360 Model 30, Functional Characteristics

Data processing systems provide rapid methods of processing the varied types of information used in today's commercial and scientific communities. Significant increases in the operating efficiency of many businesses have resulted from use of data processing systems. As the information-handling requirements of a business grow, expansion of data processing facilities becomes more and more desirable. As the advantages of a data processing system become evident (through use in initial applications), users apply this powerful tool to other areas of their business. Expansion of present operations and extension to new application areas are made easier when:

1. Methods of directing the system (programming) are modified rather than completely reworked.
2. Input/Output devices required in new applications can be connected to present equipment.
3. Storage capacity can be increased without the need for ordering an entirely new central processing unit.
4. Various input/output and communications devices that can be attached to the system are available.

These advantages are all offered in IBM System/360. Modest requirements can be met through use of an IBM System/360 Model 30 with 8,192 bytes of main storage.

Note: A *byte* is a position of storage in which eight bits of information can be stored, such as an alphabetic character, or two numeric characters.

A variety of input/output devices can be attached to this system. As data processing needs increase, storage capacity can be enlarged, and additional input/output devices can be attached. Modification of programs used on IBM System/360 is made easier because the instructions used for the smaller storage and I/O configurations apply to the larger-capacity configurations.

For example, programs used on different models of IBM System/360 can be run on an IBM System/360 Model 30 (within the limitations set forth in *IBM System/360 Principles of Operation,* Form A22-6821).

Data Processing System

Let's consider the over-all operation of a data processing system. The basic configuration of a data processing system is shown in Figure 2.

A program—series of instructions that control system operations—is contained in a storage area, which is usually in the central processing unit. The program is usually written in the following manner:

1. The job to be run is fully defined.
   Some typical questions that must be answered are:
   What types of results are required?
   What input/output devices are needed?
   What exceptional conditions can occur?
2. A pictorial block diagram (flow chart) is constructed to represent the job to be performed.
3. The programmer then writes detailed instructions in *symbolic* form. Note that the system performs

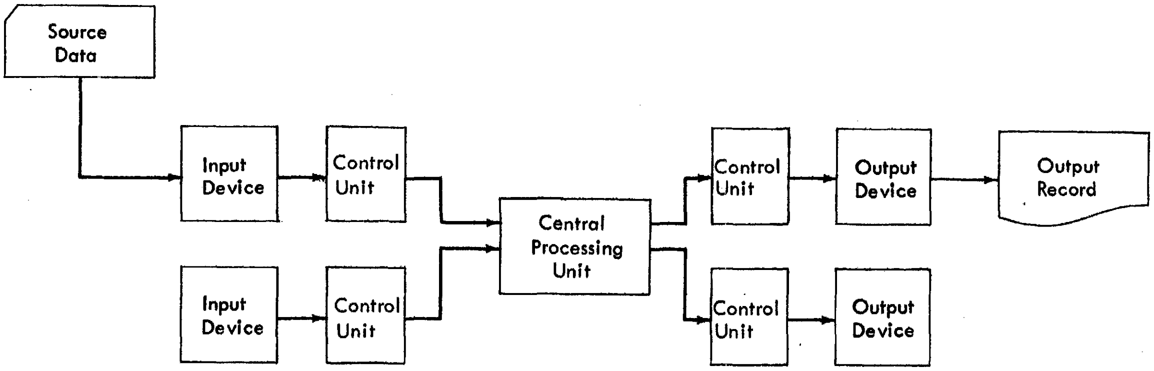


Figure 2. Basic Data Processing System

operations under control of *machine language* instructions. For the programmer, however, machine language instructions are more difficult to write than are symbolic instructions. Programs written in symbolic form are also much more easily modified than are actual machine language programs.

4. An assembly program, provided by IBM, is loaded into the system through an input device (such as a card reader). The symbolic program is then loaded in a similar manner. The assembly program transforms the symbolic instructions into machine language, producing an *object* program that is used in the actual job runs.

Once the object program is produced and loaded into storage, information (source data) that is related to the job (such as payroll data) can be entered into the system, and the job can be run. The source data is entered into an input device (Figure 2). An instruction in the program directs that this information be sent from the input device to the processing unit. The data is processed. That is, operations such as logical decisions and arithmetic functions are performed. Result data is then sent to one or more output devices where meaningful records (printed reports, punched cards, etc.) are produced. This phase of the operation is also under control of the program.

In System/360, input/output devices are controlled by control units (as shown in Figure 2).

IBM System/360 Model 30 Input/Output Control

Let's consider, in a general way, how IBM System/360 Model 30 controls the input/output (I/O) functions already described. A standard I/O interface is provided for the attachment of a wide variety of I/O devices to System/360. The *interface* is a standard set of electrical connections through which signals are sent that allow for exchange of data and control information between the CPU (Central Processing Unit) and attached I/O devices. The interface is called *standard* because any present or future device can be attached to the system as long as it complies with the specifications of the interface.

Recall that control of I/O functions in a data processing system is determined by instructions in the CPU program. In system/360 the transfer of data between CPU and I/O devices and control of the I/O devices is handled through channels. These channels coordinate the operation of I/O devices through program control.

There are two types of channels: *Selector* and *multiplexor*. One distinguishing feature of a selector channel (special feature on Model 30) is that it is used to sustain operation of one device at a time. A variety of devices can be attached to a selector channel, but it is most useful when used with high-speed I/O devices. For example, the high-speed transfer of information from a magnetic disk storage device to the CPU can be advantageously handled by a selector channel.

The data-handling capability of the multiplexor channel is *normally* lower than that of the selector channel, although the multiplexor channel on Model 30 can presently handle the same data rates of I/O devices attached to the selector channel (due to fastest I/O device presently attachable—170 kc). The advantage of the multiplexor channel is that it can service the data flow and control requirements of several slow-speed I/O devices at a time. This kind of operation is accomplished in *multiplex mode* (sometimes called data interleaved mode). For example, the data sent from two serial card readers to the CPU can be handled at one time in multiplex mode by the multiplexor channel.

The selector channel is usually used for high-speed data rate I/O devices because selector-channel data transfer operations *overlap* CPU operations while CPU facilities are *used* for multiplexor-channel data transfer operations.

Interruptions and I/O Control

To allow for the coordination of overlapped I/O operations and provide for a smooth flow of productive processing, it is necessary to provide a means of switching from one program to another. In the System/360, an interrupt system is provided for this purpose.

An *interrupt* causes a program to be suspended and allows some alternate operation to be started.

Let's follow the progress of a typical I/O operation as it occurs in the IBM System/360. Assume that instructions are being executed in the problem program state and a point is reached at which it is desired to perform an I/O operation. I/O operations are controlled by the *supervisor* program (sometimes called *control* program or *monitor* program). Therefore, it is necessary to get out of the problem program mode and into the supervisor program mode to start the I/O operation. A SUPERVISOR CALL instruction in the problem program effects this change. The I/O operation is then started from a series of instructions contained in the supervisor program. After this series of instructions has been executed, the supervisor program will return control to the problem program, which can continue processing while the I/O operation is taking place. Upon completion of the I/O operation, the problem program is interrupted so that the supervisor can determine whether any abnormal conditions were detected by the channel or control unit and take appropriate action. If the operation was completed successfully, control is again returned to the problem program so it can continue its processing.

Note that the instructions in the supervisor program are not part of the main problem program. Therefore a way must be provided to return to the right place in the problem program after the supervisor routine is completed. We could state this another way and say that the problem *program status* must be stored for use after the supervisor routine has been completed. PSW's (Program Status Words) are used to record this status information. When the SUPERVISOR CALL instruction is executed, the current PSW that relates to the problem program in progress is stored. It is replaced by a new supervisor call PSW. The new supervisor call PSW indicates (among other things) to the system that the system is operating in supervisor mode. When the supervisor routine is completed, the system is returned to the problem state by loading the problem program PSW that was previously stored. This new PSW then indicates (among other things) the instruction that is to be used next in the problem program. Actually a number of PSW's are used to shift system status for reasons other than entering or leaving supervisor mode for I/O operations. The operation just presented, however, describes their general function: one current PSW indicates the over-all active system state; stored PSW's retain, for future use, a record of the system state to which they apply.

I/O Control Units

I/O devices used in System/360 communicate with the channels through control units. The control unit provides the logical capability necessary to operate and control an I/O device and adapts the characteristics of each device to the standard form of control provided by the channel.

A control unit may be housed separately or it may be physically and logically integral with the I/O device.

From the user's point of view, most functions performed by the control unit can be merged with those performed by the I/O device.

Implementation of I/O Operations

In an I/O operation, as soon as the transition to the supervisor state is effected, the supervisor program takes over control of the operation. The program routines of the supervisor program are supplied by IBM (though they *may* be written by the user). Specifications, such as device designation, related to the I/O operation to be performed are written by the user and entered into the system.

Within the supervisor program is a START I/O instruction. This instruction indicates the channel and the I/O device on that channel that is to be used. Assume that the channel and device are free to perform the operation specified. Before this time a CAW (Channel Address Word) has been set up by the supervisor program in main storage locations 72-75. In this CAW is the main storage address of the first CCW (Channel Command Word) to be used in this operation.

Among the information contained in the CCW are:

1. The command to be sent to the control unit to which the desired I/O device is connected.
2. The area of main storage (if any) to be used in the operation.
3. The amount of data (number of bytes), if any, that is to be transferred in the operation.

Storage

Basic main storage capacity of IBM System/360 Model 30 is 8,192 bytes. Model 30 main storage capacity can be increased to 65,536 bytes. A *byte* of information is composed of eight bits plus a check bit. For the moment we can ignore the check bit and investigate the kinds of information that can be stored in a byte. The bit positions of a byte are numbered:

0 1 2 3 4 5 6 7

Each position can be at a logical 1 or a logical 0 state. This type of arrangement lends itself to a binary-number type of representation. When a bit position is a logical 1, the corresponding binary value is given. For example, the digit 2 can be represented in a byte by setting only bit position 6 to a logical 1:

Bit position	0	1	2	3	4	5	6	7
Bit position value	0	0	0	0	0	0	1	0
Binary value represented	128	64	32	16	8	4	2	1

This arrangement allows for up to 256 characters to be represented by one byte. In Model 30 each byte can contain one numeric digit (four low-order bits), two numeric digits (one digit in the four high-order bits, the other digit in the four low-order bits), or an alphabetic or special character (all eight bits are used).

When two numeric digits are contained in a byte we call this *packed decimal.* During arithmetic opera-

	BYTE								BYTE							
Represented Binary Value	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
Bit Value	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit Position	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7

Figure 3. Binary Representation of Address 8191

tions, such as addition, two bytes can be fetched from storage (one byte from each of the two fields to be added). Note that with access to only one byte from each of the two fields, two digits from each field are added. In packed decimal format, the sign is located in the low-order four bits of the low-order byte.

A byte can also contain status information. For example, a status byte sent to the channel from a control unit can indicate, by the bit values (0 or 1) of each bit in the byte, items such as:

1. The unit is busy executing a previous command.

2. A unit check has occurred. For example, a printer may be out of forms, or the stop key has been pressed.

3. A control-unit end indication has occurred. This means that the control unit has completed functions required of it in the operation.

Basic cycles in the IBM 2030 Processing Unit of System/360 Model 30 are either RW (Read/Write) or RCW (Read/Compute/Write). The RW cycle is used when information (one byte at a time) is read out of and back into storage. This cycle requires 1.5 microseconds. The RCW cycle has the additional compute function that is used in arithmetic operations, and the entire RCW cycle takes 2.25 microseconds. Note, however, that early systems have a 2-microsecond RW and a 3-microsecond RCW cycle. In this publication, timings related to a system with a 2-microsecond RW cycle, are shown in parentheses following the 1.5-microsecond system timing. For example:

The RCW cycle requires 2.25-microseconds (for 2-microsecond CPU, 3 microseconds).

Storage is binarily addressable up to 65,536 bytes in the Model 30. Suppose that it is desirable to address position 8,191. This address can be contained in two bytes as shown in Figure 3. You can determine the address represented in Figure 3 by adding the binary values that correspond to the bit positions that are set to 1:

$$\begin{array}{r}
4096\\
2048\\
1024\\
512\\
256\\
128\\
64\\
32\\
16\\
8\\
4\\
2\\
+\quad 1\\
\hline
8191
\end{array}$$

Although 16 bits are actually used in the storage address register in the Model 30, address lengths of up to 24 bits are allowed in order to maintain compatibility with other System/360 models.

Two general areas of storage are referenced in this publication: main storage and auxiliary storage. Problem programs are stored in main storage.

Auxiliary storage is composed of two storages: multiplexor storage and local storage. The multiplexor storage contains the unit control words used in multiplexor channel operations. Each unit control word maintains a record of the operation of the I/O unit to which that unit control word applies.

Local storage contains such items as the sixteen general-purpose and the four floating-point registers and certain other CPU working areas. Refer to Appendix B for further information on multiplexor and local storages.

Relationship to Other Models of the IBM System/360

The compatibility rule does not apply to detail functions for which neither frequency of occurrence nor usefulness of results warrants identical action in all models. In the following case, the operation of the Model 30 differs from that specified in the IBM System/360 Principles of Operation Form A22-6821.

If an error occurs on the execution of a CCW located at address FFF8 on a 64K Model 30, the updated CCW address stored in the CSW, is zero. If the Error Recovery Program subtracts 8 from this invalid address, a program check may result from attempting to use a maximum negative CCW address in recovery.

IBM System/360 Model 30, Characteristics

The IBM System/360 Model 30 is a basic model of the System/360. It offers extensive data processing and data-communication facilities with compatibility to other models of System/360, as set forth in *IBM System/360 Principles of Operation,* GA22-6821.

Figure 4 presents System/360 Model 30 characteristics (Figure 5 for 2-microsecond CPU). Figure 6 depicts the characteristics of I/O channels attached to the Model 30 (Figure 7 for 2-microsecond CPU). Notice in Figures 6 and 7 that the maximum number of I/O units attachable to the multiplexor channel is 224. CPU's with 8K main storage positions, however, handle a *maximum* of 32 I/O units on the multiplexor channel. CPU's with 16K or more main storage positions can handle a maximum of 96 I/O units on the multiplexor channel. However, an optional, no-cost feature can be used with Models E30 (32K) and F30 (64K) to increase the number of I/O units possible on the multiplexor channel to 224. For further information, refer to the *Input/Output Channels* section of this publication.

Characteristic	Speed (in microseconds)	Data Width Bits (Bytes)
Basic Machine Cycle	.75	—
Main Storage:	1.5	8 (1)
Model C30 --8192 Bytes		
Model D30 -- 16384 Bytes		
Model DC30 -- 24576 Bytes		
Model E30 -- 32768 Bytes		
Model F30 -- 65536 Bytes		
Registers Accessible to Programmer:		
Sixteen General Registers *	6	32 (4)
Four Floating-Point Registers *	12	64 (8) Double Precision
	6	32 (4) Single Precision
System Control: Read Only Storage (ROS)	.75	—

* These registers are in local storage (a storage area that is in addition to the main storage capacity).

Figure 4. IBM System/360 Model 30 CPU Characteristics

IBM 2030 Processing Unit

The IBM 2030 Processing Unit provides arithmetic, logic, and control functions for the IBM System/360 Model 30. A 1.5-microsecond storage cycle (2 microseconds on early system) is used to access main core storage, which is available in capacities of 8192, 16384, 24576, 32768, or 65536 bytes. The Central Processing Unit (CPU) has a one-byte- (eight-bit-) wide data flow. Controlling functions are implemented by a 0.75-microsecond (1 microsecond in early systems) Read-Only Storage (ROS).

Sixteen general registers and the four floating-point registers are contained in a separate area of core storage (local storage section of auxiliary storage). Standard features of the Model 30 include: the standard System/360 instruction set, 8192 bytes of core storage, the multiplexor channel, and the system control panel (console). The standard System/360 instruction set provides I/O instructions and the basic logical and fixed-point binary arithmetic instruction for manipulation of data.

Characteristic	Speed (in microseconds)	Data Width Bits (Bytes)
Basic Machine Cycle	1	-
Main Storage:	2	8 (1)
Model C30 -- 8192 bytes		
Model D30 -- 16384 bytes		
Model E30 -- 32768 bytes		
Model F30 -- 65536 bytes		
Registers Accessible to Programmer:		
Sixteen General Registers*	8	32 (4)
Four Floating Point Registers*	1o	64 (8) double precision
	8	32 (4) single precision
System Control: Read Only Storage (ROS)	1	-

* These registers are in local storage (a storage area that is in addition to the main storage capacity).

Figure 5. IBM System/360 Model 30 CPU Characteristics (for CPU with a 2-Microsecond RW Cycle)

Channel	Maximum Data Rates (in kilobytes per second)	Maximum Control Units per Channel @	Maximum I/O Units per Channel %
Selector Channel 1	275 *	8	256
Selector Channel 2	275 ‡ * 180 #	8	256
Multiplexor Channel Multiplex Mode	31	8	224
Burst Mode (Non-Interleaved)	180		

Notes:

* Depends upon other operations and data rates of attached devices. For control units or devices that have sufficiently fast interface response times, higher channel rates may be achieved (due to the theoretical maximum rate of 333kb on a selector channel). Multiplexor channel operation limits the maximum aggregate rate for the two selector channels to 500kb. Also multiplexor channel operations are affected by selector channel operations.

@ The term control unit, as used here, is a device requiring its own cable connection to a standard I/O interface.

% Depends upon size of main storage. Can be up to 224 units on the multiplexor channel only in models E30 and F30.

‡ The maximum rate for selector channel two, if selector channel one is not running, is the same as selector channel one.

# The maximum rate for selector channel two if selector channel one is running at 333kb.

Figure 6. IBM System/360 Model 30 Channel Characteristics

The multiplexor channel has 32 subchannels (in 8K CPU) that permit the simultaneous operation of up to 32 slow-speed serial I/O devices in multiplex mode, or the operation of a single higher-speed device, with no multiplexing in burst mode.

IBM 2030 Processing Unit Special Features

These features are explained in more detail later in this publication or in *IBM System/360 Principles of Operation,* Form A22-6821.

Instruction Sets

Three instruction sets are available for the Model 30. The *standard set* (not a special feature) plus the *decimal feature* compose the *commercial set,* which provides for the decimal arithmetic and editing facilities important in many business applications.

The *standard set* plus the *floating-point arithmetic feature* become the *scientific set,* which provides the additional computing power required for many scientific and statistical applications.

The *universal set,* composed of the *standard set* plus the *decimal and floating-point arithmetic,* and *storage protection* (two instructions) special features, is provided for the user having both commercial and scientific applications.

Figure 8 illustrates how these sets relate to each other.

IBM 1401/1440/1460 Compatibility

With the 1401, 1440, 1460 Compatibility special feature, the Model 30 can execute 1401, 1440, 1460, or System/360 instructions. This feature reduces or eliminates the reprogramming effort necessary for application conversion from the IBM 1401, 1440, or 1460 to the System/360, Model 30. Refer to *IBM System/360 Model 30 1401/1440/1460 Compatibility Feature,* GA 24-3255, for a further description.

Channel	Maximum Data Rates (in kilobytes per second)	Maximum Control Units per Channel @	Maximum I/O Units per Channel %
Selector Channel 1	250 *	8	256
Selector Channel 2	250 + 180 #	8	256
Multiplexor Channel Multiplex Mode	23.8	8	224
Burst Mode (Non-Interleaved)	180		

Notes:

* Depends upon other operations and data rates of attached devices. For control units or devices that have sufficiently fast response times, higher channel rates may be achieved (due to the theoretical maximum rate of 275kb on a selector channel). Multiplexor channel operation limits the maximum aggregate rate for the two selector channels to 375kb. Also, multiplexor channel operations are affected by selector channel operations

@ The term <u>control unit</u>, as used here, is a device requiring its own cable connection to a standard I/O interface.

% Depends upon size of main storage. Can be up to 224 units on the multiplexor channel only in models E30 and F30.

+ The maximum rate for selector channel two, if selector channel one is not running, is the same as selector channel one.

# The maximum rate for selector channel two if selector channel one is running at 250kb and the multiplexor channel is not running.

Figure 7. IBM System/360 Model 30 Channel Characteristics (for a CPU with a 2-Microsecond RW Cycle)

IBM 1620 Compatibility

With this feature, the IBM System/360 Model 30 can execute 1620 or System/360 instructions. Therefore, the 1620 Compatibility feature reduces or eliminates the reprogramming effort necessary for application conversion from the IBM 1620 to the IBM System/360 Model 30. Refer to *IBM System/360 Model 30 1620 Compatibility Feature,* GA24-3365, for a further description.

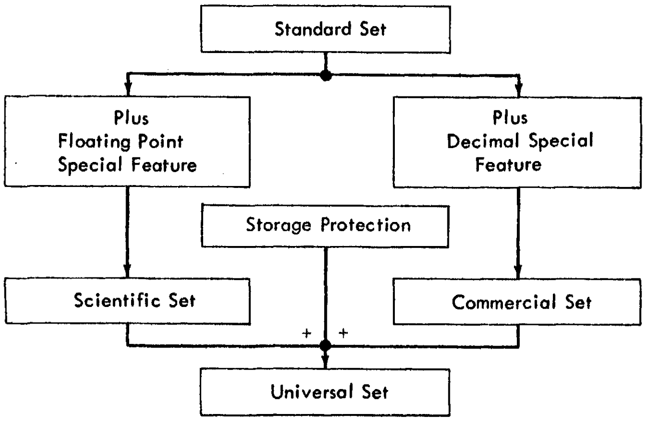


Figure 8. IBM System/360 Model 30 Instruction Sets

Storage Protection

This feature provides the ability to protect storage in blocks of 2048 bytes as specified by the programmer. With this feature, a programmed protection key prevents the writing of data into a protected area of core storage, thereby preventing one program from destroying another. Protection against reading data from an I/O device into a protected area is also provided. Protection patterns or keys are established in the supervisor mode.

Interval Timer

This feature provides the ability to decrement a program-controlled count at a fixed rate. Automatic program interruption occurs when the count passes through zero.

This feature can be used for job accounting by measuring the duration of time for each job, for interrupt to prevent a run-away job from gaining control of the system, for time stamping, and for polling a communication network on a regular basis (for example, every minute, every half-hour, etc.).

Direct Control

Control of a variety of special equipment (including non-IBM devices) is possible with this feature. Two instructions, READ DIRECT and WRITE DIRECT, and six distinct interrupt lines are included with this feature. Direct control is used to pass controlling and synchronizing information between the CPU and the special external devices. Data transfers are normally handled over the multiplexor or selector channels.

Since the external interrupt feature functions are included in this feature, the external interrupt feature cannot be installed.

External Interrupt

The external interrupt feature provides fast program response for time dependent operations for which the data-transfer function is already provided. This feature has six distinct interrupt lines.

If the direct control feature is installed, then this feature cannot be installed.

Channel-to-Channel Adapter

To interconnect two System/360 channels, using one control-unit position on each of the connected channels, only one channel-to-channel adapter is required. This feature allows data transfers between two System/360 processing units.

Selector Channels

Selector channels permit high-speed I/O operations to be overlapped with processing. Eight control-unit positions are provided for each selector channel. Up to two selector channels can be attached. See the Input/Output Channels section of this publication for operation considerations.

IBM 1051 Attachment

This feature provides for the direct attachment of IBM 1050 system devices to the IBM 2030 Processing Unit. See the *IBM 1050 Documentary Console* section of this publication for further details.

Basic data flow for Model 30 is shown in Figure 9.

The IBM 2030 Processing Unit contains all CPU features, the multiplexor and selector channels, core storage, and CPU power supplies.

Basic elements of the data flow, such as the instruction counter, storage address register, and instruction register, are employed in the conventional manner. The instruction counter provides the core storage address of the next sequential instruction to be executed. The storage address register receives address information from several sources and causes the proper storage location to be accessed during processing or input/output operations. The instruction register holds each instruction as it is fetched and while it is interpreted for execution.

Core storage is integrated with the CPU in the Model 30. The Model 30 uses a 1.5-microsecond read/write (RW) cycle (2 microseconds in early systems) in which one eight-bit byte is accessed from main storage. This RW cycle can be expanded, permitting logical and compute operations to occur on a byte of data during a single access time. This expanded read/compute/write (RCW) cycle time is 2.25 microseconds (3 microseconds in a CPU with a 2-microsecond RW cycle).

In addition to the main storage that is used to contain instructions and data, the System/360 Model 30 makes use of additional locations of core storage to accomplish its functions. Local storage is used to contain general-purpose registers, floating-point registers, and additional scratch-pad areas for the CPU. Multiplexor storage is used to contain information pertaining to the subchannels of the multiplexor channel. Figure 10 shows the number of bytes of main storage, local storage, and multiplexor storage, and the number of subchannels provided in each model of the 2030 processing unit. Although the local storage and multiplexor storage are physically built as part of main storage, they are separately addressed and are not directly available to the programmer. (Refer to Appendix B.)

The additional Multiplexor Subchannels feature can be used only on Models E30 and F30 to increase the

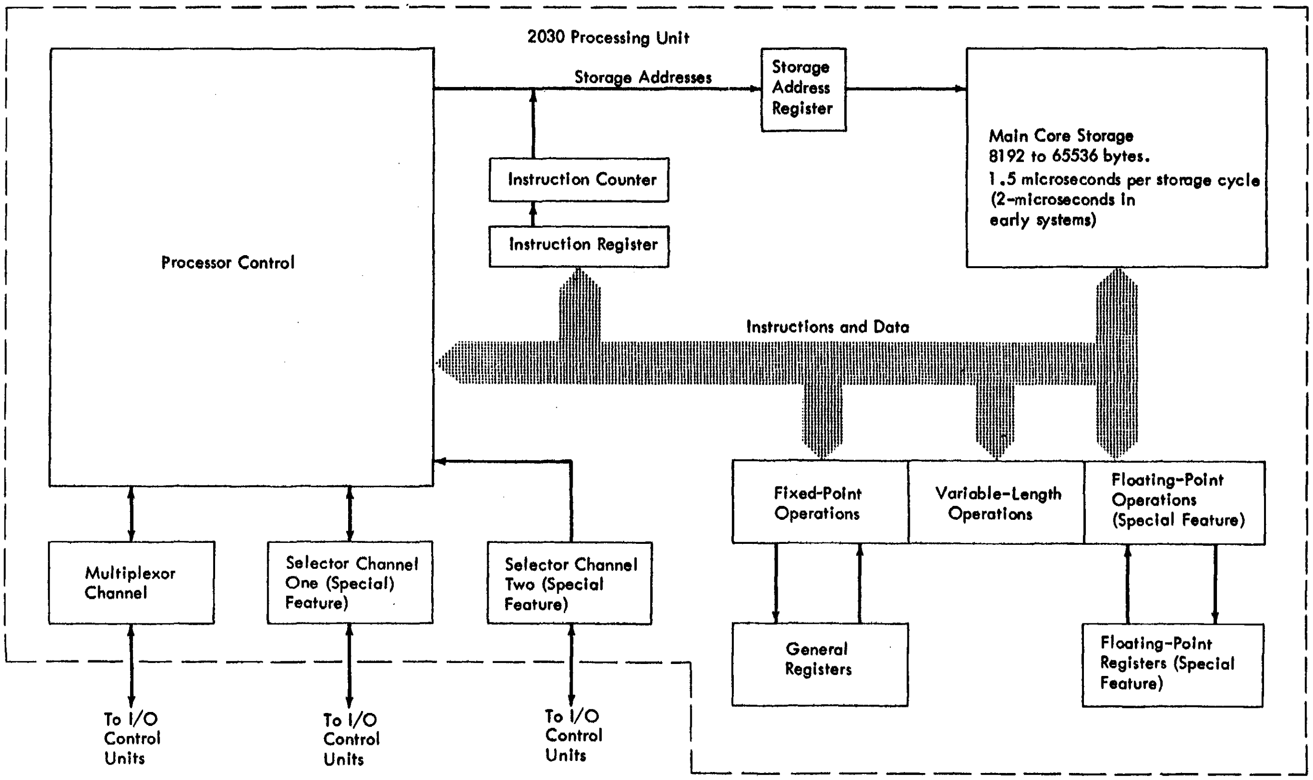


Figure 9. Basic Data Flow

System/360 Model	Main Storage (Bytes)	Local Storage (Bytes)	Multiplexor Storage (Bytes)	Subchannels
C30	8,192	256	256	32
D30	16,384	256	768	96
DC30	24,576	256	768	96
E30	32,768	256	768	96
F30	65,536	256	768	96

Figure 10. IBM System/360 Model 30 Storage Allocations

number of subchannels to 224. This feature makes it possible to operate a large number of low-speed communication lines and terminals. Control units that require shared-path subchannels (for example, tape drives and disk storage units) cannot be attached to the multiplexor channel when this feature is installed. This feature is an optional, no-charge item for Models E30 and F30.

The 16 general registers have multiple functions and are key elements in address manipulations in all models of the System/360. These registers are used:

- As fixed-point binary accumulators,
- As indexing registers,
- As sources for addresses, and
- For shifting and logical operations.

The four floating-point registers are utilized in all floating-point ^ ithmetic operations.

The input/output channels of the Model 30 are completely integrated with the CPU. The Model 30 provides up to three I/O channels: two selector and one multiplexor. The selector channels operate in an overlapped fashion with the CPU program, accessing storage for one RW cycle for each byte of data transferred. The multiplexor channel operates in either *burst mode,* in which one I/O device requires the full capacity of the channel, or *in multiplex mode,* in which several I/O operations are sustained on several subchannels of the channel.

The 2030 processing unit data flow utilizes eight-bit registers, which are connected to a system of data transfer busses, and an Arithmetic Logic Unit (ALU) capable of handling two 8-bit inputs at a time. During packed-decimal arithmetic operations, for example, two pairs of decimal digits from the two data fields can be operated upon at the same time yielding a two-digit result byte for each cycle of the operation. In packed-decimal operations, the sign is contained in the four low-order bits of the low-order byte.

The basic data-flow registers have the following designations and functions (Figure 11):

Name	*Usual Function*
I, J, U, V, and T	Address registers.
M, N	Main storage address registers.
L	Holds length of data field.
D	General-purpose data register.
R	Storage data register.
G and S	Status registers: retain machine conditions and status for testing by the microprogram control routines.
H	Priority status register.
A	A-entry to the ALU.
B	B-entry to the ALU.
F	External interrupt register (console interrupt section standard).
Q	Storage protection key (special feature).
W, X	Read-only-storage address registers.
FW, FX	Read-only-storage back-up address registers for multiplexor channel break-in.
GW, GX	Read-only-storage address registers for either selector channel (special feature) operations.

These registers, the arithmetic and logic unit, and the data-transfer busses are manipulated by the control element, the read-only-storage microprogram (part of the circuitry and not written by the problem programmer), during instruction execution.

Additional registers are added to the bus system when the associated special features are installed.

The control functions in the Model 30 are executed by Read-Only Storage (ROS), which contains sequences of micro-instructions. These micro-instructions control the sequence of data manipulations through the Model 30 data flow, and to and from core storage. Each CPU instruction is interpreted and executed by a series of micro-instructions accessed from the read-only-storage unit. Functions such as indexing of storage address are executed as part of these sequences.

Note: The programmer need not concern himself with the read-only-storage microprogram.

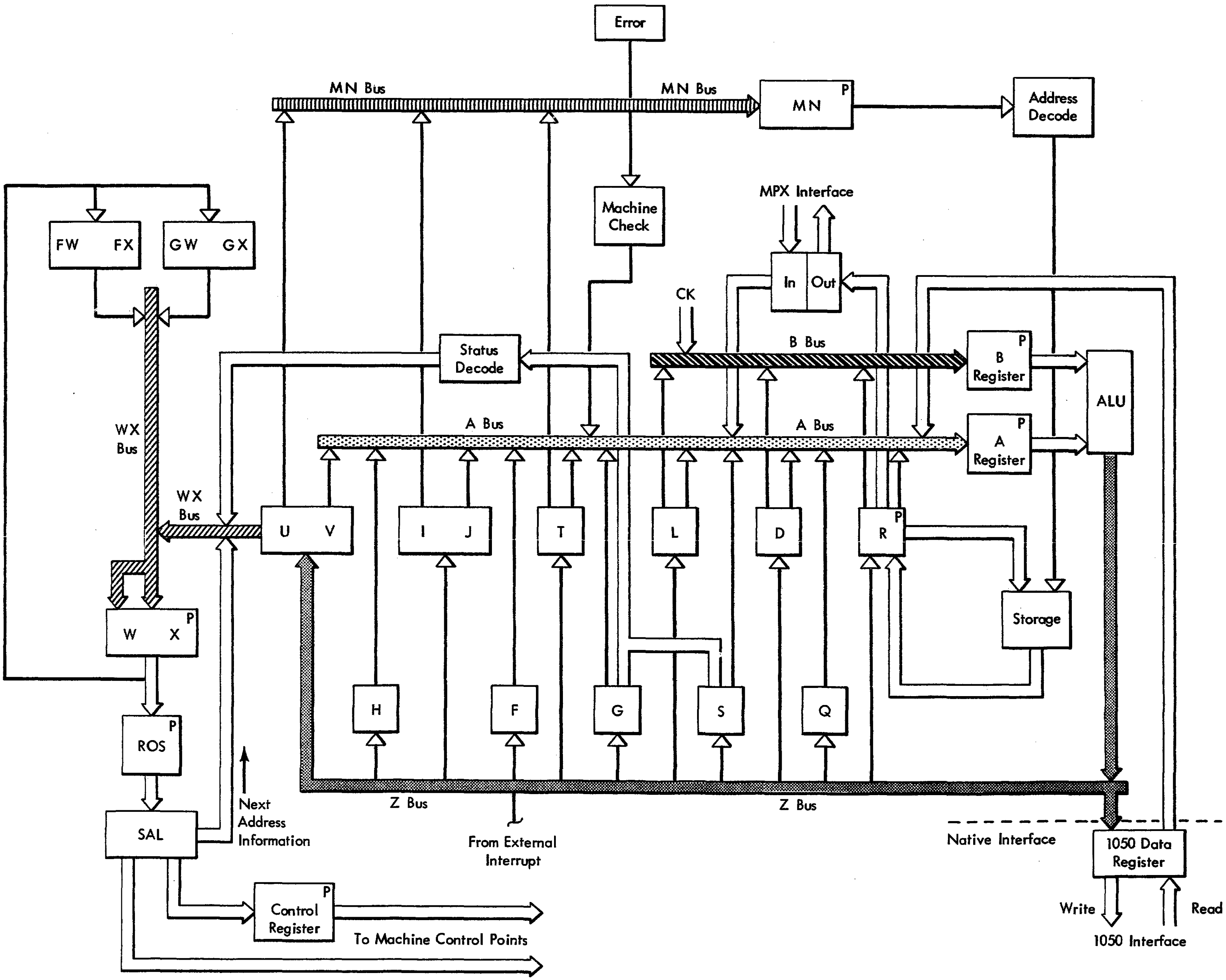


Figure 11. IBM 2030 Data Flow

Input/Output Devices

See *IBM System/360 Model 30, Configurator,* GA24-3232, and *IBM System/360 Input/Output Configurators,* GA22-6823, to determine what I/O devices are available.

Input/Output Channels

The IBM System/360 can have two types of input/output channels: *multiplexor* and *selector.*

Operations on the multiplexor channel are performed either in burst mode for high-speed I/O devices, or in multiplex mode for slower-speed I/O devices (Figure 12).

When operating in the burst mode, a single I/O device captures the multiplexor channel and does not release it from the time it is selected until the last byte has been serviced.

In the multiplex mode, the single data path of the channel can be time-shared by a large number of low-speed I/O devices operating simultaneously. The channel multiplexes data to or from them, on demand in groups of bytes as determined and specified by the particular I/O device being serviced.

The multiplexor channel allows attachment of up to eight control units, with provision to control up to 96 I/O devices (32 with 8K CPU) simultaneously.

Refer to Figures 6 and 7 for multiplexor channel data rates.

At the user's option (no cost), the number of multiplexor subchannels (in Models E30 and F30 only) can be increased to 224. This expansion makes it possible tó operate a large number of low-speed communication lines and terminals. However, control units that require shared-path subchannels (for example, tape drives and disk storage units) cannot be attached to the multiplexor channel when this expansion to 224 subchannels is installed in the system.

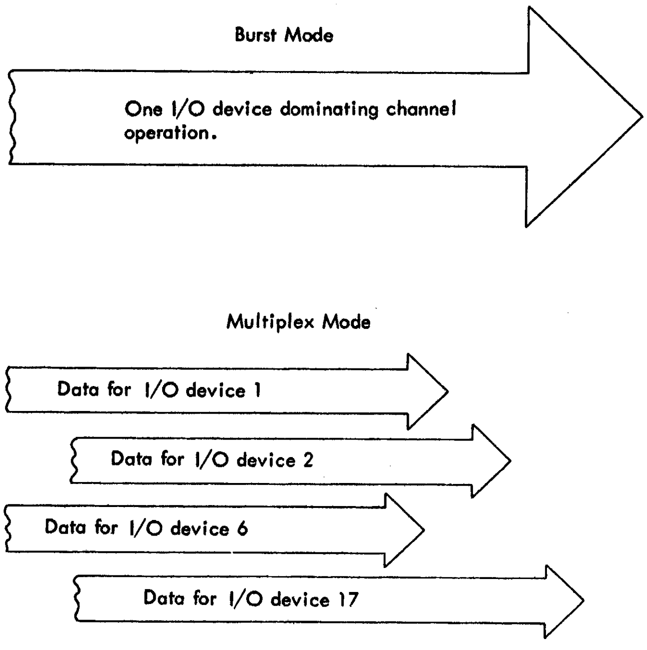


Figure 12. Channel Data Modes

The two selector channels (each one a special feature) are capable of handling high-speed I/O units. The data flow from these units is overlapped with processing cycles. Each selector channel can accommodate up to eight control units and can address up to 256 I/O units. These channels operate only in burst mode. Only one I/O unit per selector channel can be engaged in the transfer of data at one time. For selector-channel data rates, refer to Figures 6 and 7. Both the multiplexor channel and the selector channel operate from the same instruction and command formats used for all System/360 models. The START I/O instruction is used to initiate command sequences to be performed by the channel and to specify the channel and I/O unit to which the sequence applies.

The formats of the START I/O instruction and the Channel Command Words (CCW), used to cause the performance of I/O functions by the channels, are described in *IBM System/360 Principles of Operation,* Form A22-6821. Further details, pertaining to particular I/O devices, can be found in the publications describing those devices. Refer to *IBM System/360 Bibliography,* GA22-6822, for a listing of the publications describing these devices.

The commands: READ, READ BACKWARD, WRITE, CONTROL, and SENSE are handled by the channels and directed to the units attached. The command TRANSFER IN CHANNEL (TIC) is handled as an internal channel function.

The three other CPU I/O instructions implemented for the multiplexor and selector channels are: TEST I/O, HALT I/O, and TEST CHANNEL. Also, the storage protection function (special feature) is provided in the I/O area as well as for CPU main storage.

Channel Restrictions on the Model 30

1. Command chaining must be used for operation of all I/O devices attached to the IBM 2841 Storage Control (IBM 2302 Disk Storage, IBM 2311 Disk Storage, IBM 2321 Data Cell, IBM 2314 Direct Access Storage Facility, or IBM 7320 Drum Storage). If desired, command chaining can be used with any IBM System/360 Model 30 I/O device.

2. Initiation of a burst-mode operation on the standard multiplexor channel prevents any further instruction execution until the data transfer is satisfactorily completed.

3. Execution of instructions is suspended until the completion of data transfers for buffered devices (such as 1443, 1445, and devices controlled by the 2821) attached to the multiplexor channel.

4. Data chaining cannot be used with I/O devices that transfer data to the CPU before the required time for the data-chaining operation can be completed (for example, a 90kc device).

5. Assume that an IBM 2841 Storage Control is attached to one channel and an IBM 2841 Storage Control is attached to a second channel. If devices attached to these control units are time-dependent upon command chaining, then simultaneous operation of these devices on both channels may result in overrun and the operation will then have to be repeated.

6. Calculation of channel loading values for System/360 Model 30 is described in *IBM System/360 Model 30 Channel Characteristics and Functional Evaluation*, GA24-3411.

Multiplexor Channel

General Organization and Operation

The multiplexor channel of the Model 30 is composed of 32 subchannels (96 in a system with 16K or more storage position) each of which is capable of controlling one input/output operation concurrent with other subchannel operations, selector channel operations, and CPU processing. The multiplexor channel is shown conceptually in Figure 13.

Each subchannel has its functional control elements contained in its Unit Control Word (UCW). These are contained in MPX (multiplexor) storage. The CPU data flow registers and special read-only-storage, microprogram routines are utilized to perform the multiplexor channel operations. These operations consist of updating the main storage address to or from which data is moved, decrementing the count for each byte transferred, and of appropriate ending routines (such as storing of status). Each UCW contains 64 bits of information.

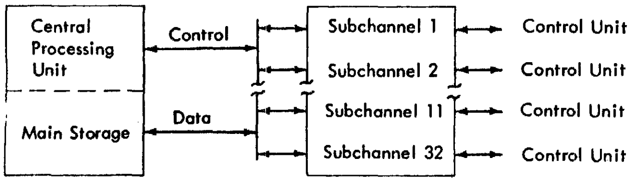


Figure 13. Conceptual Multiplexor Channel Organization

As each device requests service, its unit control word is brought out to the channel, used, updated, and restored when the required operation has been completed. Hence, a running record of the operation being performed by each operating device is stored in the UCW that pertains to that I/O device.

Each request for service for an I/O unit causes a suspension in CPU processing at the end of any RW or RCW cycle, and initiates a series of microprogram steps which:

1. Cause the CPU data flow registers to be stored (temporarily) in local storage,

2. Cause the proper subchannel information to be read out from the appropriate UCW location in the MPX storage,

3. Effect the data-access cycle and the updating of the UCW count and data address fields,

4. Restore the updated UCW into the MPX storage,

5. Re-establish the CPU registers from temporary storage, and

6. Resume the CPU processing routine.

Note: This multiplexor *share cycle* requires a total of 62.25 microseconds (steps 1 through 6). In a 2-microsecond CPU, the time is 83 microseconds.

Step 3 is immediately repeated if another request for data service is detected from the same unit. This multi-byte process continues until the device has been satisfactorily serviced. This may approach full burst mode, depending on the device. If, upon completion of step 4, another request for a share cycle is outstanding, step 5 is eliminated, and the loop is closed and begun at step 2. This affords a faster servicing loop, eliminating CPU store and restore cycles.

When the Model 30 operates the multiplexor channel in the burst mode, no instruction execution occurs. Selector-channel operations already in progress continue with the data-access cycles overlapped with the multiplexor channel burst operations.

Multi-Byte and Burst Modes on the Multiplexor Channel

The multiplexor channel is designed so that multi-byte bursts of data can be transferred between main storage and I/O units designed to operate in this fashion. After the transfer of the first byte, a minimum of 3.75 microseconds per byte (5 microseconds in a CPU with a 2-microsecond RW cycle) is added to the share cycle time for the particular I/O unit. The byte time to be added is actually the data-transfer-cycle rate of the device. This permits the multiplexor channel to operate (like a selector channel) at a *maximum* data-handling rate of 267kc (200kc in a CPU with a 2-microsecond RW cycle) or to transfer data in short bursts to prevent a device from monopolizing the channel. This multi-byte technique can be advantageously used when, for example, the IBM 2821 Control Unit is attached to the channel together with other units that require the multiplex mode of operation.

Priority of Device Operation

The priority with which I/O service requests are processed depends upon the sequence of the device attachments to the standard interface cable of the multiplexor channel. This is illustrated in Figures 14 and 15.

In Figure 14, the priority for acknowledgment of service requests is C, B, A. That is, unit C is serviced first in case of simultaneous requests from either of the other two units. In Figure 15, the priority is B, C, A, because unit B is now the first unit encountered on the select signal line. The assignment of this priority is done at installation time.

The method used to service a device on the multiplexor channel requires that, in order to obtain maximum throughput, devices must be attached to the channel in a certain sequence.

When a device is ready to send or receive a byte of data, then that device raises a request-in line in the interface cable. The channel then sends the select signal over the interface cable. This signal is sent serially to each control unit connection on the interface cable. Each control unit, if it does not require service, allows the select signal to proceed to the next control unit. When the select signal reaches a control unit that requires service, further propagation of the select signal is blocked; that is, it does not proceed to any other control unit on the cable (see Figures 14 and 15). The control unit at which the signal was stopped then secures temporary use of the interface data and signal paths so thât data or status information can be transferred. After the data or status information has been handled, the interface becomes available to the other units. Polling starts again when another request for service is recognized. If simultaneous requests for two units occur, the one closer (electrically) to the channel is serviced first. Therefore, priority of servicing is in the same order as the physical positioning of the control units on the select line of the I/O interface. An I/O interface connection can be used by a single control unit and attached I/O unit, a single control unit controlling multiple I/O units (one at a time), or multiple control units each of which controls one I/O device. (Refer to the *Control Unit Connection to Standard I/O Interface* section.)

Priority of servicing devices on the channel is determined by the following basic considerations:

1. Buffered units can be delayed without losing data. The information is contained in the buffer until needed and, hence, operation does not depend upon the speed of operation of the actual device that is buffered. For example, a 1403 print cycle does not affect the movement of data from main storage to the 1403 buffer. Once the 1403 buffer is loaded, the transfer of data is complete as far as the channel is concerned.

2. Synchronous unbuffered devices, once started, cannot be delayed without loss of data. For example, suppose that a serial unbuffered card reader has started a read cycle. Because the data read from the card is unbuffered, the channel must accept that data as it is read.

3. Asynchronous unbuffered units operate, on demand, for each character cycle. Examples here are the 1442 punch and 1050 documentary console units. These units do not overrun, but they lose speed if not serviced at their maximum speed.

Because synchronous unbuffered devices cannot be delayed, they must be placed first, in order of priority, on the channel. If simultaneous requests occur, from both a buffered and a synchronous unbuffered device,

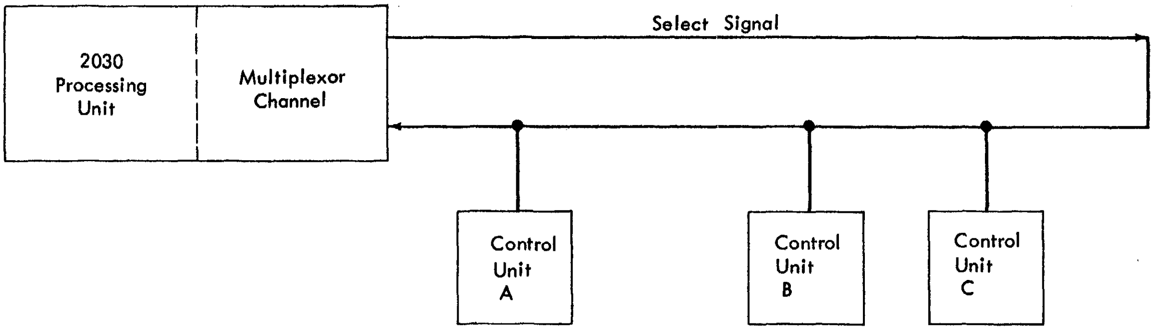


Figure 14. Priority Servicing

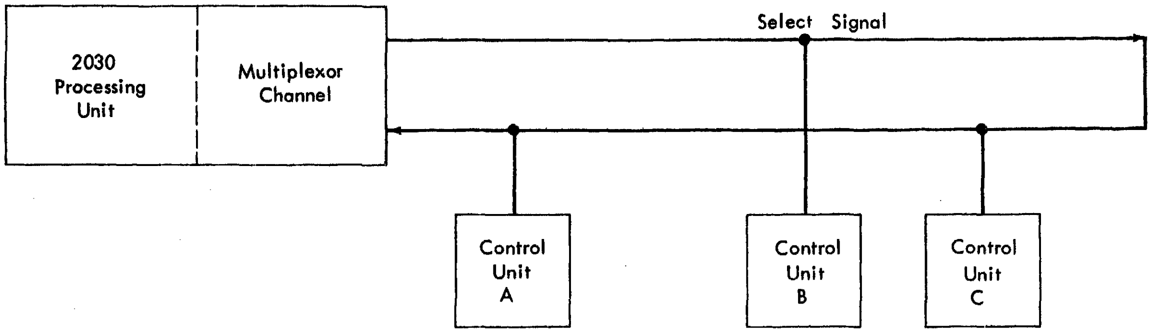


Figure 15. Priority Servicing

then the select signal is stopped by the unbuffered device before it reaches the buffered device. The unbuffered device can then proceed and not have to wait for a possible buffered device operation to be completed. This is desirable because the synchronous unbuffered device *must* have access to the channel when it has a data byte ready for transfer.

Also, higher-speed synchronous unbuffered devices require priority over slower-speed synchronous unbuffered devices. The higher-speed unbuffered devices have a shorter time in which they can wait before they *must* transfer a byte of data. Therefore, the higher-speed synchronous unbuffered devices should be positioned first, in priority, within the synchronous unbuffered device group.

A further condition must be considered for the buffered device group because there are basically two kinds of devices in the buffered group:

1. Asynchronous
2. Synchronous

An example of an asynchronous device is the 1403 printer. For the 1403 printer, there is a specific waiting time before printing can begin. If loading of the buffer is completed after the waiting time, then printing can begin. Asynchronous buffered devices should be placed last in priority within the buffered group.

On the other hand, synchronous buffered devices should be placed first, in order of priority, within the buffered group. An example here is the IBM 2540 Card Reader. In 2540 reader operations, card feeding is under control of mechanical clutches, whose engaging points are ⅓ cycle apart. Assume that card-feed cycle is in progress in the read feed. Toward the end of the cycle, at a predetermined point (that is, device end), the 2540 reader can receive another command to read so that the contents of its buffer will be sent to the channel. If, for some reason, this buffer transmission is delayed past the next normal clutch point, the clutching mechanism cannot be engaged until the following clutching point (⅓ cycle away, in time). In this case, card reading speed drops to ⅔ of full speed. Therefore, this synchronous buffered device should be placed in a position of higher priority than asynchronous buffered devices.

The IBM 2520 Card Read-Punch is handled somewhat differently than other devices. Card reading in the 2520 is synchronously unbuffered and serial, although card punching is buffered. Therefore, if only one 2520 is used, and both reading and punching are performed, then that 2520 must be placed between the serial unbuffered devices and the buffered devices.

If however, more than one 2520 is used and all 2520's are punching *and* reading, then the unit of highest priority (the card reader, which is unbuffered) should be started first (assume that the punch buffer is already loaded) and *then* the second card punch should be started. This situation is rare, however, because it implies that more than one 2520 is used in a punch-back operation (that is, punching is being done in the card previously read).

If the 2520 is only reading or only punching, it can be placed in the channel priority arrangement according to the rules for I/O priority already described.

Summary of Priority Attachment

Devices should be connected to the multiplexor channel interface cable in the following order of descending priority for most efficient operation:

1. *Synchronous unbuffered serial devices* (for example, 2501 and 2702). Within this group, the devices should be placed in descending priority order of their waiting time intervals. The devices with shortest waiting time are placed first within this group.
2. *Synchronous buffered units* (e.g., 2540 and 2520). Within this group, devices with the shortest waiting time should be placed first.
3. *Asynchronous buffered units* (e.g., 1443 and 1403). Within this group, devices with the shortest waiting time should be placed first.

4. Burst-mode devices which are time-dependent on command chaining (e.g., 2311).

5. Other burst-mode devices.

6. Asynchronous unbuffered devices (e.g., 1050 documentary console).

As a special exception to this priority attachment, the channel-to-channel adapter is placed, by its design, at the highest priority position on the cable even though it is a burst-mode device.

The 1050 console does not count as one of the eight possible control unit attachments. However, it can be given either the highest or lowest priority to all other possible device attachments. It is suggested that it be given the lowest priority such that its adapter will not delay the *select* signal propagation in command chaining some other unit on the cable.

For more detailed considerations of priority attachments of I/O units, refer to *IBM System/360 Model 30 Channel Characteristics and Functional Evaluation,* GA24-3411.

Control-Unit Connection to Standard I/O Interface

Up to eight control units can be connected to a channel in System/360 Model 30. However, connection of control units to the I/O interface can be thought of in one of three ways, depending upon the units involved:

1. A single control unit that controls *one* I/O unit can be connected to the channel. An example is the IBM 1443 Printer. The control unit for the 1443 is contained in the 1443 itself. When the 1443 is connected to a channel, one of the eight I/O interface positions is used.

2. A single control unit that services the requirements of *several* I/O units (one at a time) can be connected to the channel. An example of this arrangement is an IBM 2841 Storage Control Unit and attached direct-access storage units. The 2841 is a single control unit that provides for servicing of only one of its attached I/O units at a time. (Up to eight access mechanisms can be attached to the 2841). Again, the 2841 requires use of one of the eight positions on the I/O interface. Here, however, multiple units are serviced (one at a time) by the control unit.

3. The third situation occurs when *several* control units are contained in one unit that is itself called a control unit. Each of the separate control units services one I/O device. An example is the IBM 2821 Control Unit, which concurrently services IBM 1403 or 1404 Printers, and the IBM 2540 Card Read-Punch. Each of the attached I/O devices has its own control unit contained in the 2821. The 2821 uses only one of the eight positions on the I/O interface.

Note that while not more than eight interface attachments can be made to a channel, the number of I/O units attached can be greater than eight. For example, one interface adapter is used to attach a 2841 control unit which can control up to eight access mechanisms.

Also, note that, as a special case, the IBM 1050 Documentary Console devices do *not* use up one of the eight possible connections to the channel.

Unit Addressing Method

Before a command can be sent to the control unit of a device, the device must be addressed. The address is derived from a START I/O instruction and consists of 11 bits of information. The channel's address is contained in the three high-order bits, and the address of the device is in the eight low-order bits, as follows:

	Channel Address	*Device Address*
Bit position	0 1 2	3 4 5 6 7 8 9 10

Notice here that up to 256 different addresses can be developed in the eight low-order bits. That is, the addresses in the eight low-order bits can range from:

Binary Value	*Decimal Value*
00000000	0
to	to
11111111	255

Any of the 256 addresses can be used to designate a device on a selector channel. In IBM System/360 Model 30 there can be a maximum of two selector channels (special features). The two selector channel addresses are:

Selector Channel	*Channel Address*
1	001
2	010

Therefore, the 11-bit addresses used with selector channel devices have the following ranges:

Selector Channel One Addresses	*Selector Channel Two Addresses*
001 0000 0000	010 0000 0000
to	to
001 1111 1111	010 1111 1111

Recall that up to eight connections can be made to the standard I/O interface. This restriction, however, does *not* limit, to eight, the number of attached I/O units. For clarification of this subject, refer to the *Control Unit Connection to Standard I/O Interface* section of this publication.

A different situation exists for device addressing on the multiplexor channel. Device addressing on the multiplexor channel is dependent upon how the device and its control unit operate with the channel. For devices that have exclusive use of a control unit, the high-order bit of the unit address is set to a value of zero. Examples of such devices are:

1. *The 1443 printer.* The control unit for the 1443 printer is in the 1443 printer and it is for the exclusive use of the 1443.
2. *The 1403 printers and 2540 card read-punch.* Each 1403 printer, attached to a 2821 control unit, has a control unit (for its exclusive use) contained in the 2821. If three 1403 printers are attached to a 2821, then there are three separate control units in the 2821, one for each 1403. If a 2540 is attached to the same 2821, then the 2540 reader and 2540 punch each has its own separate control unit contained in the 2821.

The high-order bit of the unit address also has a value of zero for devices that operate simultaneously, but all of which use the same control unit. This situation occurs for a communications multiplexor, which controls a number of terminals that are operating simultaneously.

For these two kinds of devices (those that have exclusive use of a control unit or those that operate simultaneously with a single control unit), the device addresses range, in a System/360 Model 30 with 8192 positions of main storage, from:

Binary Address	*Decimal Value*
0000 0000	0
to	to
0001 1111	31

The restriction of 32 device addresses (maximum) is caused by the amount of special storage (multiplexor storage) available for channel use in a Model 30 with 8,192 positions of main storage.

All other System/360 Model 30 processing units have at least 16,384 positions of main storage. These other models have 96 unit addresses available for use by this category of unit on the multiplexor channel. Hence unit addresses (with a high-order bit value of zero) in a Model 30 with 16,384 or more positions of main storage range from:

Binary Address	*Decimal Value*
0000 0000	0
to	to
0101 1111	95

The multiplexor channel address (three high-order bits of the 11-bit address) is 000. Hence, the valid ranges for the addresses (with high-order bit of unit address set to a zero value) are:

Processing Unit Main-Storage Positions	*Multiplexor Channel Valid Addresses*
8192	000 0000 0000 *to* 000 0001 1111
16384 (or more)	000 0000 0000 *to* 000 0101 1111

Consequently, the following addresses are invalid:

Processing Unit Main-Storage Positions	*Multiplexor Channel Invalid Addresses*
8192	000 0010 0000 *to* 000 0111 1111
16384 (or more)	000 0110 0000 *to* 000 0111 1111

Note that System/360 Model 30 processing units with 32,768 or more positions of main storage can have a special feature that permits addressing of up to 224 units. This feature, however, excludes use of shared control units and their method of addressing, which we shall now consider.

Certain types of I/O units (such as magnetic tape drives) share a single control unit. For these devices, only one of the I/O units can transmit data at a time. For example, consider two tape drives (A and B) attached to a control unit. If tape drive A is transmitting data, then tape drive B cannot transmit data until tape drive A has completed its operation. Here, only one multiplexor channel subchannel is provided for the control unit that controls the two tape drives. For this situation, the device address must have its high-order bit set to a value of one. The next three high-order bits designate the shared control unit of the attached devices. The low-order four bits designate the address of devices attached to the shared control unit. The addresses for shared control units range from:

Multiplexor Channel Address	*Shared Control-Unit Address*	*Device Address**
000	1000	XXXX
	to	
000	1111	XXXX

*XXXX can range from 0000 to 1111 for each value of the shared control unit address.

Notice that only three bits are used to address the shared control unit. The number of shared control units that can be addressed is then:

Binary Address	*Decimal Value*
000	0
to	to
111	7

We have considered two kinds of addresses on the multiplexor channel:

1. Those in which the high-order bit of the unit address has a value of zero.
2. Those in which the high-order bit of the unit address has a value of one.

An important point to note is that, in certain cases, the unit using a shared control unit and the unit using a single control unit (that is, the two types of addresses considered) could conceivably use the same subchannel in the multiplexor channel. That is, to the multiplexor channel, the unit address 0000 0001 is interpreted in the same way as unit address 1001 XXXX (where the X's represent the address of a device attached to a shared control unit).

Now if both of the units, to which these addresses apply, were operating at the same time, they would both use the same UCW (Unit Control Word) in multiplexor storage. Because a subchannel can keep track of only one unit's operation at a time, an attempt to keep track of two operations simultaneously with the same subchannel (UCW) would cause meaningless information to be stored in the special subchannel storage. Therefore, the first eight subchannels are shared on a mutually exclusive basis between the first eight single control-unit addresses and the eight possible shared control-unit addresses.

Hence, if a unit address, listed in a column of the following table, is used on the multiplexor channel, then the corresponding address in the other column cannot be used:

Single Control Unit	*Shared Control Unit*
0000 0000	1000 XXXX
0000 0001	1001 XXXX
0000 0010	1010 XXXX
0000 0011	1011 XXXX
0000 0100	1100 XXXX
0000 0101	1101 XXXX
0000 0110	1110 XXXX
0000 0111	1111 XXXX

These are the only *mutually exclusive* addresses for the multiplexor channel.

At the installation of the system, the customer engineer sets in each I/O device and control unit the bit address desired for a unit, within the limits imposed by the foregoing rules.

Refer to *Appendix B* for I/O unit addresses that correspond to unit control word addresses in the multiplexor storages.

Selector Channels

Two selector channels (each a special feature) are available on the IBM System/360 Model 30.

When the program calls for an I/O operation, the CPU loads a CCW into the channel and initiates the operation. The CPU may then proceed with its program while the channel controls the transfer of data between storage and the I/O unit. During operation, the selector channel *steals* a storage cycle whenever a byte of data must be transferred into or out of storage. When the transfer of data has been completed, the channel interrupts the CPU and informs it of the results of the operation by storing a Channel Status Word (CSW) in storage.

Selector channels operate only in burst mode, stealing storage cycles as they are needed.

The selector channel is designed to be used primarily with high-speed devices such as magnetic tape and disk file units. However, if desired, any of the low-speed devices equipped with the System/360 standard I/O interface can be operated on the selector channel.

Selector Channel Organization and Operation

Each selector channel contains registers that hold the command code, maintain the data address, hold the byte count of data to be transferred under the current CCW, and hold the flags to control the current channel action and condition its next action. The data address and byte count are used and updated for each byte transferred. The flags are used to control further action when a transfer of data for a CCW is complete, or to indicate conditions of some other kind of termination. The next channel command word address is maintained in local storage to provide the storage location of the next CCW on command-chaining and data-chaining operations.

In addition, each selector channel has a data register (buffer) to hold data in its passage between main storage and the I/O device.

In the Model 30 during selector-channel read/write I/O cycles, the byte count and data address registers are updated independently of the CPU. Only one storage cycle is used to accomplish the actual data transfer between main storage and the respective channel. When either data chaining or command chaining is initiated, however, the control sequences are provided by microprogram routines within read-only storage. During these channel operations, the CPU program is interrupted for the length of time required to establish the next CCW address and read out the next CCW.

The selector-channel data buffer register in the Model 30 provides one byte of buffering in addition to that which may be offered by the device. For example, this buffer when empty accepts a byte immediately when it is offered by the device and holds it (pending its access to main storage). In the meantime, another byte is being processed by the device. Access to main storage may be delayed as much as 2.25 microseconds (3 microseconds in a CPU with a RW cycle of 2 microseconds) by an RCW cycle before an RW cycle is allocated to the channel. By the time this transfer from buffer to main storage is completed, the buffer may immediately accept the next character for transfer on the next RW cycle if the device has indicated it is waiting.

Time-sharing of main storage by two selector channels is indicated in Figure 16. This operation assumes that no multiplexor channel requests are occurring. If the multiplexor channel is operating, the capacity of the second selector channel is reduced because of a circuit that prevents the selector channels from using all available storage cycles and locking out the multiplexor channel. Note also that when the selector channels are operating, the maximum data rate of the multiplexor channel is reduced accordingly.

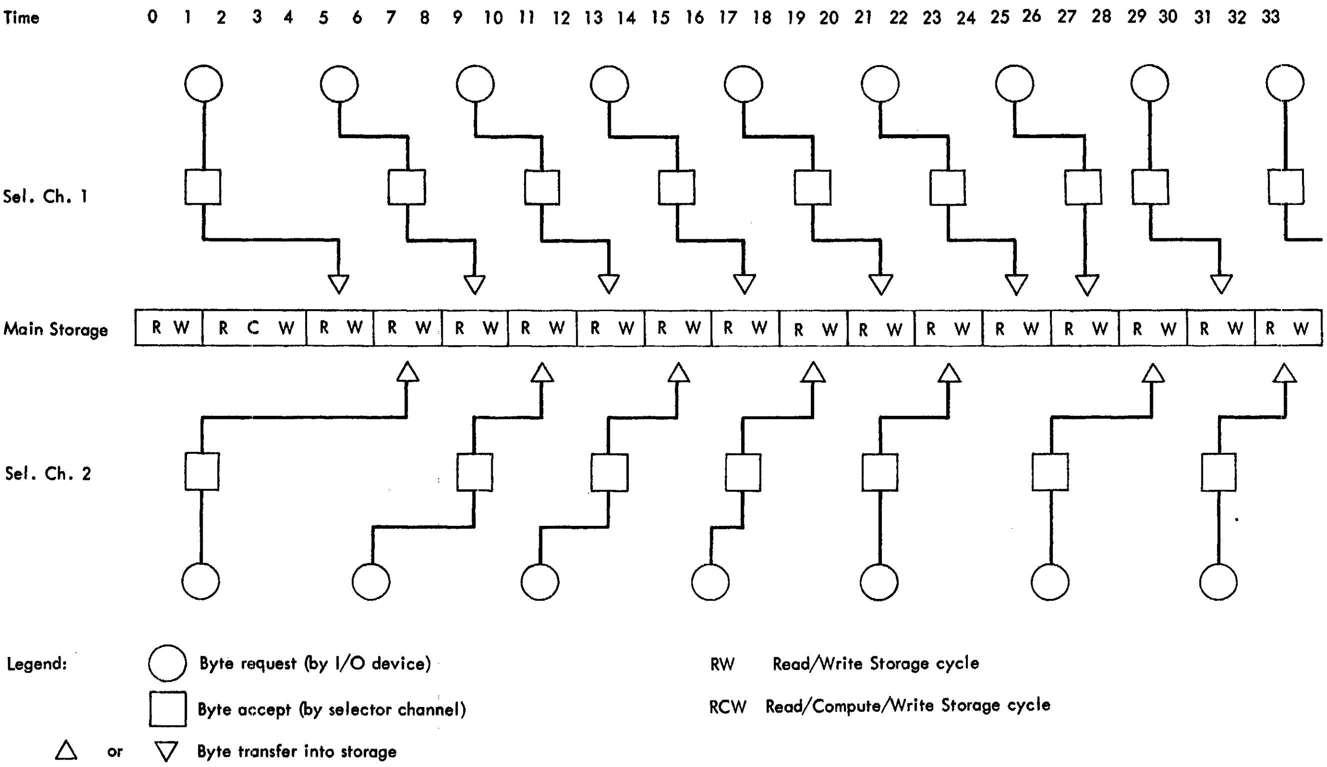

Time
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
Sel. Ch. 1
Main Storage
R W | R C W | R W | R W | R W | R W | R W | R W | R W | R W | R W | R W | R W | R W | R W | R W | R W
Sel. Ch. 2
Legend:
Byte request (by I/O device)
Byte accept (by selector channel)
or
Byte transfer into storage
RW Read/Write Storage cycle
RCW Read/Compute/Write Storage cycle

Figure 16. Time Sharing for Two Read Operations

Channel Loading and Interaction

Data Transfer

When both kinds of channels (multiplexor and selector) are operating simultaneously, the CPU is, in fact, the multiplexing channel. Selector-channel storage-cycle stealing causes the stoppage of the CPU program for 1.5 microseconds (2 microseconds in a CPU with a 2-microsecond RW cycle). Thus, every selector-channel data-access cycle adds this time to any simultaneously operating microprogram, such as the multiplexor-share or multiplexor-burst microprogram. The multiplexor channel is limited, in simultaneous operations, to cycles left available by selector-channel operations in progress. A storage priority scheme allocates storage cycles to the selector channel and the multiplexor channel such that when all three channels are in operation concurrently the selector channels do not prevent storage cycle accesses by the multiplexor channel. When the channels are operating, they take priority over the CPU, and the CPU uses the cycles that remain after the requirements of the channels are satisfied.

When both the multiplexor and the selector channels are operating concurrently, the maximum data rate of the multiplexor channel is reduced. This rate reduction is different for different modes of operation on the selector channels. It is necessary for the programmer to consider the aggregate data rates of operating I/O units (fastest case) in order to prevent the possibility of overruns (more storage requests than storage cycles available). He should also consider the interference caused by data chaining and command chaining. The procedures used to determine channel-loading limits are described in *IBM System/360 Model 30 Channel Characteristics and Functional Evaluation,* GA24-3411.

Command Chaining and Data Chaining

Command or data chaining in the selector channel is done by forced branching to a microprogram that has priority over the multiplexor microprogram. Thus, when chaining occurs in the selector channel, multiplexor channel cycle time is increased by the cycle time required for this chaining. Furthermore, the selector-channel chaining time may be elongated if the other selector channel steals main-storage cycles for data transfers.

For information on chaining times and interactions of other operations with chaining, refer to *IBM System/360 Model 30 Channel Characteristics and Functional Evaluation,* Form A24-3411.

Model 30 CPU Timings and Priorities

The physical organization of the Model 30, with its single-byte access per main-storage cycle design, results in serial-by-byte operation.

The Model 30, while interpreting and executing a program instruction, accomplishes a number of read/write storage cycles. A complete storage cycle (i.e., a read *and* write) must be performed before the CPU allows a channel to access main storage for data. An interrupt, however, can only occur when the preceding instruction is finished and the next instruction is not yet started. Certain interrupts have priority over others when two or more occur simultaneously. The lower priority interrupt will be held pending until the higher priority interrupt is completed. Therefore, the data flow and the main core storage of the Model 30 are time-shared among several functions. Channel, interval timer, and programming priority are determined as follows:

1. *Data transfer of selector channels.* Channel 1 has priority in case of simultaneous requests.
2. *Selector-channel chaining.* Channel 1 has priority in simultaneous requests, but once an operation is started on this priority, no other operation of equal priority can interrupt.
3. *Multiplexor-channel service requests.*
4. *Interval-timer storage requests.* On the Model 30, the interval-timer storage locations are updated only between CPU instructions.
5. *Execution of program steps.*

Storage overrun conditions can occur if I/O channel operations are programmed beyond the response capability of the system. For further information, refer to *IBM System/360 Model 30 Channel Characteristics and Functional Evaluation,* GA24-3411.

Instruction Timing Information

Instruction timing charts (Figure 17) provide average instruction times, in microseconds, for the IBM System/360 Model 30. (For the CPU with a 2-microsecond RW cycle, see Figure 18.) All times for instructions that reference main storage include the time required for generating an address using one general register (base) and the displacement value. For those instructions which may include an index register (all RX format Instructions), 4.5 microseconds (6.0 microseconds in a CPU with a 2-microsecond RW cycle) must be added to the basic time given in the charts if an index register is specified.

Those instructions which may use an index register are highlighted by an asterisk in the chart.

The instructions that may be indexed a second time are highlighted by an asterisk in the chart.

The following assumptions were used in the development of these instruction timing charts.

Variable-Length Field Instructions

1. For decimal-add and decimal-subtract instructions, the first operand (the destination field) is assumed to be greater than, or equal to, the length of the second operand (the source field).
2. For decimal-add and decimal-subtract instructions, recomplementation time is not included in the times given.
3. In the EDIT AND MARK (EDMK) instruction, an address is stored once. That is, this instruction is used with a single field or a line with only one numeric field rather than a complete print line.
4. In the TRANSLATE AND TEST (TRT) instruction, it is assumed that a non-zero byte from the Translate and Test table is found.

Floating-Point Instructions

The instruction times for the floating-point instructions depend upon the number of hexadecimal digits that are preshifted and postshifted, as well as the number of times recomplementation of a result occurs. The times given in the instruction timetables for floating-point instructions are a weighted average of these variables.

Legend for Timing Formulas (Figures 17 and 18)

B	Number of pairs of characters compared in finding the highest-order unequal pair.
H	Number of significant hexadecimal digits in the binary form of the number being converted.
N	Total number of bytes in field.
N_1	Total number of bytes in first operand.
N_2	Total number of bytes in second operand.
V	For divide decimal and multiply decimal. $V = N_1 - N_2$, which is the number of bytes in the multiplier or quotient.
M	Maximum of N_1 and N_2.
*	Index register may be specified.
DEC	Decimal Arithmetic feature.
FP	Floating-Point Arithmetic feature.
PROT	Storage Protection feature.
RD/WD	Read Direct/Write Direct feature.

INSTRUCTION	SEE LEGEND	FORMAT	MNEMONIC	TIME (B ≠ 0)
Add		RR	AR	22
Add*		RX	A	29
Add Decimal	DEC	SS	AP	$45 + 4N_1$
Add Halfword*		RX	AH	27
Add Logical		RR	ALR	23
Add Logical*		RX	AL	30
Add Normalized (Long)	FP	RR	ADR	61
Add Normalized (Long)*	FP	RX	AD	69
Add Normalized (Short)	FP	RR	AER	44
Add Normalized (Short)*	FP	RX	AE	52
Add Unnormalized (Long)	FP	RR	AWR	59
Add Unnormalized (Long)*	FP	RX	AW	68
Add Unnormalized (Short)	FP	RR	AUR	42
Add Unnormalized (Short)*	FP	RX	AU	51
AND		RR	NR	23
AND*		RX	N	30
AND		SI	NI	15
AND		SS	NC	33 + 4N
Branch and Link		RR	BALR	Branch = 18 No Branch = 15
Branch and Link*		RX	BAL	23
Branch on Condition		RR	BCR	Branch = 9 No Branch = 7
Branch on Condition*		RX	BC	Branch = 17 No Branch= 16
Branch on Count		RR	BCTR	Branch = 14 No Branch = 19 Count only = 11
Branch on Count*		RX	BCT	Branch = 18 No Branch = 23
Branch on Index High		RS	BXH	Branch = 38 No Branch = 37
Branch on Index Low or Equal		RS	BXLE	Branch = 38 No Branch = 37
Compare		RR	CR	20
Compare*		RX	C	27
Compare Decimal	DEC	SS	CP	45 + 4M
Compare Halfword*		RX	CH	26
Compare Logical		RR	CLR	9 + 3 B
Compare Logical*		RX	CL	17 + 3 B

Time shown in microseconds.

Figure 17. IBM System/360 Model 30 Instruction Timing Chart (for CPU with 1.5-Microsecond Storage Cycle), Part 1 of 4

INSTRUCTION	SEE LEGEND	FORMAT	MNEMONIC	TIME (B $\neq$ 0)
Compare Logical		SI	CLI	16
Compare Logical		SS	CLC	33 + 4 x total number of bytes processed.
Compare (Long)	FP	RR	CDR	65
Compare (Long)*	FP	RX	CD	57
Compare (Short)	FP	RR	CER	35
Compare (Short)*	FP	RX	CE	42
Convert to Binary*		RX	CVB	$89 + .75H + 3H^2$
Convert to Decimal*		RX	CVD	$46 + 18H + 1.5H^2$
Divide		RR	DR	413
Divide*		RX	D	420
Divide Decimal	DEC	SS	DP	$34 + V (54N_2 + 60)$
Divide (Long)	FP	RR	DDR	1619
Divide (Long)*	FP	RX	DD	1665
Divide (Short)	FP	RR	DER	291
Divide (Short)*	FP	RX	DE	301
Edit	DEC	SS	ED	$38 + 7N_1 + 9N_2$
Edit and Mark	DEC	SS	EDMK	$45 + 7N_1 + 9N_2$
Exclusive OR		RR	XR	23
Exclusive OR*		RX	X	30
Exclusive OR		SI	XI	16
Exclusive OR		SS	XC	33 + 4N
Execute*		RX	EX	25 + Executed Instruction Time
Halt I/O		SI	HIO	42 + Channel Response Time
Halve (Long)	FP	RR	HDR	85
Halve (Short)	FP	RR	HER	46
Insert Character*		RX	IC	16
Insert Storage Key	PROT	RR	ISK	15
Load		RR	LR	17
Load*		RX	L	24
Load Address*		RX	LA	17
Load and Test		RR	LTR	21
Load and Test (Long)	FP	RR	LTDR	32
Load and Test (Short)	FP	RR	LTER	20
Load Complement		RR	LCR	21
Load Complement (Long)	FP	RR	LCDR	32
Load Complement (Short)	FP	RR	LCER	20
Load Halfword*		RX	LH	22
Load (Long)	FP	RR	LDR	29
Load (Long)*	FP	RX	LD	38

Time shown in Microseconds

Figure 17. IBM System/360 Model 30 Instruction Timing Chart (for CPU with 1.5-Microsecond Storage Cycle), Part 2 of 4

INSTRUCTION	SEE LEGEND	FORMAT	MNEMONIC	TIME (B ≠ O)
Load Multiple		RS	LM	12 + 12 x Number of Registers Loaded
Load Negative		RR	LNR	21
Load Negative (Long)	FP	RR	LNDR	32
Load Negative (Short)	FP	RR	LNER	20
Load Positive		RR	LPR	21
Load Positive (Long)	FP	RR	LPDR	32
Load Positive (Short)	FP	RR	LPER	20
Load PSW		SI	LPSW	28
Load (Short)	FP	RR	LER	17
Load (Short)*	FP	RX	LE	26
Move		SI	MVI	13
Move		SS	MVC	31 + 3N
Move Numerics		SS	MVN	31 + 4N
Move with Offset		SS	MVO	$32 + 2N_1 + 3N_2$
Move Zones		SS	MVZ	31 + 4N
Multiply		RR	MR	228
Multiply*		RX	M	235
Multiply Decimal	DEC	SS	MP	$34 + V(21N_2 + 35)$
Multiply Halfword*		RX	MH	75
Multiply (Long)	FP	RR	MDR	460
Multiply (Long)*	FP	RX	MD	472
Multiply (Short)	FP	RR	MER	199
Multiply (Short)*	FP	RX	ME	208
OR		RR	OR	23
OR*		RX	O	30
OR		SI	OI	16
OR		SS	OC	33 + 4N
Pack		SS	PACK	$32 + 2N_1 + 3N_2$
Read Direct	RD/WD	SI	RDD	18 + External Delay
Set Program Mask		RR	SPM	10
Set Storage Key	PROT	RR	SSK	15
Set System Mask		SI	SSM	16
Shift Left Double		RS	SLDA	85
Shift Left Double Logical		RS	SLDL	83
Shift Left Single		RS	SLA	56
Shift Left Single Logical		RS	SLL	54
Shift Right Double		RS	SRDA	76
Shift Right Double Logical		RS	SRDL	74
Shift Right Single		RS	SRA	51
Shift Right Single Logical		RS	SRL	49

Time shown in microseconds.

Figure 17. IBM System/360 Model 30 Instruction Timing Chart (for CPU with 1.5-Microsecond Storage Cycle), Part 3 of 4

INSTRUCTION	SEE LEGEND	FORMAT	MNEMONIC	TIME (B ≠ O)
Start I/O		SI	SIO	90 + Channel Response Time
Store*		RX	ST	25
Store Character*		RX	STC	16
Store Halfword*		RX	STH	19
Store (Long)*	FP	RX	STD	38
Store Multiple		RS	STM	13 + 12 x Number of Registers Loaded
Store (Short)*	FP	RX	STE	26
Subtract		RR	SR	22
Subtract*		RX	S	29
Subtract Decimal	DEC	SS	SP	$45 + 4N_1$
Subtract Halfword*		RX	SH	27
Subtract Logical		RR	SLR	23
Subtract Logical*		RX	SL	30
Subtract Normalized (Long)	FP	RR	SDR	57
Subtract Normalized (Long)*	FP	RX	SD	65
Subtract Normalized (Short)	FP	RR	SER	42
Subtract Normalized (Short)*	FP	RX	SE	50
Subtract Unnormalized (Long)	FP	RR	SWR	55
Subtract Unnormalized (Long)*	FP	RX	SW	64
Subtract Unnormalized (Short)	FP	RR	SUR	40
Subtract Unnormalized (Short)*	FP	RX	SU	49
Supervisor Call		RR	SVC	44
Test and Set		SI	TS	17
Test Channel		SI	TCH	30 + Channel Response Time
Test I/O		SI	TIO	83 + Channel Response Time
Test Under Mask		SI	TM	15
Translate		SS	TR	31 + 6N
Translate and Test		SS	TRT	39 + 6N
Unpack		SS	UNPK	$32 + 3N_1 + 3N_2$
Write Direct	RD/WD	SI	WRD	17
Zero and Add	DEC	SS	ZAP	$43 + 4N_1$

Time shown in microseconds

Figure 17. IBM System/360 Model 30 Instruction Timing Chart (for CPU with 1.5-Microsecond Storage Cycle), Part 4 of 4

Interval Timer

The Model 30 Interval Timer (special feature) operates at a fixed cycle rate of 16.7 milliseconds (60-cycle system power-supply input) or 20 milliseconds (50-cycle power). The microprogram controls the decrementing of the timer.

The interval-timer microprogram requires 7.5 to 13.5 microseconds (10 to 18 microseconds in a CPU with a 2-microsecond RW cycle) per count depending upon whether there is a carry in the count. This cycle occurs asynchronously with respect to the stored program and I/O operations.

Back-up register is provided with the timer feature to accumulate automatically a count of up to 16 intervals of time, if main storage cannot be accessed because of prolonged I/O or direct control operations. This feature permits a delay of up to 277 milliseconds between timer count references without loss of the count.

I/O Interruption

An I/O interruption requires about 78 microseconds (104 microseconds in a CPU with a 2-microsecond RW cycle) for execution. This includes: the storing of the old PSW, the read-out of the new (I/O) PSW, and the formation of the CSW (Channel Status Word).

Other Program Interruptions

Other program interruptions require 41 microseconds (54 microseconds in a CPU with a 2-microsecond RW

INSTRUCTION	SEE LEGEND	FORMAT	MNEMONIC	TIME (B ≠ 0)
Add		RR	AR	29.
Add*		RX	A	39.
Add Decimal	DEC	SS	AP	60 + 5 N_1
Add Halfword*		RX	AH	37.
Add Logical		RR	ALR	30.
Add Logical*		RX	AL	40.
Add Normalized (Long)	FP	RR	ADR	105.
Add Normalized (Long)*	FP	RX	AD	115.
Add Normalized (Short)	FP	RR	AER	65.
Add Normalized (Short)*	FP	RX	AE	75.
Add Unnormalized (Long)	FP	RR	AWR	88.
Add Unnormalized (Long)*	FP	RX	AW	98.
Add Unnormalized (Short)	FP	RR	AUR	56.
Add Unnormalized (Short)*	FP	RX	AU	66.
AND		RR	NR	30.
AND*		RX	N	40.
AND		SI	NI	20.
AND		SS	NC	44 + 5N
Branch and Link		RR	BALR	Branch = 24 No Branch = 19
Branch and Link*		RX	BAL	35
Branch on Condition		RR	BCR	Branch = 14 No branch = 9
Branch on Condition*		RX	BC	Branch = 22 No branch = 21
Branch on Count		RR	BCTR	Branch = 20 No branch = 25 Count only = 15
Branch on Count*		RX	BCT	Branch = 25 No branch = 30
Branch on Index High		RS	BXH	Branch = 52 No branch = 51
Branch on Index Low or Equal		RS	BXLE	Branch = 52 No branch = 51
Compare		RR	CR	26.
Compare*		RX	C	39.
Compare Decimal	DEC	SS	CP	60 + 5 M
Compare Halfword*		RX	CH	36.
Compare Logical		RR	CLR	12 + 4 B
Compare Logical*		RX	CL	22 + 4 B

Time shown in microseconds.

Figure 18. IBM System/360 Model 30 Instruction Timing Chart (for CPU with 2-Microsecond Storage Cycle), Part 1 of 4

INSTRUCTION	SEE LEGEND	FORMAT	MNEMONIC	TIME ($B \neq 0$)
Compare Logical		SI	CLI	20.
Compare Logical		SS	CLC	44 + 5 x total number of bytes processed.
Compare (Long)	FP	RR	CDR	87.
Compare (Long)*	FP	RX	CD	97.
Compare (Short)	FP	RR	CER	55.
Compare (Short)*	FP	RX	CE	65.
Convert to Binary*		RX	CVB	$118 + H + 4H^2$
Convert to Decimal*		RX	CVD	$61 + 24H + 2H^2$
Divide		RR	DR	550
Divide*		RX	D	560.
Divide Decimal	DEC	SS	DP	$45 + V\ (71N_2 + 80)$
Divide (Long)	FP	RR	DDR	2180
Divide (Long)*	FP	RX	DD	2190
Divide (Short)	FP	RR	DER	390
Divide (Short)*	FP	RX	DE	400
Edit	DEC	SS	ED	$50 + 9\ N_1 + 11\ N_2$
Edit and Mark	DEC	SS	EDMK	$60 + 9\ N_1 + 11\ N_2$
Exclusive OR		RR	XR	30.
Exclusive OR*		RX	X	40.
Exclusive OR		SI	XI	21.
Exclusive OR		SS	XC	44 + 5 N
Execute*		RX	EX	25 + Executed Instruction Time
Halt I/O		SI	HIO	55 + Channel Response Time
Halve (Long)	FP	RR	HDR	102.
Halve (Short)	FP	RR	HER	50.
Insert Character*		RX	IC	21.
Insert Storage Key	PROT	RR	ISK	18.
Load		RR	LR	22
Load*		RX	L	32.
Load Address*		RX	LA	25.
Load and Test		RR	LTR	28.
Load and Test (Long)	FP	RR	LTDR	42.
Load and Test (Short)	FP	RR	LTER	25.
Load Complement		RR	LCR	28.
Load Complement (Long)	FP	RR	LCDR	42.
Load Complement (Short)	FP	RR	LCER	25.
Load Halfword*		RX	LH	28.
Load (Long)	FP	RR	LDR	39
Load (Long)*	FP	RX	LD	49

Time shown in Microseconds

Figure 18. IBM System/360 Model 30 Instruction Timing Chart (for CPU with 2-Microsecond Storage Cycle), Part 2 of 4

INSTRUCTION	SEE LEGEND	FORMAT	MNEMONIC	TIME (B ≠ O)
Load Multiple		RS	LM	16 + 16 X Number of Registers Loaded
Load Negative		RR	LNR	28
Load Negative (Long)	FP	RR	LNDR	42
Load Negative (Short)	FP	RR	LNER	25
Load Positive		RR	LPR	28
Load Positive (Long)	FP	RR	LPDR	42
Load Positive (Short)	FP	RR	LPER	25
Load PSW		SI	LPSW	37
Load (Short)	FP	RR	LER	23
Load (Short)*	FP	RX	LE	33
Move		SI	MVI	18
Move		SS	MVC	41 + 4N
Move Numerics		SS	MVN	41 + 5N
Move with Offset		SS	MVO	$41 + 3N_1 + 3N_2$
Move Zones		SS	MVZ	41 + 5N
Multiply		RR	MR	304.
Multiply*		RX	M	313.
Multiply Decimal	DEC	SS	MP	$45 + V (28N_2 + 54)$
Multiply Halfword*		RX	MH	100.
Multiply (Long)	FP	RR	MDR	1050.
Multiply (Long)*	FP	RX	MD	1060.
Multiply (Short)	FP	RR	MER	310.
Multiply (Short)*	FP	RX	ME	320.
OR		RR	OR	30.
OR*		RX	O	40.
OR		SI	OI	21.
OR		SS	OC	44 + 5N
Pack		SS	PACK	$41 + 3N_1 + 3N_2$
Read Direct	RD/WD	SI	RDD	24 + External Delay
Set Program Mask		RR	SPM	13.
Set Storage Key	PROT	RR	SSK	19.
Set System Mask		SI	SSM	21.
Shift Left Double		RS	SLDA	113
Shift Left Double Logical		RS	SLDL	110
Shift Left Single		RS	SLA	74
Shift Left Single Logical		RS	SLL	71
Shift Right Double		RS	SRDA	101
Shift Right Double Logical		RS	SRDL	98
Shift Right Single		RS	SRA	68
Shift Right Single Logical		RS	SRL	65

Time shown in microseconds.

Figure 18. IBM System/360 Model 30 Instruction Timing Chart (for CPU with 2-Microsecond Storage Cycle), Part 3 of 4

INSTRUCTION	SEE LEGEND	FORMAT	MNEMONIC	TIME (B ≠ O)
Start I/O		SI	SIO	120 + Channel Response Time
Store*		RX	ST	32.
Store Character*		RX	STC	21.
Store Halfword*		RX	STH	25.
Store (Long)*	FP	RX	STD	49.
Store Multiple		RS	STM	17 + 16 x Number of Registers Loaded
Store (Short)*	FP	RX	STE	33.
Subtract		RR	SR	29.
Subtract*		RX	S	39.
Subtract Decimal	DEC	SS	SP	60 + 5 N_1
Subtract Halfword*		RX	SH	37.
Subtract Logical		RR	SLR	30.
Subtract Logical*		RX	SL	40.
Subtract Normalized (Long)	FP	RR	SDR	105.
Subtract Normalized (Long)*	FP	RX	SD	115.
Subtract Normalized (Short)	FP	RR	SER	65.
Subtract Normalized (Short)*	FP	RX	SE	75.
Subtract Unnormalized (Long)	FP	RR	SWR	88.
Subtract Unnormalized (Long)*	FP	RX	SW	98.
Subtract Unnormalized (Short)	FP	RR	SUR	56.
Subtract Unnormalized (Short)*	FP	RX	SU	66.
Supervisor Call		RR	SVC	58.
Test and Set		SI	TS	22
Test Channel		SI	TCH	40 + Channel Response Time
Test I/O		SI	TIO	110 + Channel Response Time
Test Under Mask		SI	TM	21.
Translate		SS	TR	41 + 7 N
Translate and Test		SS	TRT	51 + 8N
Unpack		SS	UNPK	41 + 3 N_1 + 3N_2
Write Direct	RD/WD	SI	WRD	22.
Zero and Add	DEC	SS	ZAP	57 + 5 N_1

Time shown in microseconds

Figure 18. IBM System/360 Model 30 Instruction Timing Chart (for CPU with 2-Microsecond Storage Cycle), Part 4 of 4

cycle) execution time. A supervisor call interrupt requires 44 microseconds (58 microseconds in a CPU with a 2-microsecond RW cycle).

Diagnose Instruction

In IBM System/360 Model 30, the DIAGNOSE instruction is used to branch to a read-only-storage address, specified by the B_1 and D_1 fields in the DIAGNOSE instruction.

Certain compatibility operation codes are activated by the setting in the I_2 field of the DIAGNOSE instruction. For further information, refer to the *IBM System/360 Model 30 1401/1440/1460 Compatibility Feature,* Form A24-3255. The format and general description of the DIAGNOSE instruction are described in *IBM System/360 Principles of Operation,* GA22-6821.

Diagnostic Scan-Out Area

A diagnostic scan-out area is provided for System/360 Model 30 in main storage locations:

Decimal Address	*Hexadecimal Address*
128	80
129	81
130	82
131	83
133	85
134	86
135	87
137	89
138	8A
139	8B

In other models of System/360, the diagnostic scan-out area may be larger, but it always starts at address 128 (decimal).

When you write programs for Model 30, do not reference the main storage diagnostic scan-out addresses with your program. If programs written for the Model 30 are to be run on a larger System/360 model, then these programs should be written so that the larger-capacity System/360 model diagnostic scan-out area is not addressed by the program.

In System/360 Model 30, the scan-out area provides for logging information related to various check conditions. When certain check conditions occur, the contents of the machine check register are placed in storage location 128 (decimal address). If any of the following bits, of byte 128, are set to a value of one, then the corresponding check condition has occurred:

Address 128	
Bit Position	*Check Condition In*
0	A-register (input to ALU)
1	B-register (input to ALU)
2	MN-register (main storage address register)
3	Read-only-storage control register
4	Sense amplifiers (for read-only-storage)
5	Read-only-storage address register
6	R-register (main storage data register) or low 4 bits of Q register (storage protect feature only).
7	ALU (arithmetic logic unit)

Bytes 129, 130, and 131 (decimal addresses) are used for log-out when multiplexor channel check conditions are detected. Here, if one of the machine check register positions is set to a value of one when the multiplexor channel check occurs, then byte 129 is set to correspond to the machine check register setting, as follows:

Address 129	
Bit Position	*Check Condition In*
0	A-register
1	B-register
2	MN-register
3	ROS control register
4	Sense amplifiers (for read-only-storage)
5	Read-only-storage address register
6	R-register
7	ALU

However, if the multiplexor-channel check condition does not affect the machine check register, then byte 129 is set with a *catalog number* that indicates the kind of interface sequence or response (that is, time-out) check that is detected in the multiplexor channel. Catalog numbers are used by the customer engineer, who has listings of these catalog numbers and references to the sources of check conditions to which they apply.

Also, an *indicator byte* is logged-out in location 130 (decimal address) as a result of a multiplexor-channel check condition. The contents of this byte are:

Bit Position	*Indication*
0 through 3	Accumulated log-out count (that is, indicates the number of log-outs that have occurred).
4	Set to a value of one, each time a log-out occurs.
5	Indicates type of information logged-out in byte 129. If bit 5 is set to a value of zero, then the contents of the machine check register are stored in location 129. If bit 5 is set to a value of one, then a catalog number is stored in byte 129. The catalog number relates to the microprogram source of the check condition.
6 and 7	Binary coded results of interface inspection and any necessary reset:

Bits 6 and 7	
Binary value	*Indication* .
00	Interface clear at check occurrence.
01	Halt I/O reset required and successful
10	Malfunction reset (selective reset) used and successful.
11	Interface locked up (that is, resets do not function properly).

Byte 131 contains the unit address of the unit that is involved in the check condition on the multiplexor channel.

Bytes 133, 134, and 135 are used for selector-channel 1 log-out conditions in the same manner as bytes 129, 130, and 131 are used for the multiplexor channel. That is, byte 133 is set with the contents of the machine check register or with a catalog number, depending upon the type of error that occurred. Byte 134 bit positions have the same information stored as does byte 130, but for check conditions on selector-channel one. Byte 135 contains the unit address of the unit involved in the check condition on selector-channel one.

Bytes 137, 138, and 139 are used for selector-channel 2 log-outs. Information in byte 137 is either the contents of the machine check register or a catalog number, depending upon the check condition involved. Byte 138 bit positions have the same information as byte 130, but for check conditions on selector-channel two. Byte 139 contains the address of the unit involved in the check condition on selector-channel two.

Reset of any or all parts of the log-out locations is determined by programming *in the diagnostic program used* (that is, these areas are not reset automatically but must be programmed to reset).

IBM 2030 System Control Panel

The System Console information (pages 35 through 43) is obsolete and has been replaced by *IBM System/360 Model 30 Operating Guide,* Order No. A24-3373.

(Pages 36-43 deleted)

IBM 1050 Documentary Console

The IBM 1052 Printer Keyboard is the basic console keyboard printer for the System/360 Model 30. Communication between operator and program is, therefore, effected through the 1052. Besides the 1052, a variety of IBM 1050 console devices is available to increase the flexibility of the system. These devices (including the 1052) are attached to the IBM 2030 Processing Unit through an IBM 1051 Control Unit and the 1051 Attachment, which is located in the 2030.

The maximum number of 1050 console devices attachable through this feature are:

- One keyboard
- Two printers
- One reader (either card or paper tape)
- One punch (either card or paper tape).

These devices can be operated separately from the 2030 (even if CPU power is off) on a 24-hour basis. Additional use-rental is not charged for the 2030 while the 1050 devices are used in this manner. This kind of operation is called *off-line* (off-line signifying that the 1050 devices are not being used in operations that affect the 2030). Off-line operation provides for media conversion (such as transferring data from card to paper tape), card or paper tape listing or preparation, and other 1050 operations that do not depend on 2030 functions. Interdependent operations between the 2030 and the console devices are termed *on-line* operations. An example is an operator request for information from the 2030 by means of the IBM 1052 Printer Keyboard.

Additional flexibility is gained through an inexpensive data communications link (when a 1051 model 1 with the master station feature is used). Here the local 1050 devices (physically located near the 2030) can exchange information with up to 26 remote 1050 terminals. This communication link can be operated off-line only. That is, messages sent and received by the local 1050 devices are not controlled and not sent or received by the 2030. The local 1050 I/O devices not being used in the data communications link can, however, be operated on-line with the 2030. This operation can occur at the same time as the off-line data communications operation.

To distinguish between data communications operations and local operations, we use the terms *line loop* and *home loop*. *Home loop* 1050 device operations can be on-line to the 2030 or off-line, as already described, among the local 1050 devices. *Line loop* refers to operations on the communications line. These operations can occur only between local 1050 devices and remote 1050 terminals that all operate off-line to the 2030.

To summarize:

1. *On-line* refers to operations between devices and the 2030.
2. *Off-line* refers to operations that do not involve the 2030.
3. *Home loop* refers to operations on a local basis, either on-line or off-line.
4. *Line loop* refers to communications line operations.

Because the usage of I/O devices can become critical in time-dependent applications, it is important that the programmer fully understands the meanings of the terms on-line and off-line.

Multiplexor Channel Address

The unit address of the 1050 console is a fixed multiplexor channel address. In a 2030 with 8,192 positions of main storage, the 1050 console address is 1F (hexadecimal). System/360 Model 30 Processing Units that have 16,384 or more positions of main storage use address 1F or 5F for the 1050 console (one or the other, not both). This arrangement allows for the possibility of two systems connected through the channel to channel adapter. For example, if a system with 16,384 storage positions is connected to a system with 8,192 storage positions, then the address both systems use for the 1050 console is 1F. Hence communication is provided to a console from either system.

If a system has the 224 subchannels feature, then the console address can be 1F, 5F, or DF (only one, not any combination). The 1050 attachment can be selectively assigned to be first or last in terms of multiplexor channel polling priority.

Address 09 can also be used in any 2030 for the 1050 address on the multiplexor channel. (Use of this address excludes use of addresses 1F, 5F, or DF for the 1050.) The desired 1050 address is set by the Customer Engineer at installation time.

Configurations

All 1050 documentary console devices are attached to the 2030 through the 1051 Attachment in the 2030. Automatic translation is provided between the 1050 PTTC/EBCD and the 2030 EBCDI code. Alter, display, and hexadecimal conversion functions are performed by programming.

The IBM 1051 Control Unit controls the local 1050 devices. Within the 1051 are the various 1050 I/O adapters, controls, and power supplies. The IBM 1057 Card Punch is an exception in that it contains its own power suply.

Either of two models of the 1051 can be used with the 2030. The 1051 Model N1 (home loop only) provides for on-line or off-line operation of local 1050 I/O components. The 1051 Model 1 also provides for on-line or off-line operation of local 1050 components on the home loop and provides for line-loop operation of local 1050 devices with remote 1050 terminals.

Note: This communication link is off-line to the 2030 only.

The IBM 1052 Printer Keyboard has all the necessary control switches and lights for normal 1050 operations only and for operations involving the 1050 and 2030. For possible configurations not requiring a 1052, a 1050 switch unit is required.

Various configurations of IBM 1050 devices with the 2030 are possible. Three examples of the configurations are presented here.

- Basic printer keyboard
- Maximum home loop
- Off-line communication (line loop).

Basic Printer Keyboard Configuration (Figure 22)

This configuration operates on-line with the 2030 or off-line except when the 2030 is in an emergency power-off condition.

Note: All configurations can operate off-line, even with normal power off. If the 2030 emergency-pull switch is operated, power to 1050 documentary console devices is lost. However, the CE can bypass the emergency-pull switch so that 1050 devices can be operated with the emergency-pull switch operated. If 1050 devices are so operated, in emergency-pull bypass, power is dropped to these devices when the emergency-pull switch is reset by the CE. Here, the CE must return the 1050 devices out of emergency-pull bypass mode.

Maximum Home-Loop Configuration (Figure 23)

With this configuration, the program can select and read from a 1054 or 1056 reader. Also, the following operator-initiated inputs to the 2030 can be accommodated:

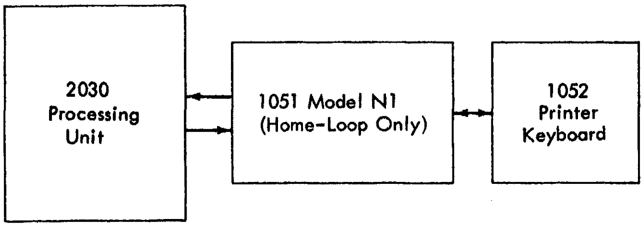


Figure 22. Basic Printer Keyboard Configuration

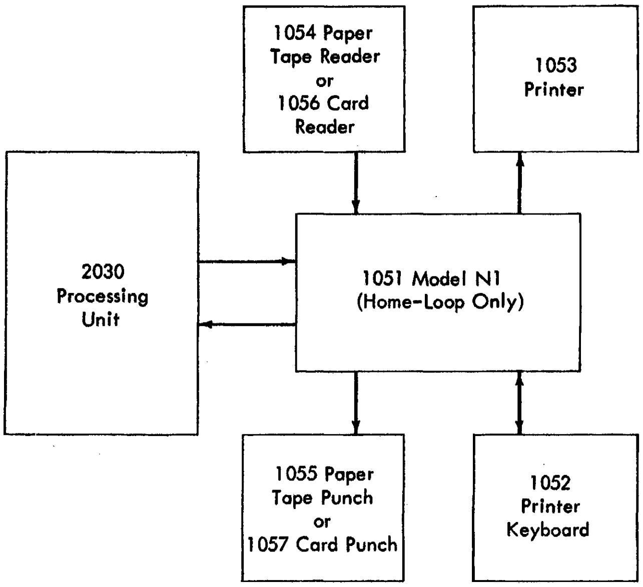


Figure 23. Maximum Home-Loop Configuration

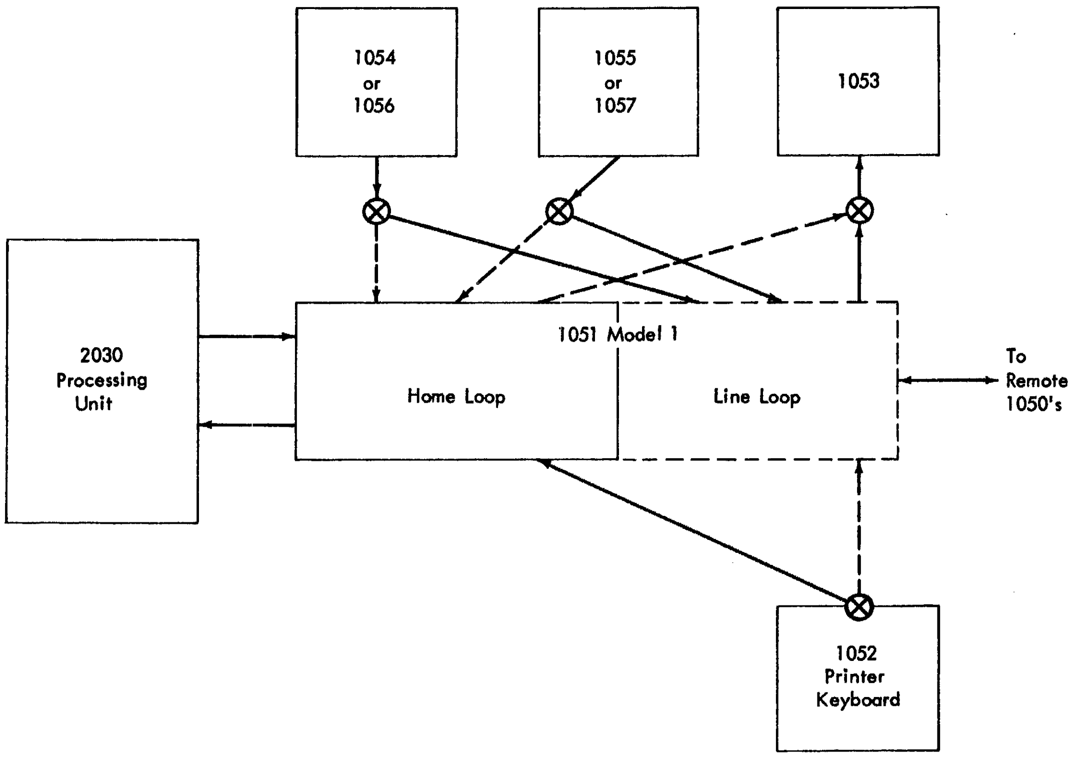
1. Keyboard to 2030.
2. 1054 Paper Tape Reader to 2030.
3. 1056 Card Reader to 2030.
4. Split input to 2030. That is, data is entered from the keyboard alternately with data from the 1054 or 1056 all in a single read operation.

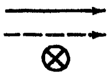
If the 1050 Home Component Recognition feature is used, the program can select one or any combination of the three types of outputs (1052, 1053, and 1055 or 1057, for example). It can change the output selection any place in the data stream during any single write operation. The program can change ribbon color and line-feed spacing at any point in the data stream when the 1051 Automatic Ribbon Shift and Line Feed Select feature is installed.

This complete 1050 system can operate off-line, performing any normal 1050 home-loop operations not requiring the 2030.

Off-Line Communications Configuration (Figure 24)

The home-loop and line-loop communications functions shown in Figure 24 can be performed simultaneously if the same I/O component is not required by both loops.





⊗ Switch Control (Position of ⊗ on this diagram indicates function but not necessarily physical location of switch).

Figure 24. Off-Line Communications Configuration

IBM 1050 Lights, Switches, and Functional Keys

The following describes the 1050 lights, switches, and functional keys for on-line 1050 home-loop operation.

The Keyboards shown in Figures 25, 26, 27, and 27.1 indicate graphics controlled by particular keys. The actual keyboards, however, do not show the lower-case alphabetic characters. The IBM 1050 Keyboard Printer models that can be used with System/360 Model 30 are:

1. Model 3 (Figure 25) is used whenever the IBM 1051 Model 1 Control Unit is used in the console configuration.
2. Model 5 (Figure 26) is used with the IBM Model 1051 N1 Control Unit when 1050 I/O components or features requiring switches (in addition to the 1052 printer keyboard) are used in the system.
3. Model 6 (Figure 27) is used with the IBM 1051 Model N1. Model 6 does not provide the switch-control capability for adding other 1050 I/O devices or features that require switches.

   The 1052 Model 6 can be obtained only on an "as available" basis. (See your IBM representative.)

   If the Systems Console Attachment Feature is installed in the 1051 Model N1 and in the 1052 Model 6, certain functions are inoperative:

   a. When a command is sent to the Model 6 to tab, backspace, or line feed, the Model 6 spaces.

   b. No action occurs when a ribbon shift and line feed select is received by the Model 6.

   c. Tab, backspace, and line feed keys do not function.

   Any programs that provide for any of these functions will operate, but the functions will not be performed.
4. The 1052 Model 8 (Figure 27.1) is used with the IBM 1051 Model N1 and replaces the 1052 Model 6 as the standard console keyboard. Model 8 provides the basic mechanisms and functions necessary in a system console environment. This model provides switch control for the basic IBM 1052 Printer-Keyboard, but it does not provide the capability for adding other 1050 I/O devices or features that require switches. The 1052 Model 8 uses only the commands used by the 1052 Model 6. Therefore, no additional coding is necessary to support the 1052 Model 8.

   The Tab, Backspace, and Line-feed keys are blank, and the functions associated with these keys are not provided in the 1052 Model 8. Also, no action occurs when a ribbon shift and line feed select is received by the 1052 Model 8. Any problem program written to control these functions operates with Model 8, even though the functions are not performed. Whenever one of the codes representing one of these functions is sent to the Model 8, a space with no printing occurs. Prefix codes are accepted by the 1051 with attached 1052 Model 8, but both the prefix character and the printable character in the prefix sequence are ignored.

   The following items are standard on the Model 8:

   a. A 13⅛ inch pin-feed platen,

   b. A 12½ inch (maximum) printing line,

   c. Six lines per inch line-feeding, and

   d. Character spacing of ten per inch.

   No additional special features are available for the Model 8.

   Also, the following functions and manual controls are removed or inoperative on the Model 8:

   a. Left and right margin set (the left and right margins are fixed),

   b. Single-double index lever,

   c. Paper release bar,

   d. Tab clear-set lever,

   e. Ribbon shift lever,

   f. End-of-line bell, and

   g. Pressure feed rolls

   The Systems Console Attachment Feature is required (in the 1051 Model N1) to attach a 1052 Model 8.

CPU Connect Switch

CPU On: In the CPU-on position, this switch connects the 1050 to the CPU. If all dc power is on in the 1051 when this switch is thrown to the ON position, a 1050 operational signal results. A transition from 1050 not-operational to 1050 operational initiates a 1050 ready interrupt with the device-end bit on in the Channel Status Word (CSW).

CPU Off: In the OFF position, this switch takes the 1050 system completely off-line. Any 1050 read or write commands are then rejected with condition code 3 (device not operational). When the CPU connect switch is returned to the home or ON position, a ready condition interrupt with device-end status is initiated.

Request Key

Pressing the request key causes an attention status to be established in the 1051 attachment. The 1051 attachment holds this attention status until the 1051 attachment becomes idle and available, at which time an attention interrupt is initiated in the CPU.

Proceed Light

The proceed light indicates that the 1051 attachment channel is available for operator-initiated keyboard and/or Reader-2 (that is, the other attached card or paper-tape reader, *not the 1052*) input through the 1051 home loop. When the proceed light is on, the keyboard is unlocked and an interlock is removed from Reader 2.

System Program/Duplicate Switch

Program Position: Two-character program-control sequences from any source cause the proper component control. The two-character sequences are not printed or punched. During a read command, neither character of a two-character control sequence is stored.

With the 1051 Home Component Recognition feature, an output other than the first printer must have its home-component-recognition latch turned on by a two-character program control sequence as well as having its assignment switch in the HOME position to satisfy the output select and ready interlock to the 1051 attachment during write commands.

Duplicate Position: With this switch in the DUPLICATE position—

1. Manual component assignment is required, and the 1051 Home Component Recognition special feature is not effective.
2. The prefix code is not printed or stored, but it is punched. The following numeric or alphabetic character is then stored, printed, and punched.
3. The output select and ready interlock to the 1051 attachment is not dependent upon any of the output home-component-recognition latches being on.

Auto-Fill Switch (with the 1051 Auto-Fill Character Generation Special Feature)

In the ON position, fill characters (idle code) are automatically generated by the 1051 during the execution time of printer functions such as new line and tab. Write commands are interlocked during the same period of time. On read commands, the idle codes are not read into storage.

In the OFF position, fill characters (idle code) are not generated.

System Attend/Unattend Switch (1051 Model 1 Only)

This switch must be in the ATTEND position for all on-line operations, or the 1051 will not indicate an operational condition to the 1051 attachment in the 2030.

1050 Intervention-Required Light (on 2030 System Control Panel)

This light is turned on whenever a command execution is terminated with an intervention-required condition. It is reset by the next 1050 read or write command, or by a 2030 system reset.

1050 Request Light (on 2030 System Control Panel)

This light is turned on whenever the 1050 request key on the 1052 is pressed. It is reset when attention status is recognized by the attachment and accepted into the unit status register.

I/O Assignment Switches

The I/O assignment switches transfer the various 1050 I/O devices to the desired 1050 loop, or they disconnect them from the 1050 completely. When the 1050 home loop is switched to on-line operation (CPU connect switch on), the devices that are to be made available to the CPU must have their switches in the HOME position.

Alternate-Code Key

When the alternate-code key is held down while a numeric key is pressed on the 1052, a 1050 control character is generated.

Alternate Code–Zero (Cancel): Whenever the alternate code key and the zero key are pressed, a unique cancel character is generated that terminates the keyboard entry with channel-end, device-end, and unit-exception statuses. The cancel character is not read into storage.

Alternate Code–Five (EOB): Whenever the alternate code key and the 5-key are pressed, an EOB (End-of-Block) character is generated. This initiates a normal end to the keyboard entry. The EOB character is not read into storage. The alternate code *Six* EOT (End-of-Transmission) produces the same functions as alternate code five (EOB).

For a description of other standard or optional 1050 lights, switches, and manual controls refer to *IBM 1050 Operator's Guide,* GA24-3125.

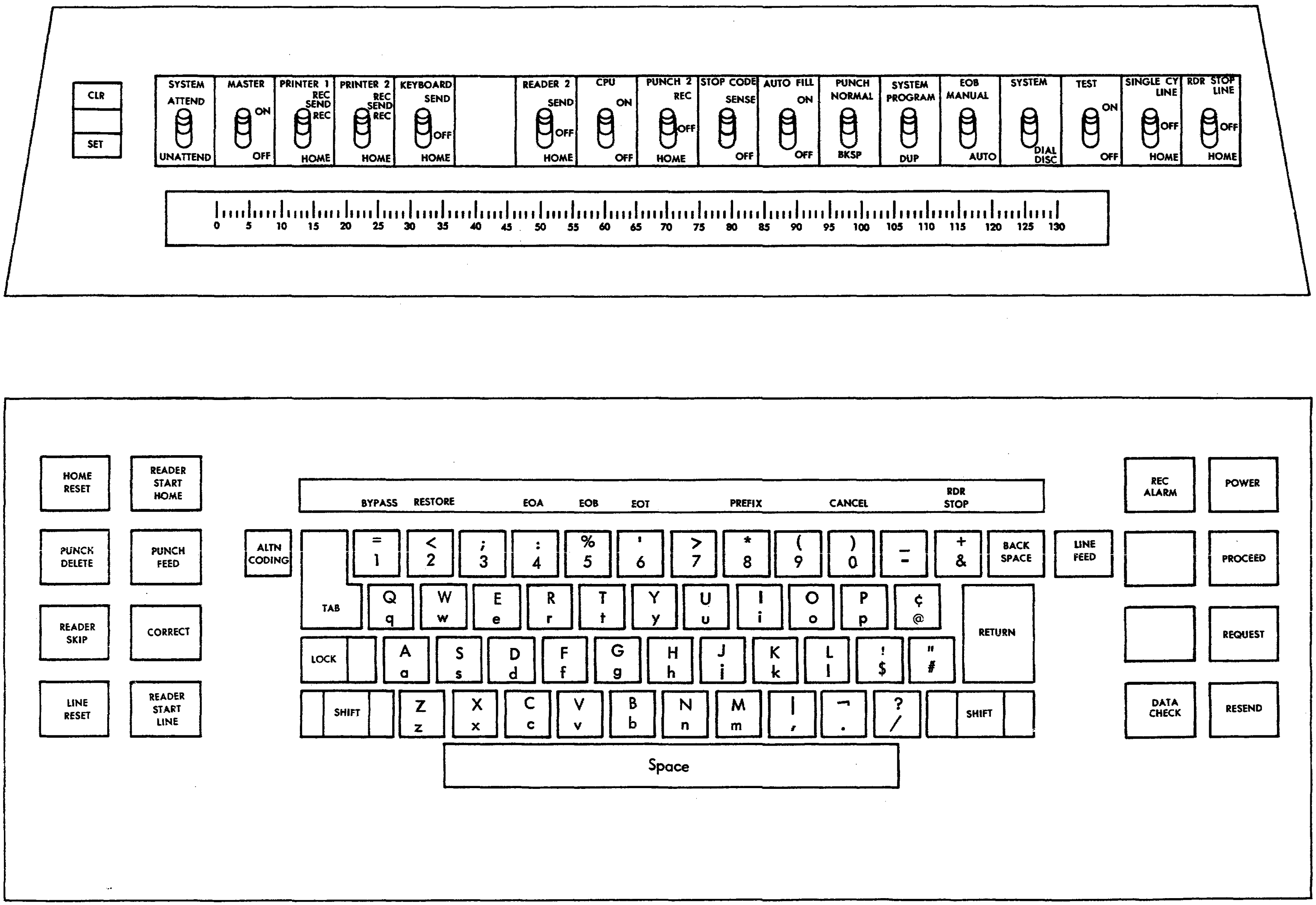


Figure 25. IBM 1052 Model 3 Keyboard and Switch Panel

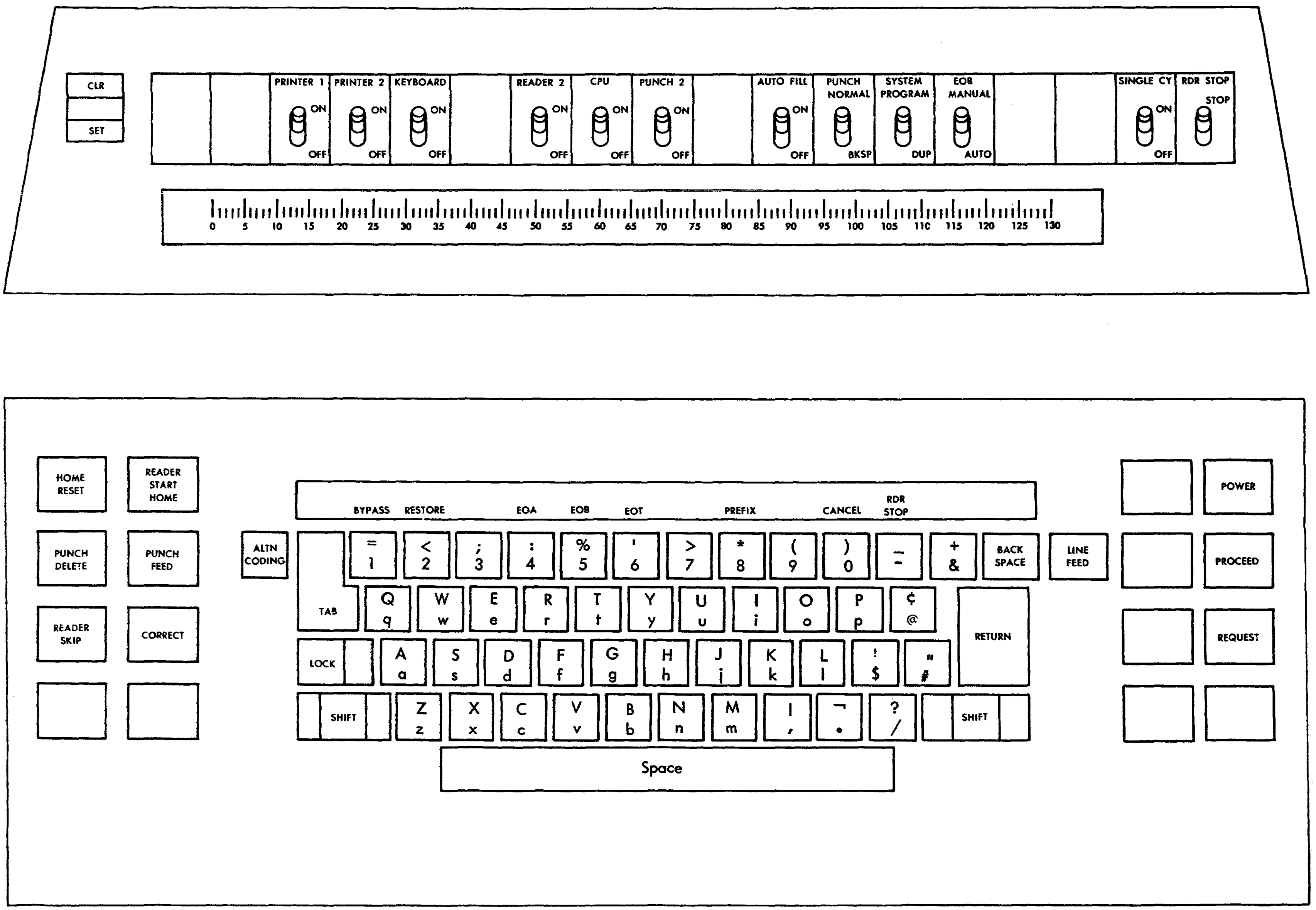


Figure 26. IBM 1052 Model 5 Keyboard and Switch Panel

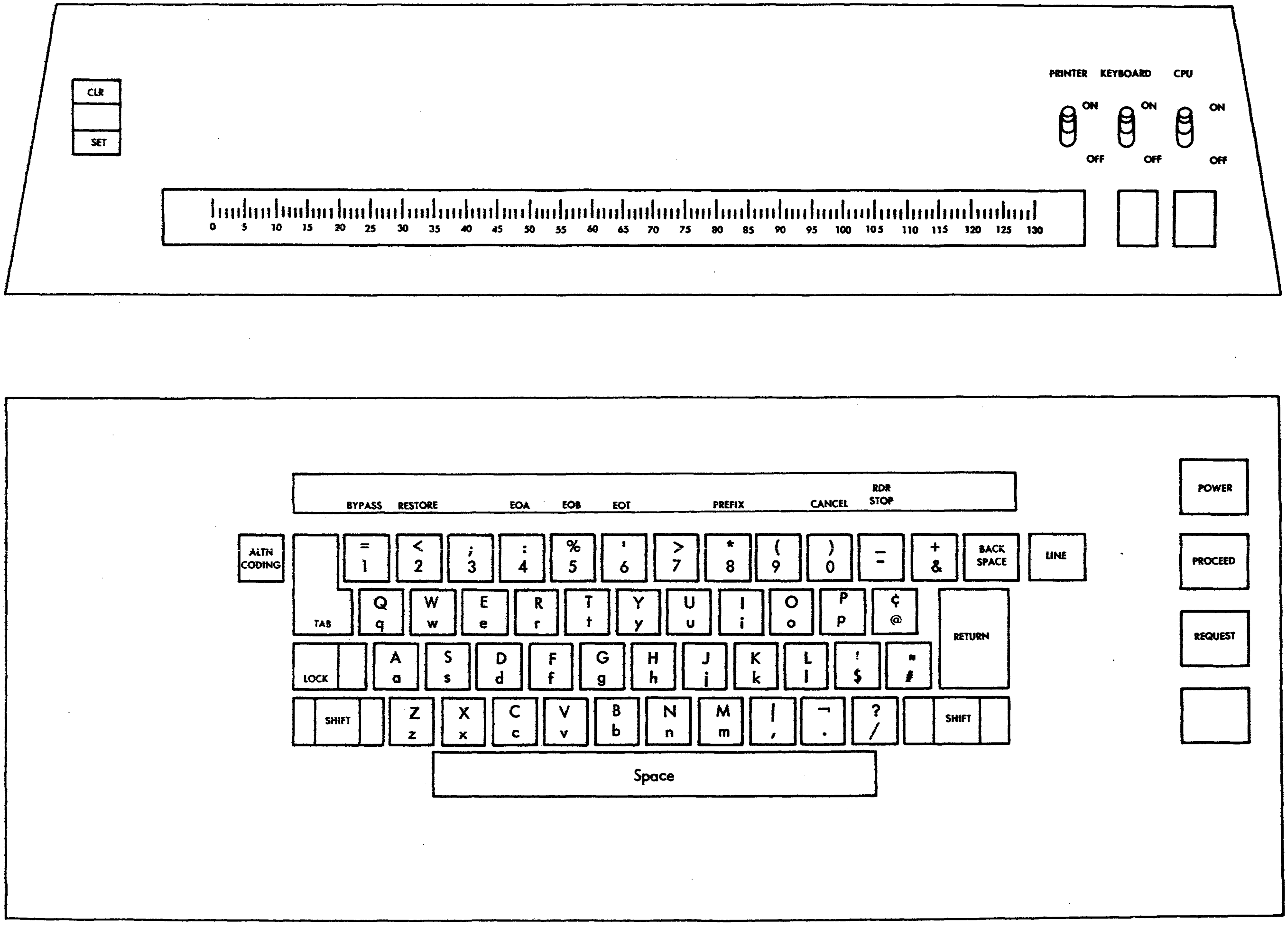


Figure 27. IBM 1052 Model 6 Keyboard and Switch Panel

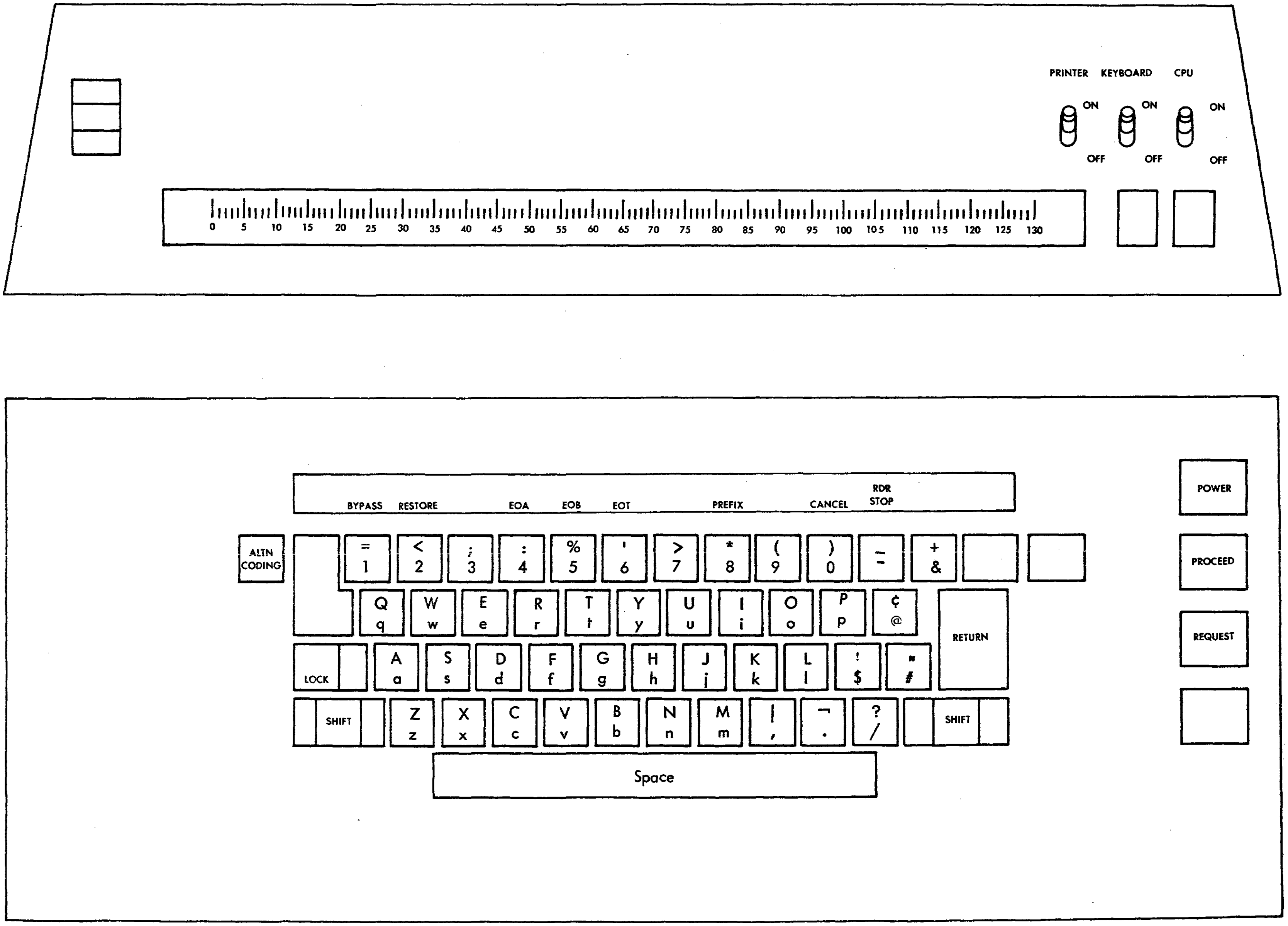


Figure 27.1 IBM 1052 Model 8 Keyboard and Switch Panel

Commands Initiated by Start I/O

The following is a list of the 1050 console home-loop commands executed from start I/O. The CCW command byte is shown.

Command	*CCW Command Byte*								
	Bits	0	1	2	3	4	5	6	7
Read Inquiry		0	0	0	0	1	0	1	0
Read Reader 2		0	0	0	0	0	0	1	0
Write		0	0	0	0	0	0	0	1
Write with Auto New Line		0	0	0	0	1	0	0	1
No-Op		0	0	0	0	0	0	1	1
Sense		0	0	0	0	0	1	0	0

The channel functions and checks associated with the START I/O instruction (CAW format checking, CCW format checking, etc.) are performed by the normal multiplexor channel microprograms. The data transfer, status manipulation, 1050 interface control, and control-unit and device-level functions are performed by unique 1050 console microprograms.

The following command descriptions, in general, cover the control-unit or device-level execution of initial selection, run or data transfer, and ending procedures.

Read Inquiry Command

This command is normally issued in response to an operator-initiated attention interrupt from the 1050 console. The operator initiates this by pressing the request key.

Initial Selection

If the 1050 power is off, or the 1050 CPU connect switch is in the OFF position (see switch interpretation under *IBM 1050 Lights, Switches, and Functional Keys*) or if the attachment is in CE mode, the START I/O instruction is terminated with condition code 3 (device not operational).

If the 1050 power is on, the CPU connect switch is in the ON position, and the attachment is in run mode and idle:

1. The read-inquiry command is initiated and the operation is terminated with condition code 0.
2. The proceed light is turned on at the 1052.
3. The 1052 keyboard is unlocked, and a holding or interlock condition is removed from Reader 2.

Operator Initiation of Inquiry

When the proceed light comes on, the operator can:

1. Enter data from the keyboard after making sure that the keyboard switch is in the HOME or ON position.
2. Enter data from Reader 2 by pressing the home-reader start key. The Reader-2 switch must be in the HOME or ON position.
3. Enter split data: in other words, enter data alternately from the keyboard and reader (either paper tape or card).

Either keyboard data or reader data can be entered first but for this example let us assume keyboard data is entered first. The desired data is entered from the 1052 keyboard. Data entry from the reader is then initiated by pressing the home-reader start key on the 1052 (the Reader-2 switch on the 1052 must be in the HOME or ON position). Once the reader start key is pressed, the 1052 keyboard is locked. Data transfer from the reader is stopped and the keyboard is unlocked by either a reader stop code punched in the card (or perforated tape) or by pressing the home-reader stop key on the 1052. This process can be repeated until an ending operation is initiated from either the keyboard, reader, or channel.

Operation Checking

During the read inquiry operation, the attachment is checking for the following conditions:

1. 1050 power on, and
2. CPU connect switch on.

If any one of these conditions is not satisfied, the operation is terminated with an I/O interrupt with channel end, device end, and unit check in the CSW. A subsequent sense operation indicates intervention-required (bit 1 of the sense byte). The 1050 intervention-required light is turned on.

The keyboard is then locked, the proceed light is turned off, and a hold condition is returned to Reader 2. An automatic carrier return and line feed are attempted to monitoring printers, *attempted* because conditions may prevent the carrier return and line feed (for example, 1050 power off).

End Operation: End of Block Character (Normal End)

From Keyboard

The End-of-Block (EOB) character is supplied from the keyboard by pressing first the alternate code key and then the 5-key while still holding down the alternate code key. EOT (that is, 6-key) functions in the same manner as EOB.

From Reader 2

For paper-tape reading, the EOB character must be punched in paper tape immediately after the last data character to be transferred to storage. For card reading, the EOB character can be punched in the card or generated by the trailing edge of the card (whichever occurs first).

The EOB character is not read into storage. Channel-end and device-end status and an interrupt condition are established. The proceed light is turned off. The keyboard is locked. A hold condition is placed on Reader 2. An automatic carrier return and line feed is initiated to any 1050 printer copying the input. The EOT (End-of-Transmission) character (alternate code/ 6) is an alternative to EOB and functions in the same manner as EOB.

End Operation: CCW Byte Count Zero, No Data Chaining (CDA)

This is a channel-initiated end. If the data was being entered from the keyboard, the proceed light is turned off, the keyboard is locked, a hold condition is placed on Reader 2, and channel-end and device-end status and an interrupt condition are established. An automatic carrier return and line feed is initiated to any 1050 printer copying the input.

If the data was being read from Reader 2, a channel-end status and an interrupt condition are established. The attachment remains busy until EOB is detected. No more data characters are then transferred to storage. If the count was N, and data is from the keyboard, the N+1 character prints (if a printable character) but is not transferred to storage. If data was being entered from Reader 2, all printable characters up to EOB/EOT print (unless the printers are deselected or in the bypass mode).

At EOB, device-end status and an interrupt condition are established. The proceed light is turned off. The keyboard remains locked. A hold condition is placed on Reader 2. An automatic carrier return and line feed is initiated to any 1050 printer copying the input.

End Operation: Cancel Operation

A cancel operation can be initiated from the keyboard during a read inquiry at any time after the proceed light comes on, provided the keyboard still has control (the reader start key has not been pressed). A cancel operation is executed by pressing the alternate code key and the zero key while still holding the alternate code key down.

The read inquiry operation is then terminated. Channel-end, device-end, and unit-exception status and an interrupt condition are established. The proceed light is turned off. The keyboard is locked, a hold condition is placed on Reader 2, and an automatic carrier return and line feed is initiated. The cancel character is not transferred to storage. Programming determines what is done to the data characters transferred to storage *before* the cancel character is generated.

Read Reader 2 Command

This read command is a non-operator solicited read command directed specifically to Reader 2.

Initial Selection

If any of the following conditions exist, the START I/O instruction is terminated with condition code 3 (device not operational):

1. 1050 power off or 1050 CPU connect switch off.
2. Reader-2 switch is not in the HOME or ON position, or Reader-2 interlocks are not satisfied.
3. Attachment is in the CE mode.

If the above conditions do not exist (1050 is operational and Reader 2 is selected and ready), the read-reader-2 command is accepted and initiated and:

1. The START I/O instruction is terminated with condition code 0.
2. Reader 2 is automatically started.
3. The keyboard remains locked.

Operation checking

During the read-reader-2 operation, the attachment is checking for the following conditions:

1. 1050 power on,
2. CPU connect switch on,
3. Reader-2 switch ON or HOME, and Reader-2 interlocks satisfied.

If any of these conditions are not satisfied, the operation is terminated with channel-end, device-end, and unit-check. Intervention required is presented in bit 1 of the sense byte if a subsequent sense operation is executed. The 1050 intervention-required light is turned on. Reader 2 is stopped without advancing to the EOB character. An automatic carrier return and line feed are attempted to any 1050 printers selected to copy the input.

End Operation

1. *End of Block (EOB) Character.* For paper tape, the EOB character must be punched in paper tape. For cards, the EOB character can be punched in the card or is generated by the trailing edge of the card (whichever occurs first).

   The EOB character is not read into storage. Channel-end and device-end status and an interrupt condition are established (assuming no command chaining). Reader 2 is stopped and a hold condition is returned to it. The keyboard remains locked. An automatic carrier return and line feed is initiated to any 1050 printers selected to copy the input. EOT functions in the same manner as EOB.
2. *CCW Byte Count Zero (No CDA).* This is a channel-initiated end. A channel-end status and an interrupt condition are established. The attachment remains busy until EOB is detected. No more data characters are then transferred to storage.

   At EOB, device-end status and an interrupt condition are established. Reader 2 is stopped and a hold condition is returned to it. The keyboard remains locked. An automatic carrier return and line feed are initiated to any 1050 printers selected to copy the input.

Write Commands

Initial Selection

If any of the following conditions exist, the START I/O instruction is terminated, rejecting the write command with condition code 3 (device not operational):

1. 1050 power is off or 1050 CPU connect switch is off.
2. The attachment is in CE mode.

If the foregoing conditions do not exist (1050 is operational and home-loop is on-line), the write command is accepted and initiated and:

1. The START I/O instruction is terminated with condition code 0.
2. The keyboard remains locked.
3. A hold condition is maintained on Reader 2.

Operation Checking

During write operations, the attachment is checking for the following conditions:

1. 1050 power on,
2. CPU connect switch on,
3. Any output selected and ready.

If any of these conditions are not satisfied, the operation is terminated with the channel-end, device-end, and unit-check. A subsequent sense operation indicates intervention required (sense bit 1). If card punching was in progress, the card is not released. An automatic carrier return and line feed are attempted to any 1050 printers selected to copy.

End Operation

The end operation is initiated by the CCW byte count going to zero. Channel-end and device-end status and an interrupt condition are established (assuming no command or data chaining). If an automatic carrier return and line feed was specified in the write command, it is initiated to any 1050 printer selected to copy.

If a card punch was selected, one of the following must be provided at the end of the data field by the program to release the last card:

1. EOB character (pluggable option in the 1057 to function – release – but not punch, or function – release – and punch in the card).
2. A prefix character followed by an H.

Carrier Return

When operating on-line, if the carrier reaches the right margin without a carrier return signal from the program, the carrier returns automatically and a single line feed occurs. Printing is suppressed during the carrier return.

Sense Command

There are no 1050 conditions to be tested for acceptance of a sense command. The sense byte is assembled by the attachment and is transferred to storage, and the operation is terminated. Channel-end and device-end status and an interrupt condition are established.

Sense Byte

Bit	*Condition*
0	*Command Reject* — a command not valid to the 1050 console was detected during the previous START I/O instruction, or an attempt was made to command-chain.
1	*Intervention Required* — The previous command could not be performed or completed, and some type of manual intervention is required. Conditions such as the following cause intervention required: 1. 1050 power is off. 2. No output device is selected or ready (write command). 3. CPU connect switch is off. 4. Attachment is in CE mode. 5. Reader-2 is not selected and ready (read Reader-2 command). 6. Attend/unattended switch is in UNATTEND position (1051 Model 1 only).
2	*Bus-Out Check — Not used.*
3	*Equipment Check* — A data error was detected during a read operation, or an equipment malfunction was detected.
4-7	*Not used.*

NO OP Command

The NO OP is a control-immediate-type command that performs no 1050 attachment or device level function. If the subchannel and 1050 attachment are not busy, the NO OP is executed. No additional 1050 attachment or device level checking is performed (that is, the 1050 need not be operational for the NO OP to be executed). If the command chaining flag is not on, condition code 1 is set and channel end and device end statuses are presented in the channel status word (CSW). If the command-chaining flag is on, condition code 0 is set and chaining is subsequently attempted to the next command.

1057 Considerations

The 1057 remains in upper- or lower-case mode at the end of each operation, depending upon the case of the last character punched. If the 1057 is left in upper-case mode (at the end of an operation) and the first character of the next write command is a lower-case character, the 1057 punches the intended lower-case character in upper-case code.

Therefore, programs should be written so that the last character of each message is known. Alternately, a downshift character may be entered as the first character of each message.

Home Component Recognition

This function requires the Home Component Recognition feature in the 1051. With the Home Component Recognition feature and the 1051 program duplicate switch in the PROGRAM position, the program can select and deselect any combination of 1050 output devices in the home loop and change the selection at any point in a given write operation.

Write Commands

With the Home Component Recognition feature in the 1051, the write commands function as follows.

Initial Selection Initiated by Start I/O Instruction

Printer 1 (either 1052 or 1053) is automatically selected.

Execution of Write Commands

If the write data is to be directed to an output device or devices other than Printer 1, the initial data bytes must consist of the required prefix-character numeric-character (recognition codes) sequences to turn on the desired output devices followed by prefix character and numeric 5 to turn off Printer 1.

As an example, assume the output data is to be copied on Printer 2 and Punch 2.

The output data, then, is: Prefix 2 Prefix 4 Prefix 5 TEXT. Printer 1 is turned off by the Prefix 5 sequence. The text output is copied by Printer 2 and Punch 2.

If the write data is to be directed to another output device or devices in addition to Printer 1, the initial data bytes must consist of the required prefix character numeric-character sequences to turn on the desired output devices.

As an example, assume the output data is to be copied on Printer 1 and Punch 2.

The output data is Prefix 4 TEXT. The text is copied by Printer 1 and Punch 2.

Output-device switching can be initiated at any point within the data stream of a given write command regardless of data chaining. The new output devices must be prefixed on before all of the previous output devices are prefixed off.

Output-select and ready-testing are performed during the execution of all write commands. If at any point during the execution, no output device indicates select and ready (not switched to HOME or ON, home-component-recognition latches not on, or interlocks not satisfied), the operation is terminated with unit-check status.

End Operation

At the end of each write command, all 1050 output devices are deselected (all output home-component select latches are reset off).

Read Inquiry Command

With the Home Component Recognition feature active in the 1051, the read inquiry command functions as follows.

Initial Selection Initiated by Start I/O Instruction

Printer 1 (either 1052 or 1053) is automatically selected for monitoring.

Execution of Read Inquiry Command

If the operator desires another or an additional output to monitor during the inquiry operation, he can:

1. Prefix outputs on and off from the keyboard before entering data, or
2. Put the PROG/DUP switch in the DUP mode and manually switch-select the desired monitoring outputs. (He must return the switch to program mode before entering EOB.)

If the operator presses the home reader start key to enter his inquiry from Reader 2, the adapter continuously tests for Reader 2 select-and-ready during that portion of the inquiry operation.

End Operation

At the end of each read inquiry command, all 1050 output devices are deselected.

Read Reader-2 Command

With the Home Component Recognition feature active in the 1051, the read reader-2 command automatically selects Printer 1 for monitoring the input. If no monitoring is desired, the prefix sequence to deselect the printer must be punched in the first card for each command or the Printer 1 assignment switch must be turned off.

Off-Line 1050 Functions

Home Component Recognition operates in its normal manner when the 1050 is operated off-line.

IBM 1050 I/O Prefix Selection

Printer 1 on	Prefix 1
Printer 2 on	Prefix 2
Punch 2 on	Prefix 4
Printer 1 off	Prefix 5
Printer 2 off	Prefix 6
Punch 2 off	Prefix 8
*Ribbon Shift up	Prefix A
*Ribbon Shift down	Prefix B
*Single-Line Feed	Prefix C
*Double-Line Feed	Prefix D

*Requires Automatic Ribbon Shift and Line Feed Select feature in the 1051.

Note: When the 1050 is operating on-line, Reader 1, Reader 2, and Punch 1 are always automatically selected ON.

Miscellaneous Operations

Write to Card Punch

It is possible to execute one or multiple chained or unchained write commands to the same card. If a write command of N characters (N<80) is issued and the last character is not an EOB character, N columns are punched and the card stops in the punch station in position to punch the N + 1 column.

The last write command to any given card must include an EOB character (in the data stream) after the last character to be punched to release the card. In other words, if the last write command consisted of M characters to be punched, the M + 1 character must be an EOB and the CCW count must be M + 1.

It is also possible to punch multiple cards during a single write command with or without data chaining. For example, assume that the write-data stream consisted of N characters—EOB—M characters—EOB, etc. N characters would be punched in the first card, and M characters would be punched in the second card, etc. The attachment would be interlocked to prevent writing by the 1051 during the release of one card and the registering of the next card.

As alternatives to the EOB character, either the EOT character or the Prefix H control sequence also causes the card to be released.

On the standard card punch, a patch panel accessible to the customer engineer permits assignments of the 1050 control codes *as a group* to perform as follows:

1. Function only,
2. Punch only,
3. Function and Punch,
4. No action.

The Operator Panel special feature for the card punch allows the above-mentioned assignments on an individual control character basis by the customer.

Operations Terminated by Intervention Required

If the execution of a command is terminated by an intervention-required status:

1. An automatic carrier return and line feed is attempted but may not be executed to monitoring printers.

   The carrier may have to be returned from the keyboard off-line after removing the cause of the intervention-required condition and before returning the 1050 to on-line operational.

2. If paper tape was being read, the paper tape reader stops within the record being read. The paper tape must be manually repositioned after removing the cause of the intervention-required condition and before returning the 1050 to on-line operational.

3. If cards were being punched or read, the punch or reader stops within the card being processed. The cards must be manually released or ejected.

Test I/O

If 1050 power is off, the CPU connect switch is off or the attachment is in CE mode, the test I/O command results in condition code 3 (device not operational). In all other respects, test I/O is executed as generally defined in *IBM System/360 Principles of Operation,* GA 22-6821.

Halt I/O

Keyboard Entry

If a halt I/O command is executed during a keyboard entry, the operation is terminated immediately. The keyboard is locked, and the proceed light is turned off. Channel-end and device-end status are established and an automatic carrier-return and line feed is initiated. Any chaining flags are turned off.

Write

If a halt I/O command is executed during a write command, the current write command is terminated immediately. Channel-end and device-end status and an interrupt condition are established. An automatic carrier return and line feed are initiated. Any chaining flags are turned off.

Read from Reader 2

If a halt I/O command is executed during a read command, the data transfer is terminated and channel-end status and an interrupt condition are established. The attachment and 1050 remain busy until the normal end of media is reached, at which time device-end and an interrupt condition are established. Any chaining flags are turned off. An automatic carrier return and line feed are initiated to any monitoring printers.

Condition Code Settings for 1050 Documentary Console

Figure 28 indicates the condition code settings that are dependent upon the command issued and the condition of the 1050.

CONDITION → SUB CHANNEL	SUBCHANNEL AVAILABLE							SUBCHANNEL NOT AVAILABLE		
1050 and ATTACHMENT → INSTRUCTION ↓	Idle and Available	Attention or Device-End Status Stored in Attachment and Pending	Busy Searching For Device End	CPU Connect Switch Off or 1050 Power Off or Attachment In CE Mode	Reader-2 Not Selected and Ready	Attention or Ready Condition at 1051 Interface But Not Yet Stored in Unit Status	Command Invalid	Busy	Interupt Status Stored in Attachment	Interupt Status Stored in Channel
START I/O										
READ RD 2	0	1 (b)	1 (c)	3	3	0 (f)	1 (g)	2	2	2
READ INQ	0	↓	↓	3	0 (e)	↓	↓	↓	↓	↓
WRITE	0	↓	↓	3	0 (e)	↓	↓	↓	↓	↓
NO-OP (With no CMD Chaining)	1 (a)	↓	↓	1 (a)	0 (e)	↓	↓	↓	↓	↓
NO-OP (With CMD Chaining)	0	↓	↓	0	0 (e)	↓	↓	↓	↓	↓
SENSE	0	↓	↓	0	0 (e)	↓	↓	↓	↓	↓
TEST I/O	0	1 (d)	1 (c)	3	0 (e)	1 (d)	N.A.	2	1 (h)	1 (h)
HALT I/O	1 (i)	1 (i)	1 (i)	3	1 (i)	1 (i)	N.A.	1 (i)	1 (i)	0

Notes:

(a) Device-end and channel-end unit status in CSW.

(b) Busy (+ attention or device end) status in CSW.

(c) Busy unit status alone status in CSW.

(d) Attention or device-end unit status in CSW.

(e) Condition not tested as part of instruction execution.

(f) Request condition is stored until the attachment becomes idle and available again, and attention status can be established.

(g) Unit-check unit status in CSW and command reject in sense register.

(h) Channel-end status alone or in combination with other unit-status bits (device end, unit check, unit exception) in CSW.

(i) All-zero status stored in CSW. Final status will be presented subsequently.

N.A. = Not applicable

Figure 28. Condition Code Chart for 1050 Documentary Console

The following automatic translation is performed between the PTTC/EBCD (Perforated Tape and Transmission Code/Extended Binary Coded Decimal) and the System/360 EBCDI code. For the PTTC/EBCD code refer to Figure 29.

Input

The following codes are translated on input:

26 Lower-case alphabetic (a through z)
26 Upper-case alphabetic (A through Z)
10 Numeric (0 through 9)
26 Special graphics #@/,-$&.=<;:%'>*)"(¢?+ !_ |¬
4 Printer control codes (line feed, new line, tab, backspace)
1 Space/blank
93 Total

All other 1050 control codes are deleted and not entered into storage.

Output

The following codes are translated on output:

26 Lower-case alphabetic (a through z)
26 Upper-case alphabetic (A through Z)
10 Numeric (0 through 9)
26 Special graphics #@/,-$&.=<;:%'>*)"(¢?+ —!|¬
1 Space/blank
16 1050 Control codes
105 Total

UPPER CASE						
	PTTC/EBCD					
Character	B	A	8	4	2	1
Space						
A	B	A				1
B	B	A			2	
C	B	A			2	1
D	B	A		4		
E	B	A		4		1
F	B	A		4	2	
G	B	A		4	2	1
H	B	A	8			
I	B	A	8			1
J	B					1
K	B				2	
L	B				2	1
M	B			4		
N	B			4		1
O	B			4	2	
P	B			4	2	1
Q	B		8			
R	B		8			1
S		A			2	
T		A			2	1
U		A		4		
V		A		4		1
W		A		4	2	
X		A		4	2	1
Y		A	8			
Z		A	8			1
+	B	A				
¬	B	A	8		2	1
_	B					
!	B		8		2	1
¢		A				
?		A				1
|		A	8		2	1
=						1
<					2	
;					2	1
:				4		
%				4		1
'				4	2	
>				4	2	1
*			8			
(			8			1
)			8		2	
"			8		2	1
PF (Punch Off)	B	A	8	4		
HT (Horizontal Tab)	B	A	8	4		1
LC (Lower Case)	B	A	8	4	2	
DEL (Delete)	B	A	8	4	2	1
RES (Restore)	B		8	4		
NL (New Line)	B		8	4		1
BS (Backspace)	B		8	4	2	
IL (Idle)	B		8	4	2	1
BYP (Bypass)		A	8	4		
LF (Line Feed)		A	8	4		1
EOB (End of Block)		A	8	4	2	
PRE (Prefix)		A	8	4	2	1
PN (Punch On)			8	4		
RS (Reader Stop)			8	4		1
UC (Upper Case)			8	4	2	
EOT (End of Transmission)			8	4	2	1

LOWER CASE						
	PTTC/EBCD					
Character	B	A	8	4	2	1
Space						
a	B	A				1
b	B	A			2	
c	B	A			2	1
d	B	A		4		
e	B	A		4		1
f	B	A		4	2	
g	B	A		4	2	1
h	B	A	8			
i	B	A	8			1
j	B					1
k	B				2	
l	B				2	1
m	B			4		
n	B			4		1
o	B			4	2	
p	B			4	2	1
q	B		8			
r	B		8			1
s		A			2	
t		A			2	1
u		A		4		
v		A		4		1
w		A		4	2	
x		A		4	2	1
y		A	8			
z		A	8			1
&	B	A				
.	B	A	8		2	1
-	B					
$	B		8		2	1
@		A				
/		A				1
,		A	8		2	1
#			8		2	1
1						1
2					2	
3					2	1
4				4		
5				4		1
6				4	2	
7				4	2	1
8			8			
9			8			1
0			8		2	
PF (Punch Off)	B	A	8	4		
HT (Horizontal Tab)	B	A	8	4		1
LC (Lower Case)	B	A	8	4	2	
DEL (Delete)	B	A	8	4	2	1
RES (Restore)	B		8	4		
NL (New Line)	B		8	4		1
BS (Backspace)	B		8	4	2	
IL (Idle)	B		8	4	2	1
BYP (Bypass)		A	8	4		
LF (Line Feed)		A	8	4		1
EOB (End of Block)		A	8	4	2	
PRE (Prefix)		A	8	4	2	1
PN (Punch On)			8	4		
RS (Reader Stop)			8	4		1
UC (Upper Case)			8	4	2	
EOT (End of Transmission)			8	4	2	1

Figure 29. Perforated Tape and Transmission Code/Extended Binary Coded Decimal

Extended Binary Coded Decimal Interchange Code (EBCDIC)

Hexa-deci-mal	Graphic & Con-trol Symbols EBCDIC	Punched Card Code	System/360 8-bit Code
00		12-0-1-8-9	0000 0000
01		12-1-9	0000 0001
02		12-2-9	0000 0010
03		12-3-9	0000 0011
04	PF	12-4-9	0000 0100
05	HT	12-5-9	0000 0101
06	LC	12-6-9	0000 0110
07	DEL	12-7-9	0000 0111
08		12-8-9	0000 1000
09		12-1-8-9	0000 1001
0A		12-2-8-9	0000 1010
0B		12-3-8-9	0000 1011
0C		12-4-8-9	0000 1100
0D		12-5-8-9	0000 1101
0E		12-6-8-9	0000 1110
0F	CU1	12-7-8-9	0000 1111
10		12-11-1-8-9	0001 0000
11		11-1-9	0001 0001
12		11-2-9	0001 0010
13		11-3-9	0001 0011
14	RES	11-4-9	0001 0100
15	NL	11-5-9	0001 0101
16	BS	11-6-9	0001 0110
17	IL	11-7-9	0001 0111
18		11-8-9	0001 1000
19		11-1-8-9	0001 1001
1A	CC	11-2-8-9	0001 1010
1B		11-3-8-9	0001 1011
1C		11-4-8-9	0001 1100
1D		11-5-8-9	0001 1101
1E		11-6-8-9	0001 1110
1F	CU2	11-7-8-9	0001 1111
20		11-0-1-8-9	0010 0000
21		0-1-9	0010 0001
22		0-2-9	0010 0010
23		0-3-9	0010 0011
24	BYP	0-4-9	0010 0100
25	LF	0-5-9	0010 0101
26	EOB	0-6-9	0010 0110
27	PRE	0-7-9	0010 0111
28		0-8-9	0010 1000
29		0-1-8-9	0010 1001
2A	SM	0-2-8-9	0010 1010
2B		0-3-8-9	0010 1011
2C		0-4-8-9	0010 1100
2D		0-5-8-9	0010 1101
2E		0-6-8-9	0010 1110
2F	CU3	0-7-8-9	0010 1111
30		12-11-0-1-8-9	0011 0000
31		1-9	0011 0001
32		2-9	0011 0010
33		3-9	0011 0011
34	PN	4-9	0011 0100
35	RS	5-9	0011 0101
36	UC	6-9	0011 0110
37	EOT	7-9	0011 0111
38		8-9	0011 1000
39		1-8-9	0011 1001
3A		2-8-9	0011 1010
3B		3-8-9	0011 1011
3C		4-8-9	0011 1100
3D		5-8-9	0011 1101
3E		6-8-9	0011 1110
3F		7-8-9	0011 1111

Hexa-deci-mal	Graphic & Con-trol Symbols EBCDIC	Punched Card Code	System/360 8-bit Code
40	SP	no punches	0100 0000
41		12-0-1-9	0100 0001
42		12-0-2-9	0100 0010
43		12-0-3-9	0100 0011
44		12-0-4-9	0100 0100
45		12-0-5-9	0100 0101
46		12-0-6-9	0100 0110
47		12-0-7-9	0100 0111
48		12-0-8-9	0100 1000
49		12-1-8	0100 1001
4A	¢	12-2-8	0100 1010
4B	.	12-3-8	0100 1011
4C	<	12-4-8	0100 1100
4D	(	12-5-8	0100 1101
4E	+	12-6-8	0100 1110
4F	|	12-7-8	0100 1111
50	&	12	0101 0000
51		12-11-1-9	0101 0001
52		12-11-2-9	0101 0010
53		12-11-3-9	0101 0011
54		12-11-4-9	0101 0100
55		12-11-5-9	0101 0101
56		12-11-6-9	0101 0110
57		12-11-7-9	0101 0111
58		12-11-8-9	0101 1000
59		11-1-8	0101 1001
5A	!	11-2-8	0101 1010
5B	$	11-3-8	0101 1011
5C	*	11-4-8	0101 1100
5D	)	11-5-8	0101 1101
5E	;	11-6-8	0101 1110
5F	¬	11-7-8	0101 1111
60	-	11	0110 0000
61	/	0-1	0110 0001
62		11-0-2-9	0110 0010
63		11-0-3-9	0110 0011
64		11-0-4-9	0110 0100
65		11-0-5-9	0110 0101
66	,	11-0-6-9	0110 0110
67	%	11-0-7-9	0110 0111
68		11-0-8-9	0110 1000
69		0-1-8	0110 1001
6A		12-11	0110 1010
6B		0-3-8	0110 1011
6C		0-4-8	0110 1100
6D	_	0-5-8	0110 1101
6E	>	0-6-8	0110 1110
6F	?	0-7-8	0110 1111
70		12-11-0	0111 0000
71		12-11-0-1-9	0111 0001
72		12-11-0-2-9	0111 0010
73		12-11-0-3-9	0111 0011
74		12-11-0-4-9	0111 0100
75		12-11-0-5-9	0111 0101
76		12-11-0-6-9	0111 0110
77		12-11-0-7-9	0111 0111
78		12-11-0-8-9	0111 1000
79		1-8	0111 1001
7A	:	2-8	0111 1010
7B	#	3-8	0111 1011
7C	@	4-8	0111 1100
7D	'	5-8	0111 1101
7E	=	6-8	0111 1110
7F	"	7-8	0111 1111

Control Character Representations

ACK	Acknowledge	EOT	End of Transmission	RES	Restore
BEL	Bell	ESC	Escape	RS	Reader Stop
BS	Backspace	ETB	End of Transmission Block	SI	Shift In
BYP	Bypass	ETC	End of Text	SM	Set Mode
CAN	Cancel	FF	Form Feed	SMM	Start of Manual Message
CC	Cursor Control	FS	Field Separator	SO	Shift Out
CR	Carriage Return	HT	Horizontal Tab	SOH	Start of Heading
CU1	Customer Use 1	IFS	Interchange File Separator	SOS	Start of Significance
CU2	Customer Use 2	IGS	Interchange Group Separator	SP	Space
CU3	Customer Use 3	IL	Idle	STX	Start of Text
DC1	Device Control 1	IRS	Interchange Record Separator	SUB	Substitute
DC2	Device Control 2	IUS	Interchange Unit Separator	SYN	Synchronous Idle
DC3	Device Control 3	LC	Lower Case	TM	Tape Mark
DC4	Device Control 4	LF	Line Feed	UC	Upper Case
DEL	Delete	NAK	Negative Acknowledge	VT	Vertical Tab
DLE	Data Link Escape	NL	New Line		
DS	Digit Select	NUL	Null		
EM	End of Medium	PF	Punch Off		
ENQ	Enquiry	PN	Punch On		

Special Graphic Characters

¢	Cent Sign	-	Minus Sign, Hyphen	⑀	Hook
.	Period, Decimal Point	/	Slash	⑂	Fork
<	Less-than Sign	,	Comma	⑁	Chair
(	Left Parenthesis	%	Percent		
+	Plus Sign	_	Underscore		
|	Logical OR, Absolute	>	Greater-than Sign		
&	Ampersand	?	Question Mark		
!	Exclamation Point	:	Colon		
$	Dollar Sign	#	Number Sign		
*	Asterisk	@	At Sign		
)	Right Parenthesis	'	Prime, Apostrophe		
;	Semicolon	=	Equal Sign		
¬	Logical NOT	"	Quotation Mark		
[	Left Bracket	\	Backslash		
]	Right Bracket	∧	Circumflex		

Figure 30. Extended Binary Coded Decimal Interchange Code (EBCDIC), Part 1 of 2

Hexa-decimal	Graphic & Control Symbols EBCDIC	Punched Card Code	System/360 8-bit Code
80		12-0-1-8	1000 0000
81	a	12-0-1	1000 0001
82	b	12-0-2	1000 0010
83	c	12-0-3	1000 0011
84	d	12-0-4	1000 0100
85	e	12-0-5	1000 0101
86	f	12-0-6	1000 0110
87	g	12-0-7	1000 0111
88	h	12-0-8	1000 1000
89	i	12-0-9	1000 1001
8A		12-0-2-8	1000 1010
8B		12-0-3-8	1000 1011
8C		12-0-4-8	1000 1100
8D		12-0-5-8	1000 1101
8E		12-0-6-8	1000 1110
8F		12-0-7-8	1000 1111
90		12-11-1-8	1001 0000
91	j	12-11-1	1001 0001
92	k	12-11-2	1001 0010
93	l	12-11-3	1001 0011
94	m	12-11-4	1001 0100
95	n	12-11-5	1001 0101
96	o	12-11-6	1001 0110
97	p	12-11-7	1001 0111
98	q	12-11-8	1001 1000
99	r	12-11-9	1001 1001
9A		12-11-2-8	1001 1010
9B		12-11-3-8	1001 1011
9C		12-11-4-8	1001 1100
9D		12-11-5-8	1001 1101
9E		12-11-6-8	1001 1110
9F		12-11-7-8	1001 1111
A0		11-0-1-8	1010 0000
A1		11-0-1	1010 0001
A2	s	11-0-2	1010 0010
A3	t	11-0-3	1010 0011
A4	u	11-0-4	1010 0100
A5	v	11-0-5	1010 0101
A6	w	11-0-6	1010 0110
A7	x	11-0-7	1010 0111
A8	y	11-0-8	1010 1000
A9	z	11-0-9	1010 1001
AA		11-0-2-8	1010 1010
AB		11-0-3-8	1010 1011
AC		11-0-4-8	1010 1100
AD		11-0-5-8	1010 1101
AE		11-0-6-8	1010 1110
AF		11-0-7-8	1010 1111
B0		12-11-0-1-8	1011 0000
B1		12-11-0-1	1011 0001
B2		12-11-0-2	1011 0010
B3		12-11-0-3	1011 0011
B4		12-11-0-4	1011 0100
B5		12-11-0-5	1011 0101
B6		12-11-0-6	1011 0110
B7		12-11-0-7	1011 0111
B8		12-11-0-8	1011 1000
B9		12-11-0-9	1011 1001
BA		12-11-0-2-8	1011 1010
BB		12-11-0-3-8	1011 1011
BC		12-11-0-4-8	1011 1100
BD		12-11-0-5-8	1011 1101
BE		12-11-0-6-8	1011 1110
BF		12-11-0-7-8	1011 1111

Hexa-decimal	Graphic & Control Symbols EBCDIC	Punched Card Code	System/360 8-bit Code
C0		12-0	1100 0000
C1	A	12-1	1100 0001
C2	B	12-2	1100 0010
C3	C	12-3	1100 0011
C4	D	12-4	1100 0100
C5	E	12-5	1100 0101
C6	F	12-6	1100 0110
C7	G	12-7	1100 0111
C8	H	12-8	1100 1000
C9	I	12-9	1100 1001
CA		12-0-2-8-9	1100 1010
CB		12-0-3-8-9	1100 1011
CC	⑀	12-0-4-8-9	1100 1100
CD		12-0-5-8-9	1100 1101
CE	⑂	12-0-6-8-9	1100 1110
CF		12-0-7-8-9	1100 1111
D0		11-0	1101 0000
D1	J	11-1	1101 0001
D2	K	11-2	1101 0010
D3	L	11-3	1101 0011
D4	M	11-4	1101 0100
D5	N	11-5	1101 0101
D6	O	11-6	1101 0110
D7	P	11-7	1101 0111
D8	Q	11-8	1101 1000
D9	R	11-9	1101 1001
DA		12-11-2-8-9	1101 1010
DB		12-11-3-8-9	1101 1011
DC		12-11-4-8-9	1101 1100
DD		12-11-5-8-9	1101 1101
DE		12-11-6-8-9	1101 1110
DF		12-11-7-8-9	1101 1111
E0		0-2-8	1110 0000
E1		11-0-1-9	1110 0001
E2	S	0-2	1110 0010
E3	T	0-3	1110 0011
E4	U	0-4	1110 0100
E5	V	0-5	1110 0101
E6	W	0-6	1110 0110
E7	X	0-7	1110 0111
E8	Y	0-8	1110 1000
E9	Z	0-9	1110 1001
EA		11-0-2-8-9	1110 1010
EB		11-0-3-8-9	1110 1011
EC	⑁	11-0-4-8-9	1110 1100
ED		11-0-5-8-9	1110 1101
EE		11-0-6-8-9	1110 1110
EF		11-0-7-8-9	1110 1111
F0	0	0	1111 0000
F1	1	1	1111 0001
F2	2	2	1111 0010
F3	3	3	1111 0011
F4	4	4	1111 0100
F5	5	5	1111 0101
F6	6	6	1111 0110
F7	7	7	1111 0111
F8	8	8	1111 1000
F9	9	9	1111 1001
FA		12-11-0-2-8-9	1111 1010
FB		12-11-0-3-8-9	1111 1011
FC		12-11-0-4-8-9	1111 1100
FD		12-11-0-5-8-9	1111 1101
FE		12-11-0-6-8-9	1111 1110
FF		12-11-0-7-8-9	1111 1111

Figure 30. Extended Binary Coded Decimal Interchange Code (EBCDIC), Part 2 of 2

Appendix A

Typical Model 30 Algorithms

Because there is a wide difference in the physical design of each model CPU, a variety of algorithms (rules of procedure for solving recurrent mathematical problems) are employed for the execution of many System/360 operations.

Examples of the algorithms for the *convert, divide,* and *multiply* instruction executions in the Model 30 are given.

Converts

Binary-to-decimal convert is accomplished by operating on the binary bits from the high order to the low order. Each binary bit is added into the decimal field, and then the decimal field is doubled. Thus the highest-order binary bit is doubled 31 times, the next bit is doubled 30 times, etc., so that each bit has its proper weight added into the decimal field.

Decimal-to-binary convert is accomplished by repeatedly dividing the decimal field by 16. The remainders of these divisions form the hexadecimal digits of the converted number. In both conversion processes only the significant digits of the decimal field are acted upon so that, in effect, the decimal field gets longer and longer as convert-to-decimal progresses and gets shorter and shorter as convert-to-binary progresses. In both cases, the total time required depends on the number of significant digits in the number being converted.

Divide

Fixed-point binary and floating-point divide are accomplished by subtracting (or adding) the divisor from (or to) the dividend. Before each add or subtract, the dividend field is shifted left one bit. Because a non-restoring process is used, the time required for the divide does not depend on the quotient bits that are generated. However, some variation in execution time is encountered due to sign handling and, in the case of floating-point divide, pre- and post-normalization.

Multiply

Fixed-point binary and floating-point multiplication in the Model 30 make use of a single microprogram loop that multiplies a 16-bit factor by a second factor that is 24, 32, or 56 bits long. Half-word multiply uses this loop only once to accomplish its 16 × 32-bit multiply. Full-word multiply (RR or RX) uses the loop twice to accomplish two 16 × 32-bit multiplies. The two 48-bit products are then added together (with one shifted by 16 bits) to form the final 64-bit product. Short floating-point multiply uses the loop twice to form two 16 × 24-bit products (in one use of the loop, the 16-bit factor has eight bits zero). Long floating-point multiply uses the loop four times to form four 16 × 56-bit products.

In the multiply loop, the 16-bit (multiplier) factor is held in both single and doubled form. Then successive hexadecimal digits of the multiplicand are examined, and additions or subtractions are made according to the table shown in Figure 31 (Figure 32 for CPU with 2-microsecond RW cycle).

Following the additions or subtractions required by the multiplicand digit, the partial product is shifted right four bits, and the next hexadecimal digit is examined. The time (C) required to process two hexadecimal digits of the multiplicand is 9.8 microseconds (13 microseconds for a CPU with a 2-microsecond RW cycle) plus the add or subtract times for the two hexadecimal digits. Assuming a uniform distribution of values of the hexadecimal digits, the average add/subtract time is 4.6 microseconds (6.1 microseconds for a CPU with a 2-microsecond RW cycle) per digit. Thus the average value of C is: 9.8 + 4.6 + 4.6 = 19 (25.2 microseconds for a CPU with a 2-microsecond RW cycle). However, if both hexadecimal digits in a byte of the multiplicand are zero (or in the case when the previous operation was a subtract, if both hexadecimal digits are F), no additions or subtractions are required and a single eight-bit shift of the partial product is made. This gives a value of C of 4 microseconds (5 microseconds for a CPU with a 2-microsecond RW cycle). Also, in the case when the 16-bit multiplier factor is zero, the partial product is set to zero, resulting in a value of C of 1.5 microseconds (2 microseconds for a CPU with a 2-microsecond RW cycle) for each byte of the multiplicand. In terms of this value, C, the times are:

Instruction	*Format*	*Mnemonic*	*Time*	*Time**
Multiply	RR	MR	76 + 8C	102 + 8C
Multiply	RX	M	83 + 8C	111 + 8C
Multiply Half-word	RX	MH	29 + 4C	39 + 4C
Multiply Long	RR	MDR	260 + 28C	344 + 28C
Multiply Long	RX	MD	266 + 28C	354 + 28C
Multiply Short	RR	MER	124 + 6C	158 + 6C
Multiply Short	RX	ME	131 + 6C	168 + 6C

*For CPU with 2 microsecond RW cycle.

The average times given in Figure 17 (Instruction Timing Chart) all assume C = 19 (25.2 for CPU with 2-microsecond RW cycle—see Figure 18), except for half-word multiply where both factors are assumed to be 16 bits long and so C was taken to be 19 (25.2 for CPU with 2-microsecond RW cycle), for the two low-order bytes of the multiplicand and 4 (6 for CPU with 2-microsecond RW cycle), for the two high-order bytes.

Value of Multiplicand Hex Digit	Operation Performed	Average Time Required (in microseconds)	Operation Performed (if previous operation was subtract)	Average Time Required (in microseconds)
0	—	.75	+1x	2.25
1	+1x	2.25	+2x	2.25
2	+2x	2.25	+2x +1x	3.75
3	+2x +1x	3.75	+2x +2x	4.50
4	+2x +2x	4.50	+2x +2x +1x	6.00
5	+2x +2x +1x	6.00	+2x +2x +2x	6.00
6	+2x +2x +2x	6.00	+2x +2x +2x +1x	7.88
7	+2x +2x +2x +1x	7.88	-2x -2x -2x -2x	7.88
8	-2x -2x -2x -2x	7.88	-2x -2x -2x -1x	7.50
9	-2x -2x -2x -1x	7.50	-2x -2x -2x	6.00
A	-2x -2x -2x	6.00	-2x -2x -1x	6.00
B	-2x -2x -1x	6.00	-2x -2x	4.13
C	-2x -2x	4.13	-2x -1x	3.75
D	-2x -1x	3.75	-2x	2.25
E	-2x	2.25	-1x	2.25
F	-1x	2.25	—	.75

Figure 31. Multiply Algorithm Timings

Value of Multiplicand Hex Digit	Operation Performed	Average Time Required (in microseconds)	Operation Performed (if previous operation was subtract)	Average Time Required (in microseconds)
0	—	1	+1x	3
1	+1x	3	+2x	3
2	+2x	3	+2x + 1x	5
3	+2x + 1x	5	+2x + 2x	6
4	+2x + 2x	6	+2x +2x + 1x	8
5	+2x + 2x + 1x	8	+2x + 2x + 2x	8
6	+2x + 2x + 2x	8	+2x + 2x + 2x + 1x	10.5
7	+2x + 2x + 2x + 1x	10.5	-2x -2x -2x -2x	10.5
8	-2x -2x -2x -2x	10.5	-2x -2x -2x -1x	10
9	-2x -2x -2x -1x	10	-2x -2x -2x	8
A	-2x -2x -2x	8	-2x -2x -1x	8
B	-2x -2x -1x	8	-2x -2x	5.5
C	-2x -2x	5.5	-2x -1x	5
D	-2x -1x	5	-2x	3
E	-2x	3	-1x	3
F	-1x	3	—	1

Figure 32. Multiply Algorithm Timings (for CPU with 2-Microsecond RW Cycle)

Local Storage and MPX (Multiplexor) Storage Map

An additional 256 bytes of local storage, which are not directly addressable by the program, contain the 16 general-purpose registers and the 4 floating-point registers. Local storage also contains a *scratch pad* area, which is not available to the programmer. The 32 words of the multiplexor channel Unit Control Words (UCW) are contained in 256 bytes of MPX storage (for a CPU with 8192 bytes of main storage). Local storage and the multiplexor storages comprise auxiliary storage.

The storage map (Figure 33) indicates the local storage addresses and MPX storage addresses assigned to the various functions. MPX 0 is standard on all 2030 processing units. MPX 1 and 2 are used only on models D30, DC30, E30, and F30.

The storage map in Figure 34 shows additional UCW locations. These 128 additional UCW's are used on models E30 and F30 only and are an optional feature.

Areas of local storage and MPX storage that can be displayed on the console are found by using the vertical and horizontal coordinates of the map. (CPU storage register V is located at horizontal coordinate 5x; the vertical coordinate is C.)

Methods used to display from and store into auxiliary storage are described in *IBM System/360 Model 30 Operator's Guide*, GA24-3373.

Unshared I/O unit addresses (hexadecimal) that correspond to UCW's in multiplexor storage are also shown in Figures 33 and 34. Units that share a common subchannel and use the first eight UCW's have the following addresses (not shown in Figure 33):

Unit Address (Hexadecimal)	*UCW*
08X	0
09X	1
0AX	2
0BX	3
0CX	4
0DX	5
0EX	6
0FX	7

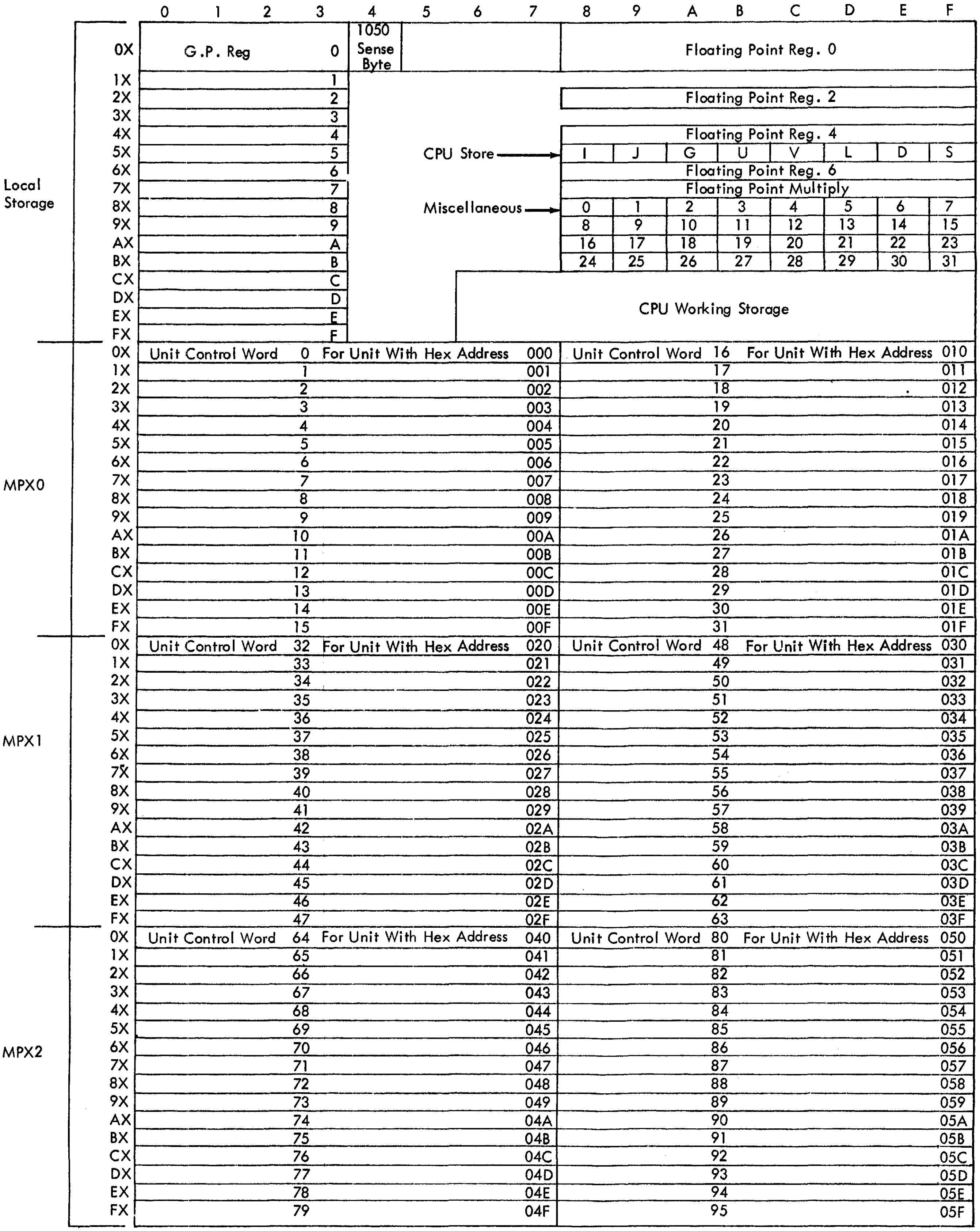
The value of X can range from 0 to F. Therefore, each X in this listing indicates a range of addresses for up to sixteen I/O devices on each multiplexor shared subchannel. For further information on device addressing, refer to the *Unit Addressing Method* section of this publication.

Local Storage Map Miscellaneous Legend (Figure 33)

0 Multiplexor channel interrupt buffer unit address.
1 Multiplexor channel T-register storage.
2 Multiplexor channel R-register storage.
3 Multiplexor channel interrupt buffer unit status.
4 Program status word bit and instruction-length code storage.
5 Selector channel 1 unit address.
6 Selector channel 1 next CCW address (high).
7 Selector channel 1 next CCW address (low).
8 Multiplexor channel unit address temporary storage.
9 1050 unit status.
10 CPU working storage.
11 CPU working storage.
12 CPU working storage.
13 CPU working storage.
14 CPU working storage.
15 CPU working storage.
16 Instruction counter (not available).
17 Instruction counter high.
18 Instruction counter low.
19
20 Selector channel chaining R-register storage.
21 Selector channel 2 unit address.
22 Selector channel 2 next CCW address (high).
23 Selector channel 2 next CCW address (low).
24 System mask.
25 Storage protection and ASCII, machine check mask, wait, and program bits of the PSW.
26
27 Condition register and program mask storage.
28 Operation code mask.
29 Selector channel chaining S-register storage.
30 Selector channel chaining U-register storage.
31 Selector channel chaining V-register storage.

CPU Store Legend (Figure 33)

Locations labelled I through S are those used for temporary storage of the corresponding registers during MPX channel operations.



		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Local Storage	0X	G.P. Reg 0				1050 Sense Byte				Floating Point Reg. 0							
	1X	1															
	2X	2								Floating Point Reg. 2							
	3X	3															
	4X	4								Floating Point Reg. 4							
	5X	5					CPU Store →			I	J	G	U	V	L	D	S
	6X	6								Floating Point Reg. 6							
	7X	7								Floating Point Multiply							
	8X	8					Miscellaneous →			0	1	2	3	4	5	6	7
	9X	9								8	9	10	11	12	13	14	15
	AX	A								16	17	18	19	20	21	22	23
	BX	B								24	25	26	27	28	29	30	31
	CX	C						CPU Working Storage									
	DX	D															
	EX	E															
	FX	F															

		0 – 7		8 – F	
MPX0	0X	Unit Control Word 0	For Unit With Hex Address 000	Unit Control Word 16	For Unit With Hex Address 010
	1X	1	001	17	011
	2X	2	002	18	012
	3X	3	003	19	013
	4X	4	004	20	014
	5X	5	005	21	015
	6X	6	006	22	016
	7X	7	007	23	017
	8X	8	008	24	018
	9X	9	009	25	019
	AX	10	00A	26	01A
	BX	11	00B	27	01B
	CX	12	00C	28	01C
	DX	13	00D	29	01D
	EX	14	00E	30	01E
	FX	15	00F	31	01F
MPX1	0X	Unit Control Word 32	For Unit With Hex Address 020	Unit Control Word 48	For Unit With Hex Address 030
	1X	33	021	49	031
	2X	34	022	50	032
	3X	35	023	51	033
	4X	36	024	52	034
	5X	37	025	53	035
	6X	38	026	54	036
	7X	39	027	55	037
	8X	40	028	56	038
	9X	41	029	57	039
	AX	42	02A	58	03A
	BX	43	02B	59	03B
	CX	44	02C	60	03C
	DX	45	02D	61	03D
	EX	46	02E	62	03E
	FX	47	02F	63	03F
MPX2	0X	Unit Control Word 64	For Unit With Hex Address 040	Unit Control Word 80	For Unit With Hex Address 050
	1X	65	041	81	051
	2X	66	042	82	052
	3X	67	043	83	053
	4X	68	044	84	054
	5X	69	045	85	055
	6X	70	046	86	056
	7X	71	047	87	057
	8X	72	048	88	058
	9X	73	049	89	059
	AX	74	04A	90	05A
	BX	75	04B	91	05B
	CX	76	04C	92	05C
	DX	77	04D	93	05D
	EX	78	04E	94	05E
	FX	79	04F	95	05F

Figure 33. Local Storage and MPX Storage Map

		0 1 2 3 4 5 6 7	8 9 A B C D E F
MPX3	0X	Unit Control Word 96 For Unit With Hex Address 060	Unit Control Word 112 For Unit With Hex Address 070
	1X	97 061	113 071
	2X	98 062	114 072
	3X	99 063	115 073
	4X	100 064	116 074
	5X	101 065	117 075
	6X	102 066	118 076
	7X	103 067	119 077
	8X	104 068	120 078
	9X	105 069	121 079
	AX	106 06A	122 07A
	BX	107 06B	123 07B
	CX	108 06C	124 07C
	DX	109 06D	125 07D
	EX	110 06E	126 07E
	FX	111 06F	127 07F
MPX4	0X	Unit Control Word 128 For Unit With Hex Address 080	Unit Control Word 144 For Unit With Hex Address 090
	1X	129 081	145 091
	2X	130 082	146 092
	3X	131 083	147 093
	4X	132 084	148 094
	5X	133 085	149 095
	6X	134 086	150 096
	7X	135 087	151 097
	8X	136 088	152 098
	9X	137 089	153 099
	AX	138 08A	154 09A
	BX	139 08B	155 09B
	CX	140 08C	156 09C
	DX	141 08D	157 09D
	EX	142 08E	158 09E
	FX	143 08F	159 09F
MPX5	0X	Unit Control Word 160 For Unit With Hex Address 0A0	Unit Control Word 176 For Unit With Hex Address 0B0
	1X	161 0A1	177 0B1
	2X	162 0A2	178 0B2
	3X	163 0A3	179 0B3
	4X	164 0A4	180 0B4
	5X	165 0A5	181 0B5
	6X	166 0A6	182 0B6
	7X	167 0A7	183 0B7
	8X	168 0A8	184 0B8
	9X	169 0A9	185 0B9
	AX	170 0AA	186 0BA
	BX	171 0AB	187 0BB
	CX	172 0AC	188 0BC
	DX	173 0AD	189 0BD
	EX	174 0AE	190 0BE
	FX	175 0AF	191 0BF
MPX6	0X	Unit Control Word 192 For Unit With Hex Address 0C0	Unit Control Word 208 For Unit With Hex Address 0D0
	1X	193 0C1	209 0D1
	2X	194 0C2	210 0D2
	3X	195 0C3	211 0D3
	4X	196 0C4	212 0D4
	5X	197 0C5	213 0D5
	6X	198 0C6	214 0D6
	7X	199 0C7	215 0D7
	8X	200 0C8	216 0D8
	9X	201 0C9	217 0D9
	AX	202 0CA	218 0DA
	BX	203 0CB	219 0DB
	CX	204 0CC	220 0DC
	DX	205 0CD	221 0DD
	EX	206 0CE	222 0DE
	FX	207 0CF	223 0DF

Figure 34. MPX Storage Map (MPX 3 through 6)

Appendix C

Extended BCD Interchange Code

The chart (Figure 35) shows the hexadecimal equivalents to the extended punched-card code. To find the card code that will be punched from a hexadecimal code of 0110 1011, find the set of four columns headed 01 (bit positions 0 and 1). Within these columns, follow the 10 column (bit positions 2 and 3) down until the bit-positions code (4567 bits) is reached. The character found here is zero zone, 3, and 8 which is a comma (,). Similarly, 1111 1110 is found to be 12 (T) zone, 11 (E) zone, zero zone, 6, 8, and 9 punches.

There are 256 characters possible in the extended punched-card code. Also see Figure 30.

Bit Positions

01	00				01				10				11			
23 4567	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	T09 18	TE9 18	E09 18	TE09 18		T	E	TE0	T0 18	TE 18	E0 18	TE0 18	T0	E0	0 28	0
0001	T9 1	E9 1	09 1	9 1	T09 1	TE9 1	0 1	TE09 1	T0 1	TE 1	E0 1	TE0 1	T 1	E 1	E09 1	1
0010	T9 2	E9 2	09 2	9 2	T09 2	TE9 2	E09 2	TE09 2	T0 2	TE 2	E0 2	TE0 2	T 2	E 2	0 2	2
0011	T9 3	E9 3	09 3	9 3	T09 3	TE9 3	E09 3	TE09 3	T0 3	TE 3	E0 3	TE0 3	T 3	E 3	0 3	3
0100	T9 4	E9 4	09 4	9 4	T09 4	TE9 4	E09 4	TE09 4	T0 4	TE 4	E0 4	TE0 4	T 4	E 4	0 4	4
0101	T9 5	E9 5	09 5	9 5	T09 5	TE9 5	E09 5	TE09 5	T0 5	TE 5	E0 5	TE0 5	T 5	E 5	0 5	5
0110	T9 6	E9 6	09 6	9 6	T09 6	TE9 6	E09 6	TE09 6	T0 6	TE 6	E0 6	TE0 6	T 6	E 6	0 6	6
0111	T9 7	E9 7	09 7	9 7	T09 7	TE9 7	E09 7	TE09 7	T0 7	TE 7	E0 7	TE0 7	T 7	E 7	0 7	7
1000	T9 8	E9 8	09 8	9 8	T09 8	TE9 8	E09 8	TE09 8	T0 8	TE 8	E0 8	TE0 8	T 8	E 8	0 8	8
1001	T9 18	E9 18	09 18	9 18	T 18	E 18	0 18	18	T0 9	TE 9	E0 9	TE0 9	T 9	E 9	0 9	9
1010	T9 28	E9 28	09 28	9 28	T 28	E 28	TE	28	T0 28	TE 28	E0 28	TE0 28	T09 28	TE9 28	E09 28	TE09 28
1011	T9 38	E9 38	09 38	9 38	T 38	E 38	0 38	38	T0 38	TE 38	E0 38	TE0 38	T09 38	TE9 38	E09 38	TE09 38
1100	T9 48	E9 48	09 48	9 48	T 48	E 48	0 48	48	T0 48	TE 48	E0 48	TE0 48	T09 48	TE9 48	E09 48	TE09 48
1101	T9 58	E9 58	09 58	9 58	T 58	E 58	0 58	58	T0 58	TE 58	E0 58	TE0 58	T09 58	TE9 58	E09 58	TE09 58
1110	T9 68	E9 68	09 68	9 68	T 68	E 68	0 68	68	T0 68	TE 68	E0 68	TE0 68	T09 68	TE9 68	E09 68	TE09 68
1111	T9 78	E9 78	09 78	9 78	T 78	E 78	0 78	78	T0 78	TE 78	E0 78	TE0 78	T09 78	TE9 78	E09 78	TE09 78

Notes: T = Twelve Zone
E = Eleven Zone

Figure 35. Extended BCD Interchange Code

Index

READER'S COMMENT FORM

IBM System/360 Model 30 Functional Characteristics GA24-3231-7

- Are you able to find information in this publication easily?
- Are the illustrations adequate, clear, and concise?
- Which of the following are included in your occupational responsibilities?

  System operating. . . . .
  Programming. . . . .
  Supervising. . . . .
  Instructing. . . . .
  Other. . . . .

- We would appreciate your comments; please give specific page and line references where appropriate.

Reply Requested?

Yes □

No □

NAME . . . .
STREET . . . .
CITY . . . .
STATE . . . . . ZIP CODE . . .

GA24-3231-7

YOUR COMMENTS, PLEASE . . . .

Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM branch office serving your locality.

Cut Along Line

Fold Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

IBM Corporation
P.O. Box 6
Endicott, N.Y. 13760

Attention: Product Publications, Dept. K10

Fold Fold

IBM®

GA24-3231-7

IBM S/360-30 GA24-3231-7

IBM