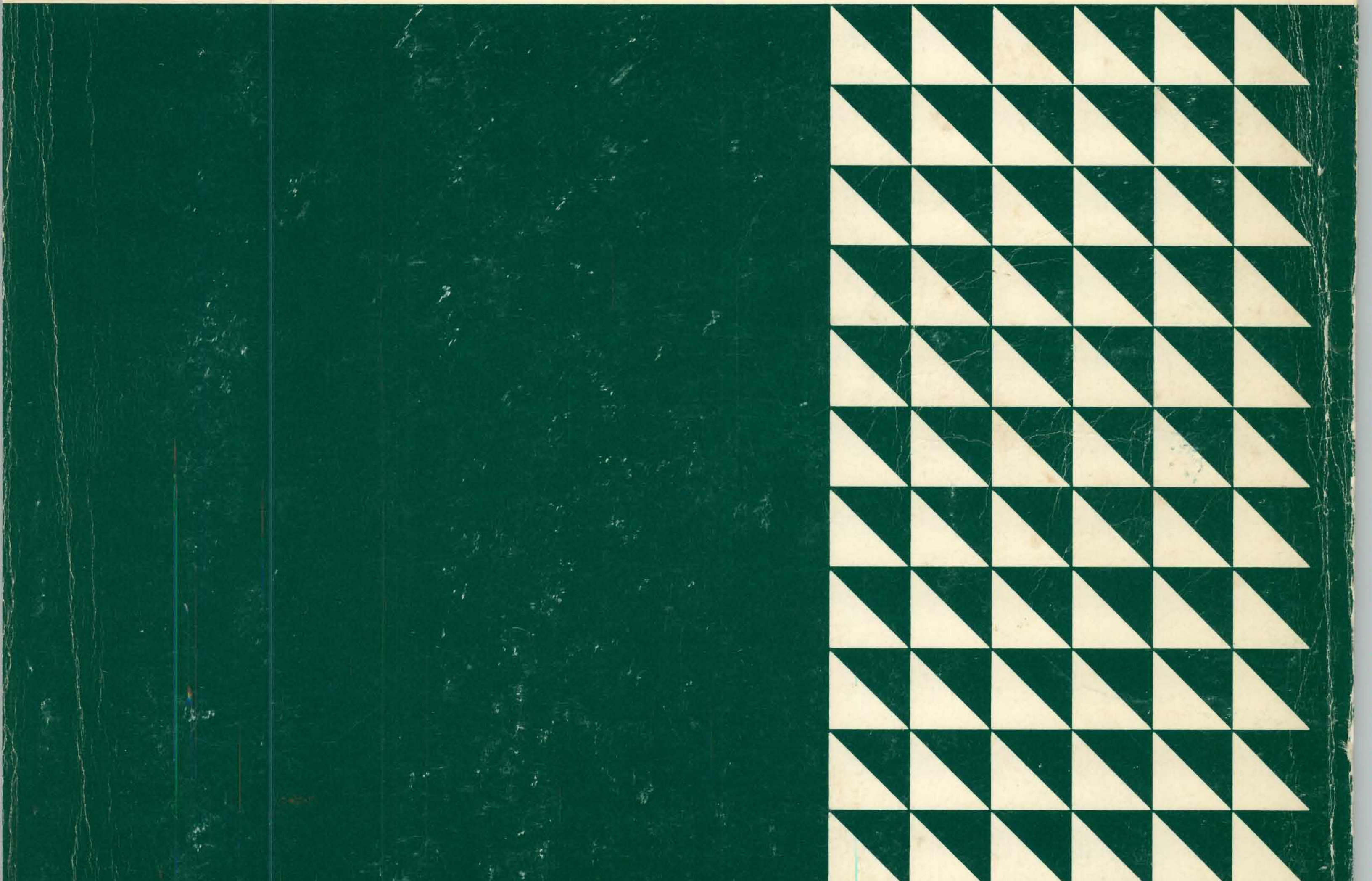
IBM

Introduction to System/360 Text



Programmed Instruction

Introduction to System/360 Text

Programmed Instruction

PREFACE

This book presents concepts, definitions and examples to the student regarding System/360.

The student is assumed to have satisfactorily completed courses in Computing Systems Fundamentals and Fundamentals of Programming.

Minor Revision (August 1970)

This publication is a minor revision of GR29-0256-2 incorporating minor changes made on pages dated (8/70). The original publication is not obsoleted.

(Reprinted October 1971)

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality. Address comments concerning the contents of this publication to IBM Corporation, DPD Education Development - Publications Services, Education Center, South Road, Poughkeepsie, New York 12602.

TABLE OF CONTENTS

INTRODUCTION

How to use this book

In the prerequisite courses you learned many things about computer systems and programming in general. This book is intended to provide specific information about the System/360 – how it computes, how it is programmed and what makes up such a computer system.

Each of the twelve sections deals with one topic. A self-evaluation quiz, with answers, at the end of each section, provides a review before starting a new topic. As you check the quiz you will see a frame number reference so that you can review the material for the questions you've missed.

The other directions are general, but very important. The self-study material is organized into a series of statements called "frames". Some frames give you information, others ask you to do something with it (think of an answer, perform a calculation, select an answer, etc.) After you respond, you can check your response against the correct one. It is printed on the left, directly below three dots (● ● ●).

a. Most of the time, you will merely think of the answer before checking it. If the type of response must be the result of a calculation or some other answer that might be lengthy, use scratch paper. Do not write in this book.

b. You may find it convenient to use a card to cover each answer before you want to check your response. If you accidentally see an answer too soon, you will have lost your chance to answer on the basis of your own knowledge.

c. If you make the wrong response to any frame, review the preceding material until you see why the printed answer is correct.

Turn to Section 1 – OVERVIEW and begin the study of Introduction to System/360.

Overview

	PERIPHERAL DEVICES FOR SYSTEM / 360	
Low Speed	Console Typewriters	
	Audio Response Units	
	Visual Display Units	
	Remote Processing Terminals	
	Magnetic Ink Character Readers	
	Optical Mark Readers	
	Optical Character Readers	
	Paper Tape Readers	
	Card Punches	
	Card Readers	
	Card Reader/Punches	
	Printers	
High Speed	Magnetic Tape Units	*
	Data Cell	*
	Disk Drives	*
	Magnetic Drum Units	*

* Also used to store volumes of data

Figure 1.

OVERVIEW

1 The design of System/360 is a sharp departure from that of previous computers, based on a decade of growth and differentiation in data processing. Most prominent points: it is a truly general purpose computer,and it is designed to function under an operating system.

There are many system features underlying these two points and many ramifications of them. The purpose of this section is to show how they provide for an unprecedented range of applications and potential for growth.

Changing needs are more the rule than the exception, in data processing. In a commercial installation for example, the volume of data to be processed steadily increases as customers and sales increase. Files must be expanded, and processing runs usually take longer. The system may need faster I/O devices.

Also, some jobs may have to be converted from batch processing [1] to in-line processing[2], just to keep up with the flow of data.

In the past, conversion to faster I/O units of a given type, or to a different type, often required installation of a new data processing system. With System/360 this is no longer true. The system includes the widest range of device types speeds, and capacities in the data processing industry.

Figure 1 shows the types of devices currently available, grouped by data transmission speed. The category "low speed" actually includes a large range of speeds. For example, the 1052 console typewriter (when it communicates with the operator) types about 15 characters per second. By contrast, the 1403 printer can print a maximum of 1400 lines of data per minute. Since each line may contain 132 characters and each character requires one byte, this amounts to 3080 bytes per second.

The range of data transmission speeds becomes immense, when the high speed devices are considered. For example, the 2301 Drum Storage transmits 1,200,000 bytes per second.

What could you conclude about the ability of System/360 to handle the high-speed I/O requirements of commercial installations?

● ● ●

System/360 can handle virtually all commercial I/O requirements.

2 As important as speed is the versatility evidenced by the variety of I/O devices listed in Figure 1. In most cases, there are several models for a given device type. In the case of remote processing terminals, there are many: This group contains the remote data collection terminals (e.g., badge readers, card readers), data communication terminals, process control terminals (monitoring and control of physical processes), and so on.

As recently as 1963, computer systems were characterized by the limited I/O devices that they could use. Thus, one system was a "card system", another was a "tape system", etc. The ability of System/360 to use so many different I/O devices is due to the concept of the "standard interface".

In data processing jargon, an interface is a boundary between systems or parts of a system. An I/O interface, then, is the boundary between an I/O device and a channel. Across this boundary must flow signals that control I/O device operation as well as the I/O data itself. The past limitation on the use of I/O devices was due to the fact that each type of device required a different series of control signals for its operation. Having the CPU

[1] Batch processing – collecting records of transactions for a given period of time, then processing them in one group.

[2] In-line processing – processing individual transactions as soon as they are reported. In-line processing will require different I/O devices (direct access storage devices), if they are not already being used.

STANDARD INTERFACE

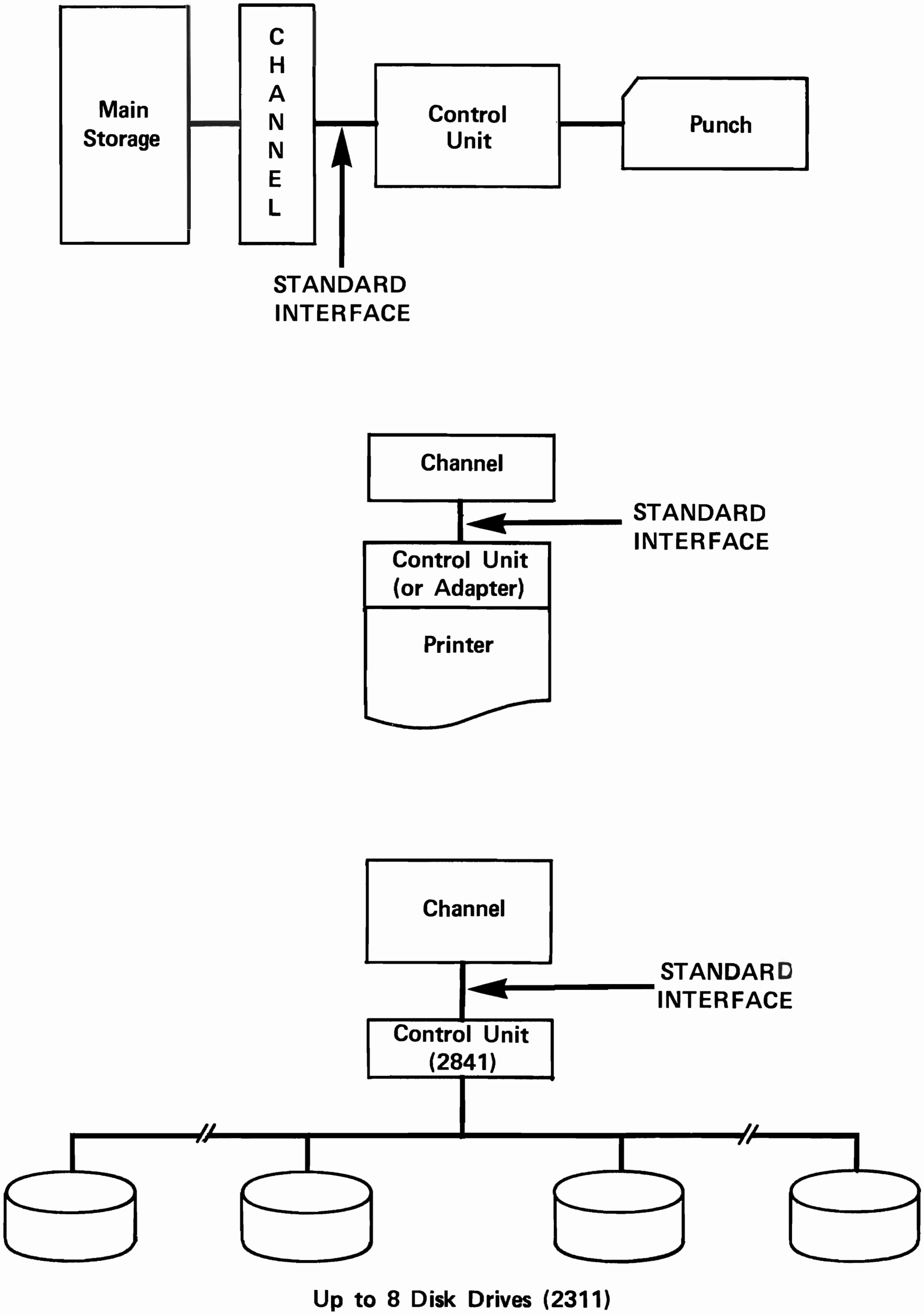


Figure 2.

work with many different I/O devices would mean using too many instructions from the available set, just for I/O.

The design departure of System/360 lies in having each type of I/O device associated with its own control unit, and in connecting the control unit to the channel (three examples are shown in Figure 2). Then, in response to any one of a few, standard, I/O orders sent out by the channel, the control unit can transmit the series of specific control signals required by its I/O device. Any kind of I/O device, whose control unit is designed to respond to the standard signals available at the I/O interface, can be used with System/360.

Questions:

1. Why is the System/360 I/O interface called a standard interface? How does a control unit respond to it?

2. Is there any limit to the number of different kinds of I/O devices that can be designed to operate with System/360?

●●●

1. The I/O interface is formed by an I/O control unit and a channel which sends out standard I/O commands. The control unit responds to these commands by transmitting specific control signals required by its I/O device.

2. Not so far as we know. System/360 is truly an "open-ended" computer system: Any conceivable I/O device can be designed to work with it.

3 Returning to the example of a commercial installation, we find that versatile, high-speed I/O devices alone do not satisfy all needs. Commercial data processing is characterized by a high proportion of I/O activity and a low proportion of CPU activity. If the CPU must wait while its command to read a card is executed, then process the data, then wait while its output command is executed, it may spend a significant amount of time in idleness. For example, a study of one commercial application shows that the CPU was idle about 40% of the time. System/360 solves this problem, and part of the solution is its use of channels.

Channels are small computers. They use main storage to hold programs which they select and execute, in response to initial instructions from the CPU. It is the execution of one of these programs which results in the channel commands to the control unit.

In a real sense, the channels relieve the CPU of the burden of controlling I/O.

Once a CPU instruction has initiated a channel program, the CPU can return to processing. It can start another input operation and resume processing data for that job, or even for another data processing job. Thus, more than one set of input and output devices can be connected to channels and, through concurrent processing and I/O operations, the computer can appear to be doing more than one job at once.

By keeping the CPU busy in this way, System/360 can greatly increase throughput – the rate at which jobs can be handled by a computing system. This ability to input, process, and output data, concurrently, is called "process overlap".

So far, we have discussed three design characteristics of System/360 which are extremely important to commercial users:

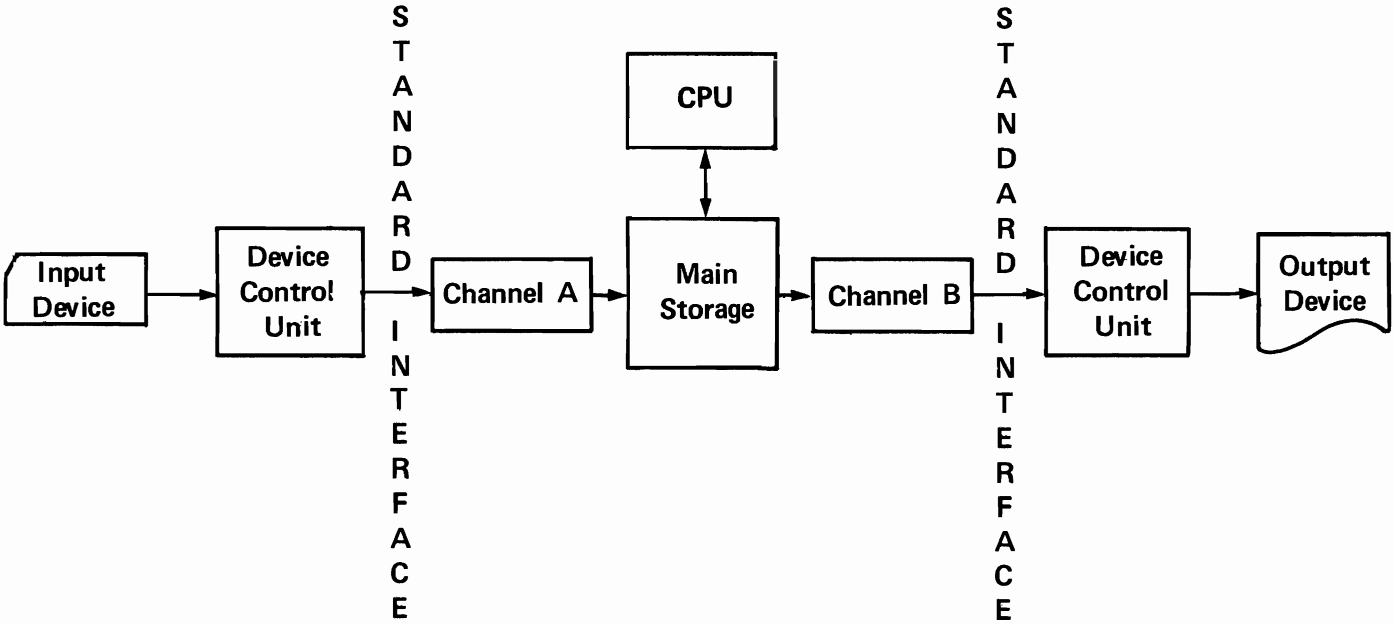
1. .... I/O devices.

2. No limit to the variety of possible I/O devices, through the .... concept.

3. Greatly increased throughput, by the use of channels to allow ....

●●●

high-speed
standard interface
concurrent processing and I/O operations, or
process overlap.



CPU
Input Device
Device Control Unit
STANDARD INTERFACE
Channel A
Main Storage
Channel B
STANDARD INTERFACE
Device Control Unit
Output Device

Figure 3.

4 The channels, as you may recall from Computing Systems Fundamentals, are of two types: Selector Channels and Multiplexor Channels. They are distinguished by the way they operate. A selector channel transmits all of the bytes of information available in one data record (from an input device) or one area of storage (to an output device) in one operation. This is called "burst mode".

A multiplexor channel transmits one byte of information to, or from storage, in one operation. This is called "byte mode".

A selector channel always operates in burst mode. A multiplexor channel may operate in either byte mode or burst mode.

Certain control units (e.g., those for remote terminals) can handle more than one I/O device of a given type. Also, more than one control unit may be connected to a given channel, either selector or multiplexor. Considering our discussion of process overlap, which do you think is usually the case?

a. Each channel is connected to one control unit, which in turn is connected to one I/O device.

b. Each channel is connected to more than one I/O device, through one or more control units.

●●●

b.

5 It is possible, therefore, for a multiplexor channel operating in byte mode to be connected to a number of I/O devices and to transmit a byte of data first from one, then from another and so on. This is called "byte interleaving".

Considering the range of speeds of transmission in Figure 1, which group of devices do you think would be involved with byte interleaving through a multiplexor channel?

●●●

low-speed devices

6 By contrast, a multiplexor channel operating in burst mode, or a selector channel, would be used with .... I/O devices.

●●●

high-speed

7 Figure 3 shows a simplified system configuration. If the input unit is a Card Reader and the output unit is a Tape Drive, what type of channel is A, and what is B?

●●●

A. multiplexor
B. selector, or multiplexor in burst mode

8 One additional point: One multiplexor channel can be connected to control units for as many as 256 I/O devices (depending on types of devices). Since a large number of these devices can be operating at the same time, only one multiplexor channel can be, or need be, attached to a given system.

Each selector channel can also address as many as 256 I/O devices. Since only one I/O device can use the channel at any one time, provision is made for attaching as many as six selector channels to a system, in addition to a multiplexor.

All selector channels might be reading or writing, or some reading and others writing, with or without simultaneous operation of the multiplexor, concurrent with processing by the CPU. Process overlap is feasible with (all/some) .... of the I/O devices and channels connected to the system.

●●●

all

9 We have been working our way into the system, from I/O units, through channels, and at last we come to the Central Processing Unit. CPU speed has traditionally been of most concern in scientific data processing, and it still is: To a large extent, scientific programs are characterized by relatively few I/O operations and many processing steps (the converse of the traditional commercial picture).

SYSTEM/360 CENTRAL PROCESSING UNITS

MODEL		20	25	30	40	44	50	65	67-1	67-2	75	85	91
	4K	X											
Core	8K	X		X									
Size	12K	X											
	16K	X	X	X	X								
	24K		X	X									
K=1024 bytes	32K		X	X	X	X							
	48K		X										
	64K			X	X	X	X						
	128K				X	X	X	X					
	256K				X	X	X	X	X	X	X		
	512K						X	X	X	X	X	X	
	768K							X			X		
	1024K							X	X	X	X	X	X
	2048K									X		X	X
	4096K											X	X
	6144K												X
Additional Large Capacity Storage 1024-8192K							X	X			X		
Bytes per Access		1	2	1	2	4	4	8	8	8	8	16	8
Access Time (micro-sec.)		3.6	1.8	1.5	2.5	1	2	.75	.75	.75	.75	1.04	.75

Figure 4.

CPU's have even been designed specifically for scientific and not commercial use, concentrating on speed. But changes in commercial requirements, again due to growth, have begun to cloud the distinction. Items:

1. High-speed I/O and simultaneous channel operation can overwhelm a CPU with data.

2. A number of applications have arisen under the general heading of Management Science. These include linear programming, to maximize profitability of operations, PERT and Critical Path Method, to keep track of operations, and various forecasting programs, to predict success of contemplated operations. These programs closely resemble the analytical types found in scientific data processing: They require more storage if they are to run at all, and they require a fast processor to run efficiently.

Thus, System/360 was designed so that any user could convert from a CPU with moderate main storage size and processing speed to another, faster processor, with more main storage.

The range of storage sizes and speeds (time to access data in main storage) is sampled in Figure 4 (note that "K bytes" means 1024 bytes). Each byte of data in storage is addressable by the program, regardless of the size of storage or the number of bytes accessed in a single CPU operation. Large as the range of core storage sizes is, it can be expanded still more, in some models, by the addition of Large Capacity Storage (LCS). The range of additional storage, from 1024K to 8192K, is achieved by adding one or more modules of 1024K to 2048K bytes. Each byte in LCS is also directly addressable by the program. LCS storage addresses start after the last main storage address.

What is the maximum amount of directly addressable storage currently available on System/360?

●●●

9216K (Model 65 or 75, with 8192K of LCS)

Put another way, it is possible for a 4K user to expand his main storage capacity more than 2,000-fold, without exhausting the capacity of System/360.

10 At this point, it is appropriate to contrast LCS to the storage supplied by the high-speed I/O devices. Some of the latter also can hold large amounts of data. But they are not extensions of main storage, in that each byte is not addressable by a program: They transmit or record data records, each of which may contain many bytes of information. We will call these devices "auxiliary storage", when speaking of their storage function, to distinguish them from LCS.

What types of devices are included in the phrase, "auxiliary storage"?

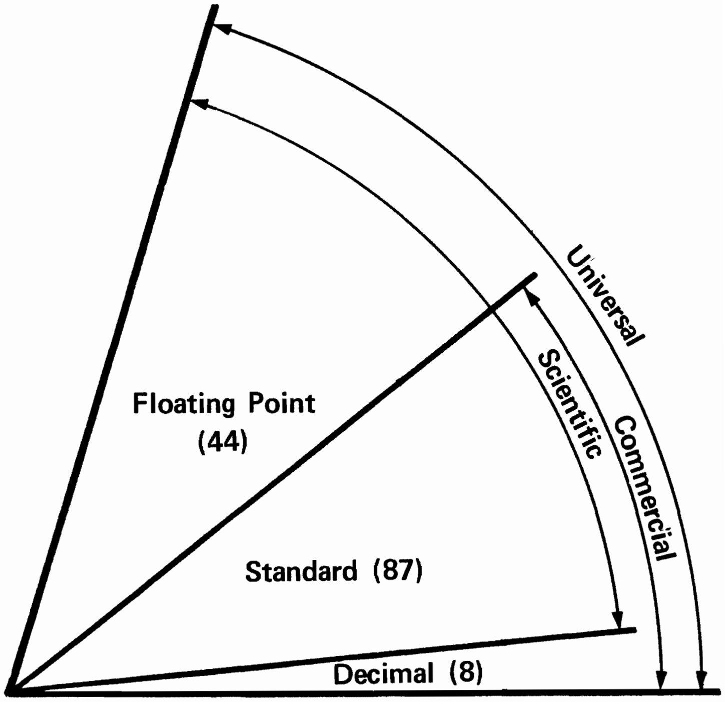
●●●

magnetic tape equipment
disk drives, magnetic drum storage, and the Data Cell.

11 Along with the immense increase in storage capacity, across the current model range, goes a significant increase in CPU power: Eight times as many bytes are accessed, in each operation, in less than one/fourth of the time. But the availability of greater storage size and processor efficiency, needed for the concurrent running of more than one program or the introduction of new larger programs, would be meaningless if existing programs had to be rewritten. This is particularly important to the commercial user, since many of his programs have a long, useful life.

So, from the standpoint of user growth, this is one of the most significant advantages of System/360: Programs that operate on one model (and whose execution does not depend on internal timings or the relationship between internal timings and I/O speed) will operate on most other models that have the same configuration.

What does this business about timing mean? Some programmers try to squeeze the last bits of efficiency from a CPU, by:



IBM SYSTEM/360 INSTRUCTION SET

Figure 5.

a. Coding instructions in a special sequence, carefully arranged to coincide with the beginning or end of an I/O operation.

b. Using an internal clock, called a "timer", to interrupt the execution of a series of instructions (on the assumption that they have had time to establish a desired data condition) and branching to another series. This use of internal timing merely saves writing a few instructions.

Any gain from the use of such techniques is wiped out by loss of compatibility: Compatible programs can be tested on one model of the system, at a programming location, then put directly into the production processing lineup at a main installation.

The system configuration limitation is simply that both systems must have enough core storage to hold the program and data, and the I/O devices and features actually used by the program must be equivalent.

What are the limitations on compatibility between models of System/360?

● ● ●

For a program to be compatible from one system to another, it must not depend on a given system's internal or I/O timing, and the systems must have equivalent configurations.

12 At the outset, we defined System/360 as a truly general purpose system. An important characteristic of such a system is the ease with which a user can do what he wishes with data. This relates to the variety of operations that the CPU can perform and is evident in the instruction set with which the user writes his programs.

The ability to perform many kinds of operations on data is called "logical power", and System/360 has the greatest logical power of any IBM computer.

For scientific data processing – solving problems in science, engineering, and other areas of technology – data is organized in storage in fixed-length words (32 bits). Operations can be performed on halfwords, fullwords, or doublewords. Solid state circuitry allows these binary fields to be added, subtracted, compared, etc., at extremely high speeds.

For commercial applications, data is organized in storage as strings of 8-bit bytes. These strings of bytes (fields) may be of various lengths. A different group of operations, called "variable field length" (VFL) operations is required to handle them.

The System/360 standard instruction set contains instructions for both fixed-point (binary) and variable field length operations.

If a decimal instruction set feature is added, the commercial user can perform decimal arithmetic (VFL) directly, and can edit (punctuate) the output.

If a floating-point arithmetic feature is added, the scientific user can perform high-speed arithmetic with very large (or very small) numbers.

The instruction set that we have been describing is represented in Figure 5. The number of instructions of each type is represented in parentheses. Note that they can be grouped into larger categories:

a. The standard instructions, plus those of the decimal feature, comprise the "commercial" set.

b. The standard instructions, plus those of the floating-point feature, comprise the .... set.

c. All of the instructions together comprise the .... set.

● ● ●

scientific
universal

13 The data that these instructions operate on comes into the system in 8-bit bytes, regardless of its final organization in storage. Instructions will be chosen for the computer program, depending on what we have specified as the contents of each byte. A byte can represent:

a. An 8-bit binary field.

b. Any one of 256 characters of data, including positive or negative digits from 0-9.

c. Two decimal digits. (Since four bits can represent any digit from 0-9, 8 bits can represent two digits).

d. Two hexadecimal digits.

For scientific data processing, bytes are combined into groups of two, four, or eight, to hold binary numbers.

The IBM standard code for the 256 characters that a byte can represent is called the Extended Binary Coded Decimal Interchange Code (EBCDIC). It includes all of the letters of the alphabet (both upper and lower case), the digits (from 0 to 9), and many special characters ($, @, punctuation marks, mathematical symbols, etc.). These nowhere near exhaust 256 possibilities: The 256-character set will accommodate many more special purpose characters, as they are invented and become useful in data processing.

The 8-bit byte is also used to represent the characters in the American Standard Code for Information Interchange (ASCII), a code devised by a group of computer users. An instruction written by the programmer is all that is needed to cause System/360 to accept this code.

Choose an item (from a. to d., above) to match each of the following descriptions of the contents of a byte:

a. The letter Q.

b. Two digits F, 3, of a number to the base 16.

c. Two digits 3, 4 of a number to the base 10.

d. Eight bits of information, each signifying the presence or absence of a condition.

●●●

a. A character of data
b. Two hexadecimal digits
c. Two decimal digits
d. A binary field

14 Which of the System/360 characteristics and features listed below are important, chiefly, to the scientific user, to the commercial user, or to both?

High-speed I/O devices
Wide variety of I/O devices, with standard interface
Channels
Compatibility among models
Large Capacity Storage
High-speed processors
Powerful instruction logic
High-speed fixed-length (binary) operations
VFL operations
Standard Instruction Set
Decimal Instruction Feature
Floating-Point Instruction Feature
256-character code
Editing

●●●

Scientific

High-speed Binary Operations
Floating-Point Instruction Feature

Commercial

High-speed I/O devices
Channels
Compatibility among models
VFL operations
Decimal Instruction Feature
Editing

Both

Variety of I/O devices
Large Capacity Storage
High-speed processors
Powerful instruction logic
Standard Instruction Set
256-character code

SYSTEM / 360 OPERATING SYSTEM

15 An "operating system" is a group of programs, in residence in the computer system, which facilitates every phase of system operation. It is only partly correct to say that an operating system performs many of the tasks that were formerly the responsibility of the programmer or the operator: Since System/360 was designed to work with an operating system, its resources are used more efficiently than those of any prior system.

The two basic parts of the operating system are the control program and the processing programs. You are already familiar with some of the latter programs – the language processors. For System/360, they include Assembler Language, FORTRAN, COBOL, RPG, and PL/I. Others are the utility programs (card-to-tape, tape-to-print, etc.) and the user's own data processing programs, called "problem programs". As problem programs are written and tested, they are stored in the system's program library. In a moment we will see how the system handles them.

The control program is the backbone of the operating system. It performs job management, task management, and data management (i.e., scheduling of a data processing job, handling routine tasks within the job, and handling the input and output data). Of these, task management is the central function, since it oversees the others. In fact, the task management part of the control program is called the "supervisor", and it is always located in main storage.

Assume that all of the problem programs needed by an installation have been stored in the system library and production processing may begin. Here is an illustrative series of events:

a. The operator initiates system operation.

b. The supervisor brings a job control program into main storage from the job management section of the system library.

c. The job control program reads control cards for the first job, assigns I/O devices for it, and notifies the supervisor that a given job is to be run.

d. The supervisor brings in the specified problem program from the library and starts its execution.

e. When the first instruction to input data is executed by the CPU, the supervisor is notified and it brings in one or more input routines from the data management section of the system library. An input routine issues an instruction to the channel, which begins the input operation. The supervisor then returns control of the CPU to the problem program.

f. The CPU continues to execute instructions in the problem program, if any, until the I/O operation in the channel is complete. The channel then signals the supervisor.

g. The supervisor returns CPU operation to the problem program, and processing continues.

h. When the job is done, the supervisor calls in job control to identify the next job.

Keep in mind that the CPU merely fetches an instruction from storage, decodes it, accesses the data that it calls for, performs the indicated operation, then goes on to the next instruction. The immense capability of the system lies in the dynamic interaction of the problem program and the control program.

Control programs were developed to make system operation more efficient. The system library, for example, makes it unnecessary for the operator to repetitively load programs on I/O devices – an operation that sometimes takes longer than the running time of the programs themselves.

Also, the system must not be allowed to stop altogether because of an error condition (e.g., an invalid instruction or data condition). The supervisor is able to respond by instantly unloading the offending program, printing a message to the operator describing the error condition, and initiating the next program.

Match types of programs with functions:

1. Job management
2. Task management (supervisor)
3. Problem program

Functions

a. Performs the data processing job.
b. Identifies the job to be done.
c. Brings in the problem program from the library.
d. Assigns I/O devices for the problem program.
e. Responds to an error condition by unloading one program and going on to the next.

●●●

1. b, d
2. c, e
3. a

16 As you may have guessed, the supervisor is also the key to multiprogramming – the fast switching from one job to another that gives the appearance of running several jobs simultaneously. This operation, along with the development of many kinds of terminals, direct access storage devices, and an internal timer, complete the picture of System/360 as a general purpose system: They provide the means to work in the third major application area – communications.

Communications applications include all situations where the input devices are more than 2000 feet from the central processing unit. They are usually much farther away: Transcontinental hookups have been in operation for some time. They are characterized by many low-speed remote terminals, many input lines, a high-speed processor, and a direct access device for storing problem programs. The system's internal timer is used to keep track of the system time used by each program.

Take banking as an example: While a regular processing run is going on, a request for information about a customer's account comes over the line from a terminal in a branch office. This causes an interruption, and the supervisor responds by branching the CPU to a program that processes such inquiries. The program calls for the supervisor to initiate an input operation, to get data on the specific customer from the direct access storage device. After the program prepares a reply to the inquiry, it signals the supervisor to initiate the output operation which will send it back to the terminal. Finally, the supervisor returns the CPU to the point at which the original program was interrupted.

All of the preceding happens so fast that the effect on main program throughput is barely noticeable.

What are the system features that are important to communications?

●●●

Many types of terminals, and a variety of I/O.
Multiprogramming
Direct storage devices
The ability to connect many I/O lines (multiplexor channel)
An internal timer

Note:

At this point, you should fill in your notes for this section and then take the self-evaluation quiz.

Self-Evaluation Quiz
Overview

QUESTIONS

1. a. What characterizes I/O devices that are always connected to a multiplexor channel?

   b. Give four examples.

2. To what I/O or auxiliary storage devices would a selector channel be connected?

3. Define:

   a. Burst mode
   b. Byte mode

4. Describe the function of a channel and a control unit, with respect to the "standard interface".

5. Describe the activities of the CPU during the execution of a program.

6. What are the three main activities of the control program?

7. What are the limitations on program compatibility?

8. What are three kinds of System/360 instruction sets called, and what are they composed of?

9. What are the two data codes acceptable by System/360?

10. What can be represented by an 8-bit byte?

11. List five System/360 features or characteristics that are chiefly important to scientific, commercial, and communications applications.

ANSWERS Frame Reference

1. a. Low speed (5)

   b. Any four of the following:

      1) Console typewriters
      2) Audio Response Units
      3) Visual Display Units
      4) Remote Processing Units
      5) Magnetic Ink Character Readers
      6) Optical Mark Readers
      7) Optical Character Readers
      8) Paper Tape Readers
      9) Card Punches
      10) Card Readers
      11) Card Reader/Punches
      12) Printers

2. Magnetic Tape Units (5,10)
   Data Cell
   Disk Units
   Magnetic Drum Units

3. a. Burst mode - transmission by a channel of all of the bytes of information in one data record, or one specified area of storage, in one operation. (4)

   b. Byte mode - transmission by a channel of one byte of information (to or from storage) in one operation.

4. A channel responds to commands from the CPU by sending out orders across the I/O interface. The control unit responds to orders by transmitting the series of specific control signals required by its I/O device. (2,3)

5. The CPU fetches an instruction from storage, decodes it, accesses the data that it calls for, performs the indicated operation, then goes on to the next instruction. (15)

6. a. Job management - identification and scheduling of jobs, I/O device assignment. (15)

   b. Task management (supervisor) - handling and initiating programs and routines, including I/O operations. (15)

   c. Data management - handling input and output data.

7. For a program to be compatible from one system to another, it must not depend on a given system's internal or I/O timing, and the systems must have equivalent configurations. (11)

8. a. Scientific - standard set, plus floating-point set. (12)

b. Commercial - standard set, plus decimal set.

c. Universal - all of the sets, together.

9. a. Extended Binary Coded Decimal Interchange Code (13)
b. American Standard Code for Information Interchange

10. a. Any one of 256 characters, including a positive or negative digit from 0 to 9. (13)

b. Two decimal digits.

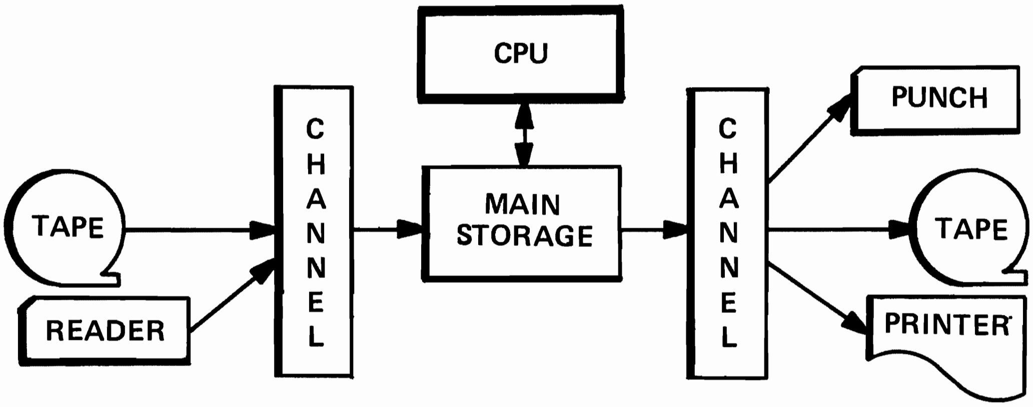
c. Two hexadecimal digits

d. An 8-bit binary field.

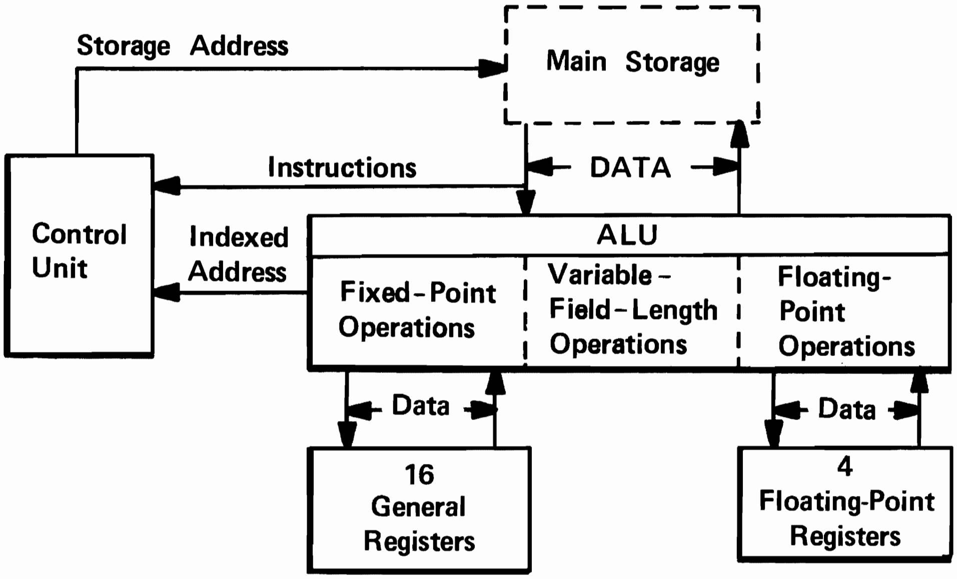
11. Any five, where checked: (14,16)

	Scientific	Commercial	Communications
High-speed I/O Devices		x	
Variety of I/O Devices	x	x	x
Terminals		x	x
Selector Channel		x	
Multiplexor Channel		x	x
Direct Access Storage			x
High-speed CPU	x	x	
Large Capacity Storage	x	x	
Internal Timer			x
Compatibility		x	
High-speed Binary Operations	x		
VFL Operations		x	
Floating-point Instruction Feature	x		
Decimal Instruction Feature		x	
Editing		x	

Central Processing Unit



TYPICAL DATA PROCESSING UNIT
Figure 1.



CENTRAL PROCESSING UNIT LOGIC FLOW
Figure 2.

CENTRAL PROCESSING UNIT

Overview

1 Historically, logical design of computers has dictated that separate designs be used for scientific and commercial computers. Scientific computers had high processing and low I/O requirements. Processing speed was important and relatively raw (unedited) data was adequate output for most applications.

The commercial computers, on the other hand, had high I/O and relatively low processing requirements. Character data such as names, addresses, descriptions, etc., was important. The output had to be readable and of high quality. Thus, the editing of the data, that is, the grouping, spacing, punctuation and the use of such items as dollar signs, asterisks and percent signs, was an important requirement of the computer.

Today the situation has changed so that the requirements for scientific and commercial applications are similar in many ways. Scientific computation has need for high I/O and editing and the use of character data. Commercial computing has shown the need for speed of processing and computing characteristics formerly found only in scientific applications.

The S/360 logical design was formed with these overlapping requirements. The Central Processing Unit (CPU) and the data handling capabilities of the CPU provide an insight into how the S/360 meets the requirements of both scientific and commercial data processing.

The diagram in Figure 1 depicts a typical S/360 data processing unit. The heart of the system is the Central Processing Unit (CPU). The function of the CPU is to fetch instructions from and perform the indicated operations on data stored in main storage.

●●●

2 Figure 2 illustrates the component parts of the CPU and the logic flow within the CPU.

The two main sections in the CPU are: 1) the control unit (system control) and 2) the arithmetic and logical unit (called ALU).

From the illustration (Figure 2), you should be able to see some of the functions of the control unit. They are:

1. All reference: o main storage, whether for instructions or for data, are made by the control unit.

2. The control unit addresses main storage and causes the instruction to be fetched and sent to the control unit. The instruction is then decoded by the control unit and executed.

The instruction is brought out of main storage to the .... unit. The control unit decodes the ....

All addresses are supplied to the main storage by the.......... unit.

●●●

control
instruction
control

3 The CPU is made up of two sections. They are the .... and ....

●●●

control unit
arithmetic and logical unit (ALU)

4 Let us look now at the arithmetic and logical unit.

In general, the ALU contains the circuits necessary for arithmetic operations such as add, subtract, multiply and divide. It also contains the circuitry necessary for logical operations such as comparing, moving, bit testing, shifting and editing.

Arithmetic operations treat the data as numbers. Logical operations do not.

●●●

5 As can be seen from the CPU Logic Flow illustration, the ALU can do:

1. Variable-field-length operations.
2. Fixed-point operations involving fixed-length fields.
3. Floating-point operations.

In your own words, what is the function of the arithmetic and logical units of a computer?

....

●●●

The function of the ALU is to perform operations on the data fields, and carry out the operations defined in the instructions being executed.

6 List some of the arithmetic functions performed by the ALU.

●●●

add, subtract, multiply, divide

7 List some of the logical functions performed by the ALU.

●●●

Move, compare, shift, edit

8 What three types of operations can the ALU do?

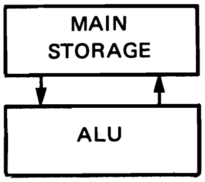
●●●

1. Variable-field-length operations.
2. Fixed-point operations.
3. Floating-point operations.

9 Each of these operations has certain advantages and characteristics associated with them. In looking at the ALU, we will examine these advantages and the types of data associated with these operations. We will also examine the characteristics of the instructions generally associated with the three types of operations.

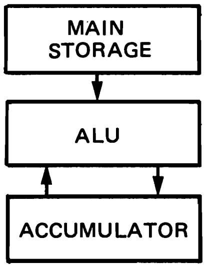
●●●

10 In looking at the ALU, let us first consider variable-length fields as used in many commercial computers of the past. Two main concepts were used. The storage-to-storage concept was used by computers of the 1401 family. In it, the data fields were brought out of main storage, operated upon, and the results went back into main storage.



STORAGE—TO—STORAGE CONCEPT

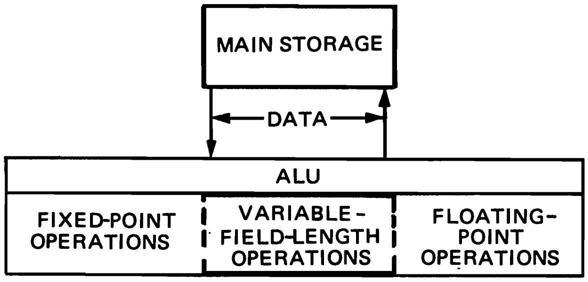
Other computers such as those of the 702-705 family used a storage-to-accumulator concept. The accumulator was a small storage device. The medium could be core storage, vacuum tubes or transistorized registers. In the storage-to-accumulator concept, one of the data fields would be in main storage and the other would be in an accumulator. Both fields would be brought out to the ALU, operated upon, and the result would go back into the accumulator.



STORAGE—TO—ACCUMULATOR CONCEPT

●●●

11 For its variable-field-length operations, the System/360 uses the storage-to-storage concept.



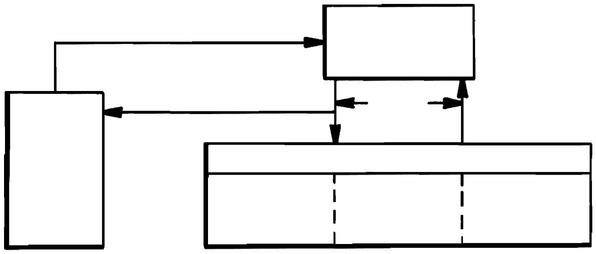
●●●

12 Sections of the System/360 necessary for a variable-field-length operation, are shown in this frame. Label the blocks as to:

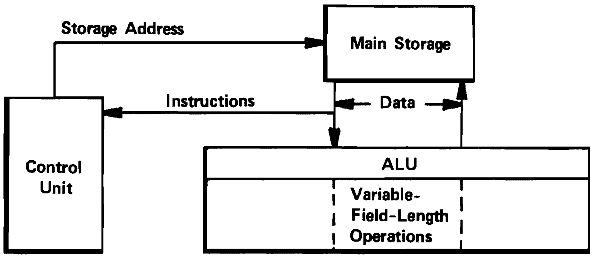
Control Unit
Main Storage
ALU
Variable-Field-Length Operations

On the lines connecting the blocks, indicate whether they are:

Storage address
Instructions
Data

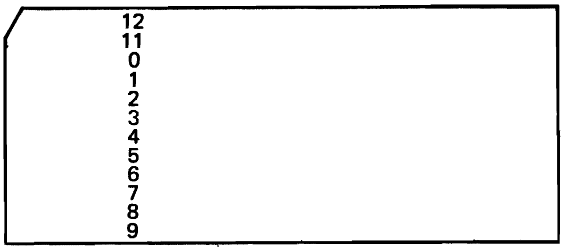


●●●



13 Usually, variable-field-length operations involve character data and, as such, these operations are used principally in commercial applications.

You will recall that in an IBM punched card, data is stored in Hollerith code. In Hollerith code the card takes the familiar pattern shown here:



with 12 rows and 80 columns.

Each column may contain a character or a digit.

A digit 5 would be stored by punching a hole in row .... of a column.

●●●

5

14 Holes in rows zero through 9 are called digit punches.

To code a letter, a combination of zone and digit punches is used. In Hollerith code, a hole in the ...., ...or...........row is called a zone punch.

●●●

12
11
zero

15 Note that the zero row may contain a zone or a digit punch.

The letter "A" is made up of a 12- and a 1-punch, a "B" of a 12 and a 2. What punches make up a C?

●●●

A 12- and a 3-punch.

16 The letters "J" through "R" are coded using an 11 - punch with a digit punch one through 9. The letters "S" through "Z" use a zero zone and digits 2 through 9. Hollerith coding is also known as zoned decimal format.

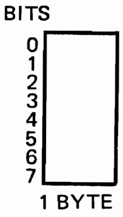
S/360 uses the .... code within main storage

●●●

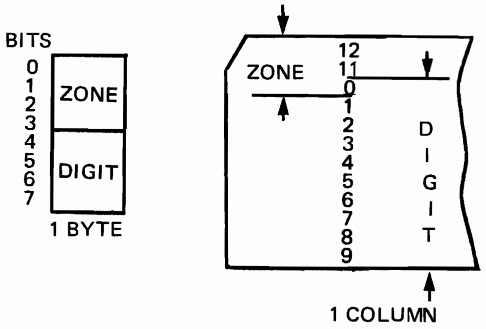
EBCDIC (Extended Binary Coded Decimal Interchange Code) (or Extended BCD)

17 The EBCDIC format for character data represents digits in a zoned decimal format with one character per byte.

For a minute, let us stand a byte (8 bits) on end. Bit 0 will be at the top and bit 7 at the bottom as shown in this illustration:



Now divide the byte in half by drawing a line across it. If we label the top half ZONE and the bottom half DIGIT, we can readily see the resemblance between the byte and a card.



●●●

18 The coding structure of EBCDIC is the same as the coding structure of the card. Each contain a .... and .... portion.

●●●

zone
digit (decimal)

19 Looking first at the digit portion of the byte, we see that .... bits must be used to represent the decimal digits 0-9.

●●●

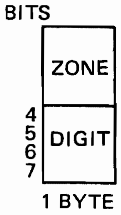
four (bits 4 through 7)

20 This is acceptable because the decimal digits 0-9 may be represented by their .... equivalents in the digit portion of the byte.

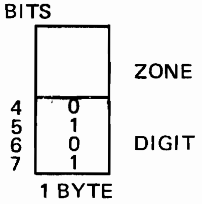
●●●

hexadecimal (binary) (decimal 0 is 0000
1 is 0001
2 is 0010
3 is 0011, etc.)

21 How would the digit 5 be represented in bits 4 through 7?



●●●



The zone portion of the card code may be similarly represented by the upper half of the byte (bits 0 through 3).

22 Remember that on the card, a 12, 11 and a zero zone were used to define letters and a no-zone condition defined digits. For example, a 12-punch (zone) and a 1-punch (digit) represent an A, an 11-punch and a 2-punch represent a K, and a 3-punch alone (no-zone) represents the digit 3. The same situation exists in the byte. Here bits .... through .... represent zone and no-zone conditions.

●●●

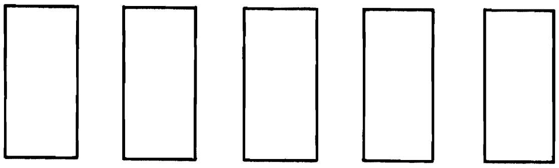
0
3

23 In EBCDIC, a 12-zone is represented by a hexadecimal (hex) C, and 11-zone by a hex D, a zero-zone by a hex E and a no-zone by a hex F.

(You will recall that:
hex C = 1100
hex D = 1101
hex E = 1110
hex F = 1111)

Split each byte shown into zone and digit sections and show the bit representation of the following numbers and letters:

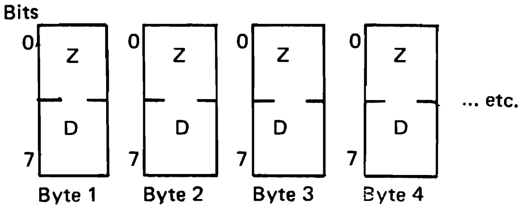
3, 7, A, C, G



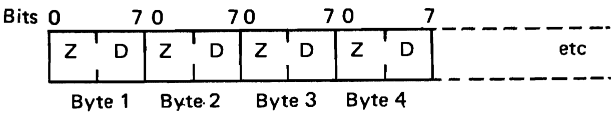
●●●

	1	1	1	1	1
Z	1	1	1	1	1
	1	1	0	0	0
	1	1	0	0	0
	0	0	0	0	0
D	0	1	0	0	1
	1	1	0	1	1
	1	1	1	1	1

24 We have been showing the byte in a vertical position for convenience of comparison with the card data format. Remember, the byte may represent data in other formats. Some of these formats require more than one byte. Because of these additional requirements, it is conventional to show the bytes strung out end to end.



Conventionally, these bytes are shown like this:



●●●

25 In addition, the bit structure within a byte is represented by two hexadecimal digits.

Hexadecimal is a shorthand representation of a binary number containing .... bits.

●●●

4

26 What would be the hexadecimal representation of the following characters:

1111	1001	1111	0111	1100	0100

●●●

F	9	F	7	C	4

27 You can recognize these characters as the numbers and letters ...., ... and ....

●●●

9
7
D

28 We have been working with unsigned numbers. Generally, unsigned numbers are assumed to be positive. But, in many cases, numbers may be positive or negative, and to distinguish one from the other, plus and minus signs are used. In S/360 the sign of a number is stored as the zone portion of the low-order byte. Negative numbers are indicated by a hex D. Hex D corresponds to the 11-punch in punch card practice. Thus, a –17 looks likethis:

F	1	D	7

All other zones (hex C, E and F) are always assumed positive.

Using a hex C for a plus (+) sign and a hex F for an unsigned number, diagram the following numbers:

a. 023
b. +175
c. –329
d. –871
e. +007

●●●

a.

F 0	F 2	F 3

b.

F 1	F 7	C 5

c.

F 3	F 2	D 9

d.

F 8	F 7	D 1

e.

F 0	F 0	C 7

29 In S/360, the zoned decimal numbers must be in another format for the variable-field-length arithmetic and edit operations. This format is called the "packed" format.

Here the flexible byte may be used to store two digits.

Packed decimal format provides increased arithmetic performance and improved rate of data transmission.

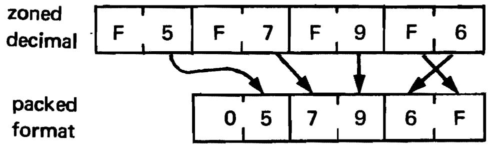
Give the hex representation of the 5796 in zoned decimal:

●●●

F 5	F 7	F 9	F 6

30 In packed format, the zones are removed from the numbers (except for the low-order number) and then the numbers are compressed (packed) into a shorter field.

For example, using the previous example, packing produces this result.



Three actions occur:

1. The numbers are packed from 4 bytes into 3 bytes.

2. In the low-order byte, the zone and digit are interchanged.

3. Any unfilled high-order portions in the packed field are padded with zeros.

Pack the following data into a 5-byte field:

zoned decimal	F 3	F 7	F 1	F 4	F 8
packed format					

●●●

0 0	0 0	3 7	1 4	8 F

31 Let us look into the purpose of action 2. Remember that in a previous frame we had lookebd at signed numbers. In the zone decimal number format, the sign of the number is stored in the zone portion of the .... order digit.

●●●

low

32 Action 2 thus provides us with a means of retaining the sign of the number when doing a PACK operation. Using a hex C for a plus sign, a hex D for a minus sign, and a hex F to indicate an unsigned number, show the zone-decimal format and the packed format for the following numbers in 3 byte fields.

a. + 123
b. - 456
c. - 789
d. + 024
e. 680

●●●

	Zoned-Decimal			Packed		
a.	F 1	F 2	C 3	0 0	1 2	3 C
b.	F 4	F 5	D 6	0 0	4 5	6 D
c.	F 7	F 8	D 9	0 0	7 8	9 D
d.	F 0	F 2	C 4	0 0	0 2	4 C
e.	F 6	F 8	F 0	0 0	6 8	0 F

33 We have been using a hex C to designate a plus sign. In S/360 variable-field-length arithmetic operations, a positive result is always signed with a hex C. Thus, the hex C is said to be the standard plus sign.

Which of the following hex zones designate a plus sign and which designate a minus sign?

Which one is used as the standard plus sign and which one as the standard minus sign?

a. hex C
b. hex D
c. hex E
d. hex F

●●●

1. Plus sign hex C
 hex E
 hex F
2. Minus sign hex D
3. Standard plus sign hex C
4. Standard minus sign hex D

34 It can be seen that packing results in a field made up of an odd number of digits and a sign.

digit	digit	digit	digit	digit	sign

What are two benefits derived from the use of packed fields in variable-field-length arithmetic?

●●●

1. Increased arithmetic performance.
2. Improved rate of data transmission.

35 Variable-field-length operations normally work on data in the .... or .... format.

●●●

EBCDIC (zoned decimal) (character)
packed decimal

36 The operations to be performed on these forms of data are specified within certain instructions. Let us now look at the characteristics of these instructions.

Variable-field-length operations in S/360 use the s..........-to-s.......... concept.

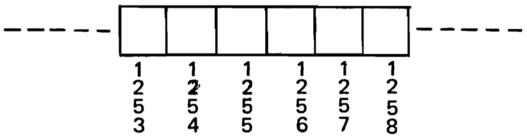
●●●

storage
storage

37 In the storage-to-storage concept, the data fields involved in the operation are both in main storage. Thus, in an instruction made up of an operation code and two operands, each operand must refer to its respective storage address. A storage address always refers to a single byte.

In S/360, a field is always referenced by the address of the high-order (leftmost) byte.

If a field in main storage in S/360 extends from main storage address 1253 to main storage address 1258 (as shown in the figure below),



this field is always referenced by the address.

a. 1253
b. 1255
c. 1258

●●●

a.

38 Variable-length fields can start at any byte location in main storage. However, there must be some way of indicating to the system the length of the fields. S/360 specifies the length of these fields by a length code in the machine language instruction.

Variable-length fields can start at .... byte location in main storage. Their length is specified by (in your own words) ....

●●●

any
a length code in the machine language instruction

39 The length code can be either 4 or 8 bits long, depending on the instruction. The length code is in binary. As a result, the maximum data field length can be either 16 or 256 bytes. The value of the code is one less than the total number of bytes.

Length code of 0000 = 1 byte
Length code of 1111 = 16 bytes
Length code of 11111111 = 256 bytes

A length code of 0111 would specify a variable field length of how many bytes? ....

● ● ●

8

40 So far, we have discussed variable field length data formats in storage and the length of the field in bytes. Now let us discuss the format of the instructions which control the operations on the data.

Instructions may be represented in three forms: actual, symbolic and explicit.

The "actual" (or machine language) form of the instruction is the form in which the instruction is stored within the computer ready for execution. This is the output of the language translator programs (which will be covered later) and is never written by the programmer.

A symbolic instruction is one in which a data field is given a symbolic name (label) which identifies the field meaningfully.

For example: RECEIPTS, ISSUES, TOTAL. These symbols are converted to addresses and length codes by the language translator program.

Finally, an explicit instruction is one in which the address and length (in bytes) of a field are explicitly coded by the programmer.

Actual (or machine language) coding is not done by the programmer in S/360. We will therefore confine our discussions to the symbolic and explicit forms of the instructions.

In what three forms may an instruction be represented?

● ● ●

1. actual
2. symbolic
3. explicit

41 The symbolic and explicit forms of the instructions have the same general format.

OP-CODE 1ST-OPERAND, 2ND OPERAND

The Op-Code (Operation Code) is a mnemonic abbreviation for the operation to be performed on the fields represented by the first and second operands.

For example, A, S, M and D stand for the arithmetic operations ...., ..., ... and .... respectively.

● ● ●

add
subtract
multiply
divide

42 In addition, the direction of data flow is the same in both the symbolic and explicit forms of the instruction. Then too, these forms may be combined within an instruction.

Let us examine the symbolic instruction first and then compare it to the explicit form.

● ● ●

43 We have seen that in order to define a variable-length field, we must specify the .... and .... of the field.

● ● ●

address (location)
length

44 When using a symbolic programming language, the label the programmer assigns to a field must have two attributes. That is, the labels must define the named field by implying its .... and ....

●●●

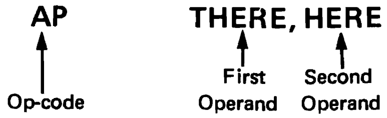
address (location)
length

45 An instruction is made up of an .... and two ....

●●●

op-code
operands

46 Here is an example of symbolic coding.



(AP means Add Packed.)

This example shows an instruction with two symbolic labels.

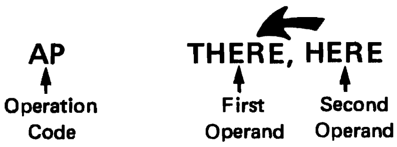
What attributes does each label have?

●●●

1. address (location)
2. length

47 Now let us look at some of the general characteristics of variable-field-length instructions.

For the S/360, the direction of operations in variable-field-length instructions is always from the second operand to the first. For example:



In this instruction, the number stored at the symbolic main storage address label "HERE", is added to the number stored at "THERE" in the ALU, and the result is stored in the location "THERE".

HERE and THERE identify the address of the .... (high/low)-order byte of the fields.

●●●

high

48 In addition to the address, a symbolic label must also define the implicit field length of a variable-length field.

The maximum length field defined by a symbolic label is determined by the instructions in which it will be used. The maximum length field may be either .... or ...., depending on the instruction used.

●●●

16
256

49 The minimum length field for a storage-to-storage operation must be .... byte(s).

●●●

one

50 Now let us take a look at the explicit format of an instruction instead of the symbolic format. In the explicit format, the address and the field length of each operand are defined.

AP THERE, HERE
SYMBOLIC CODING

AP 17 (5, 11), 473 (4, 11)
EXPLICIT CODING

The two instructions shown here will generate exactly equivalent machine instructions in our program. In explicit coding, the programmer actually defines the address and length of the fields involved in the operation.

Explicit coding is not used very often but can provide coding advantages under some programming situations. Symbolic labels are ordinarily very much easier to use and are more meaningful to the programmer.

Before proceeding further, let us look at the addressing scheme used in S/360.

●●●

51 In S/360, a main storage address is defined in base-displacement form. The base is the reference point from which we determine the data address. The reference point may be any address in main storage, 10,756; 87,329; 1,329,483; etc. Now the data location is determined with respect to this base (reference point).

To the base is added a number (from 0 to 4095) which is called the displacement. Thus the data address in main storage is the sum of the .... and the ....

●●●

base
displacement

52 The base is stored in one of the general registers and the displacement is stored in the instruction itself. The control unit decodes the instruction, goes to the proper base register (the general register containing the base) and then adds the displacement to the base to form the storage address.

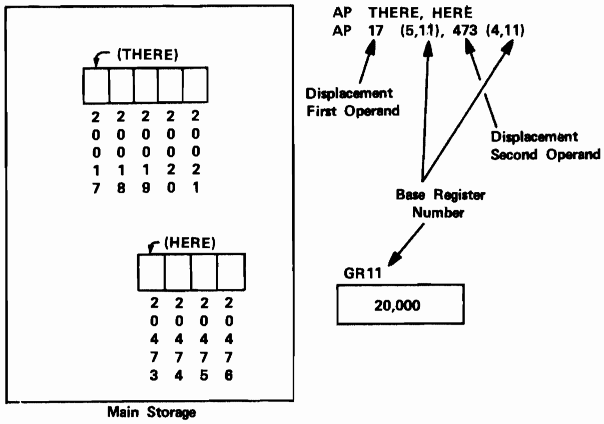
Base-displacement addressing was selected for the S/360 for several reasons. Some of these are:

1. The program may directly access a very large main storage (16,777,216 bytes).

2. Fewer and shorter standard formats were required for the instructions because only 4 bits are needed to point to the general storage address in the computer.

3. Programs could be easily relocated, that is, they could be assigned to any starting main storage address in the computer.

Let us look at an example of how base-displacement addressing works.



This diagram represents two fields in storage. The instruction in which these fields are used is shown in both symbolic and explicit form. These fields are:

1. THERE with a displacement of 17, using a base contained in base register 11 and a length of 5 bytes.

2. HERE with a displacement of 473, using a base contained in base register 11 and a length of 4 bytes.

The instruction is decoded by the ....

●●●

control unit

53 To determine the location of the first operand (THERE), the control unit recognizes that the displacement is 17 and the base is contained in base register 11. (General Register 11). The control unit adds the displacement to the base to determine the address of the data field.

In the example shown, what is the address stored in the base register?

●●●

20,000

54 The first operand is THERE and has a displacement of 17. What is its address?

●●●

20,017

55 The address of a field in S/360 refers to the .... (high/low)-order byte of the field.

●●●

high

56 The address of the second operand, HERE, is determined in the same manner. This field has a displacement of 473 and uses base register 11. What is the address of HERE?

●●●

20,473

57 Although the same base register was used in one example, it was done so only for the sake of simplicity. Each field could have a different base register.

One use for the 16 g.......... r.......... in S/360 is as base registers.

●●●

general
registers

58 We have seen how an address is generated using the base-displacement principle. In addition, in order to define a field we must also specify its ....

●●●

length

59 The op-codes are prepared in mnemonic form to make them easy to remember. Likewise, the definition of a field may be abbreviated. What would you say would be the abbreviations or mnemonics for base, displacement and length?

●●●

B
D
L

60 The language translator program requires that these be grouped and punctuated in a specified order when defining fields.

The abbreviations are grouped as follows:

D (L, B)

When explicitly coded in an instruction.

Subscripts are used to distinguish the operands.

$D_2 (L_2, B_2)$

would refer to the ...., ..., and .... of the .... operand.

●●●

displacement
length
base
second

61 The Add instruction would have the following format in explicit coding:

AP $D_1 (L_1, B_1), D_2 (L_2, B_2)$

Identify the op-code, the second operand and the first operand in the above instruction statement.

●●●

Op code	AP
Second operand	$D_2 (L_2, B_2)$
First operand	$D_1 (L_1, B_1)$

62 In the preceding example, the displacements (D_1 and D_2) may have any value from 0 to 4095. The fields lengths (L_1 and L_2)

may be any number of bytes from 1 to 16. The registers (B_1 and B_2) containing the base addresses are specified by the numbers which identify the registers used and may be 0 to 15.

●●●

63 The instruction shown might be coded

AP 17 (5, 11), 473 (4, 11)

What is the address of the field identified in the second operand?

●●●

473+ (the contents of base register 11).

64 We remind you that the base could be any address up to the number of bytes available in your specific machine (65,535 in the largest Model 30).

Yes, 65,535, because in S/360 the address of the first byte is 0 (zero). Thus the highest address in a 65,536 byte main storage would be 65,535.

What is the displacement of the first operand field?

●●●

17 bytes

65 What is the length of the field defined by the first operand?

●●●

5 bytes

66 Remember, this is programmer coding so that the actual length of the field, not its length code, is specified in the instruction.

What is the length of the field specified by the second operand?

●●●

4 bytes

67 Where will the sum be found after the addition is completed?

●●●

in the first operand field.

68 There is one additional form of the storage-to-storage instruction. We have been looking at symbolic and explicit coding which refers to the addresses of fields of data. There are some cases in which only one byte of data is required for an operation. For example, initializing, setting, resetting, and testing switches, and inserting code characters or special symbols such as a dollar sign or an asterisk into a field.

Convenience in coding and reduced execution time could be gained if this byte did not have to be addressed but was available immediately. There is such a form of instruction and the byte of data is referred to as Immediate data.

What would be the mnemonic for Immediate data?

●●●

I

69 The immediate data is always the second operand and is always defined explicitly.

MVI $D_1(B_1), I_2$

Why is no length code required in the first operand? (in your own words)

●●●

I_2 is always only one byte of data.

70 I_2 is called "immediate" data because it does not have to be addressed by the instruction. The reason it does not have to be addressed is that it is stored within the actual instruction itself and is immediately available.

This instruction format is called the storage immediate instruction.

●●●

71 Note that a symbolic label may be used for the first operand.

For ease of coding, the symbolic language translator permits the definition of the byte of immediate data in any convenient S/360 data format including binary, character, or hexadecimal.

●●●

72 Let's review what we've covered concerning variable length field instructions:

1. May operate on 1 to 16 bytes or 1 to 256 bytes of data depending on the specific instruction.

2. Operate on data from the second operand to the first operand.

3. Have both operands specify main storage locations.

4. Specify main storage locations using base-displacement addressing.

5. Must have the length of at least one field explicitly defined, depending on the specific instruction used.

6. May store one byte of immediate data within the instruction itself when using the storage-immediate instruction format.

●●●

73 Another type of operation available in the ALU for S/360 is the ability to work with fixed-length or word data.

Fixed-field-length operations are performed on binary data. Speed of execution is the principal benefit derived from binary operations. High-speed arithmetic is therefore the largest area of usage for fixed-field-length operations.

Here the computer is working like a scientific computer or "word machine". More and more commercial applications are taking advantage of this S/360 capability.

Fixed-field-length operations are of two types, fixed-point operations and floating-point operations. Let us look first at fixed-point operations.

Fixed-point operations (also known as word operations) are all done using fixed-length fields. These fields may be 2, 4 or 8 bytes in length depending on the particular instruction.

Fields 2, 4 and 8 bytes long are known as a ...., ... and ...., respectively.

●●●

halfword
fullword
doubleword

74 Variable-length fields have a length of.......... or .... bytes, depending on the instruction.

●●●

1 to 16
1 to 256

75 Normally, fixed-point data consists of signed whole numbers. The high-order bit of a signed number is always the sign. A zero bit represents a plus sign and a 1 bit represents a minus sign.

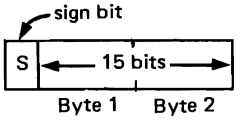
How many bits are available for numeric data in a halfword?

●●●

15

Did you forget that the high-order bit is a sign bit?

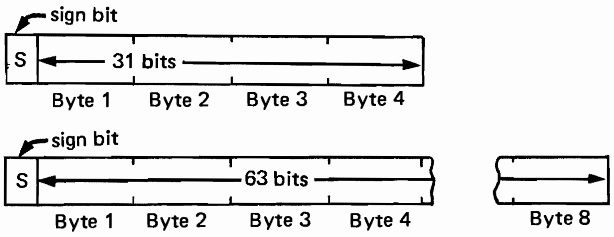
76 Here two of those versatile bytes are working together to form a halfword of data.



Unlike the independent bytes containing character data in variable-field-length operations, in fixed-point operations, the bytes cooperate and the data may extend over more than one byte.

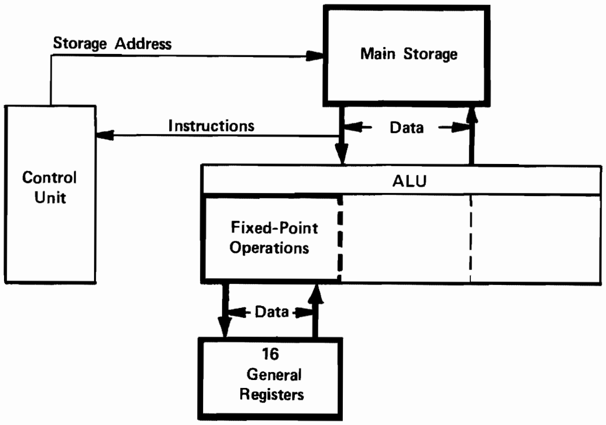
Diagram a fullword (word) and a doubleword in the form shown for the halfword above.

●●●



77 In fixed-point operations, no length specification is required as the specific instruction used will determine whether a halfword, word or doubleword will be required.

So far, we have been discussing formats of fixed-length data. Let's now consider the types of operations that can be performed on fixed-field-length data. Because fixed-length data may reside in either main storage or registers, we can operate in either the storage-to-accumulator, or accumulator-to-accumulator concept.



●●●

78 When operating on fixed-length fields (such as halfwords, words, or doublewords), the S/360 uses the storage-to-accumulator concept and the accumulator-to-accumulator concept. These fixed-length operations use binary operands. For use as accumulators, the S/360 has .... registers available to the programmer. As these registers can be used for purposes other than accumulating, they are called ....

●●●

16
general registers

79 When working with fixed-length operations, the S/360 uses a s..........-to-r..........concept and a r..........-to-r..........concept.

●●●

storage
register
register
register

80 For use as accumulators, the programmer has available 16 .... These registers are numbered 0-15 and are addressed in an instruction by their decimal register number.

●●●

general registers

81 A register is nothing more than a small piece of very high-speed storage used in conjunction with the ALU and the control section of CPU. In S/360, registers are made up of 32-bit positions. This, then, determines the length of a "word" in S/360.

Since a byte contains .... bits, ... bytes are required to make up a word (or fullword).

●●●

8
4

82 It is important to note that there are certain restrictions on the use of general registers. For example, the Disk Operating System uses registers 1, 13, 14 and 15 during input and output operations. When an I/O operation is called for during a program, any data that the programmer has accumulated in these registers will be destroyed. Thus, while the programmer can use these registers, it is probably not desirable to do so.

83 To use register 2 as an accumulator, what address is given?

●●●

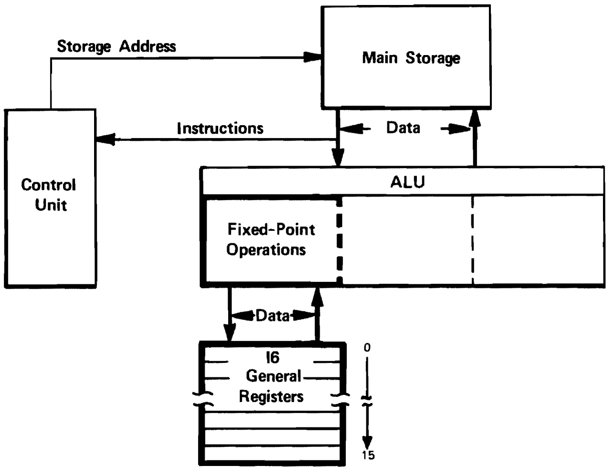
2

84 General registers 0-15 are all one word in length. How many bytes may be contained in a general register? ....

●●●

4

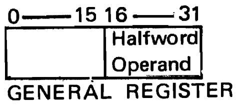
85 With sixteen general registers, sometimes both fixed-length binary operands will be in the general registers. In these cases, another data flow concept is used. The S/360 can do a register-to-register (accumulator-to-accumulator) operation.



●●●

86 A register-to-register operation is the fastest operation possible on the S/360 computer. These operations may involve single or double registers depending on the requirements of the specific instruction.

Being a word in length, a general register can be used to contain a halfword data field. Data fields are sometimes referred to as operands.



As can be seen in the preceding figure, the bits of a general register are numbered left to right starting with the number 0. Also, we can see that a halfword operand is placed in the low-order bits (16-31) of a general register.

●●●

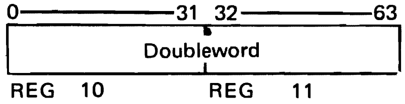
87 None of the general registers 0-15 can contain a doubleword. For those operations that use a doubleword operand, such as fixed-length divide, a pair of adjacent registers is used. In these cases, an even-odd pair of registers (such as 2-3 or 6-7) is used, and the even register is addressed.

With general register 10 specified, which two general registers would be used in a fixed-length divide operation? ....

●●●

10 and 11

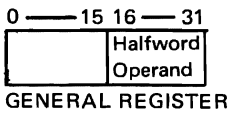
88 In the preceding question, bits 0-63 of the doubleword would be in the registers as shown below.



●●●

89 Number the bit positions of the general register below. Also show where a halfword operand would be placed.

●●●

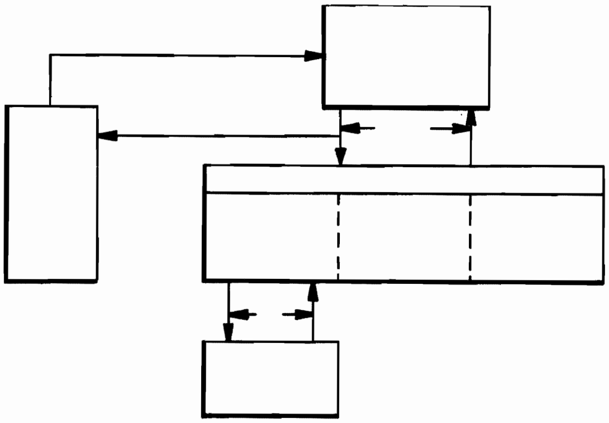


90 Sections of the S/360 necessary for fixed-length operations, are shown in this frame. Label the blocks as to these five items:

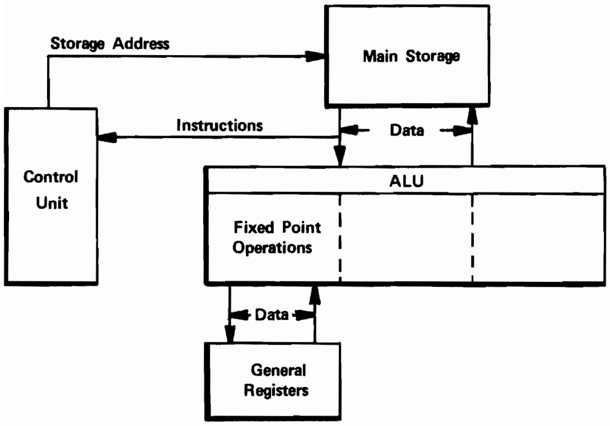
Control Unit, ALU, Main Storage,
Fixed-point Operations, General Registers

On the lines connecting the blocks indicate whether they are:

storage addresses
instructions
data



•••



91 We have been discussing the types of operations permitted with fixed-point data. Now let's look at the instructions involved.

The instructions that handle fixed-point operations can be characterized rather simply.

Fixed-point operations use the r..........-to-r.......... and the s..........-to-r.......... concepts.

•••

register
register
storage
register

92 In the register-to-register concept, both data fields reside in the ....

•••

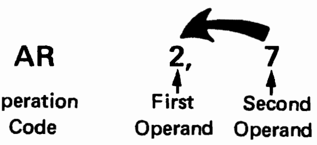
general registers (registers)

93 The instructions for register-to-register operation contain an operation code and two operands. These operands are the numbers of the general registers containing the data. As with variable-length fields, the direction of operation is from the .... operand to the .... operand.

•••

second
first

94



(AR means Add Register-to-register.)

In this instruction, the binary data in register .... is added to the data in register .... and the sum is stored in register ....

•••

7
2
2

95 Because this is a fixed-field-length operation, no length code is required in the instruction.

What is the length of the data involved in this operation?

•••

one word

96 Remember, a register contains one fullword of information.

•••

97 In the storage-to-register concept, one data field resides in main.......... and one resides in a..........

●●●

storage
general register (register)

98 Usually, the instructions for storage-to-register operation contain an operation code and two operands. The first operand is always a general register and the second operand is a main storage location. The direction of operation is from the .... operand to the .... operand, except when storing information from a register to main storage or when converting a binary number in a register to packed data in main storage.

●●●

Second
First

99 An example of a storage-to-register instruction is shown below

A 9, IN (3)

(A means Add storage-to-register.)

The binary number at .... is added to the number at .... and the sum is stored at ....

●●●

IN
register 9
register 9

100 The symbolic label IN may be replaced with an explicit operand. In this case, the base and displacement address is supplemented by one more address component, called an Index.

The index is used and stored in exactly the same way as the base, except that the index applies only to the instruction that references the index.

The base address is stored in a ....

●●●

register

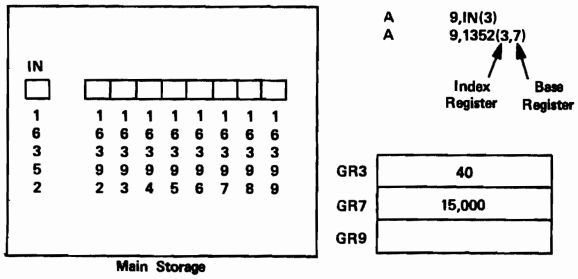
101 The index is also stored in a register.

The storage address referenced is the sum of the base + index + ...

●●●

displacement

102 Address generation using an index is an extension of base-displacement addressing. Here is an example of how it might appear.



We have seen that in base-displacement addressing, the address of the desired field was generated by .... (in your own words).

●●●

adding the displacement to the base.

103 The base is stored in a ....

●●●

general register (base register)

104 Similarly, in indexing, the index is added to the base-displacement address to determine the data address.

Thus, the data address is made up of the sum of the displacement + the base + the ....

●●●

index

105 Like the base, the index is stored in a ....

•••

general register (index register)

106 Note that in the symbolic format of the instruction, the index register is explicitly defined. The field defined is at the location of the indexed address of IN.

In our example, the field IN has been indexed by the index in register 3. If register 7 is the base register, and the displacement is 1352, what is the data address?

•••

16,392 (1352 + 15,000 + 40)

107 Indexing is useful where the processing of successive fields within a record is to be done. Modification of the program base register is not practical for this purpose because it serves as the base register for all the data and instruction addresses within a 4096 byte area of main storage.

The program may provide indexing by the required increment simply and rapidly.

In our example, assume that we wished to perform the same series of operations on the two fields of data located at IN(3) and IN(3) + 4.

With the registers set as shown, the address of the data is ....

•••

16,392 (1352 + 15,000 + 40)

108 After processing the data at that location, we wish to process more data in the same record. This data is made available by adding 4 to the index register. Its contents will now be ....

•••

109 The effective address will now be ....

•••

16,396 (1352 + 15,000 + 44)

110 The second portion of the data may now be processed.

•••

111 The explicit format of the instruction may be expressed in its general form quite readily. Like the mnemonics for base and displacement, index is also simple. It is X.

Base, displacement and index have the mnemonics ...., ... and .... respectively.

•••

B
D
X

112 Writing the instruction explicitly, the form shown here is followed:

$A \quad R_1, D_2 (X_2, B_2)$

where X is the register number containing the index amount.

In the following instruction,

A 9, 1352 (3, 7)

at what storage address will the field specified by the second operand be found?

•••

1352 + (the contents of base register 7)
+ (the contents of index register 3)

113 Let's review what we've covered concerning fixed-point instructions.

1. May operate on 2, 4, or 8 bytes depending on the specific instructions used.

2. Operate on data from the second operand to the first except when:

   a. storing data from a register to main storage.
   b. converting a binary number in a register to packed format in storage.

3. One or both operands must specify a general register.

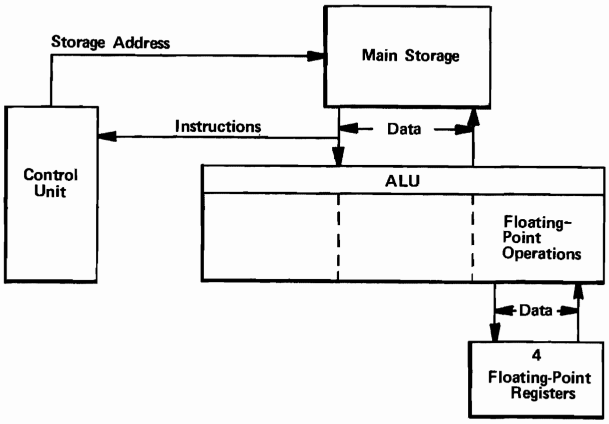
4. May index the main storage data field by using an index register when specific instructions are coded.

5. No length code is required.

●●●

114 Another fixed-field-length operation is "floating-point" operations. Floating-point operations are used almost exclusively in scientific applications.

Floating-point is available on S/360 as an ALU function. As with fixed-point operations, speed of execution can be increased through the use of registers. The ALU makes use of four special floating-point registers.



Floating-point arithmetic is most useful for expressing very large or very small numbers and operating on them with much precision. To do floating-point arithmetic, the S/360 has .... floating-point registers.

●●●

four

115 The four floating-point registers are numbered 0, 2, 4, 6. These are not the same as general registers 0, 2, 4, 6. The floating-point registers are separate registers used only as accumulators during floating-point operations.

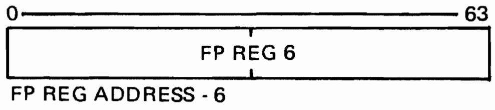
●●●

116 There are .... floating-point registers numbered ...., ..., ..., ...
The floating-point registers are not the same as (in your own words) ....

●●●

four
0
2
4
6
general registers 0, 2, 4, 6

117 The floating-point registers are doubleword registers and are addressed by their decimal register number in floating-point instructions.



Floating-point registers are .... bits long and can contain a ....

●●●

64
doubleword

118 If you are sure that you won't be using floating-point arithmetic, or if you can answer all of the following questions, skip to frame 134.

QUIZ

1. What is the advantage of floating-point representation?

2. How is a floating-point number represented in storage, for each degree of precision?

3. What is excess 64 arithmetic and what is its purpose in floating-point representation?

4. How many significant decimal digits can be kept in short precision floating-point arithmetic? How many in long precision?

●●●

119 Floating-point arithmetic provides an automatic scaling procedure and thus affects the range of numbers representable in a fixed-length field. For example, the largest positive integer that can be represented in binary, in a 32-bit word, is +2, 147, 483, 647 or a little more than 2 x 10^9. By contrast, floating-point allows us to represent +7 x 10^{75} in the same 32-bit word.

Floating-point (slightly/moderately/vastly) .... increases the range of numbers representable in a fixed-length field.

●●●

vastly (by a factor of about 3.5 x 10^{66}).

120 A floating-point number has two parts, an exponent and a fraction, signified by (E,F).

Study the first three examples below, before deriving the fourth:

Decimal Number		Representation as Decimal Fraction times a power of 10
373.645	=	.373645 x 10^3
27.56	=	.2756 x 10^2
.0099	=	.99 x 10^{-2}
147562.	=	....

	Floating-Point (E,F) Notation
=	3, .373645
=	2, .2756
=	-2, .99
=	....

●●●

.147562 x 10^6
6, .147562

121 Reviewing the parts of a floating-point number, in the last example, 6 is the .... and .147562 is the ....

●●●

exponent
fraction

122 The exponent or the fraction can be either positive or negative. Study the first three examples below, before completing the last two:

Decimal Number	E,F Notation
+.00746	-2, +.746
-.000567	-3, -.567
-1744.223	+4, -.1744223
-15280.06	....
+.00007	....

●●●

+5, -.1528006
-4, +.7

123 At this point, it should be noted that S/360 performs its floating-point arithmetic in hexadecimal. Decimal numbers used in calculations must first be converted to hexadecimal and then put in floating-point form. When this is done, the fraction is a series of .... digits to the right of a hexadecimal point. The exponent is a power of ....

●●●

hexadecimal
16

124 A hexadecimal floating-point number is derived in the same way as a decimal floating-point number. Study the first three examples in the following figure; then complete the rest.

Decimal Number	Hexadecimal Equivalent	Representation as a Hexadecimal Fraction times a power of 16	Hex Floating-Point Number
149.25	95.4	$.954 \times 16^2$	+2, + .954
-1710.828125	-6AE.D4	$.6AED4 \times 16^3$	+3, - .6AED4
.001	.00418937	$.418937 \times 16^{-2}$	- 2, + .418937
-4096	-1000	....	....
.00001	.0000A7C5AC	....	....

●●●

$-.1 \times 16^4$	+4, -.1
$+.A7C5AC \times 16^{-4}$	-4, + .A7C5AC

125 In the preceding examples, we have seen that four items of information are needed to describe a floating-point number:

a. The sign of the exponent.
b. The value of the exponent.
c. The sign of the fraction.
d. The value of the fraction.

Here's how these items are represented in a 32-bit word:

The sign of the fraction is used as the sign of the entire word. It is in the leftmost bit (bit 0). A zero in this position signifies a positive number.

The absolute value of the fraction is represented by six hexadecimal digits in bits 8-31 (four bits per hex digit). The hexadecimal point is not represented. The computer "knows" that the point is supposed to be to the left of the high-order fraction digit, and keeps track of it throughout arithmetic operations.

The sign and the fraction of a floating-point number are shown below. What are they?

0------ 1001 0101 0100 0000 0000 0000

●●●

sign is + (0 in leftmost bit position)
fraction is .954000

126 With 24 bits used for the fraction and 1 used for its sign, 7 bits are left for representing the signed exponent. As a signed binary number, these 7 bits could hold exponent values from -64 to +63.

Circuit design considerations, however, make it advantageous not to carry another sign bit in addition to the usual one in the leftmost bit position. Accordingly, a constant is added to each exponent to make the range 0 to 127. It need not be signed, because it is always positive.

From the negative and positive limits shown above, what would this constant have to be?

●●●

64

127 When an exponent is changed by the addition of a constant, it becomes a "characteristic". Since the floating-point characteristic is produced by adding 64 to the exponent, it is said to result from "excess 64 arithmetic".

What is the excess 64 arithmetic and what is its purpose?

●●●

Excess 64 arithmetic is the addition of 64 to the exponent of a floating-point number, changing it into a characteristic. Its purpose is to change the exponent range (-64 to +63) into a positive characteristic range of 0 to 127.

128 Can any characteristic from 0 to 127 be represented in a 7-bit binary number?

●●●

yes. $N_{max} = 2^n - 1 = 2^7 - 1 = 127$

129 If the characteristic of a floating-point number is 10, what was the exponent?

●●●

-54

130 A word in storage is pictured on the next page. If it contains a floating-point number, what is the characteristic in binary, the sign, and the fraction in hexadecimal?

1110 0011 0010 1110 0110 1001 0000 0001

●●●

99,–,.2E6901

131 In the answer to the last frame, we showed the sign separate from the characteristic and expressed the characteristic as the value of a seven-bit binary number. Actually, a computer printout of storage always shows the contents of a word as a series of hex digits, one for each four bits. How would the printout look, for the word in the preceding frame?

●●●

E32E6901

132 The term "precision" refers to the number of digits (or bits) required (in registers, work areas, etc.) during a calculation, in order to retain the desired number of significant digits in the result. The more significant digits required, the .... (more/fewer) digits (or bits) or precision are required.

●●●

more

133 The precision of a floating-point number depends on the number of significant digits in its fraction. S/360 provides for two degrees of precision, called "short precision" and "long precision."

Short precision is provided by the 32-bit word mentioned in the previous examples. The six hexadecimal digits in the fraction carry enough precision to represent six significant decimal digits.

Very often, more than six significant digits are required. This calls for long precision: The floating-point number occupies a doubleword (64 bits) The sign and characteristic bits, in the doubleword format, are identical to those in short precision. All of the additional 32 bits are used to represent additional hex fraction digits. With the extra digits, the fraction carries enough precision to represent up to 15 significant digits in decimal quantities.

Describe the format of a floating-point word for each case: short precision and long precision.

How many significant decimal digits can be represented in each case?

●●●

Short precision: The floating-point number occupies one 32-bit word. Bit 0 carries the sign (a 0-bit signifies a positive number), and bits 1-7 carry the characteristic. Bits 8-31 carry the fraction as six hexadecimal digits.

Long precision: The floating-point number occupies one 64-bit doubleword. Bits 0-7 carry the sign and the characteristic as in the short precision format. The remaining 56 bits carry the fraction as 14 hexadecimal digits.

Six significant decimal digits can be represented in short precision and fifteen in long precision.

134 We have seen how the S/360 operates upon various forms of data. To take advantage of the various modes of operation of the S/360, separate instruction sets are available.

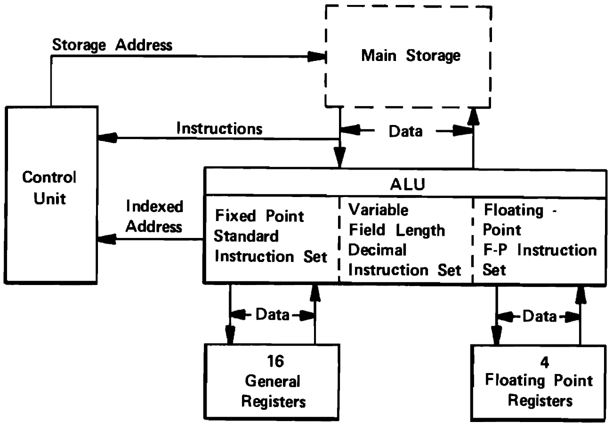
The Standard Instruction Set is supplied as a standard feature on all machines. The Standard Instruction Set supports the fixed-point operations and some of the variable-length field logical operations.

Variable-length field arithmetic and edit operations are supported by an optional instruction set called the Decimal Feature Instructions.

Floating-point operation is also an optional feature. The instruction set is called the Floating-Point Feature Instructions.

The combination of Standard and Decimal instruction is called the Commercial Instruction Set. The Standard and Floating-Point instructions combine to form the Scientific Instruction Set. All three instruction sets make up the Universal Instruction Set.

The general areas of usage of the instruction sets are illustrated in the figure below.


Storage Address
Main Storage
Instructions
Data
Control Unit
Indexed Address
ALU
Fixed Point Standard Instruction Set
Variable Field Length Decimal Instruction Set
Floating - Point F-P Instruction Set
Data
Data
16 General Registers
4 Floating Point Registers

●●●

135 We have examined the CPU and the control unit and the ALU within the CPU. We have examined the general characteristics of the various types of instructions and operations and the types of data on which they operate.

Within the ALU we were introduced to the general registers. The S/360 has 16 general registers, each one word long and four floating-point registers, each a doubleword long. Their speed and versatility are a major feature of the S/360.

In the explicit coding of instructions, we have discussed the base registers, the index registers, and the general registers (for data in register-to-register and storage-to-register operations).

They are called general registers because they are used for other purposes besides accumulating. They are really general purpose registers.

If you recall, in the illustrations of explicit addressing in instructions, the address of a base (B) or index (X) register was identical to the addressing of general registers in register-to-register operations.

They are indeed the same general registers. Thus, the base register and index register usage are two other main uses of general registers.

Variable-field-length and fixed-field-length operation capabilities contribute significantly to the tremendous span of applications to which the System/360 may be directed.

●●●

You have completed this section. At this point you should fill in your notes and take the self-evaluation quiz.

Self-Evaluation Quiz
Central Processing Unit

QUESTIONS

1. What are three identifying characteristics of a commercial computer?

   a. ______________________

   b. ______________________

   c. ______________________

2. What are three identifying characteristics of a scientific computer?

   a. ______________________

   b. ______________________

   c. ______________________

3. The CPU is composed of:

   a. ______________________

   b. ______________________

4. List the major functions of the control unit.

   a. ______________________

   b. ______________________

5. List the two major functions of the ALU.

   a. ______________________

   b. ______________________

6. List the three types of operations handled by the ALU.

   a. ______________________

   b. ______________________

   c. ______________________

7. What is the concept used in S/360 variable-field-length operations.

   ____

8. What type data is usually handled?

   ____

9. What is the principal coding structure used in variable-field-length operations?

   ____

10. What are the hexadecimal and binary representations of the following number + 4672?

11. Give hexadecimal representation of the same number (+4672) in packed format.

12. In what three forms may instructions be represented?

13. What is the general format of S/360 variable-field-length instructions?

14. What is the normal direction of operation in the instruction?

15. What are the two characteristics or attributes that a symbolic operand has?

16. How much data can be operated on by a variable-field-length instruction?

17. How many base registers are available in S/360 for addressing?

18. What range of values may the displacement have?

19. Given below are the base and displacement for several fields. What is the effective address for each?

		General Registers	
a.	B5,D320	GR3	14,250
b.	B3,D105	GR4	17,600
c.	B7,D0	GR5	325,450
d.	B4,D4090	GR6	1,250,740
e.	B6,D3275	GR7	87,000

20. What is immediate data?

21. What is the principal characteristic of fixed-field-length operations?

22. What are the types of fixed-field-length operations handled by the CPU?

23. Describe the form of fixed-point data.

24. Diagram (for fixed-point operations):

    a. A halfword
    b. A fullword.
    c. A doubleword.

25. Fixed-point operations work on __________ and __________ concepts.

26. What is the effect of an index register on address generation?

27. What does each of the following represent?

    a. B
    b. D
    c. L
    d. X
    e. I

28. For what purpose is floating-point representation most useful?

29. Name the three S/360 instruction sets.

30. What is included in the following?

    a. Commercial Set
    b. Scientific Set
    c. Universal Set

Self-Evaluation Quiz
Central Processing Unit

ANSWERS — Frame Reference

1. a. Character data. (1)
   b. High I/O requirements.
   c. Low processing requirement.
   d. Editing.

2. a, Speed (1)
   b. Low I/O requirement.
   c. High processing requirement.
   d. Unedited output.

3. a. Control unit (3)
   b, Arithmetic and Logical Unit (ALU)

4. a. Reference main storage (addressing). (2)
   b. Fetch and decode instructions.

5. a. Arithmetic (4)
   b. Logical

6. a. Variable-field-length (5)
   b. Fixed-point
   c. Floating-point

7. Storage-to-storage (11)

8. Character (EBCDIC) (13)

9. EBCDIC (16)

10. (26)

```
F4         F6         F7          C2
1111 0100  1111 0110  1111 0111   1100 0010
```

11. 04 67 2C (30)

12. a. Machine language (40)
    b. Symbolic
    c. Explicit

13. Op-code First-Operand,Second-Operand (41)

14. From the second operand to the first operand. (47)

15. a. Address (location) (44)
    b. Length (in bytes)

16. 1 to 16 or 1 to 256 bytes depending on the specific instruction. (48)

ANSWERS Frame Reference

17. 16 (numbered 0 to 15). (57)

18. 0 to 4095. (51)

19. a. 325,770 (53)
b. 14,355
c. 87,000
d. 21,690
e. 1,254,015

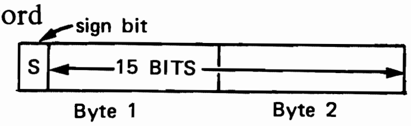
20. One byte of data stored within the instruction itself. (70)

21. Fixed field length operations work with signed binary data. (73)

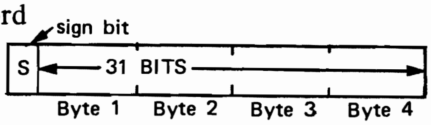
22. a. Fixed point. (73)
b. Floating-point.

23. Signed whole numbers 2, 4 or 8 bytes in length. (75)

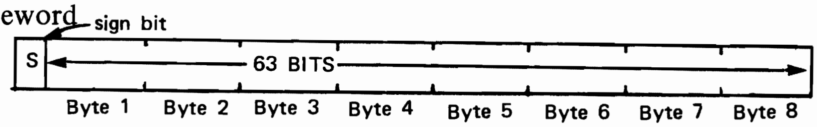
24. a. halfword (76)



b. fullword



c. doubleword



25. a. register-to-register (79)
b. storage-to-register

26. The quantity in the index register is added to the address from the base and displacement to form the effective address. (100)

27. a. Base register (general registers 0-15)
b. Displacement (0-4095)
c. Length (1-16 or 1-256)
d. Index register (general registers 0-15)
e. Immediate data (one byte)

28. Expressing very large or very small numbers. (114)

29. a. Standard Instruction Set (134)
b. Decimal Feature Instructions
c. Floating-Point Feature Instructions

30. a. Standard and Decimal Instructions (134)
b. Standard and Floating-Point Instructions
c. All three instruction sets.

Program Execution

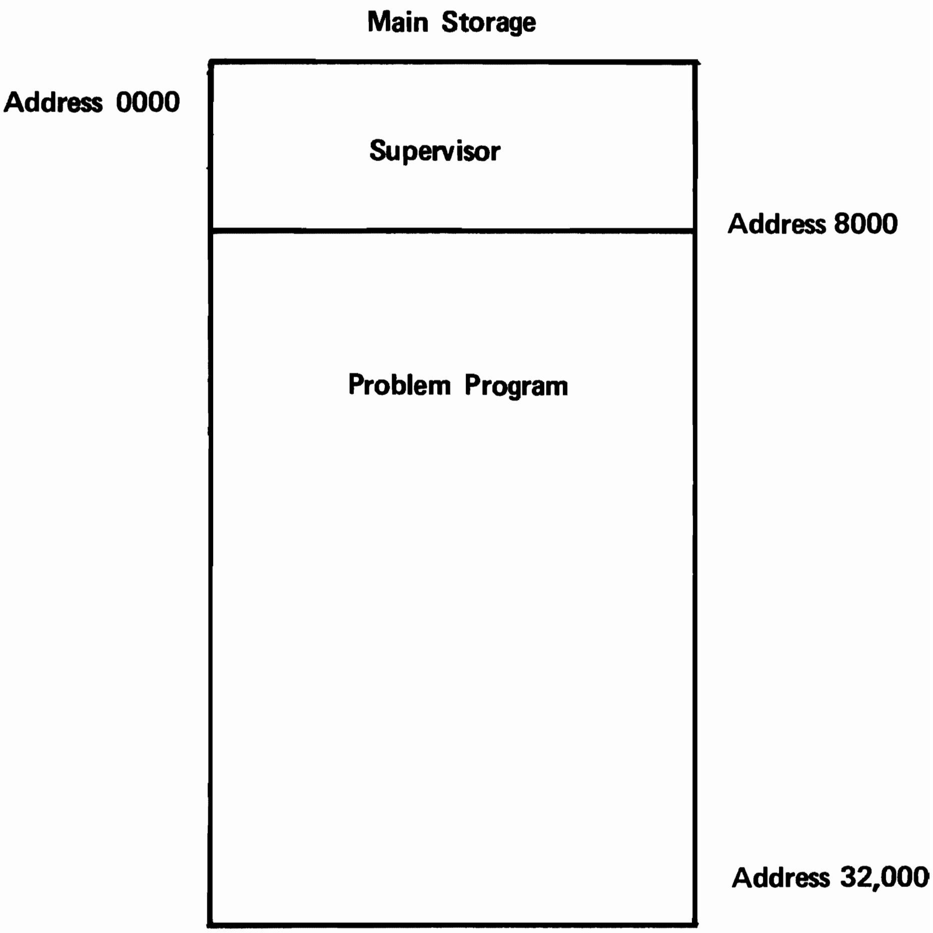

Main Storage
Address 0000
Supervisor
Address 8000
Problem Program
Address 32,000

Figure 1

PROGRAM EXECUTION

1 This topic is concerned with program execution on S/360. We shall discuss the functions of the S/360 control program, the functions and sources of problem programs, the concept of interrupts, the classes of interrupts on S/360, program states, status switching, the program status word, and the order of instruction execution.

●●●

2 In the preceding sections of this course you learned that there must be a control program in main storage at all times while the S/360 is operating. The control program performs several special functions for each of the user's programs.

Figure 1 shows the S/360 control program in main storage. It is called the .... program.

●●●

supervisor

3 The supervisor program is the S/360 control program. According to Figure 1, the supervisor occupies the (lower/higher) .... numbered addresses in main storage.

●●●

lower

4 Which are true?

a. Every S/360 must have a control program in storage while it (the computer) is in operation.
b. The S/360 control program is called the supervisor.
c. The supervisor occupies the higher storage addresses.
d. The supervisor occupies the lower storage addresses.
e. Each user program has its own unique supervisor.
f. The supervisor performs several special functions for each user program.

●●●

a, b, d, f

5 In your own words, define "control program", and identify the S/360 control program.

●●●

A control program is a special program that resides in main storage at all times, performing various special functions for user programs. The S/360 control program is called the supervisor.

6 Figure 1 shows another kind of program in main storage with the supervisor. It is called a .... program.

●●●

problem

7 Problem programs cause the computer to perform the operations required for particular "applications", that is, they make the machine do the work to get specific jobs done.

The programs you write to process payroll data, update inventory records, print bills., etc. are .... programs.

●●●

problem

8 Any program that involves input, processing, and output, and that interacts with the supervisor while it is being executed, is a

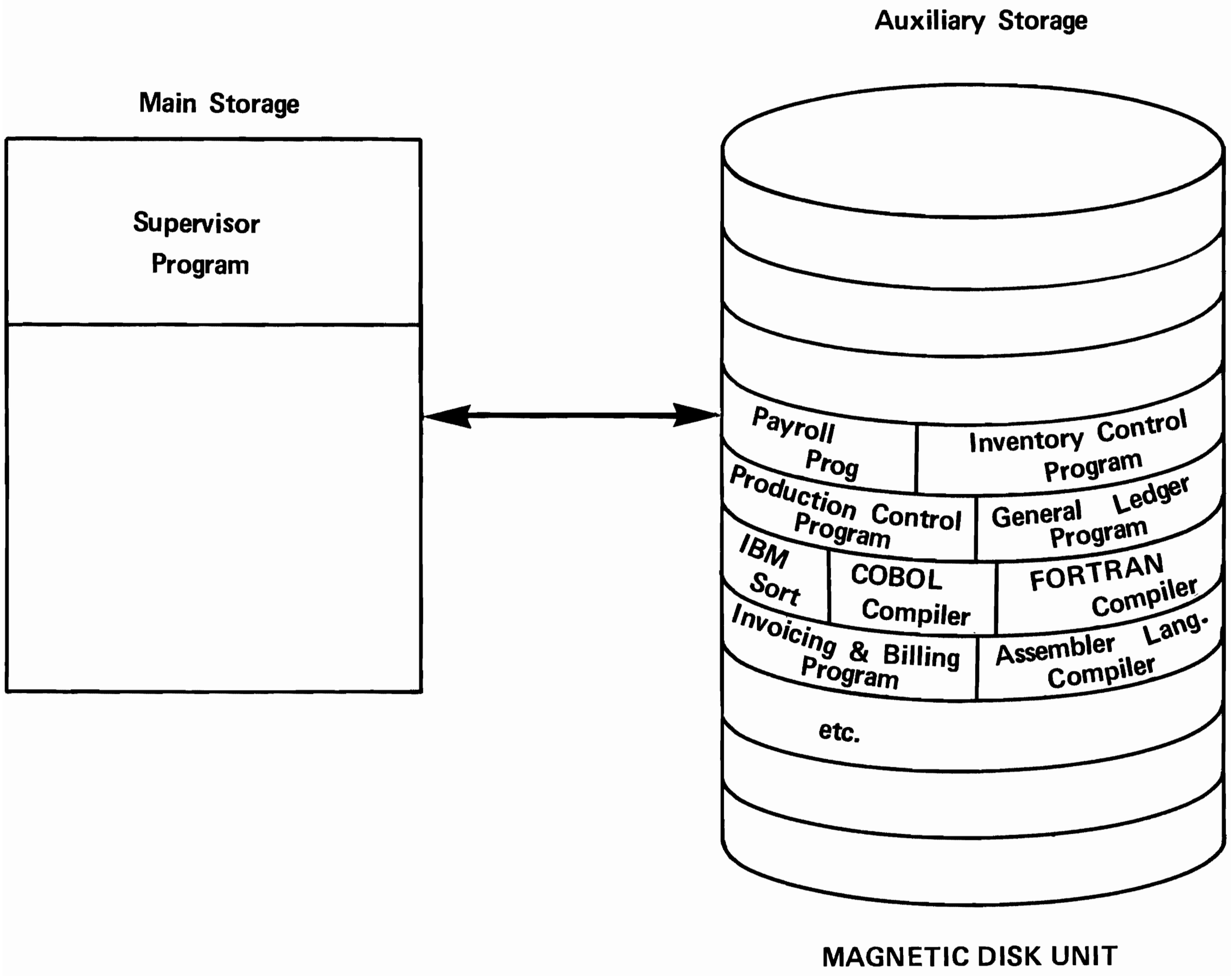

Auxiliary Storage
Main Storage
Supervisor Program
Payroll Prog
Inventory Control Program
Production Control Program
General Ledger Program
IBM Sort
COBOL Compiler
FORTRAN Compiler
Invoicing & Billing Program
Assembler Lang. Compiler
etc.
MAGNETIC DISK UNIT

Figure 2

problem program. Is a compiler a problem program? Why?

●●●

Yes. It involves input, processing, and output, and it interacts with the supervisor during execution.

9 IBM supplies compiler programs; it also supplies other useful programs such as sort programs that put data into a desired sequence, utility programs that transfer records from one medium to another (cards to tape, for example) and many others.

All IBM-supplied programs, with the exception of the supervisor itself, are problem programs. Why are they so classified?

●●●

Because they involve input, processing, and output, and they interact with the supervisor during execution.

10 Both IBM-supplied and user-written programs cause the computer to perform the operations required for particular applications. A program that causes the computer to calculate employee wages and to print checks and check statements would be for a .... application.

●●●

payroll

11 For what application would an IBM-supplied program that places records in a desired sequence be used? A program that translates source language statements into machine language?

●●●

A sort application
A compilation application

12 The term "application", then, refers to the job the program is designed to do. What is the function of a problem program?

●●●

A problem program causes the computer to perform the operations required for a particular application.

13 We have seen that there are two kinds of problem programs: IBM-supplied and user-written. In any S/360 installation there will be a number of each. All problem programs should be stored in a location from which they can quickly be loaded for execution.

Figure 2 shows a common arrangement for storing problem programs. The storage device is a ....

●●●

magnetic disk unit.

14 By keeping all problem programs on an auxiliary storage device, any given program can quickly be located and loaded into main storage for execution. The auxiliary storage usually is a magnetic disk file, but magnetic tapes sometimes are used.

Look at Figure 2. One of the programs shown has the job of locating problem programs in the auxiliary storage device and loading them into main storage. You can deduce that this program is the ....

●●●

supervisor.

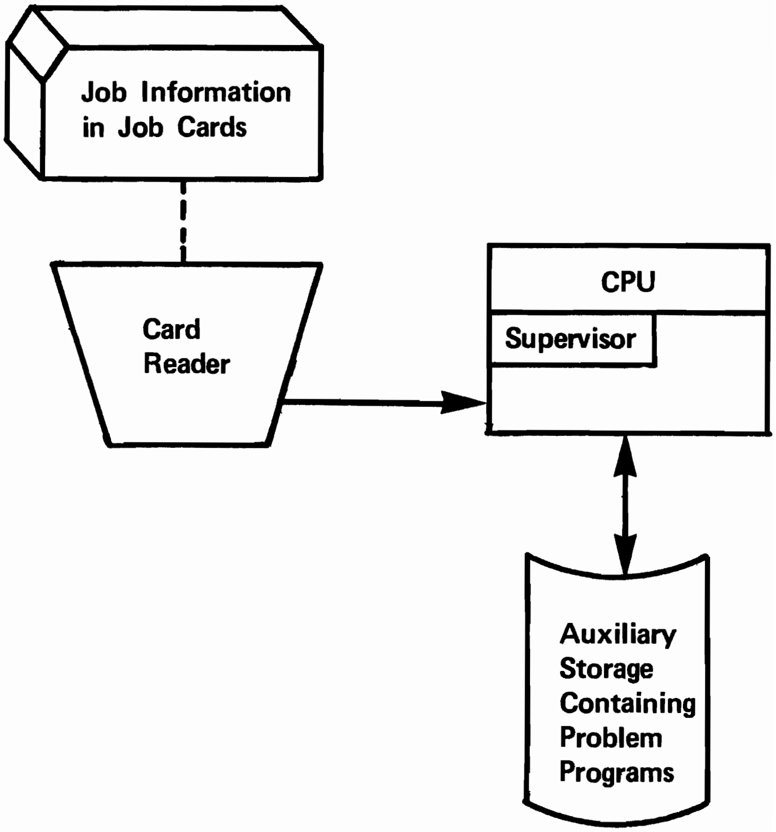

Job Information in Job Cards
Card Reader
CPU
Supervisor
Auxiliary Storage Containing Problem Programs

Figure 3.

15 The operator notifies the supervisor of the jobs he wants to do, and the sequence in which they are to be run. Figure 3 shows how this is done. Job information is recorded in .... which are read by a ....

●●●

punched cards
card reader.

16 The information contained in the punched cards (called job cards) tells the supervisor which problem programs to locate and load, and in what sequence they are to be scheduled. The supervisor locates and loads

a. IBM-supplied programs only.
b. user-written programs only.
c. both IBM-supplied and user-written programs.
d. neither IBM-supplied nor user-written programs.

●●●

c. Both IBM-supplied and user-written programs.

17 In effect, then, the supervisor schedules the execution of IBM-supplied and user-written problem programs. Where does it get the information it needs to do this?

●●●

From job cards prepared by the operator.

18 Name one function of the supervisor program.

●●●

It schedules the execution of IBM-supplied and user-written programs.

19 We noted previously that the supervisor performs various functions for each user program. (User programs, in this sense, means problem programs, including IBM-supplied.) As we have said, each problem program interacts with the supervisor. One reason for this interaction is the fact that the supervisor is designed to perform certain functions that are common to all problem programs.

Which of these is common to all problem programs?

a. Table lookup operations.
b. Input/output operations.
c. Editing operations.

●●●

Input/output operations.

20 You will recall that writing the instructions for input/output operations in a single program can consume up to forty percent of the programmer's time. If it were possible for all problem programs to use a common set of I/O routines, the time required to write individual problem programs would be reduced substantially.

A common set of I/O routines does exist for S/360. Each problem program has access to these routines as it is being executed. You can deduce that they are part of the .... program.

●●●

supervisor

21 A major function of the supervisor is to control and coordinate I/O functions for each problem program. To do this, it uses a set of routines that are called the Physical Input Output Control System (PIOCS).

To perform an input/output operation, a problem

program must call on the .... to use its .... routines.

●●●

supervisor
PIOCS

22 PIOCS stands for ....

●●●

Physical Input Output Control System.

23 At each point in a problem program at which an I/O operation is required, the problem program must branch to the supervisor, which then executes the appropriate PIOCS routine.

As you will see later, I/O operations can become quite complex, because of the number and kinds of I/O devices on the system, and the frequency and duration of I/O operations for a given problem. Therefore, a high degree of control and coordination is required. The program that supplies this control and coordination is the ....

●●●

supervisor.

24 Name two major functions of the supervisor.

●●●

It schedules execution of IBM–supplied and user-written programs.
It controls and coordinates I/O functions for all problem programs.

25 We now know that there is a supervisor and a problem program in main storage for each job. At any given time during the run, the CPU may be executing instructions from the supervisor, or from the problem program, depending on which is in "control" of the system at that moment.

When the CPU is executing problem program instructions it is said to be in the "problem" state. You can deduce that when the CPU is executing supervisor instructions it is said to be in the .... state.

●●●

supervisor

26 The supervisor and problem states are two of several "program states" in which the CPU may be at any time. "Program state" simply means the status of the CPU at a given time.

What is the status of the CPU when control program instructions are executed?

●●●

It is in the supervisor state.

27 What is the status of the CPU when instructions from a payroll program, or a compiler program, are being executed?

●●●

It is in the problem state.

28 The status of the CPU is different, in one or more ways, for each program state. For example, there are certain instructions called "privileged" instructions that can be executed only in the supervisor state. Among them are the S/360 I/O instructions. Why would these

S/360 PROGRAM STATES

STOPPED	or	OPERATING
RUNNING	or	WAITING
SUPERVISOR	or	PROBLEM
MASKED	or	INTERRUPTIBLE

Figure 4.

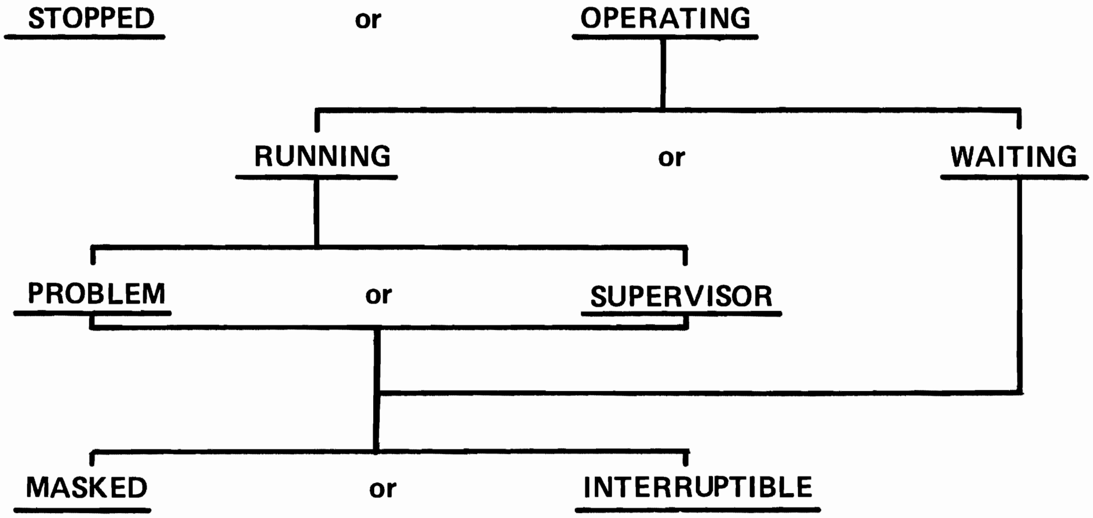


Figure 5.

instructions never be found in a problem program?

●●●

The CPU is in the problem state while executing problem programs. It could not execute these instructions.

29 The status of the CPU, then, differs among the possible program states. Name two program states and one difference between them.

●●●

Supervisor state, problem state. Certain instructions can be executed in the supervisor state, but not in the problem state.

30 Define "program state".

●●●

The status of the CPU at any given time.

31 Figure 4 lists the possible program states for S/360. It is important to note that the CPU can be in more than one state at the same time. For example, when the CPU is in either the supervisor or problem state, it also is in the running state.

Figure 5 shows the possible relationships among the S/360 program states. If the CPU is in the operating state it also will be in either the .... or .... state.

●●●

running
waiting

32 If the CPU is in the operating, running and problem states, what other state(s) could it possibly be in? (See Figures 4 and 5.)

●●●

Either the masked or interruptible state.

33 A CPU in the stopped state can be in what other state?

●●●

none

34 You will learn more about these program states in this topic.

The CPU is switched from one program state to another depending on conditions that occur during S/360 operation. A change in CPU status from one program state to another is called a status-switching operation.

For example, the CPU status changes from problem to supervisor state when the problem program requests an I/O operation. This change is a status-switching operation. After initiating the I/O operation, the supervisor causes the CPU to return to the problem program. This, too is a ....

●●●

status-switching operation.

35 Define a status-switching operation.

●●●

A status-switching operation is a change from one program state to another.

36 Each program state differs from other program states in three ways:

1. The way the CPU functions.
2. The way the CPU status is indicated.
3. The way the status is switched.

Let's consider the first of these differences, the

Figure 6.

PROGRAM STATE	CPU FUNCTIONING	STATUS INDICATION	HOW SWITCHED
Stopped	Incapable of any function.	"Manual" light on console	"Stop" key on console
Operating	Capable of executing instructions and being interrupted. Timer is updated.		
Running	______________	______________	______________
Waiting	______________	______________	______________
Supervisor	______________	______________	______________
Problem	______________	______________	______________
Masked	______________	______________	______________
Interruptible	______________	______________	______________

Figure 7.

PROGRAM STATE	CPU FUNCTIONING	STATUS INDICATION	HOW SWITCHED
Stopped	Incapable of any function.	"Manual" light oı, console	"Stop" key on console
Operating	Capable of executing instructions and being interrupted.	"System" or "Wait"light on console	"Start" key on console

Figure 8.

PROGRAM STATE	CPU FUNCTIONING	STATUS INDICATION	HOW SWITCHED
Stopped	Incapable of any function.	"Manual" light on console	"Stop" key on console
Operating	Capable of executing instructions and being interrupted.	"System" or "Wait" light on console	"Start" key on console
Running	Instruction fetching and execution proceed normally.	A zero bit in position 14 of the program status word (PSW)	I/O interrupt External interrupt
Waiting	No instruction processing. I-O and external interrupts accepted unless masked. Timer is updated.	A one bit in position 14 of the program status word (PSW)	Load PSW instruction Any interrupt

way the CPU functions. As you can imagine, in each program state the CPU can do some things it can't do in the others.

Figure 6 shows a partially completed chart of information about S/360 program states. What is a major difference between the stopped and operating states?

●●●

In the stopped state the CPU is incapable of any function. In the operating state the CPU is capable of executing instructions and being interrupted.

37 As we continue we will complete the chart in Figure 6, and explain the unfamiliar terms.

Refer to Figure 6. How is the CPU switched to the stopped state?

●●●

The stop key on the console must be pressed.

38 How can you tell that the CPU is in the stopped state?

●●●

A console light labeled MANUAL will be on.

39 In the stopped state the CPU (is/is not) .... executing instructions.

●●●

is not

40 Look at Figure 7. To switch to the operating state the operator must ....

●●●

press the start key on the console.

41 Look back at Figure 5. In the operating state the CPU will be in the .... or .... state.

●●●

running
waiting

42 Look at Figure 8. Instructions are being fetched (from storage) and executed in the (waiting/running) .... state.

●●●

running

43 Now you can see why we say that in the the operating state the CPU is capable of executing instructions. What determines whether or not it will be doing so?

●●●

Whether it is in the running or waiting state.

44 Normally the system will be in the running state, executing instructions from either the supervisor or problem program. Occasionally, however, something happens that makes it desirable for the CPU to stop executing instructions but otherwise remain in readiness. This is the waiting state (often called the wait state). How can the CPU be switched to the wait state? (See Figure 8.)

●●●

A Load PSW instruction or any interrupt can switch CPU status from running to wait.

45 We have used some terms that are unfamiliar to you, so we will explain them now.

We have said that in some program states the CPU is "interruptible", and we have made references to "interrupts". It is important that

you understand the concept and functioning of interrupts on S/360.

From time to time during operation, an exceptional condition will occur somewhere in the computer system. Obviously, these exceptional conditions must be investigated to determine what effect they may have on the system. To call the attention of the CPU to the exceptional condition, a signal called an "interrupt" is generated. The interrupt is actually a request for a change in the status of the CPU.

Here is an example. Suppose that an input/output operation is proceeding simultaneously with problem program processing. At the end of the I/O operation a signal called an interrupt will be generated, requesting a change in CPU status. Thus, the end of an I/O operation is considered an ....

●●●

exceptional condition.

46 The signal generated when an exceptional condition (such as the end of an I/O operation) occurs is called an ....

●●●

interrupt.

47 The purpose of an interrupt is to ....

●●●

request a change in the CPU status.

48 The reason for a status-switching operation at the end of an I/O operation is this: there may be additional I/O operations to be performed immediately after the one is completed. I/O operation can only be started or serviced when the CPU is in the...........state.

●●●

supervisor

49 Prior to the interrupt, while the S/360 is executing problem program instructions, the CPU will be in the .... state.

●●●

problem

50 The interrupt that occurs when an I/O operation ends requests a change in CPU status from .... to ....

●●●

problem
supervisor

51 As an example of this, let's look at a magnetic tape read operation. It is possible for data to be read incorrectly. If it does (and the computer will know), the usual procedure is for the CPU to issue instructions causing the tape to backspace and re-read the data. Often a subsequent attempt will result in correct reading.

To enable the CPU to check on the results of an I/O operation, and issue instructions for corrective action, if necessary, an .... must occur, requesting .... from .... to ....

●●●

interrupt
a change in CPU status
problem
supervisor

52 Another reason for the I/O interrupt is that there may be other I/O operations waiting to be performed. This occurs when the channel is too busy to handle them all. Some are kept pending (waiting for the channel to get around to them). Therefore, as soon as one I/O operation is completed, the supervisor must not only check to see if the just completed operation occurred correctly, but also must check to see if there are I/O operations pending. If there are, the supervisor must initiate the next one before returning control to

CLASSES OF INTERRUPTS AND SOME EXAMPLES OF EACH

I/O

Completion of transfer of data from or to an I/O device.

EXTERNAL

Operator presses interrupt key on console.
Internal timer signal.

PROGRAM - CHECK

Arithmetic overflow.
Improper specification of data.
Improper use of instructions.
Storage protection violation.

MACHINE - CHECK

Machine malfunction of other than I/O units.

SUPERVISOR CALL

Execution of a special instruction that switches the CPU to the supervisor state.

Figure 9.

the problem program.

●●●

53 We said that an I/O interrupt will switch the CPU from the problem to the supervisor state. According to Figure 8, what other change in CPU status can an I/O interrupt cause?

●●●

Waiting to run.

54 In addition to the I/O interrupt, another kind of interrupt can change the CPU from waiting to running. It is an ....

●●●

external interrupt.

55 An external interrupt occurs when the operator presses the interrupt key on the console.

Name two kinds of interrupts.

●●●

I/O
external

56 So we see that an interrupt is a request for a change in CPU status, resulting from an exceptional condition. The end of an I/O operation, and the pressing of the interrupt key, are ....

●●●

exceptional conditions.

57 Define an "interrupt".

●●●

An interrupt is a request for a change in CPU status, resulting from an exceptional condition.

58 There are five classes of interrupts on S/360. They are listed in Figure 9. We already have named two: the I/O interrupt and the external interrupt. The others are:

●●●

program-check
machine-check
supervisor-call

59 There are various exceptional conditions that result in interrupts. For example, exceptional arithmetic results cause an interrupt. By exceptional, we mean results that are too large for a given register or storage location or that otherwise depart significantly from expected results.

An arithmetic overflow results in a .... interrupt. (See Figure 9.)

●●●

program-check

60 Another type of interrupt results from a machine malfunction. For example, it is possible for the circuits to gain or lose a bit while transferring data from one location to another. Such a condition would immediately be detected by the system itself, because of the built-in checking circuits. The result would be a parity check. A parity check results in a .... interrupt.

●●●

machine-check

61 Match the following:

a. machine-check
b. external
c. I/O

d. program-check

1. the end of an I/O operation
2. arithmetic overflow
3. parity check
4. operator presses interrupt key

●●●

a. 3
b. 4
c. 1
d. 2

62 Look at Figure 9. Another type of interrupt is the ....

●●●

supervisor-call

63 A supervisor-call interrupt occurs when a specific instruction, the Supervisor Call instruction, is executed by a problem program. The purpose of this instruction is simply to cause a status-switching operation to occur, from problem to supervisor state.

There can be numerous conditions in problem programs at which a change to the supervisor state is required. Reading and writing data, loading new programs to be executed, setting the interval timer, cancelling programs being executed, and other situations call for the supervisor to assume control of the computer.

An example of this is the end-of-job condition, which occurs when a problem program is finished. Control of the computer must be turned over to the supervisor, so it can locate and load the next scheduled problem program. This could not happen unless the Supervisor Call instruction was executed by the just-finished problem program.

An end-of-job condition will result in a .... interrupt.

●●●

Supervisor-call

64 The instruction that results in a supervisor-call interrupt is the .... instruction. It would occur in a problem program at ....

●●●

Supervisor Call
end-of-job, reading, writing, loading new programs, setting the timer, etc.

65 Name the five classes of interrupts and give an example of each.

●●●

I/O: completion of transfer of data from or to an input/output device
external: operator presses interrupt key
program-check: arithmetic overflow
machine-check: parity check
supervisor-call: end of job

66 The important thing about interrupts is that directly or indirectly they cause the CPU to enter the supervisor state. This is because a major function of the supervisor is to handle all interrupts; that is, to determine the nature of the interrupt and take appropriate action.

Previously we discussed two major functions of the supervisor. They are (1) to schedule the execution of IBM-supplied and user-written programs and (2) to control and coordinate I/O functions for all problem programs.

Name a third major function of the supervisor.

●●●

It handles all interrupts.

67 Let's review what we have discussed so far about program execution on S/360. We know that there are eight possible program states for the CPU. The CPU will be in one or more program states depending on what it is doing or is capable of doing at any given time. Program states differ from one another in the way the CPU functions, the way the CPU status is indicated, and the way the CPU is switched to each program state. One way of switching program states is by interrupts, which are caused by exceptional conditions that occur during computer operation.

PROGRAM STATE	CPU FUNCTIONING	STATUS INDICATION	HOW SWITCHED
Stopped	Incapable of any function.	"Manual" light on console	"Stop" key on console
Operating	Capable of executing instructions and being interrupted.	"System" or "Wait" light console	"Start" key on console
Running	Instruction fetching and execution proceed normally.	A zero bit in position 14 of the program status word (PSW) "System" light on console	I-O interrupt External interrupt
Waiting	No instruction processing. I/O and external interrupts accepted unless masked. Timer is updated.	A one bit in position 14 of the program status word (PSW) "Wait" light on console	Load PSW instruction Any interrupt
Supervisor	All instructions are valid.	A zero bit in position 15 of the PSW	Any interrupt
Problem	All I/O instructions and a group of control instructions are invalid.	A one bit in position 15 of the PSW	Load PSW instruction
Masked	____________________	____________________	____________________
Interruptible	____________________	____________________	____________________

Figure 10.

Look at Figure 10. How is the status of the CPU indicated when it is in the running state? The waiting state?

●●●

A zero bit in position 14 of the program status word; the "System" console light.
A one bit in position 14 of the program status word; the "Wait" console light.

68 A program status word (abbreviated PSW) is a doubleword whose contents reflect the logical status of the computer with respect to the program being executed.

By logical status we mean the logical conditions that exist on the computer at any given time. Such conditions as the program state of the CPU, the address of the next instruction to be executed, and the relationship (high, low, equal) of the last data items that were compared by the program, are all logical conditions.

Where are such logical conditions recorded?

●●●

In a PSW (program status word)

69 As we shall see, there are several PSW's involved in S/360 operation. But at any one time only one PSW is in a position to influence the computer. It is called the "current" PSW. The other PSW's are kept in specific locations in main storage.

Now, to return to CPU program states and status switching, we said that the running and waiting states are indicated by a zero bit and a one bit, respectively, in position 14 of the PSW. You can deduce that we are talking about the

a. current PSW.
b. A PSW in storage.

●●●

current PSW

70 Look at Figure 10. The CPU can be switched to the wait state by an instruction involving a PSW. It is the ....

●●●

Load PSW instruction.

71 The Load PSW instruction selects a PSW from storage and moves it into another location, where it becomes the current PSW. If bit 14 of the new PSW is a one, the CPU will enter the .... state. (See Figure 8.)

●●●

wait

72 You will remember that there are certain instructions called privileged instructions that can be executed only when the CPU is in the supervisor state. The Load PSW instruction is one of them. Therefore, the CPU can be switched to the wait state by the Load PSW instruction only when the CPU is in the supervisor state.

So we see that program states can be switched by I/O interrupts (signals generated at the end of input/output operations), external interrupts (from the interrupt key on the console, and other sources), and the Load PSW instruction (which substitutes a new PSW for the current one).

●●●

73 You will learn more about PSW's as we go along. But first, let's complete the table of program states with attendant information.

We saw in Figure 5 that when the CPU is in operating and running states it also will be in either the supervisor or problem state. The major difference between the problem and supervisor states is .... (We discussed this earlier. If you have forgotten, check Figure 10.)

●●●

PROGRAM STATE	CPU FUNCTIONING	STATUS INDICATION	HOW SWITCHED
Stopped	Incapable of any function.	"Manual" light on console	"Stop" key on console
Operating	Capable of executing instructions and being interrupted.	"System" or "Wait" on console	"Start" key on console
Running	Instruction fetching and execution proceed normally.	A zero bit in position 14 of the program status word (PSW) "System" light on console	I-O interrupt. External interrupt
Waiting	No instruction processing. I-O and external interrupts accepted unless masked. Timer is updated.	A one bit in position 14 of the program status word (PSW) "Wait" light on console	Load PSW instruction Any interrupt
Supervisor	All instructions are valid.	A zero bit in position 15 of the PSW	Any interrupt
Problem	All I-O instructions and a group of control instructions are invalid.	A one bit in position 15 of the PSW	Load PSW instruction
Masked	I-O, External, and Machine-check interrupts (individually masked) remain pending. Program interrupts are ignored.	Zero bits in the system mask, program mask, and machine-check mask fields of the PSW	Set Program Mask instruction Set System Mask instruction Load PSW instruction Any interrupt
Interruptible	Interrupts of all unmasked classes accepted.	One bits in the system mask, program mask, and machine-check mask fields of the PSW	Same as "Masked" above

Figure 11.

I/O and certain control instructions can be executed in the supervisor state but not in the problem state.

74 Look at Figure 11. How can the CPU be switched to the supervisor state?

●●●

Any interrupt will directly or indirectly cause the CPU to enter the supervisor state.

75 As an example, let's consider the Supervisor Call interrupt. The Supervisor Call instruction is located in the problem program at those points at which it wants the supervisor to take over. For example, each time the problem program wants an I/O operation performed it will execute a Supervisor Call instruction, which causes a supervisor call interrupt.

As a result of this interrupt, a new PSW automatically will replace the current PSW. The CPU will be switched to the supervisor state by.......... in the new PSW. (See Figure 11.)

●●●

a zero-bit in position 15

76 How can the CPU be switched to the problem state? (See Figure 11.)

●●●

by the Load PSW instruction.

77 What condition must exist in the new PSW to cause a switch to the problem state?

●●●

a one-bit in position 15

78 The CPU must be in the supervisor state prior to being switched to the problem state. Why?

●●●

Because the Load PSW instruction can be executed only when the CPU is in the supervisor state.

79 Look at Figure 11. The last two program states are .... and ....

●●●

masked
interruptible

80 You know from our previous discussion that there are five classes of interrupts. They are:

a. input/output
b. program-check
c. machine-check
d. supervisor-call
e. external

In the preceding frame you learned that there is a masked program state and an interruptible program state.

You can deduce that when the CPU is interruptible for a class of interrupts, these interrupts are (processed immediately by the supervisor/kept pending) ....

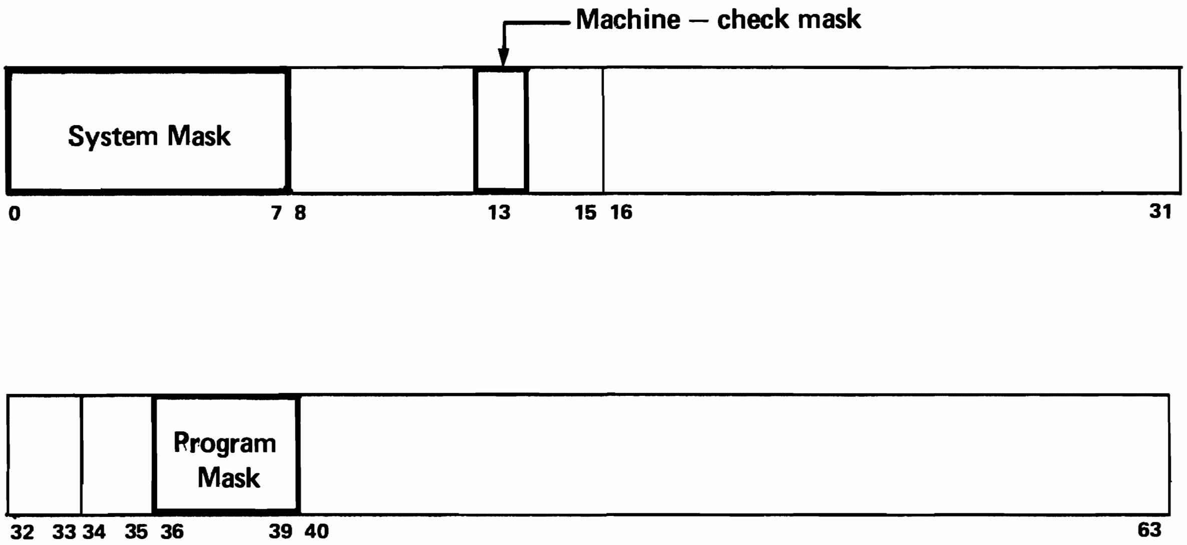
●●●

processed immediately by the supervisor

81 Conversely, when the CPU is .... against a class of interrupts, these interrupts are ....

●●●

masked
kept pending



System Mask — for I/O interrupts.

Program Mask — for Program-Check interrupts.

Machine-Check Mask — for Machine-check interrupts.

machine malfunctions

THE PROGRAM STATUS WORD

Figure 12.

82 Note that the interrupt is kept pending; it is not ignored. Although it is possible to mask an interrupt and ignore it, this would be most unusual. IBM-supplied programs will process all masked interrupts in time.

●●●

83 There are times, however, when it is desirable to mask certain interrupts, so that they do not require immediate attention. For example, as the supervisor is processing an I/O interrupt, we do not want it interrupted by another I/O interrupt. (This is possible - with multiple input/output channels and multiple I/O operations on one channel, the S/360 can have a number of I/O operations proceeding simultaneously.) To prevent this, when the supervisor is handling an I/O interrupt, it masks all other possible I/O interrupts.

When will an interrupt be kept pending?

●●●

When it is masked.

84 Figure 11 indicates how masking is accomplished. In general, how is this done?

●●●

By the presence of zero bits in the mask fields in the current PSW.

85 Figure 12 shows the PSW mask fields, and the interruptions that each field masks.

We know that signals generated at the end of I/O operations cause I/O interrupts. These are masked by zero bits in the appropriate positions of the System Mask field.

Program - check interrupts, caused by exceptional arithmetic results, and other conditions, are masked by..........

●●●

zero bits in the program mask field.

86 How is the CPU masked against machine malfunctions?

●●●

A zero bit in the machine-check mask field.

87 The CPU is masked against any class of interrupt if there is a zero bit in the appropriate position of the mask field in the PSW. The question now arises, how do the bits get changed from one to zero, or from zero to one if you are unmasking an interrupt?

Look at Figure 11. You can set zero bits in the program mask field by ....

●●●

executing the Set Program Mask instruction.

88 The Set Program Mask instruction can be executed by the problem program. It will set the bits in the program mask to ones or zeros as desired.

How are the bits set in the system mask field? (See Figure 11.)

●●●

By executing the Set System Mask instruction.

89 The Set System Mask instruction is privileged (executable in the supervisor state only). Otherwise, it functions exactly like the Set Program Mask instruction.

Look at Figure 11. Another way to mask interrupts involves another privileged instruction. It is the .... instruction.

●●●

Load PSW

90 You will remember that this instruction moves a new PSW from storage to become the current PSW. How can a new PSW affect the masked status of the CPU?

●●●

If the new PSW has zeros in any mask field positions, the corresponding interrupts will be masked.

91 Figure 11 indicates that any interrupt can result in masking. Why is this?

●●●

There is an automatic exchange of PSW's when an interrupt occurs. The mask fields in the new PSW can affect masking.

92 We have seen how the CPU can be switched to the masked state for any given interrupt. According to Figure 11, how can it be switched to the interruptible state?

●●●

By the same instructions used to mask it. The instructions would result in ones instead of zeros being placed in the mask field.

93 We have discussed the different program states of S/360, and the things about each that make it unique. In what three major ways can a program state differ from another state?

●●●

The way the CPU functions.
The way the CPU status is indicated.
The way the CPU status is switched.

94 Let's turn now to the PSW, the doubleword whose function is to reflect the logical status of the complete system. You already have learned quite a bit about the PSW. You will remember that although there are several PSW's, at any one time there is only one in a position to influence the complete system. It is the one actually in use during the execution of instructions. This PSW is called the ....

●●●

current PSW

95 You know that when a Load PSW instruction is executed, the current PSW is replaced by a "new" PSW from storage. Also, that any interrupt automatically replaces the current PSW with a new one.

Here's the reason for this PSW movement. Whenever the CPU status changes because of an interrupt, we must preserve the logical status of the system prior to the interrupt, so it can be restored after the interrupt has been handled.

To do this, would you

a. use any available PSW after the interrupt is handled?
b. store the current PSW at the time of interrupt and restore it later?
c. use the same PSW that was used during interrupt handling?

●●●

b. store the current PSW at the time of interrupt and restore it later.

96 When the current PSW is replaced by a new PSW, it is not lost. Instead, it is stored in a particular location, from which it can be obtained when desired. In this position, it is called the "old" PSW.

Match the following:

a. old PSW
b. new PSW
c. current PSW

STORAGE ADDRESSES OF PROGRAM STATUS WORDS

ADDRESS	PROGRAM STATUS WORD
24	External OLD PSW
32	Supervisor Call OLD PSW
40	Program OLD PSW
48	Machine-check OLD PSW
56	Input/Output OLD PSW
88	External NEW PSW
96	Supervisor Call NEW PSW
104	Program NEW PSW
112	Machine-check NEW PSW
120	Input/Output NEW PSW

Figure 13.

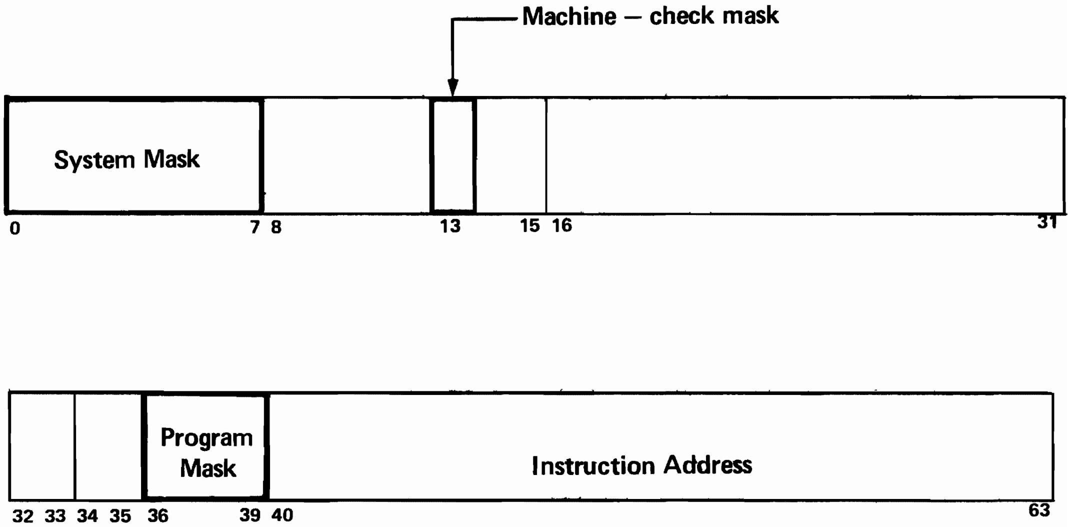


Figure 14.

1. the PSW that replaces and then becomes the "current" PSW as a result of an interrupt.
2. the PSW in use during the execution of an instruction (the one that influences system operation).
3. the PSW that was replaced, and that is stored awaiting recall.

●●●

a. 3
b. 1
c. 2

97 Each old PSW is stored in a specific storage location, dictated by the class of interrupt. Figure 13 shows these addresses. There are (how many?) .... old PSW locations.

●●●

five

98 Look at Figure 13. There are (how many?) .... new PSW's.

●●●

five

99 We know that there are five classes of interrupts. You can see that a PSW exchange occurs when an interrupt from any of the five classes occurs.

What happens, so far as the PSW's are concerned, when an interrupt occurs?

●●●

The current PSW becomes the old PSW and is stored in a storage location determined by the class of interrupt. The new PSW for that class of interrupt becomes the current PSW.

100 Define the "current" PSW.

●●●

The PSW in use during the execution of an instruction.

101 Define the "old" PSW.

●●●

The PSW that is replaced by the new PSW as a result of an interrupt, becomes the "old" PSW. The old PSW is stored in a specific location determined by the class of interrupt.

102 Define the "new" PSW.

●●●

The PSW which replaces and then becomes the "current" PSW as a result of an interrupt. The new PSW is fetched from a specific storage location determined by the class of interrupt.

103 State the function of the PSW.

●●●

The PSW reflects the logical status of the system.

104 We know that in order for the system to operate, it must have a current PSW in position to influence the computer system. For that reason, a new PSW is fetched to replace the current PSW when an interrupt occurs.

Figure 14 shows one reason for fetching a new PSW. Positions 40-63 of the PSW contain the ....

●●●

instruction address.

105 The instruction address in the new PSW tells the CPU where it will find its next

instruction. The next instruction, of course, will be somewhere in the supervisor. Can you explain why this is true?

●●●

The supervisor handles all interrupts. Immediately after an interrupt, the supervisor will take control of the computer.

106 The routine to be entered when an interrupt occurs will depend on the class of interrupt. Since there is a new PSW for each class, each new PSW indicates the address of the first instruction in the supervisor routine that handles that class.

For example, assume an interrupt occurs because you have used an instruction incorrectly, or specified some data incorrectly. The new PSW will be fetched from storage location 104. It will contain the address of the first instruction in the supervisor routine for handling which of these?

a. machine-check interrupts
b. I/O interrupts
c. program-check interrupts

●●●

program-check interrupts

107 The address of the first instruction of an interrupt handling routine is in the .... field of the ....

●●●

instruction address
PSW.

108 List the events in an interrupt situation and describe each, from the occurrence of the interrupt to the resumption of the problem program.

●●●

a. Interrupt occurs.
b. Current PSW is stored in old PSW location.
c. New PSW is fetched and made the current PSW.
d. Supervisor program is entered at location specified in current PSW instruction address.
e. Supervisor program is executed.
f. Old PSW is loaded in current PSW position.
g. Problem program is entered at location specified in current PSW.

109 In addition to indicating the address of the first instruction to be executed after an interrupt, the instruction address field serves another purpose. As each instruction is executed, the instruction address field is changed to indicate the address of the next instruction to be executed.

S/360 instructions are placed in storage in the order in which they will be executed, from lower to higher numbered addresses. Suppose the first instruction to be executed begins at address 5600. The address in the PSW will be ....

●●●

5600.

110 Assume the first instruction is 4 bytes (two halfwords) long. After the instruction is executed, the instruction address field will contain ....

●●●

5604.

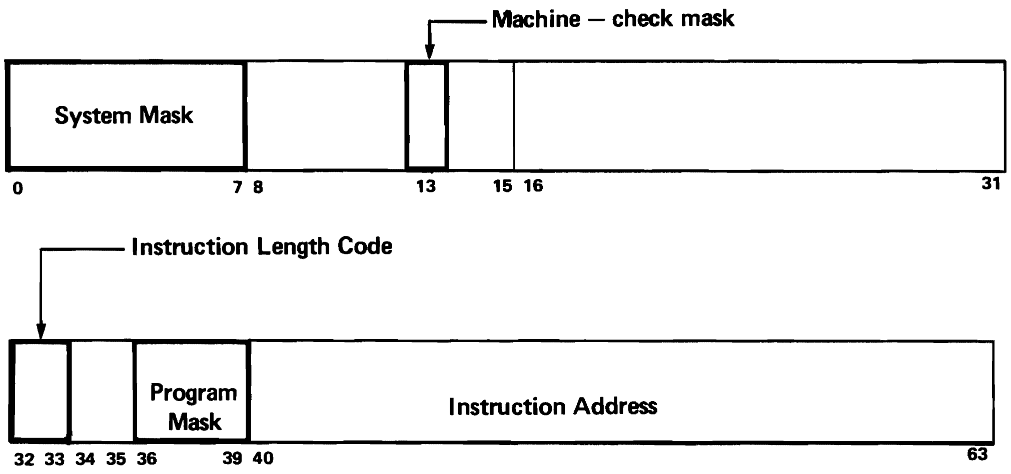
111 5604 is the address of which of these?

a. the old PSW.
b. the next sequential instruction.
c. the new PSW.

●●●

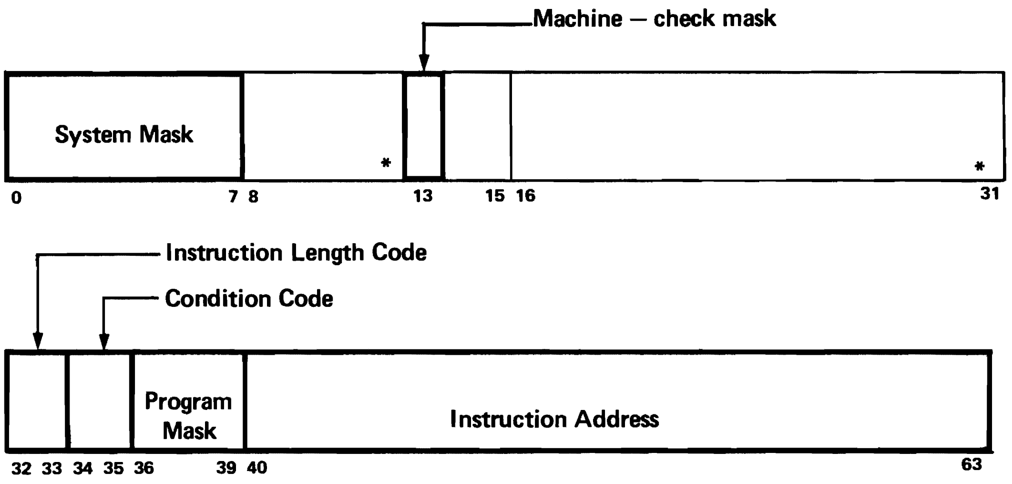
The next sequential instruction.

112 Instructions normally are executed in the sequence in which they occur in storage.



THE PROGRAM STATUS WORD

Figure 15.



THE PROGRAM STATUS WORD

Figure 16.

*These fields not discussed in this text.

However, S/360 instructions vary in length. The question is, how does the CPU know how much to increment the address of the current instruction to get the address of the next one?

The answer lies in the operation code of the instruction. Part of the operation code of each instruction indicates the length in halfwords of that instruction. This information is used by the CPU to increment the address of the current instruction. The resulting address is the address of the ....

●●●

next sequential instruction.

113 Figure 15 shows the PSW. One of the fields holds the length, in halfwords, of the last interpreted instruction, that is, the instruction currently being executed. This field is the .... field.

●●●

instruction length code.

114 The instruction length code in the PSW is derived from the operation code of the instruction just interpreted. It indicates the amount by which the instruction address must be incremented to get the address of the next instruction.

●●●

115 In what order are instructions stored?

●●●

Sequentially, from lower to higher main storage addresses.

116 Normally, instructions are executed in the order in which they occur in storage. However, there are places in every program at which the sequence may change; that is, the CPU will stop executing instructions in one sequence and switch to another part of the program, where it will begin executing another sequence of instructions. This change in sequence is called a branch.

Branching, you will remember, can be conditional or unconditional. If a branch occurs as a result of a comparison between two data items, it is a(n) .... branch.

●●●

conditional

117 Conditional branches occur frequently in the average program. They may be based on the results of arithmetic operations (whether the result is greater than, equal to, or less than zero), the coding of incoming data, and many other factors.

Branching also may be unconditional. Such a branch would occur (depending on/regardless of) .... data relationships and record codes.

●●●

regardless of

118 Define a branching operation.

●●●

A branching operation is a change in the order of execution of instructions within a program.

119 Conditional branches, as we pointed out, can be based on conditions that exist after data has been operated on. So it is necessary for the computer to remember such conditions. For example, if two data items are compared, and the first is found to be greater than the second, this information must be stored somewhere until the next instruction (the conditional branch instruction) can be fetched and interpreted.

Figure 16 shows where this information is kept. It is in the PSW, in the .... field.

●●●

condition code.

120 The condition code field reflects the conditions that exist after an instruction has been executed. If the instruction was a compare instruction, the relationship between the compared data items would be indicated by the condition code.

Nearly every S/360 instruction results in a particular condition code setting – it is not limited to compare instructions. Here are some of the conditions that can be indicated by the condition code:

a. The relationship (high, low, equal) of one quantity to another.

b. Whether a quantity is other than zero.

c. Whether a quantity is greater than, equal to, or less than zero.

d. Whether or not a start I/O operation was successful.

Describe the sequence of events that result in a conditional branch by the computer.

●●●

An instruction is executed.
The conditions that result from this instruction are recorded in the condition code field of the PSW.
A conditional branch instruction is fetched and interpreted.
The instruction sequence will be changed to a new sequence elsewhere in the program if the condition specified by the branch instruction is reflected by the condition code.

121 We have discussed several major fields in the PSW. Name and describe these fields.

Don't look at your illustrations.

●●●

The instruction address – the storage address of the next sequential instruction (NSI).
The instruction length code – the length in halfwords of the current instruction.
The program mask – the bits that permit or inhibit interrupts due to exceptional arithmetic results.
The condition code – the conditions that exist after a particular operation has been performed by the computer.

122 In discussing instruction sequencing, we said that the sequence of execution is changed by branching. We have discussed another way in which S/360 sequence of execution is changed. Can you recall what it is?

●●●

Status switching between problem and supervisor states, which is accompanied by an exchange of PSW's. The instruction address in the new PSW starts the CPU executing instructions at a new address.

123 In what order are instructions executed by S/360, and when is this order changed?

●●●

Sequentially, unless changed by branching, or interrupts.

124 Earlier in this topic we discussed some major functions of the supervisor program. They are (1) to schedule execution of IBM-supplied and user-written programs via a stream of information on Job Cards and a library of programs on an auxiliary storage device, (2) to control and coordinate I/O functions for all problem programs, and (3) to handle interrupts.

A fourth major function of the supervisor is to provide timer services.

The timer consists of a word of storage at address 80. It can be set to any desired value, and is automatically counted down (decremented) by S/360 circuitry. When the timer reaches zero, an external interrupt occurs.

The word at location 80, which records elapsed time intervals, is called the .... It is serviced by the .... program.

●●●

timer
supervisor

125 The problem program can ask the supervisor to set the timer to a predetermined figure. When the timer reaches zero, it generates a signal that interrupts the CPU, and the supervisor takes over. The supervisor will notify the problem program that the timer has reached zero, and the problem program can take appropriate action.

For example, if a particular user is to be allocated a certain amount of computer time, the timer can be set to that value. The problem program will be notified that the allotted time has been used when ....

●●●

the timer reaches zero.

126. The supervisor can interrogate the timer at any time, as requested by the problem program, without waiting for an interrupt from the timer. This permits "time stamping" each job; that is, printing out the beginning and ending for the job.

The program which initializes the timer, interrogates it, and communicates the information to the problem program is...........

●●●

the supervisor.

127 What are the major functions of the supervisor?

●●●

Scheduling execution of IBM-supplied and user-written programs.
Controlling and coordinating I/O functions.
Handling interrupts.
Providing timer services.

You have completed this section. At this point you should fill in your notes and take the self evaluation quiz.

Self Evaluation Quiz
Program Execution

QUESTIONS

1. Identify the S/360 control program.

2. Name the major functions of the S/360 control program.

3. Define "program state".

4. Name the program states of S/360.

5. Define "interrupt".

6. Name the five classes of S/360 interrupts.

7. Define PSW.

8. Define "current" PSW.

9. Describe what happens to the PSWs when an interrupt occurs.

10. Name some of the major fields in the PSW.

Self-Evaluation Quiz
Program Execution

ANSWERS — Frame References

1. The S/360 control program is called the supervisor. (2)

2. The supervisor schedules the execution of all IBM-supplied and user-written programs, controls and coordinates all I/O functions, handles interrupts, and provides timer services. (129)

3. Program state is the status of the CPU at any given time. (30)

4. Stopped or operating, running or waiting, problem or supervisor, masked or interruptible. (Figure 4)

5. An interrupt is a request for a change in the status of the CPU. (45)

6. I/O, program-check, machine-check, external, supervisor call. (65)

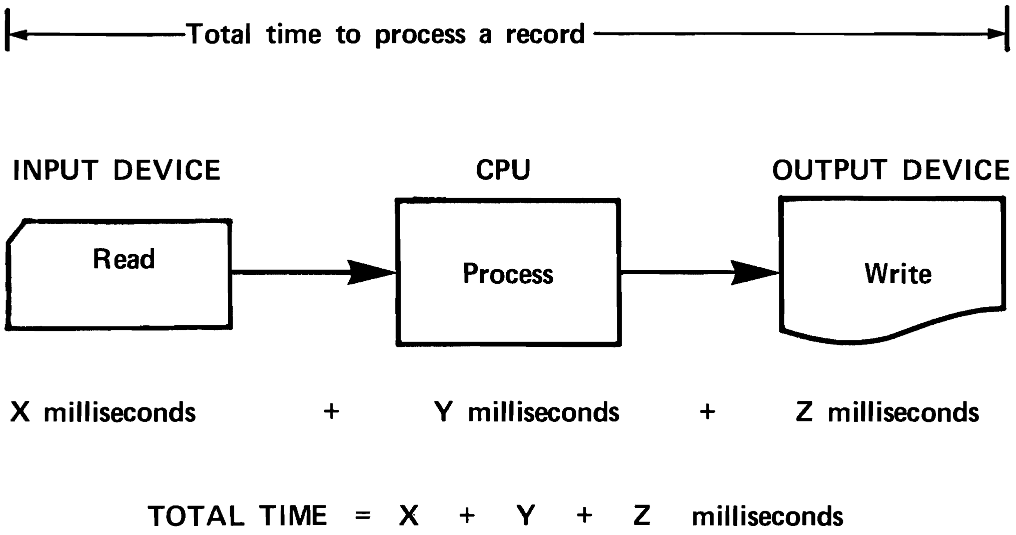
7. PSW means program status word. A PSW is a doubleword whose contents reflect the logical status of the S/360 with respect to the program being executed. (68)

8. The current PSW is the one which is in a position to influence the computer. (100)

9. When an interrupt occurs, the current PSW is stored in a storage location determined by the class of interrupt. A new PSW, also determined by the class of interrupt, is fetched from storage and becomes the current PSW. (99)

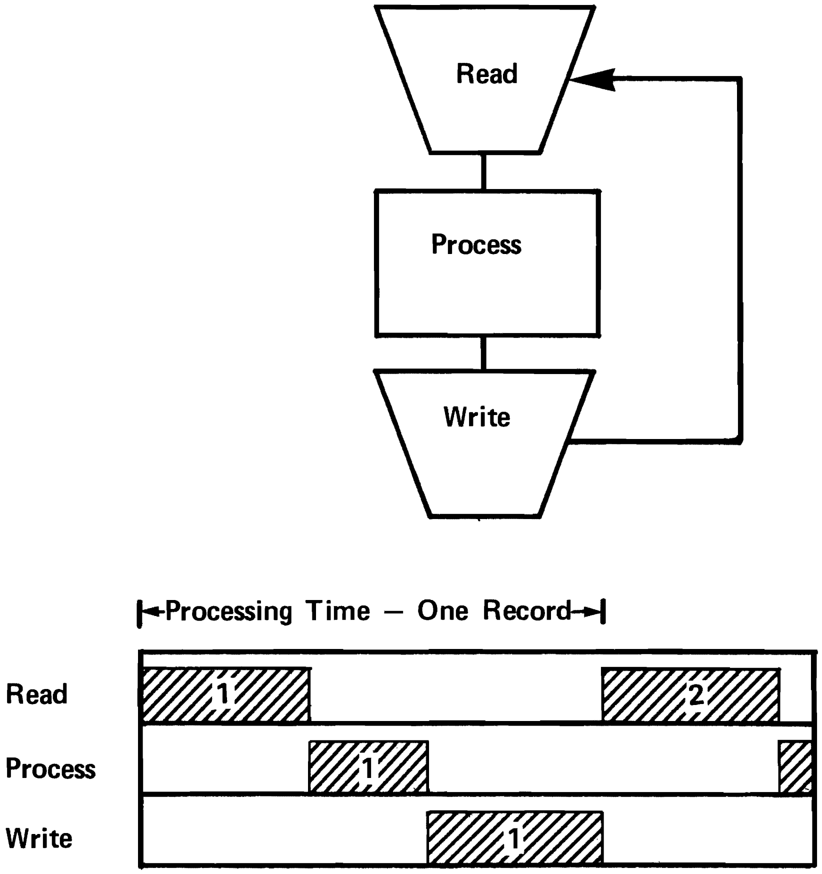
10. Program mask field, system mask field, machine check mask field, condition code field, instruction length code field, instruction address field. (121)

Input/Output Channels



RECORD PROCESS TIME

Figure 1.



NON-OVERLAPPED PROCESSING

Figure 2.

S/360 INPUT / OUTPUT CHANNELS

1 This topic is about S/360 channels. If you are familiar with the concept of overlapped processing, using two input/output storage areas for each data file, you may skip to frame number 37, where we begin discussing the operation of S/360 channels in particular.

Look at Figure 1. The diagram shows that the time taken to process a record is the sum of the individual times required to ...., ..., and .... it.

●●●

read
process
write

2 In this section, we will use the term "process" to mean the total time in which the CPU is involved in reading in, computing, and writing out each record.

If, at any one time the computer is involved in only <u>one</u> of these activities, the operation is <u>not</u> overlapped.

Look at Figure 2. The operation diagrammed at the bottom (is/is not) .... overlapped.

●●●

is not

3 The operation is not overlapped because the computer is involved in (how many?) .... activities at one time.

●●●

one

4 Non-overlapped processing is inefficient. It should be avoided if the system permits. To get overlapped processing, the computer must be involved in more than one activity at one time. If reading record 2 was concurrent with writing record 1, reading and writing would be said to be ....

●●●

overlapped.

5 We can reduce the time taken to process a record by .... two or more activities.

●●●

overlapping

6 Which activities are overlapped according to the following diagram?

```
R3    R4    R5

   P3    P4    P5

   W2    W3    W4
```

●●●

processing and writing.

7 Completely overlapped processing requires that reading, computing, and writing take place concurrently.

Complete the following chart showing the timing of read, process, and write operations in overlapped processing.

```
R    1    2    3    4

P

W
```

●●●

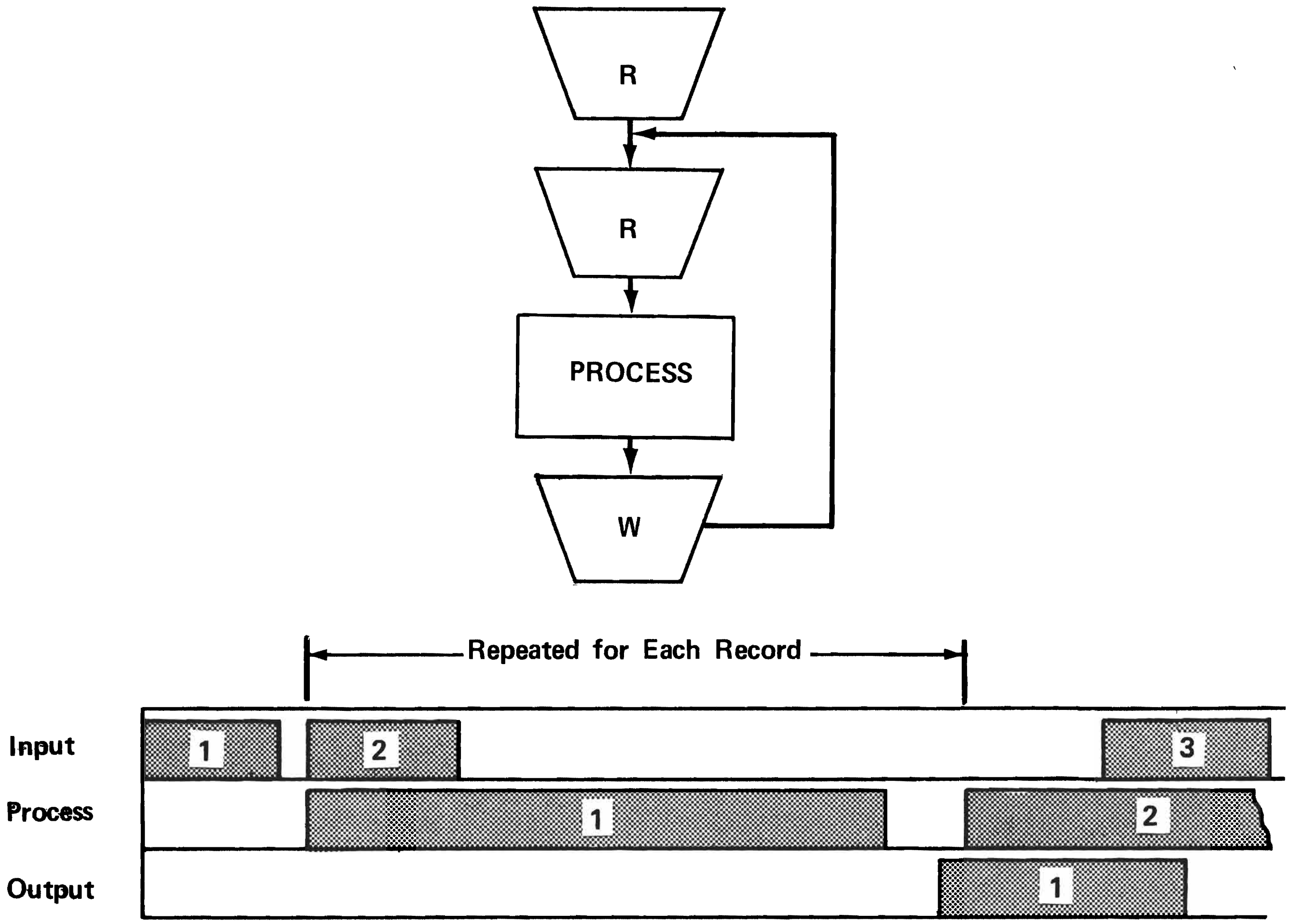


Figure 3.

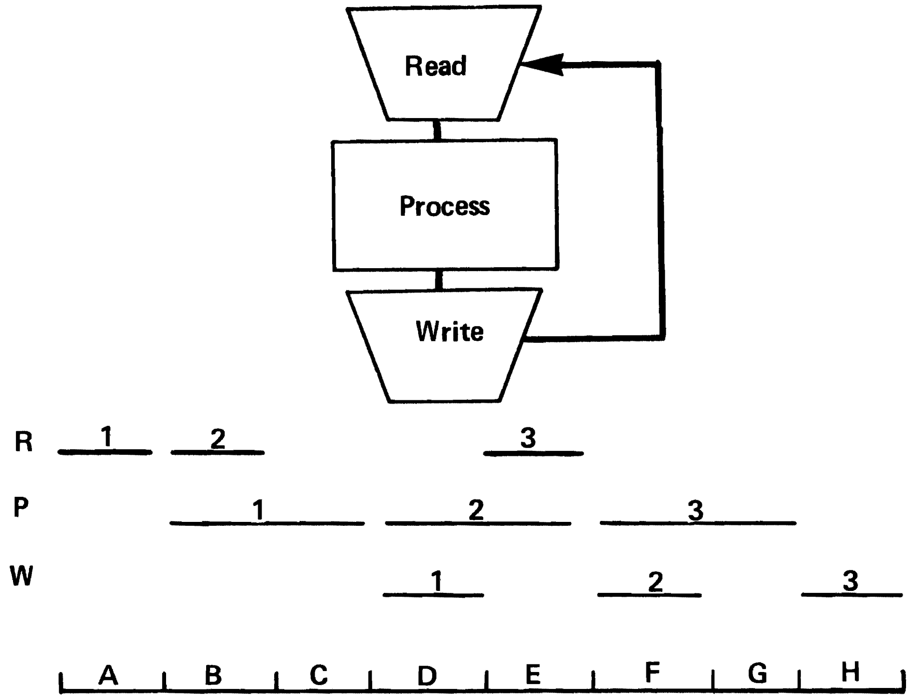


Figure 4.

R	1	2	3	4		
P		1	2	3	4	
W			1	2	3	4

8 Overlapped processing is possible only if the computer system has channels. Channels provide the means by which I/O operations may occur concurrently with computing.

It's what the program does with the channels that's important. Let's see how the program uses channels to get overlapped processing.

●●●

9 Look at Figure 3. In this diagram, record 2 is being read while data from record 1 is being processed. If record 2 is read into the same storage area in which record 1 is being processed, what will happen? (in your own words)

●●●

Record 1 will be destroyed before it is processed and written out. (Remember that entering data into core storage is destructive; old data is replaced by new.)

10 Can you suggest a way to read in the second record without interfering with the processing of the first record?

●●●

Read it into another area of storage.

11 The use of one storage area to permit reading in or writing out of data, while processing is taking place in an alternate area, is called buffering. In the diagram in Figure 3, the storage area into which record 2 is being read is acting as a ....

●●●

buffer.

12 Although it is acting as a buffer, the storage area being used is not a separate piece of hardware; it is part of main storage. The programmer sets aside those areas of main storage that are to be used as buffers. This means that he determines the sizes of buffer areas. The exact amount of storage required can be allocated. A main storage buffer can therefore be used for efficient handling of records of any ....

●●●

size, length, etc.

13 Refer to Figure 4. At D, the area occupied by record 1 is acting as an (input/output) .... buffer.

●●●

output

14 At E, Record 3 is being read into this same area. It now is an .... buffer.

●●●

input

15 Once data has been read into a storage buffer, it may be processed where it is, or moved elsewhere for computation. The first method avoids having to move data and is therefore .... (more/less) efficient.

●●●

more

16 Let's look at the first method, for an input file. Refer to Figure 4. While record 1 is being processed in the first area, record 2 is being read into the second area. When record 1 is finished with the first area (that is, it has been both processed and written out), the first area can be used again as an input buffer.

So, while record 2 is being computed in the second area, record 3 is being read into area ....

●●●

1

17 Thus, area 1 alternately is a processing area and a ....

●●●

buffer.

18 This flip-flop technique is the simplest form of storage buffering, and is the most widely used. The two areas involved usually are referred to as I/O storage areas.

A similar technique is useful with blocked records. Suppose the processing on each logical record within a block took 10 milliseconds (ms), and it took 8 ms to read in a 5-record block. The reading in of an entire block could be accomplished during the processing of (how many?) .... logical records.

●●●

one

19 On this occasion, instead of setting aside two I/O storage areas as buffers, the programmer needs to reserve only one, plus a work area which is big enough for a single logical record.

Let's assume that each logical record is 100 bytes long, and that each block has 5 logical records. If two I/O storage areas were used as buffers, how many bytes of storage would be needed?

●●●

1000

20 If one I/O storage area and one work area are used, only .... bytes of storage are needed.

●●●

600 (One 500-byte area for the input block, and one 100-byte area for the work area.)

21 No processing is done in the I/O storage area if we have reserved a work area. Logical records are moved as required for processing into the ....

●●●

work area.

22 For output, processed data can be moved from the work area to an output area. But let's concentrate on input. When the last record in the block has been moved from the input area, reading in of the next block of records can be started. By the time the first record of the next block is needed for processing, what will be the status of that block? (In your own words.)

●●●

It will have been read completely into the input storage area.

23 Note that this is possible only when the entire block can be read, in no more time than is required to process a single logical record.

●●●

24 The one buffer method uses less storage than the two buffer method. However, in addition to processing, records must be moved about in storage. Which method would require the most processing time per logical record?

●●●

The one buffer method - time required to move the record must be added to the time required to process it.

25 Special conditions may permit you to achieve overlapped processing with the one buffer method. (As we have just shown, you must be able to read in a complete block of records while you are processing one logical record.) But in general, the two buffer method is the one most often used.

●●●

26 We have seen how the use of two input storage areas helps achieve overlapped processing. But we also must consider the output. Assume we are putting out a file of blocked records, using a single output storage area. As each logical record is processed, it is moved to the next available record space in the output area. When the output area is filled, the channel is signaled and the block of records begin to move to the designated output device. Can we immediately start moving computed records into the output area?

●●●

No. We must wait for the output area to be cleared, that is, all records in the output area must have been transmitted to the output device before we can again use the area.

27 What can be done to enable processing to continue while the records are being written from an output area?

●●●

Set up a second output area and alternately fill each. The most recently filled area is written while the alternate is being filled.

28 Each data file, whether input or output, can have its own pair of I/O storage areas. Note that we said "I/O". As you know, this is an abbreviation for "input/output". It is customary to call these areas "I/O" areas, regardless of whether the area is used as an input or output area.

A file of records, each 125 bytes long, has 4 logical records in each physical block. How many bytes must each storage area have?

●●●

500

29 Each file of records involved in a job will be read/written by a specific I/O device on a particular channel. The I/O storage areas associated with the file must be (greater than/ less than/equal to) .... the largest quantity of data transmitted during a single I/O operation.

●●●

equal to

30 Define an I/O storage area.

●●●

A portion of main storage reserved for the data being received from or sent to a specific I/O unit on a particular channel. The size of the area is such that it can accommodate the amount of data transmitted during any single I/O operation.

31 What is the advantage of assigning more than one I/O area to each I/O file?

●●●

It makes possible overlapped processing by providing an area into which or from which data is transmitted while the CPU is processing data in an alternate area.

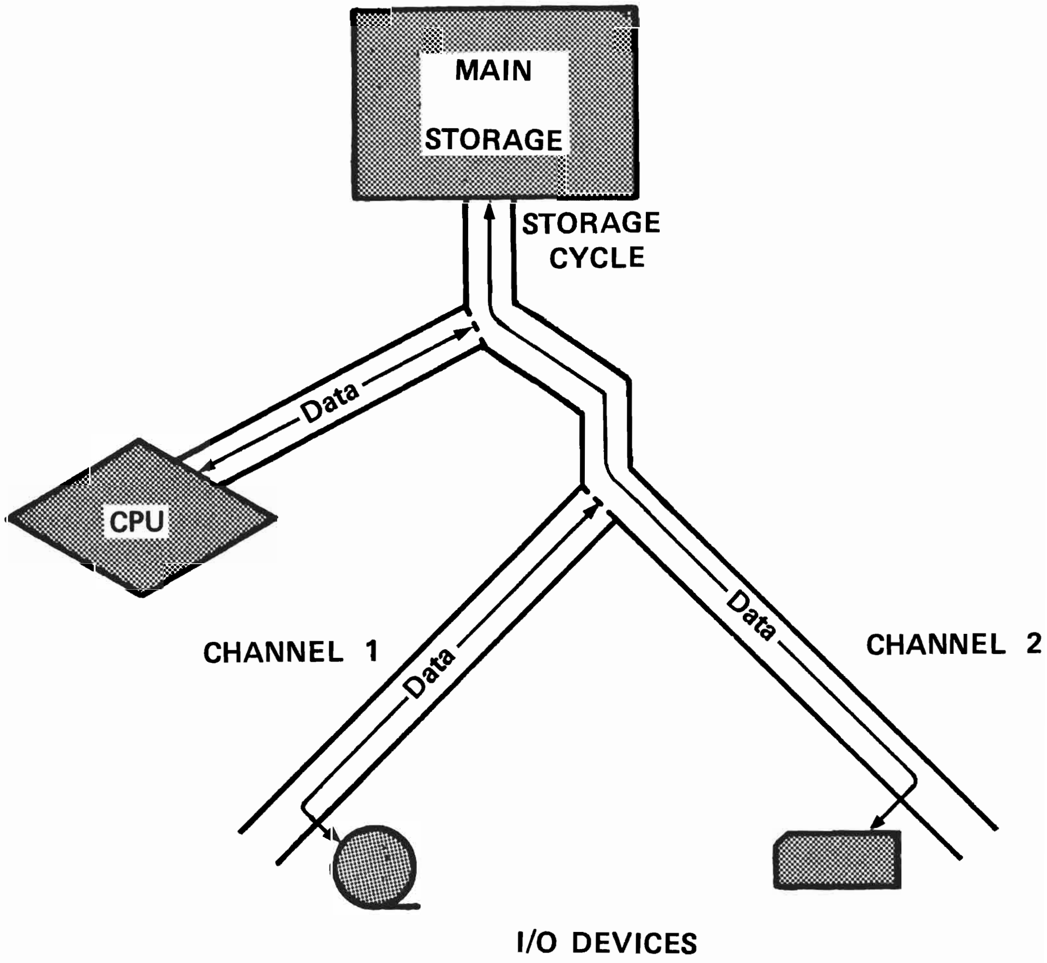

MAIN
STORAGE
STORAGE CYCLE
Data
CPU
CHANNEL 1
Data
Data
CHANNEL 2
I/O DEVICES

Figure 5.

32 We have examined two techniques for achieving overlapped processing. One uses two I/O storage areas per file and processes the input records right in the input areas. The other technique uses only one I/O area per file, plus a work area in which records are computed.

Frequently, it is necessary to add and/or delete records as we process a file. This can be done when we are processing in the I/O areas, but requires special programming that can be rather complex. If we use the work area technique, it is easy to add and delete records, but you will recall that the work area method is somewhat inefficient because (your own words) ....

●●●

records must be moved to and from the work area which takes time.

33 The two I/O area method and the I/O area with work area are .... techniques the programmer may use to achieve overlapped processing.

●●●

buffering

34 Earlier, we said that overlapped processing is possible only if the computer system has channels. Let's see what part channels play in computer operations in general, and in overlapped processing in particular.

Channels are those components in computer systems that direct and control the flow of data between I/O devices and main storage. Data flows along the channels, under control of the channels, to and from the I/O storage areas we have been discussing.

Where would you expect a channel to be connected in a computer system?

●●●

between one or more I/O devices and main storage.

35 Usually, several I/O devices share the same channel. For the time being, assume that only one of these devices will be operating at one time on each channel.

So to overlap reading from one device, say a magnetic tape, and writing on another, you would need (how many?) .... channels.

●●●

two

36 In this example, data would be entering main storage from one channel at the same time that data would be leaving storage on the other channel. In other words, we are reading on one channel at the same time we are writing on the other.

When two or more computer activities, such as reading and writing, occur concurrently, we have ....

●●●

overlapped processing.

37 During completely overlapped processing, data is being transferred between main storage and the input/output devices. In addition, the CPU is executing instructions from the problem program (or supervisor), operating on data already read in. Where are the instructions and the data being operated on?

●●●

In main storage.

38 Now look at Figure 5. It shows that only one data path enters and leaves S/360 main storage. All instructions and data moving between main storage and the CPU use this path; so does data moving between main storage and the channels.

The time required to transfer a single byte to or from main storage is called a storage cycle. Look at Figure 5. Which of the following can be working with main storage on any one storage cycle?

a. Both channels or the CPU.
b. Only one channel or the CPU.
c. Both channels and the CPU.
d. Only one channel and the CPU.

●●●

b. Only one channel or the CPU. No matter whether the byte is part of an instruction, of data being processed, or of data being read in or out, no other byte movement in or out of storage is possible, while it is being moved.

39 Most input/output devices cannot transfer data to main storage as quickly as storage can accept it. Even a disk drive, which transfers a byte every 6 microseconds (a microsecond is one millionth of a second) does not keep up with S/360 storage speeds. This means that for every storage cycle (the time required to transfer a byte to or from storage) used by the channels to get data in and out of storage, there are several storage cycles available to the CPU.

Between the transfer of successive bytes of an input/output record there is (no time/time to spare) ....

●●●

time to spare.

40 As the computer operates, it fetches instructions from main storage to the control unit, and transfers data between main storage and the arithmetic-logic unit. These activities require storage cycles, since they involve main storage. But when I/O operations occur, these activities must stop to permit each character of I/O data to enter or leave storage. I/O operations, in effect, preempt main storage so that no instruction fetching and/or data transfer to or from the arithmetic-logic unit is possible.

But, as we pointed out, there usually is time to spare between the transfer of successive bytes in an I/O record. So, during this spare time, instructions can be executed.

What enables S/360 to execute instructions between the transfer of successive bytes in an I/O record? (your own words.)

●●●

The high internal speed of S/360 and the relatively slow speed of I/O devices.

41 During I/O operations, then, storage is used alternately by the channels and the ....

●●●

CPU.

42 What computer system components use main storage?

●●●

Channels and the CPU.

43 A S/360 can have several channels. On a multi-channel machine any one storage cycle is used by the CPU or by one of the ....

●●●

channels.

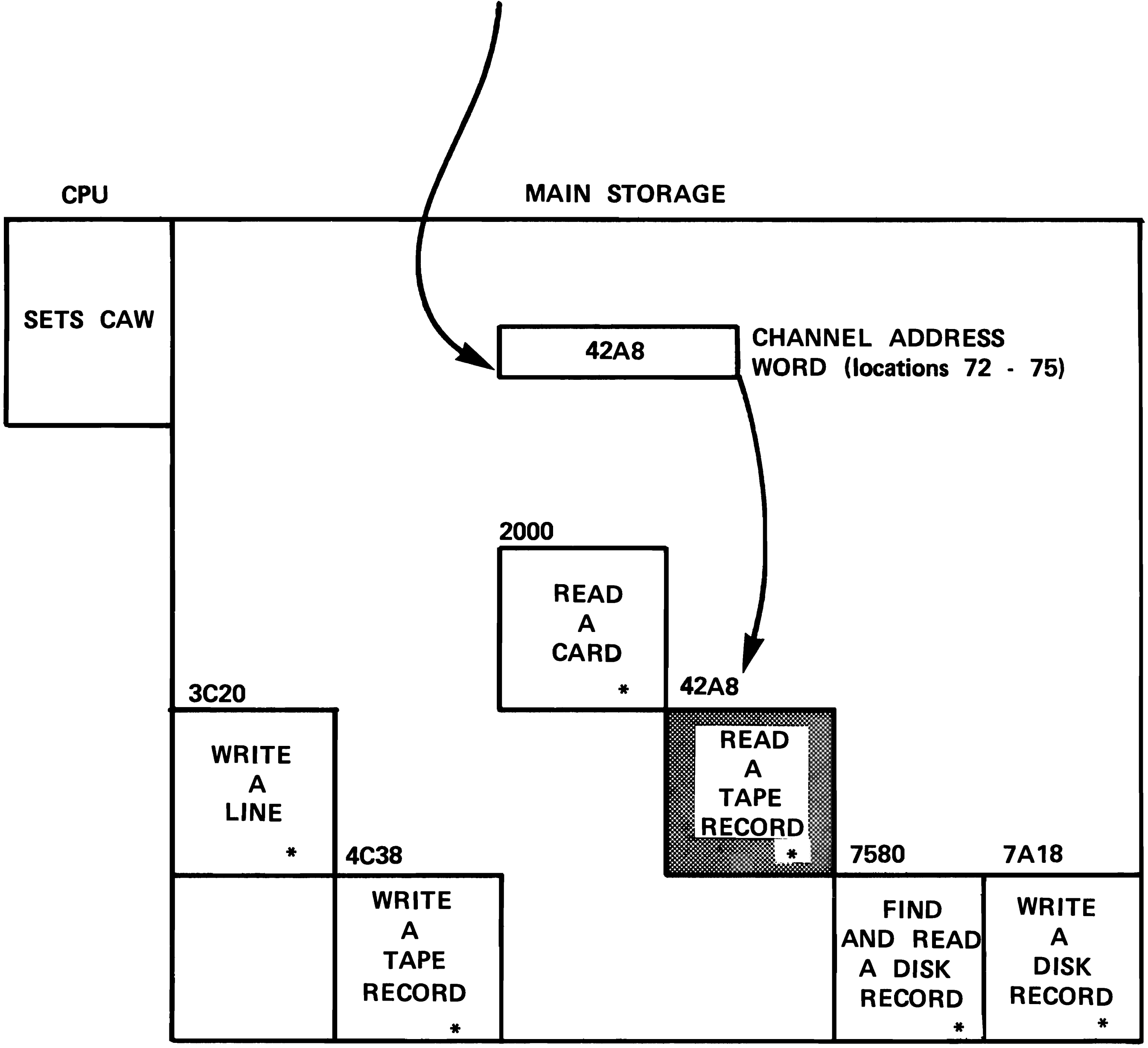
44 Thus, on any storage cycle a byte of data can be transferred between storage and one channel or between storage and another channel or between storage and the ....

●●●

CPU.

1. Program selects a job to be done by a device.

2. Supervisor puts the address of list of commands in the CHANNEL ADDRESS WORD (CAW).



* List of channel commands for indicated I/O operation.

CHANNEL OPERATION - CAW

Figure 6.

45 But a record consists of many bytes. Reading it will take many storage cycles. Between each cycle involved in reading (or writing) there will be cycles available for processing. Thus, data transfer is .... with processing.

●●●

overlapped

46 Earlier, we said that the channel's job is to direct the flow of data between I/O devices and main storage. It does this when told to by the CPU, which of course, is simply executing supervisor program instructions.

The supervisor starts each I/O operation by doing two things:

1. Issuing a Start Input Output (SIO) instruction that activates the channel and tells it which I/O device to use.

2. Placing in a special storage word the address of a list of commands that must be executed to complete the I/O operation.

True or false? An I/O routine consists of a single instruction, the SIO instruction, executed by the CPU.

●●●

False. Additional commands are required. (See (2) above.)

47 All S/360 operations require additional commands after the SIO instruction that initiates them. These commands are kept in main storage. To locate the appropriate list of commands, the channel must know the .... of the list.

●●●

address

48 The supervisor provides the channel with the address of the first command in the list of additional commands. After being activated by the SIO instruction, that is all the channel needs to carry on the I/O operation.

Once it has got the channel started, the CPU can do other work not directly related to the I/O operation. The channel continues without help from the CPU. This is an important point. Once started, the channel can operate independently of the CPU. It actually functions like a small computer on its own. It can execute special commands, taken from main storage, much as the CPU executes instructions taken from main storage. These commands control the I/O device and keep the channel going. We will say more about this later.

The channel continues without help from the CPU until it reports back to the CPU that its work is finished. When it reports back, it will interrupt the program being executed. What kind of interrupt is this?

●●●

An I/O interrupt.

49 Figure 6 shows how the supervisor tells the channel where the command list is located in main storage. Each box represents the command list for a different I/O operation. The number at the top of the box is the address of the first command in the routine.

In Figure 6 the program is to read a tape record. Its first step is to put the address of the first command in the list into the ....

●●●

channel address word (CAW).

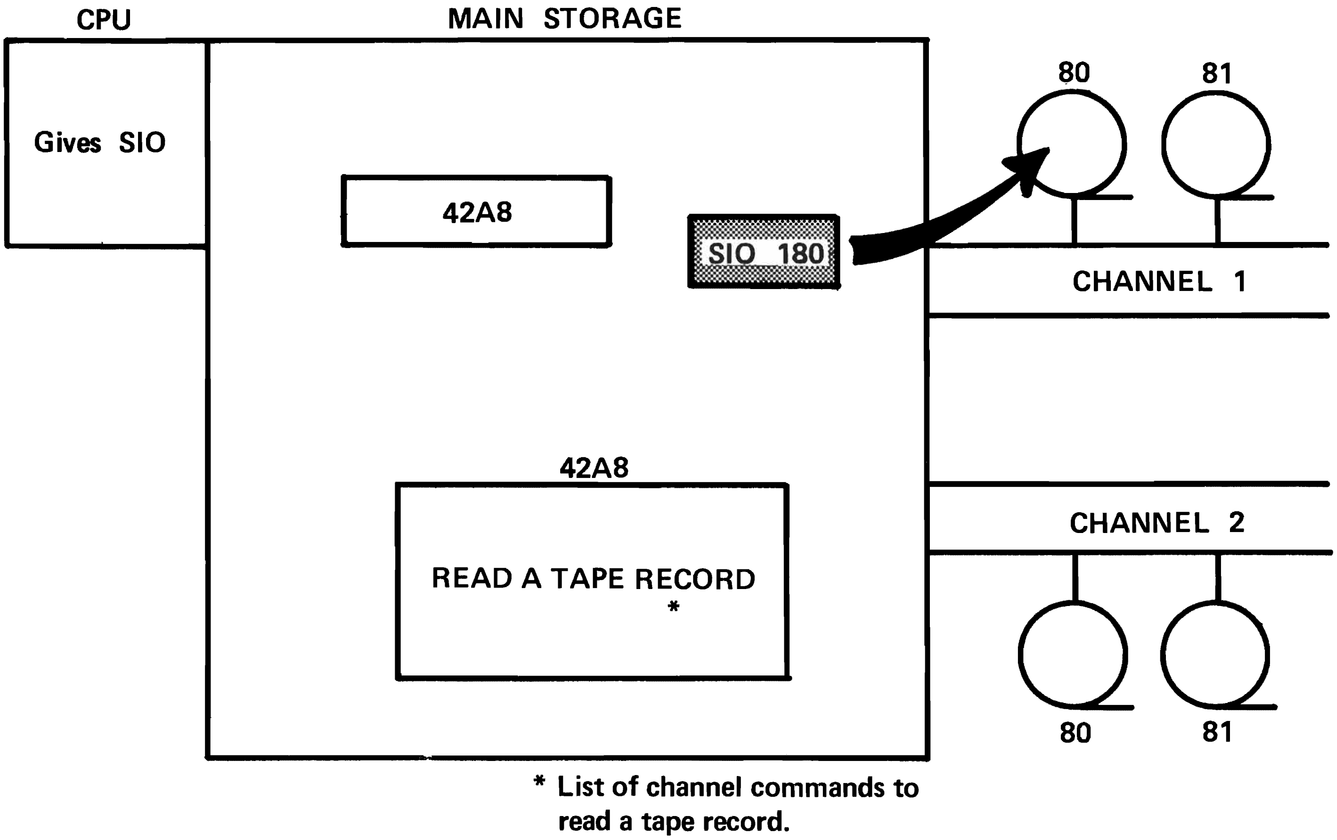
50 The address of the first command in this example is ....

●●●

42A8.

3. Program selects a suitable device to perform the operation.

4. Tells the channel to Start its Input or Output operation (SIO instruction).



CHANNEL OPERATION - SIO INSTRUCTION

Figure 7.

51 This list of commands constitutes the additional steps in the I/O operation, that we previously referred to. After putting the address of the first command in the list into the channel address word, the supervisor starts the channel operating and tells it which I/O device to use. This information is contained in the .... instruction.

●●●

Start Input Output (SIO)

52 Figure 7 shows how the SIO instruction indicates the address (channel and device numbers) of the I/O device to be used. In this example, the channel is number ...., the I/O device is number ....

●●●

1
80

53 After supplying this information the CPU continues with any other work it has to do. Input/output is controlled by ....

●●●

the channel.

54 This may seem like a complex way of doing something as simple as reading a card, particularly if you have had experience with other computers in which a single I/O instruction caused all the operations necessary to read or write a record. Actually, things haven't changed much; except for the instructions that set up the channel address word, the CPU executes just one instruction, the .... instruction.

●●●

Start Input Output (SIO)

55 But the working of the channel is more complex than on earlier computers and allows much more efficient scheduling of input and output operations.

A S/360 channel has many of the attributes of a separate computer, and as you have seen, it shares main storage with the CPU. The program that the channel executes consists of commands that are executed one at a time in the sequence in which they appear in storage.

Match the following:

a.	CPU	1.	Commands
b.	Channel	2.	Instructions

●●●

a. 2
b. 1

56 Commands can be located anywhere in main storage. They occupy words called channel command words (CCW's). The channel locates the desired list of CCW's by looking at the ....

●●●

channel address word (CAW).

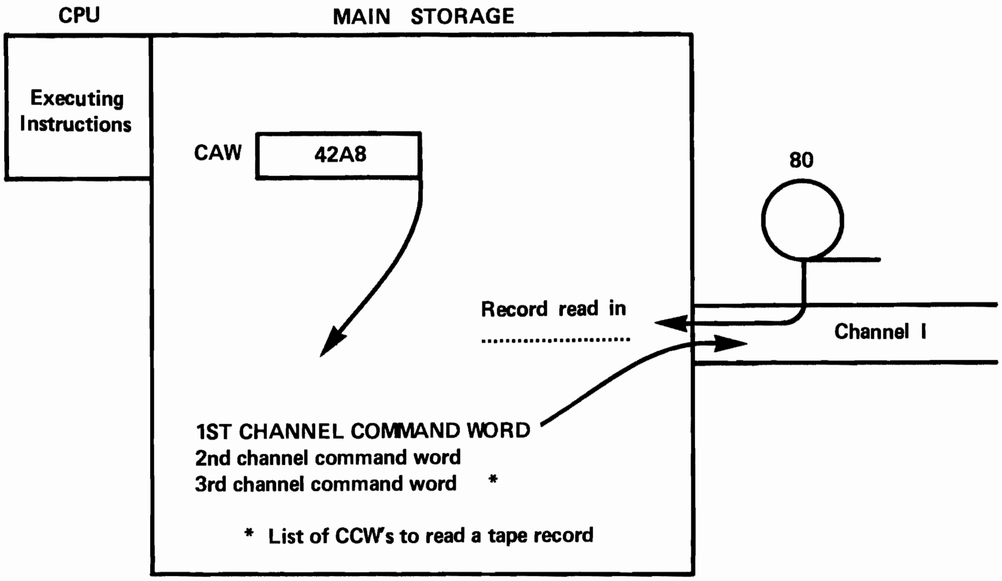
57 Note that the CAW just "points to" the first channel command word (CCW). In other words, it specifies the address of the first CCW. After finding the first CCW, the channel finds the others in successive locations. The address in the CAW isn't needed again. So there's only one CAW. But there can be any number of ....

●●●

CCW's.

5. Channel finds the address of its program in the CAW (which can then be freed for use by other characters).

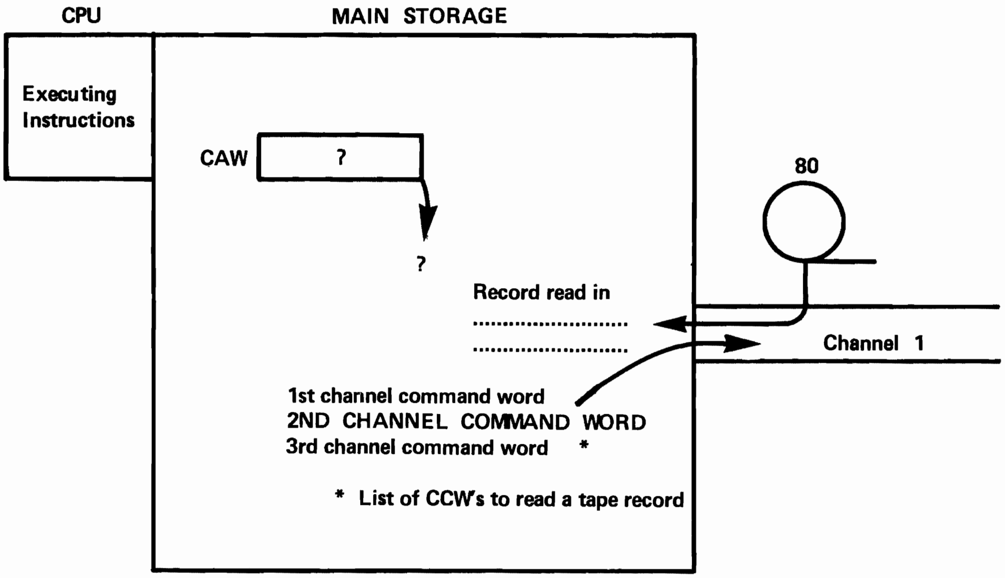
6. Channel executes the command found there.



CHANNEL OPERATION - CCW

Figure 8.

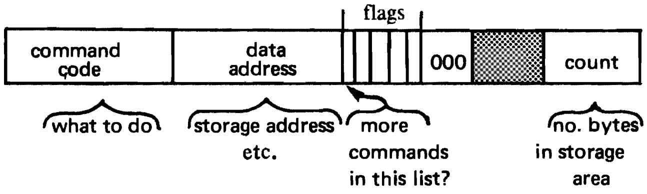
7. The channel executes commands in turn.



CHANNEL OPERATION – CCW

Figure 9.

58 Each CCW contains several items of information the channel must have to do its job. The illustration below shows the CCW format.



See if you can match the following descriptions of the CCW data fields with the captions in the above illustration.

1. The number of bytes in the storage area, hence, the physical size of the block of records being written or read.
2. The address of the first byte of the I/O area.
3. Whether there are more commands in this channel program.
4. The operation to be performed.

●●●

1. count
2. data address
3. flag
4. command code

59 Now look at Figure 8. The channel has located the first command, and in response reads in a record from device 180. The address of the first command was found in the ....

●●●

CAW.

60 The number of the I/O device was specified by the .... instruction. (See Figure 7)

●●●

SIO

61 Refer to Figure 9. After the first command has been executed, if there are more commands in the list, the next one is executed. The channel works its way through to the end of the channel program in this way. You'll see in a moment how it knows where the end is. Then, when it has finished, the channel reports back to the CPU which, throughout the channel operation, has been busily processing data.

Reporting back consists of:

1. causing an I/O interrupt, and
2. storing information about the completed I/O operation in a fixed area of storage called the channel status word (CSW). (See Figure 10.)

Do you remember what happens on an interrupt? The program sequence that was being executed is temporarily suspended. What happens to the current PSW?

●●●

It is stored in a special location, as the old I/O PSW.

62 What replaces it?

●●●

The new I/O PSW.

63 So the next instruction executed after the I/O interrupt is determined by the new I/O PSW. This instruction is the first of a supervisor routine that is executed only when there is an I/O interrupt. What would you expect this routine to do? (Your own words.)

●●●

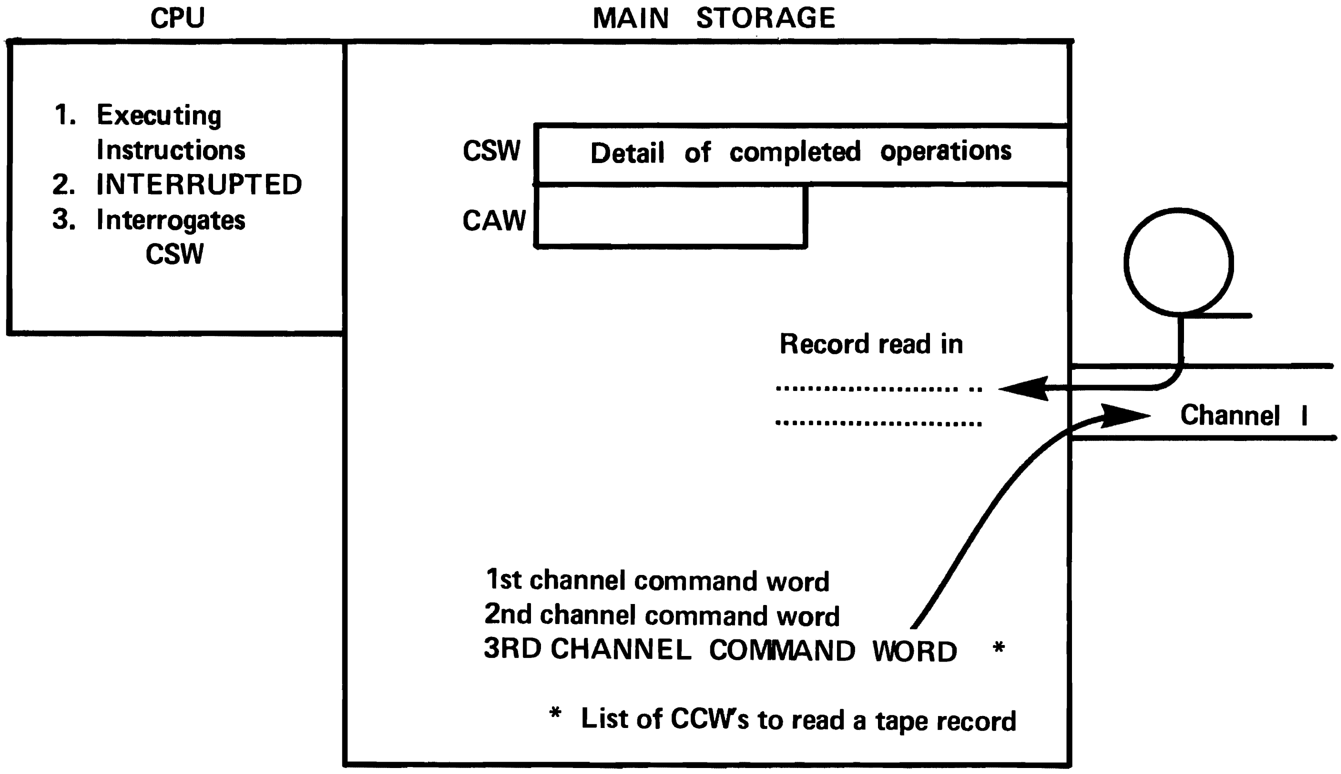
Find out which channel program caused the interrupt, and find out if the I/O operation was successful.

64 Where will this interrogation routine find information about the just-completed I/O routine?

●●●

8. When the last CCW in the chain has been executed, the CPU is informed at the first opportunity by an I/O interrupt.

9. Details about the completed operation are put in the Channel Status Word (CSW).



CHANNEL OPERATION - CSW AND I/O INTERRUPT

Figure 10.

In the CSW (channel status word).

65 There is only one channel status word. Its contents are set up by the channel program just executed. It reflects the conditions that exist in the I/O device and on the channel when the channel operation has ended. It is interrogated by the special routine we just referred to.

But we know that more than one channel operation can be going on at the same time. Suppose another channel completes its operation before the CSW is interrogated following the first one. What happens to other I/O interrupts while the CSW is being interrogated?

•••

They are masked. They will not be allowed until the interrogation for the preceding channel operation is complete.

66 The special routine that is executed when an I/O interrupt occurs also will examine the old I/O PSW. This contains the address of the device that caused the interrupt. So on an I/O interrupt, the routine will interrogate two doublewords. What are they?

•••

The CSW, and the old PSW.

67 Complete the words below, which refer to channel operations; then match the completed words with items 1, 2, 3, 4.

a. C________ A________ W________

b. C________ C________ W________

c. S________ I________ O________

d. C________ S________ W________

1. A location containing a direct command to the channel.
2. A location containing information about a completed channel operation.
3. An instruction executed by the CPU to initiate a channel operation.
4. A location containing the address of the list of commands to be executed by the channel.

•••

a. Channel Address Word - 4
b. Channel Command Word - 1
c. Start Input Output - 3
d. Channel Status Word - 2

68 Here is a list of the steps in a channel operation. Which of the following is involved in each step: CAW, CCW's, SIO, I/O PSW's, CSW?

a. CPU indicates list of commands to be executed.
b. CPU initiates channel operation.
c. Channel locates first command.
d. Channel executes channel program.
e. Channel stores information about job just completed.
f. CPU is interrupted.

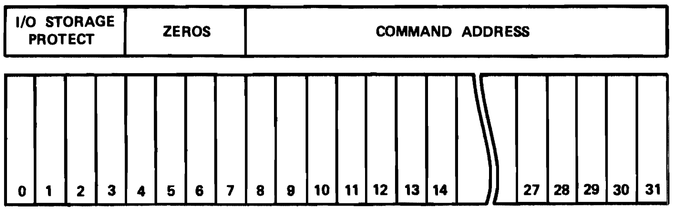
•••

a. CAW
b. SIO
c. CAW
d. CCW's
e. CSW
f. I/O PSW

69 In which of the steps in Frame 68 is the CPU involved?

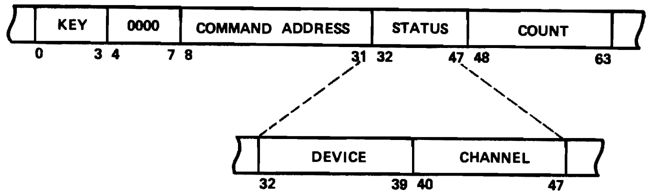
•••

a, b, f



CHANNEL ADDRESS WORD

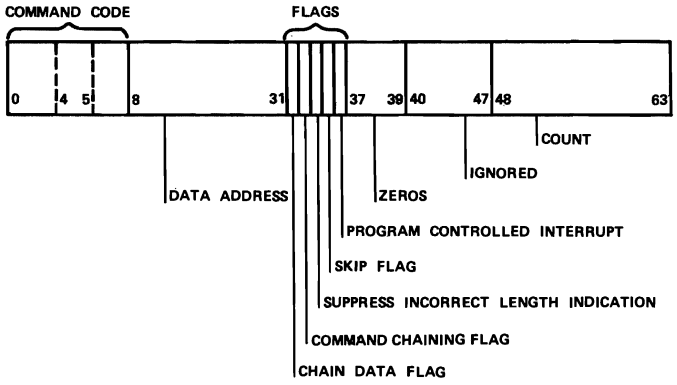
Figure 11.



BIT	DESIGNATION	BIT	DESIGNATION
32	Attention	40	Program - Controlled Interruption
33	Status Modifier	41	Incorrect Length
34	Control Unit End	42	Program Check
35	Busy	43	Protection Check
36	Channel End	44	Channel Data Check
37	Device End	45	Channel Control Check
38	Unit Check	46	Interface Control Check
39	Unit Exception	47	Chaining Check

CHANNEL STATUS WORD

Figure 12.



CHANNEL COMMAND WORD

Figure 13.

70 You will find the formats of the CAW, the CSW, and the CCW's in Figures 11, 12, and 13. These illustrations are for your information only. Don't attempt to learn the formats.

●●●

71 There is one field in the CCW that we should discuss. It is the command chaining flag, in bit position 33 of the CCW.

In an earlier frame we said we would show how the channel knows it has executed the last CCW in a list. The command chaining flag is the answer. If the CCW has a 1-bit in position 33 it indicates to the channel that there is another CCW in the list. You can deduce that if there is a 0-bit in position 33 it tells the channel that ....

●●●

this is the last CCW in the list.

72 Now, how does the programmer fit into the I/O operation? Actually, his job is simple. In his problem program he writes a macro instruction at those points at which he wants to write or read data, or do other I/O operations. There is a specific macro for each. In addition, he supplies a list of parameters for each of his data files. That's all. The channel programs, the scheduling and control programs, and any other routines required for I/O are included in IBM-supplied programs.

●●●

73 Let's sum up what we have learned about channels and overlapped processing.

a. For each input and output file, areas of storage, called I/O areas, are set aside.

b. The size of each area is such that it can accommodate the amount of data transmitted during any single I/O operation.

c. All data entering main storage from input units, or leaving main storage for output units travel through channels.

d. Channels direct and control the flow of I/O data independent of the CPU.

e. The CPU signals the channels to start an I/O operation and then returns to processing other data.

f. The channel begins executing one or more commands, that together make up the channel "program". The address of the first of these commands was placed in the CAW by the supervisor. Each command indicates whether or not there are additional commands in the program.

g. The commands executed by the channel cause data to be read in or written from main storage.

h. When the channel operation is completed, or stopped for any reason, the channel notifies the CPU by way of an interrupt.

i. Several channels can be operating concurrently with the CPU. This permits overlapped processing on the system (reading, writing, and computing simultaneously).

●●●

74 What is the function of a channel on a computer? (Briefly, in your own words.)

●●●

It directs and controls the flow of data between I/O devices and main storage.

75 In what way does a channel increase the efficiency of a computer's operations, and what enables it to do so?

●●●

After being signaled by the CPU to start an I/O operation, the channel communicates with the I/O devices and permits the CPU to continue processing data concurrently with the I/O operation. This permits the system to do overlapped processing. It can do this because of the ability of the channel to execute a limited number of I/O commands, independent of the CPU.

76 Sequence the events in the list below in the order in which they would occur.

a. Channel commands activate the control unit, which in turn activates the desired I/O.

b. Control program issues Start Input Output (SIO) instruction to the channel.

c. Data is transmitted to the available I/O area while problem program processes data in the alternate I/O area.

d. Problem program signals control program that all records in an I/O area have been processed.

e. Channel signals control program by way of the I/O interrupt that the transmission is ended.

f. Channel locates channel command words (CCW's), using the address from the channel address word (CAW), and begins executing the commands. Problem program resumes processing records, switching to the alternate area.

●●●

1. d.
2. b.
3. f.
4. a.
5. c.
6. e

77 We are not quite finished with our study of channels. We know the basic principles of their operation and the benefit derived from their use. There are a few more things about channels you should know.

For example, let's consider this. Although S/360 CPU's do not differ in logical characteristics (they do differ in power, i.e., the ability to do a given amount of work in a unit of time), the input/output units that connect to the CPU's do differ in many ways.

S/360 CPU's can be attached to many types of I/O devices. And as new applications for computer usage are developed, new types of I/O devices are developed to accommodate them. Consider the following list of existing I/O devices:

1. Card Readers
2. Card Punches
3. Card Reader/Punches
4. Paper Tape Readers
5. Printers
6. Magnetic Tape Units
7. Disk Units
8. Magnetic Drum Units
9. Data Cells
10. Optical Character Readers
11. Optical Mark Readers
12. Teleprocessing Terminals (many types)
13. Magnetic Ink Character Readers
14. Audio Response Units
15. Visual Display Units
16. Console Typewriter

●●●

78 You can readily imagine that these many I/O device types have different characteristics. For example, a card reader can read data, and can also select cards into various stacker pockets after they have been read. A magnetic tape unit can read data, and also can backspace tape, rewind tape, and unload tape, none of which a card reader can do. (Because, of course, there is no need for a card reader to have these capabilities.) Each device performs the common function of reading data, but in addition, performs other operations peculiar to itself.

A printer spaces and skips the paper form. A magnetic disk unit used for output sends its read/write heads to specific addresses on the disk pack at which data will be recorded. A punch unit selects newly punched cards into stacker pockets. What common function can each of these three devices perform?

●●●

Writing data. Each will respond to a write command.

79 The problem, then, is to provide for the many different characteristics of the many S/360 device types.

This could have been done by putting on a separate channel for each I/O device type, and giving the channel a set of commands peculiar to that type. You can imagine how expensive and inefficient this would be.

The S/360 solves this problem in a unique way. The S/360 is designed to permit any type of I/O unit, present and future, to be connected to any channel and to respond to a standard set of commands that can be executed by the channel.

Assume a given S/360 has one multiplexor channel and two selector channels. Can you connect an optical mark page reader to either channel? Will it work, if we do?

●●●

Yes. Any I/O unit can be connected to any S/360 channel, and will respond to the standard set of commands the channel can execute.

80 The design feature that permits attaching any I/O unit to any channel on a System/360 is called standard interface. Standard interface is a significant development in computer technology.

Why is it so important? As we have seen, it gives us great flexibility in connecting I/O devices. What is equally important is the standard set of commands (CCW's) with which a channel can activate and control the I/O units attached to it. Special commands for each unit are not required.

A number of input devices attached to the same channel (will/will not) .... respond to the standard read command on the channel.

●●●

will

81 In your own words, briefly define "standard interface."

●●●

A design feature of S/360 that permits attaching any I/O unit to any channel.

82 What are the advantages of standard interface?

●●●

It permits attachment of present and future I/O devices to the channels, without altering the I/O command set. Thus, any I/O unit will respond appropriately to the standard channel commands.

It simplifies the channel command structure by permitting a common set of I/O commands to communicate with the I/O units.

83 But, you may ask, how can a standard set of commands accommodate all these different I/O device types, each with some unique functions?

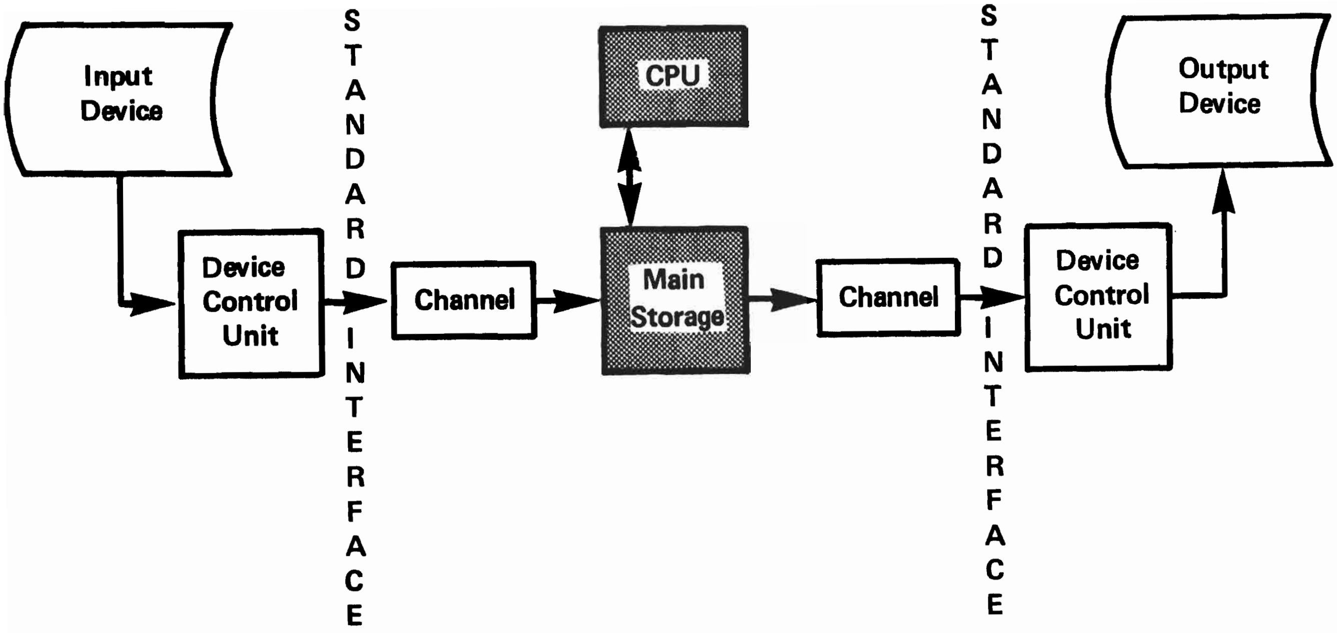

Input Device
Device Control Unit
STANDARD INTERFACE
Channel
CPU
Main Storage
Channel
STANDARD INTERFACE
Device Control Unit
Output Device

Figure 14.

The answer lies in the nature of the channel commands. They are listed below.

a. Read
b. Write
c. Read backward
d. Sense
e. Transfer in Channel
f. Control

We won't discuss these in detail. The nature of the first three is fairly obvious. They cause I/O units to read or write data, and in the case of magnetic tape, to read the tape in a reverse direction.

The Sense and the Transfer in Channel commands have to do with the mechanics of channel operation.

That leaves the Control command. Can you guess what its purpose is?

●●●

It enables the channel to cause I/O units to perform functions unique to themselves. In effect, it controls the I/O units.

84 The Control command can be structured in a variety of ways. Each different structure is designed to cause a specific type of I/O unit to perform a function unique to itself, such as stacker selection, backspacing tape, seeking disk addresses.

But now we find that we need another device, one that can analyze these variations in the structure of Control commands, and direct the I/O units accordingly. In effect, we need another little CPU, to analyze and execute some orders to the I/O units.

The unit that does this is called the control unit. Control units are the devices through which I/O units are connected to the channels.

Control units analyze ....commands and direct the I/O units accordingly.

●●●

Control

85 Where do you think the control unit would be connected in the computer system?

●●●

Between the I/O unit and the channel

86 The variations in the control command are called orders. They are analyzed by the control unit, which obtains them from the channel, and,after analysis, directs the I/O unit accordingly.

Match the following:

a.	commands	1.	control unit
b.	orders	2.	CPU
c.	instructions	3.	channel

●●●

a - 3
b - 1
c - 2

87 Earlier,we said that a S/360 design feature called standard interface permits the attachment of any type of I/O device to any channel on the system, and adapts the special characteristics of individual I/O units to the standard form of control offered by the channel. Thus, the control unit enables the channel to communicate with the I/O devices, using a standard set of commands.

In effect, then, standard interface is implemented where in the S/360? (See Figure 14.)

●●●

In the control unit.

88 A single control unit may control several devices, which may be of different types in some cases. When devices of widely varying types are to be attached, a separate control unit will be required for each.

A tape unit and a disk unit (can/cannot) .... share a control unit.

●●●

cannot

89 Six disk units (may/may not) .... share one control unit.

●●●

may

90 A maximum of 8 control units can be connected to each S/360 channel. The number of I/O devices that can be connected to a control unit depends on the type of I/O unit. For example, 8 magnetic tape drives can be connected to one magnetic tape control unit. This permits (how many)..........tape drives per channel.

●●●

64

91 The number of channels on a S/360 varies with the model number. A Model 30 can have one multiplexor and two selector channels. How many tape drives could be connected to such a system?

●●●

192

92 The control unit for card and printer equipment will accommodate up to 3 I/O devices, depending on the device. On a Model 30, then, with three channels, you could connect (how many)..........card and/or printer devices.

●●●

72 (3 devices x 8 control units x 3 channels)

93 Of course, these figures represent only theoretical maximums. Computer systems usually employ a variety of I/O units, but the total of any one type never reaches the theoretical maximum.

The main point to be made is that the considerations determining the number of I/O units that can be connected to a S/360 are:

a. The number of channels on the system.
b. The number of control units that can be connected to each channel.
c. The number of I/O units that can be connected to each control unit.

●●●

94 Item (c) in the last frame deserves some comment. The number of I/O units that can be connected to a control unit depends on the type of I/O unit. For example, as we noted, control units for magnetic tape drives accommodate up to eight units; control units for card and printer devices accommodate up to only three units.

Which is correct?

a. Control units vary with I/O device type.
b. One type of control unit serves all device types.

●●●

a

95 Remember that the channel directs the flow of information from device to main storage. The channel relieves the CPU of the need to communicate directly with I/O devices, thus allowing processing to proceed concurrently with I/O operations.

The control unit is linked to the channel via a standard interface. The standard interface provides an information format and a signal sequence common to all I/O devices.

96 Where is standard interface implemented in S/360?

●●●

In the control unit.

97 What is the physical relationship among I/O device, channel, and control unit?

●●●

The control unit links the I/O device to the channel.

98 Briefly describe the functions and operational characteristics of the control unit.

●●●

The control unit analyzes and executes orders to the I/O units, conveyed to it by channel commands.

99 What considerations govern the selection of a control unit through which a particular type of I/O device will be connected?

●●●

The control unit must be capable of handling all commands appropriate to the attached I/O units. Since no one type of control unit can handle all types of I/O units, the I/O and control units must be compatible.

100 The maximum number of control units that can be connected to a channel is..........

●●●

eight.

101 What considerations determine the number of I/O units that can be connected to a S/360?

●●●

The number of channels on the computer.

The number of control units that can be connected to each channel.
The number of I/O units that can be connected to each control unit.

102 What determines the number of I/O units that can be connected to any one control unit?

●●●

The types of I/O units to be connected. Some control units can handle more of the I/O units for which they are designed than other control units of their I/O device types.

103 All that has been said so far applies to both types of /360 channel.

The two types of the /360 channel are the multiplexor channel, intended for use with relatively slow devices such as card and paper tape readers, printers, punches, character readers, teleprocessing equipment, console typewriters etc., and the selector channel, intended for use with ....devices, such as magnetic tape or disk.

●●●

fast

104 There is only one multiplexor channel, but there can be one or more........... channels (depending on CPU model).

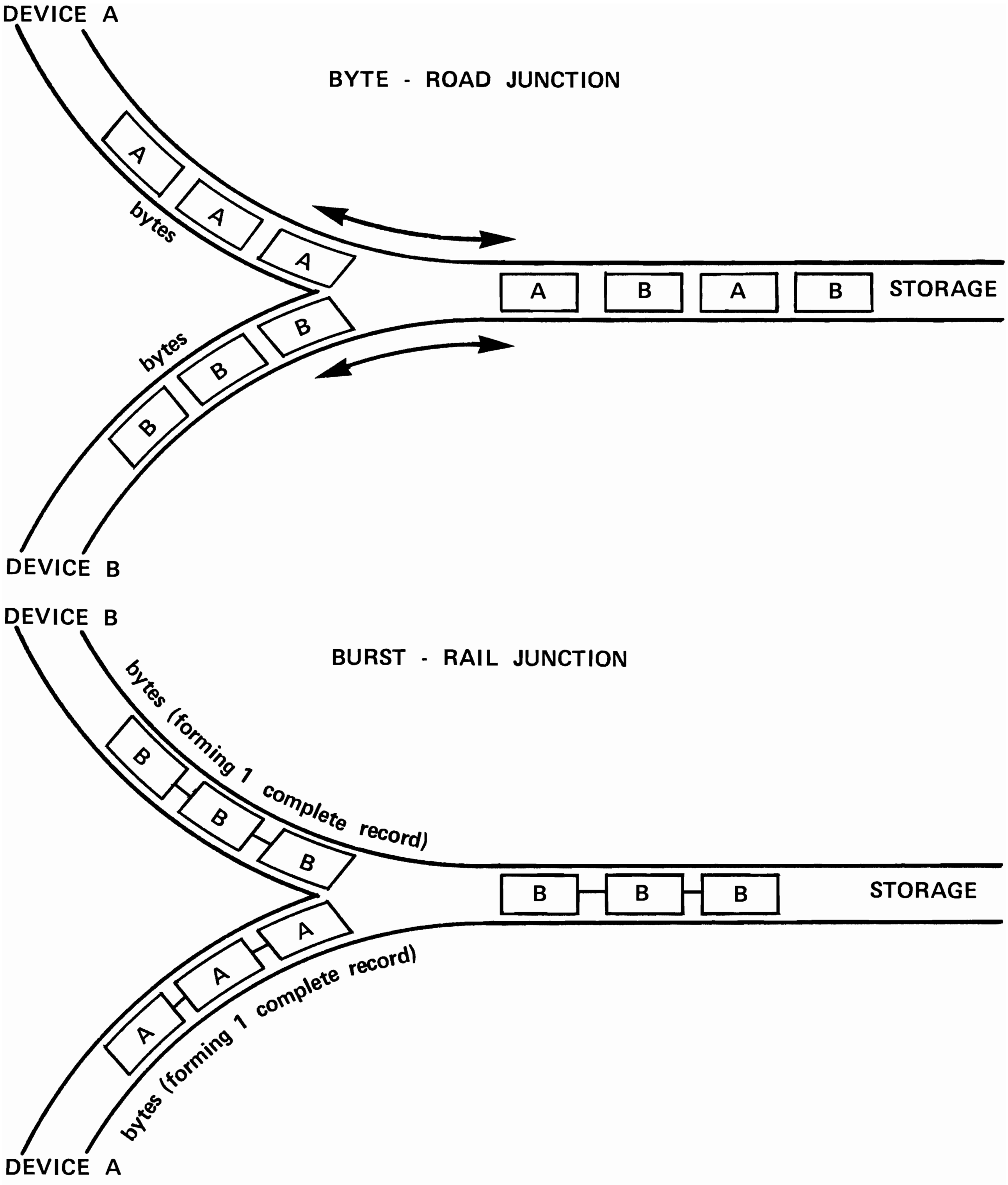
●●●

selector

105 Each type of channel gives one data path between device and ....

●●●

storage.



BYTE AND BURST TRANSMISSION MODES

Figure 15.

106 Along this path one...........is transferred at a time.

●●●

byte

107 The difference is that for slow devices there is time to spare between the transmission of successive bytes. To prevent loss of this time, several slow devices can share the path, each transmitting a byte in turn. So, on the........... channel, several devices can be operating simultaneously.

●●●

multiplexor

108 This sharing of the channel among several devices, where bytes from different devices are interleaved, is called byte interleave mode or just byte mode.

Only the...........channel can work in byte mode.

●●●

multiplexor

109 Of course, if faster devices are used on the multiplexor channel, sharing becomes impossible. When fast devices are used on the multiplexor channel, it operates in burst mode.

When in burst mode, how many I/O devices can operate at once through the multiplexor channel?

●●●

One only.

110 The selector channel, which is intended for fast devices can operate only in...........mode.

●●●

burst

111 A useful analogy to explain the difference between byte and burst mode is shown in Fig. 15.

●●●

112 A card reader connected to a multiplexor channel with other card and/or printing devices would be considered a (fast/slow)........... device and would operate in (byte/burst)........... mode.

●●●

slow
byte

113 A magnetic drum has a very high data transfer rate. It is considered a (slow/fast) ....device, and probably would be connected to a (multiplexor/selector)...........channel.

●●●

fast
selector

114 Remember that we said any I/O device can be connected to any channel. If a fast I/O device is connected to a multiplexor channel, in what mode will it operate?

●●●

Burst. (Fast devices always operate in burst mode, regardless of which type of channel they are connected to.)

115 If a card reader and a printer are connected to a selector channel, will they operate in byte mode?

●●●

No. (Burst mode is the only mode in which the selector channel can operate. Thus all devices connected to it operate in that mode.)

116 What determines whether an I/O unit is classified as fast or slow?

Basically, its data transfer rate. If the rate is sufficiently slow that it can operate in byte mode, it is considered a slow device. Fast devices operate only in burst mode.

Match the items in the columns below.

a.	slow I/O devices	1.	multiplexor
b.	fast I/O devices		channel
c.	burst mode only	2.	selector channel
d.	byte mode		
e.	2501 Card Reader		
f.	2401 Magnetic Tape Unit		

●●●

a - 1
b - 2
c - 2
d - 1
e - 1
f - 2

You have completed this section. At this point you should fill in your notes and take the self-evaluation quiz.

Self-Evaluation Quiz
I/O Channels

QUESTIONS

1. Define overlapped processing.

2. Briefly explain the flip-flop method of buffering.

3. Define "channel".

4. Explain briefly why it is necessary to have channels to do completely overlapped processing.

5. Describe the steps in a channel operation, using the proper terms or abbreviations for the programs and machine features involved.

6. Define "standard interface".

7. Give a major advantage of standard interface.

8. Where and how in S/360 is standard interface implemented?

9. What are the major operational characteristics of the multiplexor and selector channels?

Self-Evaluation Quiz
I/O Channels

ANSWERS Frame Reference

1. Two or more computer activities, such as read, process, and write, taking place concurrently. (36)

2. Two input/output areas in main storage are reserved for each data file. While data is being read into or written from one area, the data in the alternate area is being processed. (11)

3. The device that directs and controls the transmission of data between main storage and the I/O devices. (34)

4. Completely overlapped processing requires concurrent reading, writing, and processing. At any given time a channel can be either writing or reading; it cannot do both simultaneously. Thus, at least two channels are necessary – one for reading, one for writing. (36)

5. The problem program contains macro instructions where channel operations are to occur. The supervisor, which is called into operation by the macro instructions, places the address of the first of a group of Channel Command Words in the Channel Address Word. It then executes a Start Input Output instruction that initiates the channel operation and supplies the address of the I/O unit to be used. The channel successively executes the commands in the CCW's, while the CPU continues to execute problem program instructions. At the end of the channel operation an I/O interrupt occurs and the status of the channel is recorded in the Channel Status Word. The supervisor interrogates the CSW to determine if the I/O operation was successful, and takes appropriate remedial action if it was not. (76)

6. Standard interface is a design feature of S/360 that permits attaching any I/O device to any channel on the system. (81)

7. Standard interface permits a single set of I/O commands to control any I/O unit on any channel. (82)

8. Standard interface is implemented in the control unit, which interprets variations in the channel commands and sends them to the I/O units as orders. The control units link the I/O units to the channels. (96)

9. The multiplexor channel is intended for use with relatively slow devices such as card readers and punches, printers, and paper tape readers, whose data transmission rates are such as to permit the channel to operate in "byte" mode. This means that several "slow" devices can use the channel at the same time, each sending or receiving bytes that are interleaved with the others. (103,117)

   The selector channel is intended for use with relatively fast devices whose data transmission rates are too rapid to permit byte mode transmission. All bytes in a single group of data from a single I/O unit are successively transmitted on the selector channel while other I/O units on the channel remain inactive. This kind of transmission is called "burst" mode.

Card and Paper Document Handling,
And System Control, I/O Devices

SYSTEM/360 I/O DEVICES

Most can be connected to any model of S/360

1 - CARD READERS
2 - CARD PUNCHES
3 - CARD READER/PUNCHES
4 - PAPER TAPE READERS
5 - PRINTERS
6 - MAGNETIC TAPE UNITS
7 - DISK UNITS
8 - MAGNETIC DRUM UNITS
9 - DATA CELLS
10 - OPTICAL CHARACTER READERS
11 - OPTICAL MARK READERS
12 - TELEPROCESSING TERMINALS (MANY TYPES)
13 - MAGNETIC INK CHARACTER READERS
14 - AUDIO RESPONSE UNITS
15 - VISUAL DISPLAY UNITS
16 - CONSOLE TYPEWRITERS

Figure 1.

S/360 CARD AND PAPER DOCUMENT HANDLING, AND SYSTEM CONTROL, I/O DEVICES

1 In this topic, we are going to talk about two classes of I/O devices: those that handle cards or paper documents, and those that are used to control the computer system.

If you are familiar with card readers and punches, printers, and those I/O units that read documents magnetically or optically, you may skip this unit.

● ● ●

2 In the preceding topic we talked about overlapped processing and the I/O areas, channels, and control units that make it possible. We said that there is a large variety of I/O device types that can be attached to the S/360. In this topic, we will take a look at some of those types; specifically, those that handle cards or paper documents, and those that are used by the operator to control the computer system.

From the list in Figure 1, select some devices that handle paper and/or cards.

● ● ●

Card readers, and punches, printers, optical character readers, magnetic ink character readers, paper tape readers, optical mark page readers.

3 The operator must, from time to time, communicate with the computer. Also, there are times when the computer wants to signal the operator about conditons in the system. Can you name one or more devices that are used for this purpose?

● ● ●

The console
The printer keyboard (or console typewriter)

4 Input/output devices may be separated into classes, with all members of a given class performing a similar function. Thus, all card readers read punched cards, all printers imprint readable characters on paper, all magnetic tape units read and write magnetic tape.

But within any one device class there usually are several models. The differences among these models are more in speed and cost than in basic function.

The devices listed in Figure 2 all perform what function?

● ● ●

They read cards.

5 Name one or more areas in which the devices listed in Figure 2 differ.

● ● ●

Reading and punching speeds, ability to read and punch in one feed, read-punch pattern (serial vs. parallel).

6 Figure 3 lists some other I/O devices. All perform what function?

● ● ●

Printing

7 In what way(s) do the devices listed in Figure 3 differ?

● ● ●

Printing speeds, number of positions that can be printed, ability to print magnetic ink characters, print on cards.

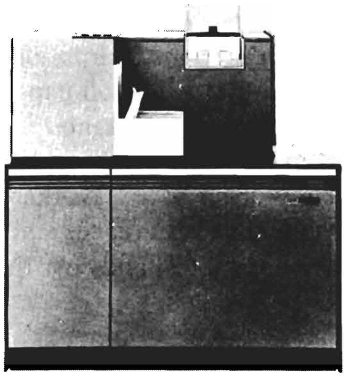
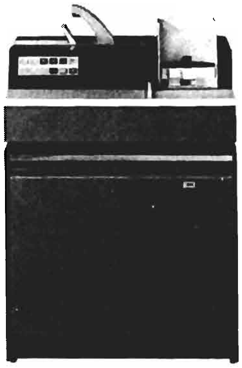
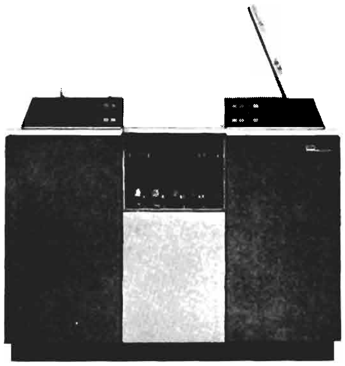
S/360 CARD READERS AND PUNCHES	
 1442 READER - PUNCH	Maximum reading speed 400 CPM Maximum punching speed 160 columns per second 1 feed for reading and punching Serial operation - column by column Also available as punch only
 2520 READER - PUNCH	Maximum reading speed 500 CPM Maximum punching speed 500 CPM 1 feed for reading and punching Serial reading Parallel punching Also available as punch only with punching speed 300/500 cards per minute
 2501 READER	Maximum reading speed 600/1000 cards per minute Serial operation
 2540 READER AND PUNCH	Maximum reading speed 1000 CPM Maximum punching speed 300 CPM Reading and punching in separate feeds Parallel operation - row by row Punch read feed available

Figure 2.

S/360 PRINTERS

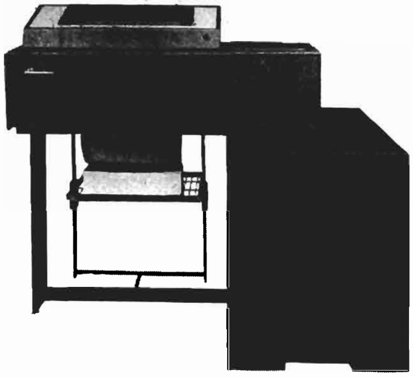
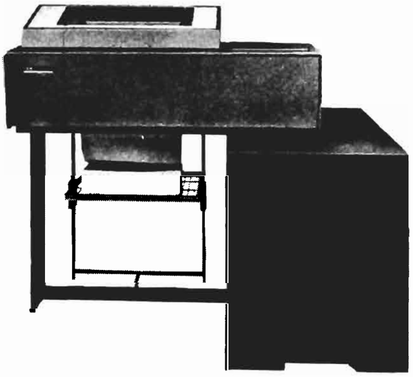
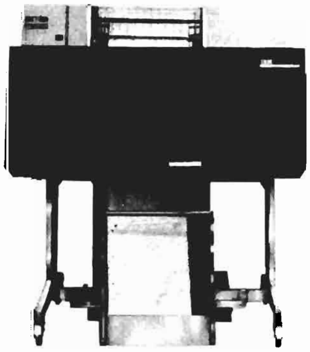
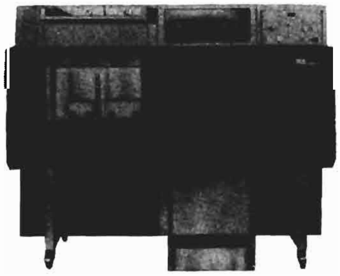
 1443	120 or 144 print positions Speed depends on character set 13/39/52/63 character set gives 600/300/240/200 lines per minute
 1445	113 print positions As 1443 but can print magnetic characters Speed depends on character set 14/42/56 character set gives 525/240/190 lines per minute
 1403	132 print positions 48 character set standard Model 2 - maximum speed 600 LPM Model 3 - maximum speed 1100 LPM Can use universal character set
 1404	Can print on cards as well as paper Maximum speed - cards - 800 CPM Maximum speed - paper - 600 LPM Optional feature allows reading of 30 columns of card

Figure 3.

8 You can see that it would be easy to get lost in a mass of performance and cost figures in selecting the I/O units for a computer system. Not only must the particular function be served (printing, punching, reading, etc.), but the particular model of that device type that does the job in the most efficient way for the least cost should be selected.

We will not attempt to evaluate all the factors that go into selecting an I/O device. But we will look at the characteristics of those device types with which this topic is concerned in conjunction with a computer system for an imaginary company.

●●●

9 The Electric Credit Sales Company has 500 branch offices, each serving about 2000 customers. Most customers pay in weekly installments. The present system is decentralized, that is, each branch office keeps the accounts for its own customers. A rapid growth rate has made it necessary to consider the use of computers to handle the increasing accounting load.

Which of these might result from using a computer?

a. Faster processing of individual records.
b. A decrease in record storage space.
c. A change in the physical environment in which data processing takes place.

●●●

All could result. Computers operate faster than humans, and data on tapes and disks takes up less space than paper documents. Also, computers usually require a dust-free environment.

10 Let's see how we can provide a suitable system for our imaginary company. Our automated system could be:

a. A central computer serving all branch locations, or
b. A small computer in each branch office.

Either approach presents a problem. A single large computer at a central location would probably cost less than 500 small computers, even if each branch office had enough work to keep its own small computer reasonably busy. Also, there is the problem of consolidating data from 500 individual computer installations into a form that is useful and usable.

On the other hand, there are certain difficulties in centralization. The main difficulty will be communication.

What must be done to data before it can be introduced into a computer?

●●●

It must be made machine readable, that is, put in a form that can be sensed by the computer input device.

11 What other consideration can you think of that affects introduction of machine readable data into a computer at a remote location?

●●●

The data must travel.

12 At the present time, customers of the Electric Credit Sales Company can pay their bills at the nearest branch, and/or receive statements of their accounts, all in a few minutes. This is possible because their records are in the shop. If master records are kept miles away from the location of the transaction, the time required to make the data machine readable and transmit it to the central computer location must be as short as possible.

Before we decide whether or not we want a centralized system, let's consider the problem of getting human language input into the machine.

Most computer input originally is transferred from source documents to .... by an operator using a manually operated punch.

●●●

punched cards

Card Readers and Punches

13 Most IBM installations use punched cards. These are read into storage by card readers. IBM manufactures several card readers that can punch cards as well as read them. To be accurate, these are really card reader-punches. In addition, there are card readers and punches that actually are separate machines mounted on one frame.

Card handling machines are either serial or parallel. A serial machine reads or punches column by column while a parallel machine operates .... by ....

●●●

row
row

14 In most computer installations, card readers feed the original input into main computer storage. Let's consider card input for our imaginary company.

What could we use as a source document, that is, the medium on which is recorded the figures of the transaction? Can you think of anything that is produced during the transaction that could be used as raw input? There is something – the paper tape on which the register prints details of the transaction. Most cash registers can easily be modified so that the tape contains a keyed-in account number as well as the amount paid.

Each transaction could be punched in a card, using the tape as a source document. The information would require, say 15 card columns.

Would we need to use one card per transaction?

●●●

Five transactions could be placed in each card.

15 How many cards would we prepare each week at all locations?

●●●

200,000

16 Assuming that each branch office punches the transactions into cards and forwards them to the central computer location where there is a 600-cpm card reader, how long will it take to read all the cards in?

●●●

About 6 hours.

17 From Figure 2, select the card reader(s) that could do the job for us.

●●●

The 2501 or the 2540 will read at 600 cpm.

18 Considering what you know of the job up to this point, which of the two possible readers would you select, and why?

●●●

The 2501, because it reads only. No punching has yet been specified.

19 Suppose that we select a central computer system and use it only for updating the account master records, which are on magnetic tape. Assume the new (updated) master records are written out on magnetic tape, and that the updated ones also are printed on an attached printer. From Figure 2, select the card reader you would use if you wanted the fastest possible card reading speed.

●●●

The 2501. Since the 2540 has punching capability as well, and we have not specified card output, you would not want to pay for unused capacity.

20 Let's assume that we do want punched card output. We've decided to punch a card for each transaction, and send it to the customer as a verification of the transaction. We find that a 500-cpm reading speed is satisfactory. Can we use the 2520 card reader-punch for card input/output? Why?

●●●

No. The input card will have 75 columns of information punched in it, representing 5 different transactions. The 2520 uses the same feed for reading and punching. Thus, any punched output must go into the input card. Obviously, we can't create a separate receipt card for each of the five customers, when we have only five free columns in each input card.

21 There are many applications in which input data requires only a portion of the card. We know that some card I/O devices have common feeds and some do not. Which type would you choose, to allow maximum utilization of card space?

●●●

One that has a common feed.

22 Often it is convenient to have computed results in the same card with the initial data. Indeed, if there is enough room, it is economical to do so, for these reasons:

a. The resulting record is in a single card. This saves on card costs. (One card holds the necessary data, instead of two.)

b. It saves on card storage space. (Only one card must be stored.)

c. It cuts down on input time for subsequent runs. (Only one card must be read.)

In order to punch into the same card from which data is read, the card I/O device must have a common feed. From Figure 2, select the units that have this capability.

●●●

The 1442 and the 2520.

23 For what reason do some S/360 card reader-punches have a common feed?

●●●

To permit punching into the same card from which data is read.

24 What are some of the advantages of punching into the same card from which data is read?

●●●

The resulting record is in a single card, which means that fewer cards are used, card storage requirements are reduced, and time required for subsequent runs is reduced.

25 Let's suppose we decide to punch a separate card for each customer of the Electric Credit Sales Company. Assuming the computer can maintain a punching speed of 2000 cpm, select from Figure 2 the card I/O device you would use, and explain your choice.

●●●

The 2540 is one possible choice. It is the only card I/O device listed that has separate feeds for reading and punching, a requirement of the job.

The 2520 is another choice. It has the fastest punching speed, but would require a separate card reader in this problem.

26 We have seen that, to permit punching into the same card from which data is read, some S/360 card reader-punches have common feeds. There is another aspect of card I/O operation that we should consider. It is the disposal of cards that have been read and/or punched.

Suppose that the computer reads in a transaction card for which there is no master record. Perhaps the transaction card has an incorrectly punched account number. Or suppose the computer finds an unreasonable dollar amount involved in a transaction. Which of these would you prefer?

a. Select the "error" card into a special compartment.
b. Print out something that identifies the card and the nature of the error.
c. Neither
d. Both

●●●

"d" is the best choice, although it may not always be possible. In any case, you would want to select the card physically, so it could be examined.

27 To permit selection of cards as they come out of the card reader (or punch), a limited number of separate compartments are available into which cards can fall. These are called "stacker pockets" and cards can be directed to specific ones by the stored program.

Where would the program send each transaction card that was in error? Why?

●●●

To a particular stacker pocket, so that it could manually be examined and the nature of the error determined.

28 There is one pocket into which all cards fall unless they are directed to one of the other pockets. To direct a card into a pocket other than the common one, the program would have to execute a(n) ....

●●●

instruction.

29 It might also be possible to print a message to the operator with more information about the error card. This would depend on the availability of on-line printing equipment. A message such as "TRANSACTION AMOUNT EXCESSIVE", or "NO MATCHING ACCOUNT NUMBER" could be stored with the program and printed as needed.

But just having the ability to select these cards and direct them to an alternate stacker pocket is a big help in identifying and correcting errors in input data.

Cards not in error would fall into (the same/a separate) .... stacker pocket from the one into which error cards fall.

●●●

a separate

30 What causes cards to fall into stacker pockets other than the one into which they normally would fall?

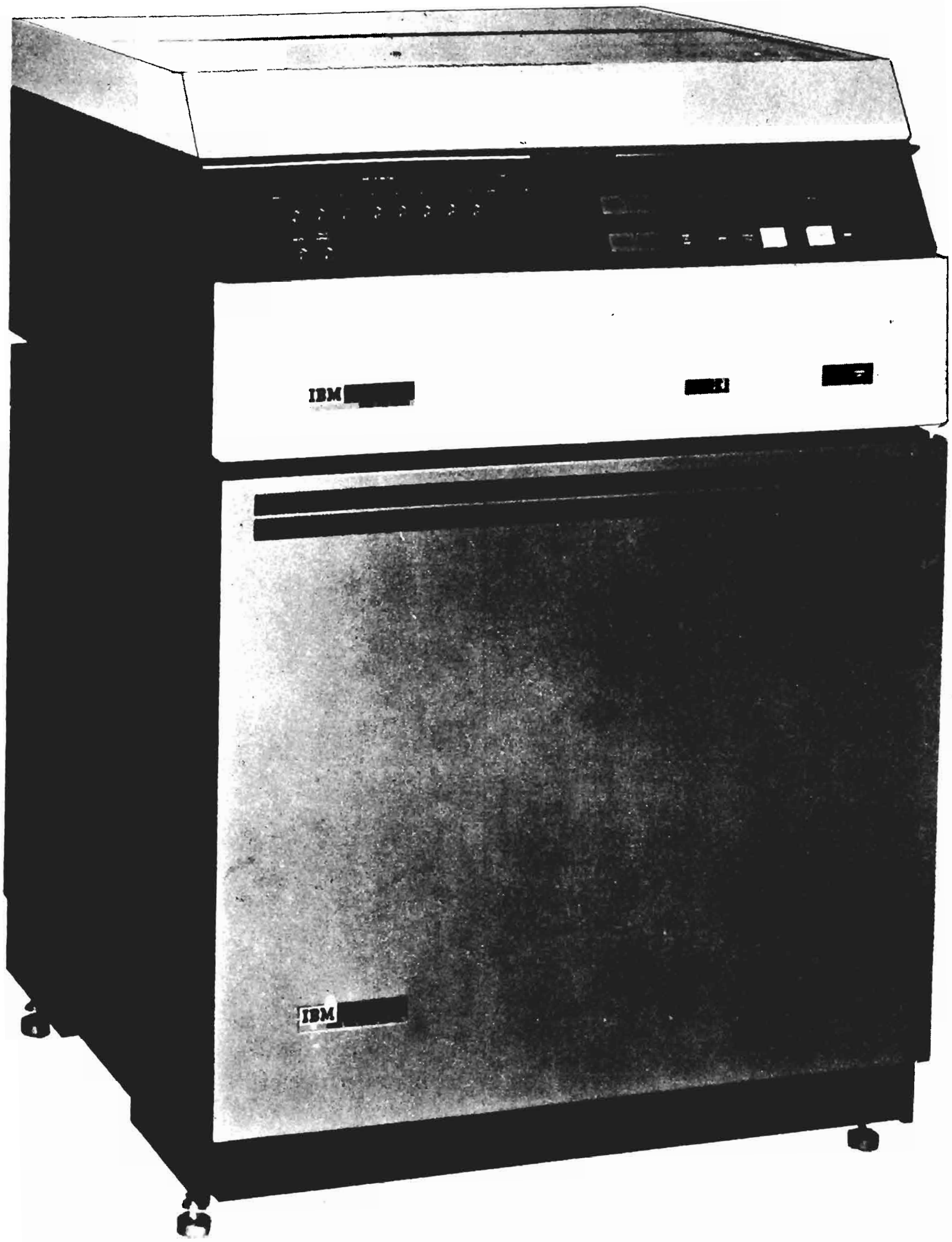
●●●

The program can issue an instruction selecting the card into an alternate pocket.

31 Suppose we are feeding two different types of input cards. After the run they are to be filed separately. How could we sort the cards during this run?

●●●

One type could be allowed to fall into the common pocket; the other could be selected into one of the special pockets. The program would include these instructions.



2671 PAPER TAPE READER and 2822 CONTROL UNIT

Can read 5-, 6-, 7-, or 8-channel tape
Maximum reading speed 1000 characters per second
Reading is optical - holes sensed by light

Figure 4.

32 Of course, as we mentioned, there are a limited number of stacker pockets available. The 2540 Card Reader-punch has two pockets for the read feed and two for the punch feed, plus one common pocket into which cards from either feed can be directed. This means that only a limited amount of selection can be done. A major selection of cards into groups would require a standard sorting operation.

●●●

33 In your own words, give the purpose of stacker pockets on S/360 card readers and punches, and the benefits we can get from their use.

●●●

They permit the selection of error cards into separate groups.

They permit the separation of cards from files that have just been read and/or punched, into different groups, depending on the number of stacker pockets available. This simplifies filing by reducing the need for sorting after the run.

34 Let's return to the problem of the Electric Credit Sales Company. We said that all customer transaction cards could be read into the system in about six hours. Is the reading time the only factor we must consider when reviewing the possibility of card input to the system?

●●●

No. Key punching time for creating input cards also is important.

35 If we use a decentralized system, there must be a card punch and someone to operate it at each location. If we use a centralized system we can get by with fewer card punches and operators, but the delay in transporting the source data, plus the punching time, could require perhaps 2 extra days between payment and updating the master records.

Paper Tape Readers

36 Many large systems achieve rapid updating using punched card input, but they need card punches and operators. Replacing a clerk with a card punch operator is not an economy. We probably can devise a better system using another method of input.

Figure 4 suggests an alternate method of input. It is the ....

●●●

paper tape reader.

37 The IBM2671 Paper Tape Reader can enter data into a computer at a speed of..........

●●●

1000 characters per second.

38 The 2671 reads characters in the paper tape by ....

●●●

light sensing.

39 The immediate problem, if paper tape is used, is getting the data punched into the tape.

Existing cash registers could be adapted or replaced so that paper tape is produced as a by-product of the over-the-counter transaction. But the cost of this solution might rule it out.

Special perforating equipment is available. It requires an operator to key in the data being punched, much the same as with a card punch. This approach might be used.

In both of the above cases there would be a delay in updating records at a central location because ....

●●●

the data would have to be physically transported to the central location.

40 Data in paper tape often is mailed to central computing locations. The tapes may be physically short, and are read individually by the paper tape reader. This requires someone to insert the short tapes into the reader. This kind of operation, if large amounts of data were involved, would be (efficient/inefficient) .... compared to the use of punched cards.

●●●

inefficient

41 There is no reason why large amounts of data could not be recorded in paper tape and transported physically to a central location. Let's consider this:

a. The 2671 reads paper tape at 60,000 characters per minute.

b. The 2540 reads cards at 80,000 characters per minute.

For volumes of data, to be recorded in an input medium at the branch office and physically transported to the central computing location, you probably would choose .... as the input medium.

●●●

punched cards

42 The reading speed of the input unit is an important factor in selecting the input data medium.

●●●

43 Transporting the paper tape physically to the central computer location is not necessarily the only way to get the data there. Paper tape is widely used in transmitting data by wire, over telephone or telegraph circuits.

Special equipment exists that permits an operator to record data on paper forms, much as a typist does, while at the same time punching the data into paper tape. Later the tape can be read by a reader attached to telephone (or telegraph) circuits and the data transmitted to remote locations. There it can be recorded again in paper tape.

This second tape could be read into a S/360 by ....

●●●

the 2671 Paper Tape Reader.

44 The fastest method of getting data to a central location would be by (mail/wire transmission) ....

●●●

wire transmission

45 Transmitting data to a central location and punching it into paper tape is one way to batch transactions so they can be processed all at the same time. As much information can be accumulated as is desired, then a computer run can be made to process all the accumulated data.

Name the I/O device that would be used to enter the data into a S/360.

●●●

The IBM 2671 Paper Tape Reader.

46 Batching transactions enables the operator to use the system for other jobs until it comes time to run one or more batches of transactions recorded in the paper tape. However, even this approach involves some delay in updating master records.

Can you suggest a way to transmit data by wire and to update each master record immediately, without batching?

●●●

Feed the data directly from the wire circuits into the computer.

47 Direct input of data to a computer from wire circuits is beginning to be widely used. You will learn more about this approach to data processing later in this course, in topics about teleprocessing and multiprogramming.

For now, let's wind up our discussion of paper tape as input.

There are several different codes in which data can be punched in paper tape. According to Figure 4, they are ....

●●●

5-, 6-, 7-, and 8-channel codes.

48 Each of these codes can be punched into paper tape and/or transmitted over wire circuits. The 2671 Paper Tape Reader can handle tape punched with which of these codes?

●●●

All can be handled by the 2671.

49 There are many considerations in selecting an input medium. Paper tape might be an excellent solution in some cases, and highly impractical in others. We have discussed only some basic considerations about paper tape.

What are the characteristics of the paper tape reader, including its maximum speed?

●●●

The paper tape reader reads data in the form of holes punched in continuous strips of paper tape, at a maximum speed of 1000 characters per second. Reading is by light sensing. Data can be punched in 5-,6-,7-, or 8-channel codes.

50 Keep in mind that if you have to keep an operator at each branch to punch the tape manually, you really haven't saved anything over punched cards. Unless there is special reason to transmit data by wire, the punched card approach would appear to be the more economical, because punched card data can be entered into the computer somewhat (slower/faster)...........than data in paper tape.

●●●

faster

51 There are, of course, other card or paper document devices we could use for input. Select some of these from the list in Figure 1.

●●●

The optical character reader, magnetic ink character reader, and optical mark page reader.

Magnetic Ink Character Reader

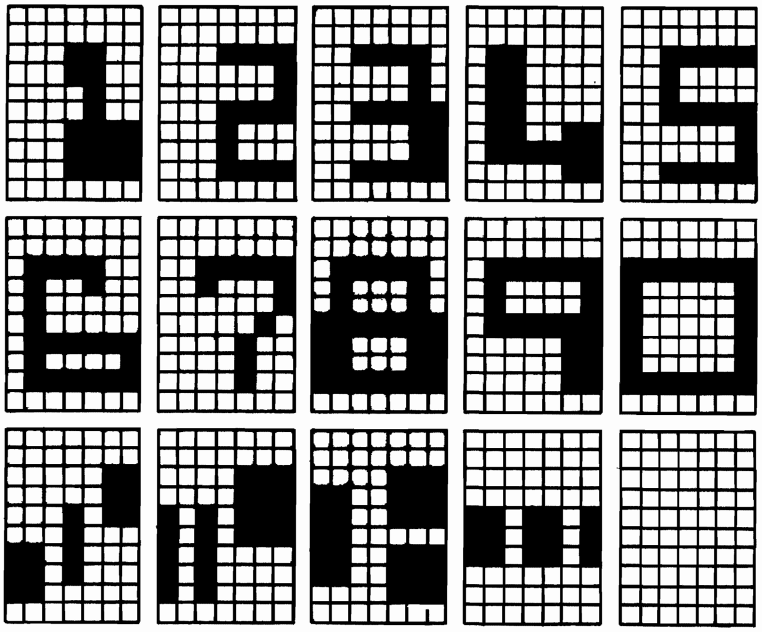
52 Each of these input units is designed to read some kind of a pattern imprinted on a paper document. We will look at each to see if it can be used in the Electric Credit Sales Company's computer system.

The magnetic ink character reader was designed primarily for bank applications. The process of reading documents printed in magnetic ink is called magnetic ink character recognition. MICR is a commonly used abbreviation for the process. The magnetic ink character reader usually is referred to as the MICR reader.

What is the form of data read by the MICR reader?

●●●

Documents printed with magnetic ink.



MAGNETIC INK CHARACTER SET

1 2 3 4 5

6 7 8 9 0

STANDARD CHARACTER SET

Figure 5.

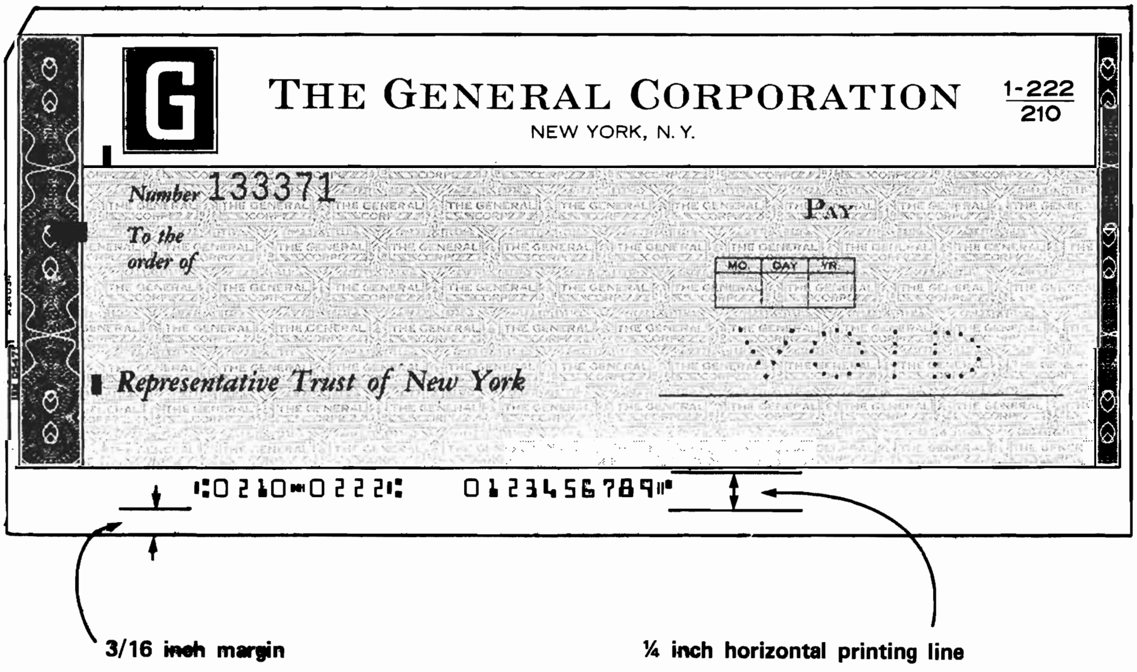


Figure 6.

53 MICR characters are both human readable and machine readable. However, each MICR character has a distinctive shape. Figure 5 compares MICR characters with ordinary printed characters.

Since you can identify each of the MICR characters they are .... readable.

●●●

human

54 Since the MICR characters can be read by a machine, they are both .... and .... readable.

●●●

human
machine

55 The distinctive shapes of MICR characters make certain their positive identification by the MICR reader.

MICR characters are printed in a special ink that can be magnetized. The magnetizing is done by the MICR reader just prior to the actual reading. At the reading station each MICR character is sensed by the reader as a distinctive magnetic pattern.

Can any character printed in magnetic ink be read by the MICR reader?

●●●

No. The characters read by the MICR reader have a predetermined shape.

56 What characteristics must a printing device have to print MICR characters?

●●●

Each character in the type set must have the shape prescribed for that character. Also, the ink with which the characters are printed must be magnetic ink.

57 Another consideration of MICR characters is the placement of characters on the document. There is a limited area on the document from which MICR data can be read. Figure 6 indicates this area to be ....

●●●

a ¼-inch horizontal band located 3/16-inch from the bottom of the document.

58 Would this limitation be a handicap to us in entering data into the Electric Credit Sales Company's computer?

●●●

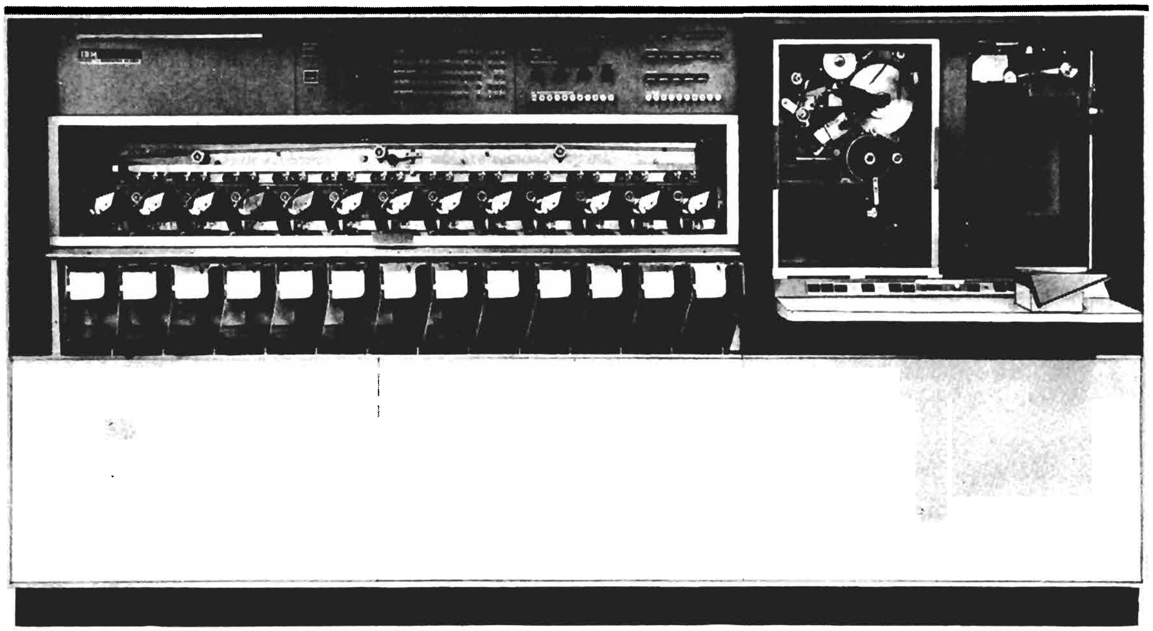
Yes. Having to record transaction data in only a small portion of an input document would be a totally unnecessary constraint, making our operation quite inefficient.

59 Looking at Figure 5, can you see any other drawback to the use of MICR documents as input to the Electric Credit Sales Company's computer?

●●●

Only numeric character, plus a few special character characters, can be identified by MICR readers. We might want to enter some alphabetic data from time to time.

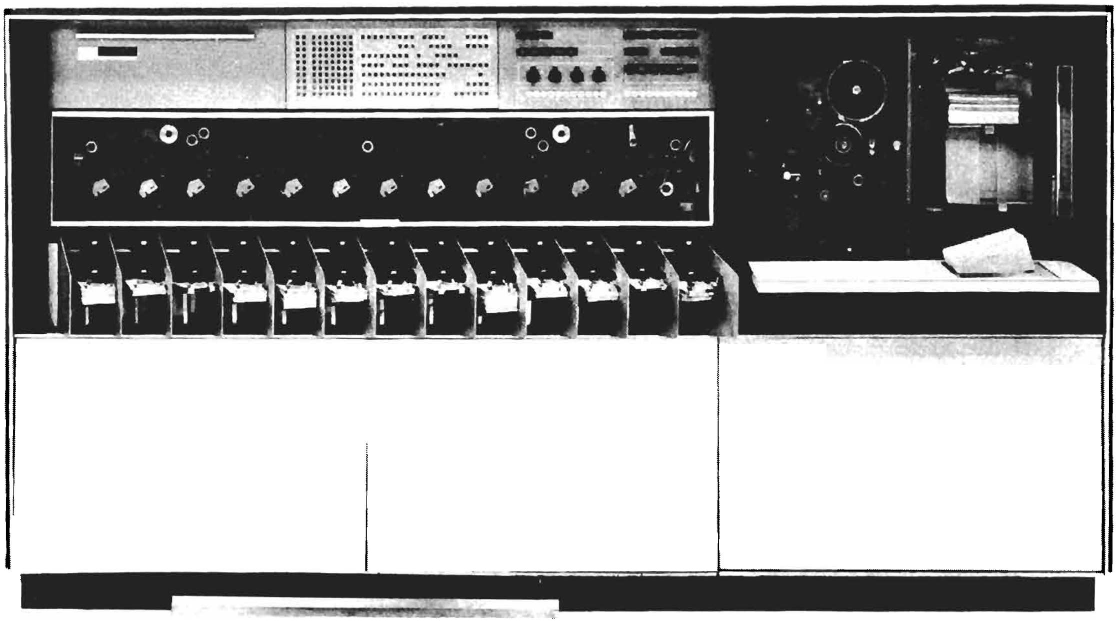
60 There are applications for which this restriction is not a handicap - in the banking applications for which MICR was developed, only a small amount of data is required to be in machine readable form. Checks, deposit slips, and withdrawal slips, imprinted with account numbers, routing codes, and transit codes, contain quite enough information to permit them to be read into a computer and processed.



13 Stacker Pockets

Reading Speed up to
950 Documents per minute

IBM 1412 MAGNETIC INK CHARACTER READER



13 Stacker Pockets

Reading Speed up to
1600 Documents per minute

IBM 1419 MAGNETIC INK CHARACTER READER

Figure 7.

Restrictions on the area in which MICR characters can be placed on a document (are/are not) .... a serious handicap in banking applications.

● ● ●

are not

61 Another advantage of MICR for the banking industry is that original documents (checks, for example) can be used for direct input of data to a computer. Billions of checks are processed annually. You can imagine the problem the banking industry would face if data from each check had to be transcribed to some other medium.

From what you have seen of bank checks, would you say that IBM MICR readers can handle checks of one size only?

● ● ●

No. IBM MICR readers can handle checks of various sizes, intermixed.

62 Figure 7 shows two models of IBM MICR readers. What significant difference exists between the two models?

● ● ●

The 1412 reads and sorts up to 950 documents per minute.
The 1419 reads and sorts up to 1600 documents per minute.

63 What feature do both models have in common?

● ● ●

Each has 13 stacker pockets.

64 We noted that IBM MICR readers can read and sort documents. You can deduce that the purpose of the stacker pockets on the readers is to ....

● ● ●

receive the documents after they have been read.

65 A check that has been read by a MICR reader could be sorted into (which) .... of the 13 stacker pockets.

● ● ●

any one

66 Separating checks into groups immediately after they have been read is one step in the process of distributing them to the banks on which they were drawn. The ability of MICR readers to .... speeds this separation process.

● ● ●

sort

67 Describe the characteristics of MICR documents and tell how and at what speed they are read by a magnetic ink character reader.

● ● ●

Numeric and a few special characters printed in special ink on documents of varying sizes are magnetized prior to reading, and sensed by a mechanism that recognizes the magnetic pattern they produce. Maximum reading speed is 1600 documents per minute.

68 MICR equipment was developed primarily for applications in the .... business.

● ● ●

banking

69 So it appears that the use of MICR documents as input to the Electric Credit Sales Company's computer would present many problems. But there are other input device types we have not yet considered.

•••

Optical Character Reader

70 We know that magnetic ink characters have distinctive shapes that are recognized by the MICR reader. Another reader that senses distinctively shaped characters is the optical character reader. You can deduce that this reader uses (magnetism/light) .... to scan and identify printed characters.

•••

light

71 Optical character readers use a mechanism involving light rays to examine and identify printed characters. To be sensed by an optical mechanism, would such characters have to be printed in magnetic ink?

•••

No

72 Optical characters don't have to be printed in magnetic ink; ordinary ink will do, so long as the contrast between the ink and the paper meets certain specifications. However, the characters themselves must have distinctive shapes. Here is an optical character set:

123456789TNCSZX⌿

Note that a few alphabetic characters, but not all, are included in this set.

The characters in this character set would be printed in (magnetic/ordinary).......... ink for reading by a(n) (optical/magnetic) .... mechanism.

•••

ordinary
optical

73 We noted that a few alphabetic characters are included in such a set. These characters are used as special symbols; they do not have their ordinary meaning as alphabetic characters. This means that an optical reader that can read only this set is a(n) (alphabetic/numeric) .... reader.

•••

numeric

74 Optical characters (do/do not)................ have distinctive shapes compared to ordinary printed characters.

•••

do

75 Since optical characters have distinctive shapes, it follows that the printing devices that print such characters must be equipped with optical character type fonts. Earlier we said that cash registers could be modified to print MICR characters. They could just as easily be equipped to print optical characters. Thus, one form of input to the central computer of the Electric Credit Sales Company could be the ...., printed in optical characters, from the cash registers.

•••

paper tapes

76 Each customer payment would be recorded on the cash register tape as the transaction occurred. The cash register could be caused to print an account number, date, and other information that would be needed to update the master record.

Reader Number	Reads Printed Numeric (1428 Font)	Reads Machine Printed Alphabetic, Numeric, & Special Characters	Reads Handwritten Numeric	Reads Optical Marks	Reads Printing From 1403 Printer	Reads Printing From Typewriter	Number of Stackers	Document Form			Document Size	Example of Reading Speed
								Printed Tape	Cards	Sheets		
1285	x				x	x		x			1 5/16" to 3 1/2" Wide	2031 ± 5% Lines Per Minute
1287	x		x	x	x	x	3	x	x	x	5.91" x 9.00" max for sheets	485 Documents Per Minute Sized 2.25" x 3.00" with 30 Machine Printed Characters Per Line
1418	x			x	x	x	3 or 13			x	3 2/3" x 8 3/4" max	420 DPM for Sheets 5 7/8" wide 300 DPM for Sheets 8 3/4" wide
1428	x	x		x	x	x	3 or 13			x	Up to 3 2/3" x 8 3/4" max	288 to 400 DPM for Sheets 5 7/8" to 8 3/4" wide

NOTE: Reading speeds vary with document size and number of characters per unit area on document

NOTE: Character placement on document requires only that margins and vertical and horizontal spacing minimums between lines and characters be observed, as specified for each type of OCR document.

Figure 8.

From Figure 8, select the IBM optical readers that could read input in this form.

● ● ●

The 1285 and 1287 can read paper tape.

77 Rolls of cash register tape could be transported physically to the central computer location and entered directly into the optical reader. Thus the source document would become direct input to the system.

Which steps in a punched card system would this eliminate?

● ● ●

Punching and verifying the cards before reading them into the computer.

78 Note that the cash register would record the transactions on the tape in random order; that is, as they occurred. To have the most efficient processing, on which of these devices should the master records be kept, and why?

a. magnetic tape
b. magnetic disks

● ● ●

Magnetic disks, because they permit random access to the records they contain. Since the transactions are in random order on the tally roll, we would have to perform a time-consuming search on magnetic tape for the particular master record we wanted. Disks permit the computer to go directly to the wanted record, without intervening search over other records.

79 We said that optical character readers can read data printed by cash registers. Figure 8 indicates two other devices that can print machine-readable characters. They are ....

● ● ●

the 1403 Printer and a typewriter.

80 Optical characters have distinctive shapes. Could a 1403 Printer, or a typewriter, print optical characters with MICR type?

● ● ●

No. It would have to have optical type.

81 As we will see later, the 1403 is a high-speed printer. It can produce large volumes of printed data in relatively short periods of time. It can print numeric, alphabetic, and a range of special characters. Typewriters print these characters too. We can readily imagine that there will be times when it would be convenient to read alphabetic characters as such, into a computer, with an optical reader. From Figure 8, you can determine whether or not there is an optical reader capable of reading alphabetic, numeric, and special characters.

● ● ●

The IBM 1428 reads all upper case alphabetic characters, as well as numeric and special characters.

82 For the Electric Credit Sales Company job, we probably could get along very well with numeric data only. The account number, date, and amount of payment appear to be enough data to update the master record. But if alphabetic data as well is to be entered, the 1428 reader will do the job.

All IBM optical character readers can read machine printed characters (from cash registers, the 1403 Printer, or a typewriter). Figure 9 shows another type of optical character. It is ....

● ● ●

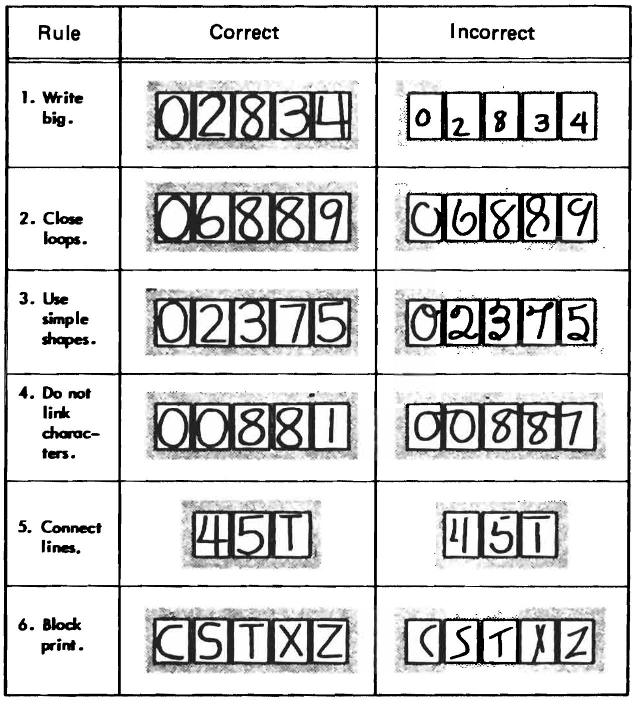
handwritten numeric.

83 Handwritten numeric characters can be read by the IBM .... (See Figure 8.)

● ● ●

1287

HANDWRITING RULES FOR HANDWRITTEN NUMERIC OPTICAL CHARACTERS



Rule	Correct	Incorrect
1. Write big.	02834	02834
2. Close loops.	06889	06889
3. Use simple shapes.	02375	02375
4. Do not link characters.	00881	00881
5. Connect lines.	45T	45T
6. Block print.	CSTXZ	CSTXZ

Figure 9.

OPTICAL MARK STYLES

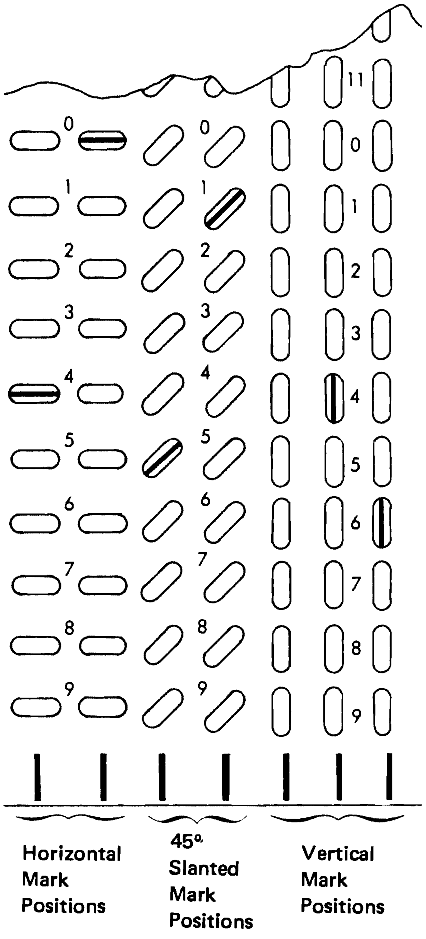


Figure 10.

84 Can just any handwritten numeric character be read by the 1287? (See Figure 9.)

●●●

Not necessarily. It must be formed according to the rules shown in Figure 9.

85 Handwritten optical characters have many interesting applications. Sales checks, merchandise orders, utility bills, telephone toll tickets, and other handwritten source documents can be processed by the 1287.

●●●

86 Optical documents vary in size and shape. As noted previously, they can be paper tape, or cut-form documents. The 1287 can read another type of document. It is .... (See Figure 8.)

●●●

cards.

87 The speed of optical readers depends not only on the type of document (tape, cards, or sheets), but on the number of characters in a given area on the document. For example, the 1287 can read 3-inch wide tape, with four 10-character lines per inch, at a speed of 2250 lines per minute.

An example of reading speed of the 1287 for 2¼" x 3" paper documents is given in Figure 8. The speed is ....

●●●

485 documents per minute.

88 It is important to note that the documents being read do not have to be all of the same size. They can be of varying sizes, so long as they do not exceed minimum and maximum allowable sizes.

●●●

89 How many characters would the 1287 enter into the computer in one minute if it were operating at the speed shown in Figure 8? Assume 12 lines of printing on each document.

●●●

174,600

90 There are maximum and minimum sizes for optical character documents. Are there restrictions on the placement of characters on the document?

●●●

Prescribed margins must be left, and vertical and horizontal spacing between characters must not exceed prescribed minimums. (See Figure 8.)

91 On what does the reading speed of optical character readers depend?

●●●

Document size and number of characters in a given area on the document. (See Figure 8.)

92 Figure 10 shows another form of data that can be read by some IBM optical readers. It is ....

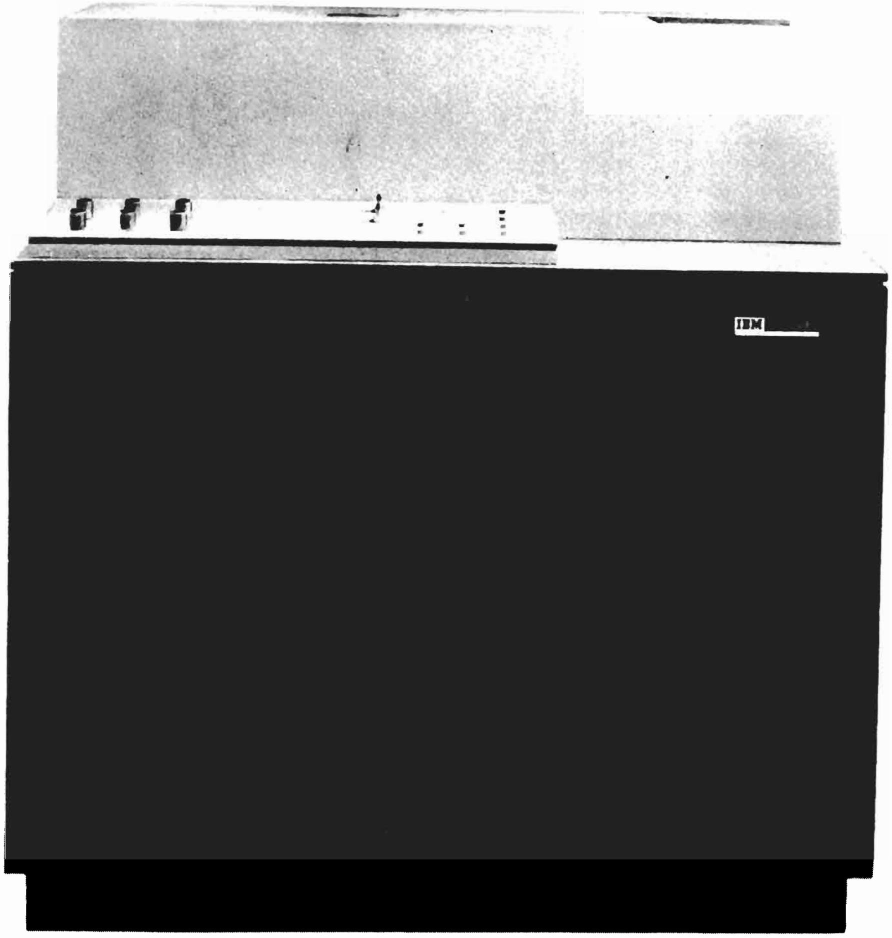
●●●

optical marks.

93 We will learn more about optical marks shortly. From Figure 8, select those optical readers that can read them.

●●●

1287, 1418, 1428



1231 OPTICAL MARK PAGE READER

Reads up to 2000 8 1/2" x 11" sheets/hour

Figure 11.

94 Describe the forms of data that can be read by IBM optical character readers and give the factors that determine maximum reading speeds.

●●●

Depending on the unit in question, optical readers can read machine-printed alphabetic, numeric, and special characters; handwritten numeric characters; and optical marks, recorded on rolls of paper tape, cards, or cut-form documents. Maximum speed varies with document size and number of characters in a given area.

95 It is worth pointing out again that there would be no extra work for the branch office staff in preparing the input to the central computer. The tally rolls are there anyway, and need only be mailed to the central location.

Optical character readers are used for many applications, as previously noted. Merchandise orders, sales checks, telephone toll tickets, and other original business documents can easily be entered via the optical character reader.

So an optical character reader appears quite feasible as an input device to the Electric Credit Sales Company's centralized computer. Considering the other I/O devices we have studied, would we be likely to select it for decentralized operations; that is, for a computer in each branch office?

●●●

Probably not, unless there were many other applications calling for optical characters. Punched card input would probably be clearly indicated for decentralized operations.

96 Actually, there are still other possibilities. They involve teleprocessing - transmitting data by wire. These will be discussed later in the course.

Optical Mark Page Reader

Figure 11 shows a picture of the IBM ....

●●●

1231 Optical Mark Page Reader.

97 Look again at Figure 10. Optical marks are

a. short straight lines located at random on an 8½" x 11" page.

b. short curved lines located in prescribed places on the page.

c. short straight pencil marks in prescribed locations on the page.

●●●

c. short straight pencil marks in prescribed locations on the page.

98 Optical marks can represent all or parts of data items. For example, an account number of more than one digit would require several mark positions, but the number of dozens of an inventory item could be represented by one mark. In either case, which indicates what the mark means? (See Figure 10.)

a. The color of the mark.

b. The position of the mark on the page.

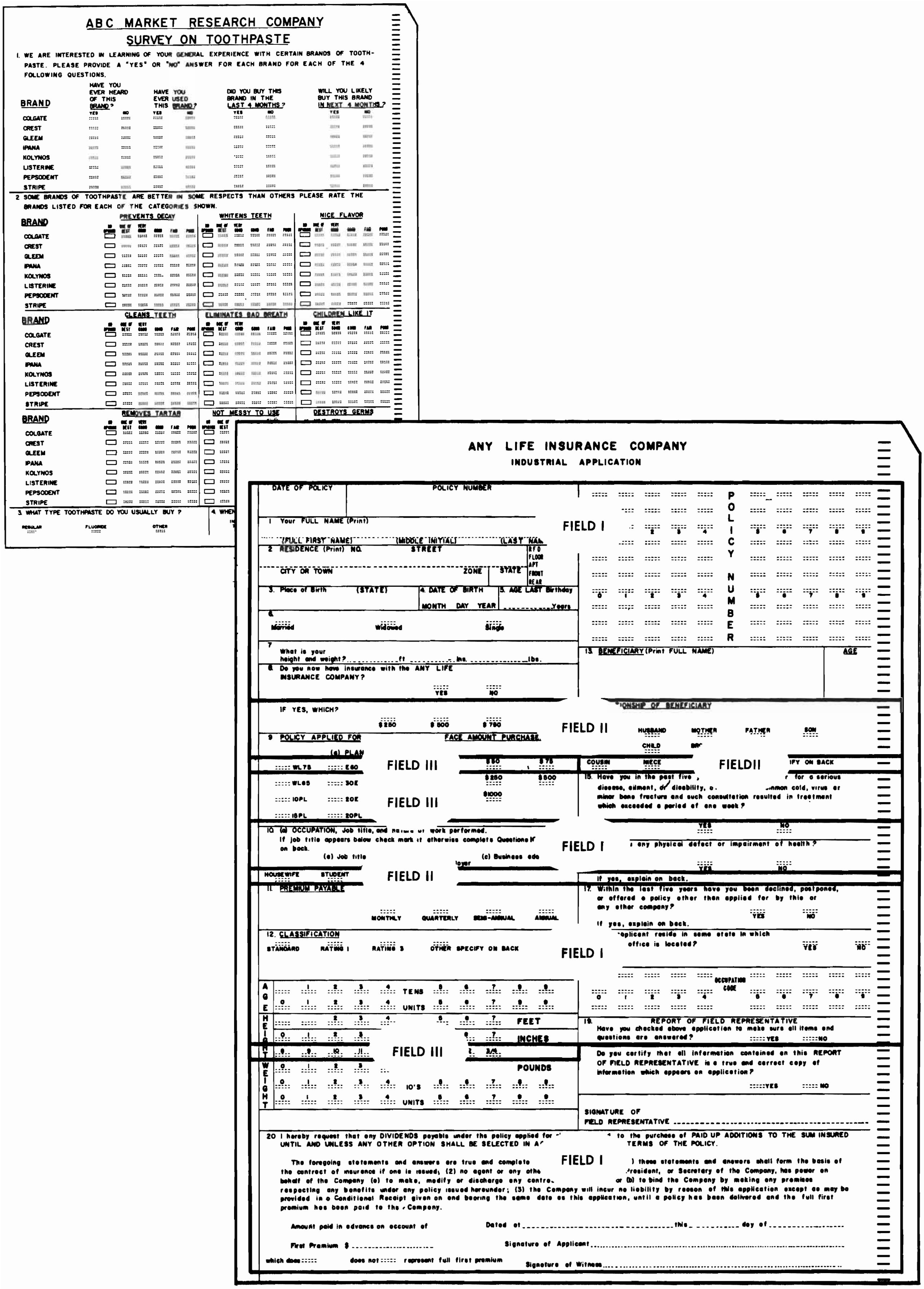
c. The spacing between marks.

●●●

b. The position of the mark on the page.

99 Optical marks are widely used in scoring tests. The test offers several possible answers to each question. There is a mark position on the page for each possible answer. The test-taker marks the position corresponding to the answer he has selected. Thus the position of the mark indicates the particular data item represented by the mark.

●●●



ABC MARKET RESEARCH COMPANY

SURVEY ON TOOTHPASTE

1. WE ARE INTERESTED IN LEARNING OF YOUR GENERAL EXPERIENCE WITH CERTAIN BRANDS OF TOOTHPASTE. PLEASE PROVIDE A "YES" OR "NO" ANSWER FOR EACH BRAND FOR EACH OF THE 4 FOLLOWING QUESTIONS.

BRAND	HAVE YOU EVER HEARD OF THIS BRAND?		HAVE YOU EVER USED THIS BRAND?		DID YOU BUY THIS BRAND IN THE LAST 4 MONTHS?		WILL YOU LIKELY BUY THIS BRAND IN NEXT 4 MONTHS?	
	YES	NO	YES	NO	YES	NO	YES	NO
COLGATE	:::::	:::::	:::::	:::::	:::::	:::::	:::::	:::::
CREST	:::::	:::::	:::::	:::::	:::::	:::::	:::::	:::::
GLEEM	:::::	:::::	:::::	:::::	:::::	:::::	:::::	:::::
IPANA	:::::	:::::	:::::	:::::	:::::	:::::	:::::	:::::
KOLYNOS	:::::	:::::	:::::	:::::	:::::	:::::	:::::	:::::
LISTERINE	:::::	:::::	:::::	:::::	:::::	:::::	:::::	:::::
PEPSODENT	:::::	:::::	:::::	:::::	:::::	:::::	:::::	:::::
STRIPE	:::::	:::::	:::::	:::::	:::::	:::::	:::::	:::::

2 SOME BRANDS OF TOOTHPASTE ARE BETTER IN SOME RESPECTS THAN OTHERS PLEASE RATE THE BRANDS LISTED FOR EACH OF THE CATEGORIES SHOWN.

BRAND	PREVENTS DECAY						WHITENS TEETH						NICE FLAVOR					
	NO OPINION	ONE OF BEST	VERY GOOD	GOOD	FAIR	POOR	NO OPINION	ONE OF BEST	VERY GOOD	GOOD	FAIR	POOR	NO OPINION	ONE OF BEST	VERY GOOD	GOOD	FAIR	POOR
COLGATE	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::
CREST	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::
GLEEM	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::
IPANA	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::
KOLYNOS	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::
LISTERINE	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::
PEPSODENT	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::
STRIPE	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::

BRAND	CLEANS TEETH						ELIMINATES BAD BREATH						CHILDREN LIKE IT					
	NO OPINION	ONE OF BEST	VERY GOOD	GOOD	FAIR	POOR	NO OPINION	ONE OF BEST	VERY GOOD	GOOD	FAIR	POOR	NO OPINION	ONE OF BEST	VERY GOOD	GOOD	FAIR	POOR
COLGATE	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::
CREST	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::
GLEEM	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::
IPANA	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::
KOLYNOS	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::
LISTERINE	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::
PEPSODENT	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::
STRIPE	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::	□	:::::	:::::	:::::	:::::	:::::

BRAND	REMOVES TARTAR						NOT MESSY TO USE		DESTROYS GERMS
	NO OPINION	ONE OF BEST	VERY GOOD	GOOD	FAIR	POOR	NO OPINION	ONE OF BEST	
COLGATE	□	:::::	:::::	:::::	:::::	:::::	□	:::::	
CREST	□	:::::	:::::	:::::	:::::	:::::	□	:::::	
GLEEM	□	:::::	:::::	:::::	:::::	:::::	□	:::::	
IPANA	□	:::::	:::::	:::::	:::::	:::::	□	:::::	
KOLYNOS	□	:::::	:::::	:::::	:::::	:::::	□	:::::	
LISTERINE	□	:::::	:::::	:::::	:::::	:::::	□	:::::	
PEPSODENT	□	:::::	:::::	:::::	:::::	:::::	□	:::::	
STRIPE	□	:::::	:::::	:::::	:::::	:::::	□	:::::	

3. WHAT TYPE TOOTHPASTE DO YOU USUALLY BUY ?

REGULAR ::::: FLUORIDE ::::: OTHER :::::

4. WHEN

ANY LIFE INSURANCE COMPANY

INDUSTRIAL APPLICATION

DATE OF POLICY POLICY NUMBER

P O L I C Y N U M B E R

2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

FIELD I

1 Your FULL NAME (Print)

(FULL FIRST NAME) (MIDDLE INITIAL) (LAST NAM

2 RESIDENCE (Print) NO. STREET RFD FLOOR APT FRONT REAR

CITY OR TOWN ZONE STATE

3. Place of Birth (STATE) 4. DATE OF BIRTH MONTH DAY YEAR 5. AGE LAST Birthday ....Years

6. ::::: Married ::::: Widowed ::::: Single

7 What is your height and weight? ....ft ....ins. ....lbs.

8 Do you now have insurance with the ANY LIFE INSURANCE COMPANY? ::::: YES ::::: NO

13. BENEFICIARY (Print FULL NAME) AGE

IF YES, WHICH? ::::: $250 ::::: $500 ::::: $750

FIELD II

'IONSHIP OF BENEFICIARY

HUSBAND ::::: MOTHER ::::: FATHER ::::: SON ::::: CHILD ::::: BR'

9 POLICY APPLIED FOR FACE AMOUNT PURCHASE.

(a) PLAN

::::: WL75 ::::: E60 FIELD III ::::: $50 ::::: $75

COUSIN ::::: NIECE ::::: FIELDII IFY ON BACK

::::: WL65 ::::: 30E ::::: $250 ::::: $500

::::: 10PL ::::: 20E FIELD III ::::: $1000

::::: 15PL ::::: 20PL

15. Have you in the past five , r for a serious disease, ailment, or disability, o. .mmon cold, virus or minor bone fracture and such consultation resulted in treatment which exceeded a period of one week? ::::: YES ::::: NO

10. (a) OCCUPATION, Job title, and nature of work performed. If job title appears below check mark it otherwise complete Questions If on back.

(a) Job title (c) Business ado loyer

FIELD I

, any physical defect or impairment of health? ::::: YES ::::: NO

HOUSEWIFE ::::: STUDENT ::::: FIELD II

If yes, explain on back.

11. PREMIUM PAYABLE ::::: MONTHLY ::::: QUARTERLY ::::: SEMI-ANNUAL ::::: ANNUAL

17. Within the last five years have you been declined, postponed, or offered a policy other than applied for by this or any other company? ::::: YES ::::: NO

If yes, explain on back.

12. CLASSIFICATION ::::: STANDARD ::::: RATING 1 ::::: RATING 3 ::::: OTHER SPECIFY ON BACK

FIELD I

'pplicant reside in same state in which office is located? ::::: YES ::::: NO

OCCUPATION CODE 0 1 2 3 4 5 6 7 8 9

AGE		1	2	3	4	TENS	5	6	7	8	9
	0	1	2	3	4	UNITS	5	6	7	8	9
HEIGHT			2	3	4		5	6	7	FEET	
	0	1	2	3			6	7	INCHES		
	8	9	10	11	FIELD III		1/2	3/4			
WEIGHT	0	1	2	3					POUNDS		
	0	1	2	3	4	10'S	5	6	7	8	9
	0	1	2	3	4	UNITS	5	6	7	8	9

19. REPORT OF FIELD REPRESENTATIVE

Have you checked above application to make sure all items and questions are answered? :::::YES :::::NO

Do you certify that all information contained on this REPORT OF FIELD REPRESENTATIVE is a true and correct copy of information which appears on application? :::::YES ::::: NO

SIGNATURE OF FIELD REPRESENTATIVE ....

20 I hereby request that any DIVIDENDS payable under the policy applied for -' ' to the purchase of PAID UP ADDITIONS TO THE SUM INSURED UNTIL AND UNLESS ANY OTHER OPTION SHALL BE SELECTED IN A' TERMS OF THE POLICY.

FIELD I

The foregoing statements and answers are true and complete) these statements and answers shall form the basis of the contract of insurance if one is issued; (2) no agent or any othe 'resident, or Secretary of the Company, has power on behalf of the Company (a) to make, modify or discharge any contra. or (b) to bind the Company by making any promises respecting any benefits under any policy issued hereunder; (3) the Company will incur no liability by reason of this application except as may be provided in a Conditional Receipt given on and bearing the same date as this application, until a policy has been delivered and the full first premium has been paid to the Company.

Amount paid in advance on account of Dated at ....this.......... day of ....

First Premium $ .... Signature of Applicant ....

which does ::::: does not ::::: represent full first premium Signature of Witness ....

Figure 12.

100 Figure 12 shows two other application areas in which optical mark documents are used. They are ....

● ● ●

market analysis
insurance

101 The value of a data item is indicated by the position(s) of the mark(s) on the page. For example, each mark on the market research form in Figure 12 rates a product by one category; thus each mark represents (all/part) .... of a data item. On the insurance form in Figure 12, each mark in field 1 represents (all/part) .... of the policy number.

● ● ●

all
part

102 There are many other applications in which optical marks are used. In every case, the item of data is represented by ....

● ● ●

the position of the mark(s) on the document.

103 The 1231 Optical Mark Page Reader reads 8½" x 11" data sheets containing 1000 optical marks. Each set of ten marks is translated into two bytes of storage. At a reading speed of 2000 sheets per hour, how many bytes of storage can be filled in an hour?

● ● ●

400,000 ((1000 x 2000) ÷ 10) x 2

104 The size of the optical mark page is ....

● ● ●

8½" x 11".

105 The maximum number of marks that can be put on one page is ....

● ● ●

1000.

106 Assume a 1231 Optical Mark Page Reader is operating at maximum speed. If each data sheet is half full of marks and each mark represents one item of data, how many individual data items are being read into the computer in an hour.? How many if the data sheet is completely full?

● ● ●

1,000,000
2,000,000

107 Describe the form of data read by the optical mark page reader and the medium on which the data is recorded.

● ● ●

Short, straight pencil marks, recorded in predetermined locations on 8½" x 11" data sheets.

108 The sensing mechanism used by the optical mark page reader involves:

a, magnetism
b. brushes in an electric circuit
c. light scanning

● ● ●

c.

109 Match the items in these lists:

a. MICR
b. OCR
c. OMP

1. Machine printed alphabetic, numeric, and special characters, printed with a special type font, or handwritten numeric characters, or optical marks, sensed by a scanning mechanism involving light.
2. Numeric and a few special characters, printed with type of a special font, read magnetically.
3. Short marks in prescribed locations on a page, sensed by a scanning mechanism involving light.

●●●

a. 2
b. 1
c. 3

110 It might be very efficient to have an optical mark page reader as input if we had decentralized operations, that is, a computer in each branch office. The clerk could mark the information on the sheet each time a transaction occurred. At an appropriate time the batched sheets could be run to update the master records.

However, overall input time is slower than some other media we have studied. A card reader at 1000 cards per minute can enter data at a rate of 4,800,000 bytes per hour. The OMP Reader can enter data at a maximum rate of 400,000 bytes per hour. This difference could be very significant if volumes of data are quite large.

Another consideration is the amount of wasted space on each mark page. It would not normally require a full page of marks to represent one transaction. So there would be some unused space. Of course, we could utilize this space by ....

●●●

putting more than one transaction on a page.

111 For a centralized computer, we still would face the problem of transporting the data to the central location. Here again, when all factors are considered, we might decide on the OMP Reader, or we might select one of the other input devices we have studied.

There are, as we previously said, many factors that go into the selection of I/O and other units for a computer system. It is entirely possible that a MICR or OMP Reader would serve satisfactorily although they were designed primarily for special purposes. Cost comparisons among possible units, plus overall applicability to this and other applications that would be run on the computer, and many other factors, would have to be considered. The final choice of devices for a computer system very often represents a compromise.

●●●

Printers

112 We have looked at card readers and punches, and the MICR, OCR, and OMP Readers. Except for the punches, all are input devices. Now let's look at one of the most widely used output devices - the printer.

Output in the form of punched cards, magnetic tape, and magnetic disks normally become input to future computer runs. Printed output normally is for human consumption.

●●●

113 Would printed output ever become input to a computer?

●●●

Yes. MICR and OCR documents can be machine printed, as we previously pointed out.

114 Printed output can become input to the computer, if the document is the proper size and the printer is equipped with the proper type set. In the case of MICR printing, of course, the ink must be magnetic.

In either case, printed output also is human readable. MICR and OCR characters, as well as standard type, are designed for complete legibility.

The printer considerations we will discuss apply to both human and machine readable printing. Insofar as the CPU is concerned, there is

EXTENDED BINARY-CODED-DECIMAL INTERCHANGE CODE (EBCDIC)
showing bit positions, bit patterns, graphic characters for EBCDIC

		00				01			
		00	01	10	11	00	01	10	11
Bit Positions 4, 5, 6, 7	0000								
	0001								
	0010								
	0011								
	0100								
	0101								
	0110								
	0111								
	1000								

←Bit Positions 0, 1
←Bit Positions 2, 3

		10				11			
		00	01	10	11	00	01	10	11
Bit Positions 4, 5, 6, 7	0000								0
	0001	a	j			A	J		1
	0010	b	k	s		B	K	S	2
	0011	c	l	t		C	L	T	3
	0100	d	m	u		D	M	U	4
	0101	e	n	v		E	N	V	5
	0110	f	o	w		F	O	W	6
	0111	g	p	x		G	P	X	7
	1000	h	q	y		H	Q	Y	8
	1001	i	r	z		I	R	Z	9

←Bit Positions 0, 1
←Bit Positions 2, 3

		00				01			
		00	01	10	11	00	01	10	11
Bit Positions 4, 5, 6, 7	1001								
	1010					¢	!		:
	1011					.	$	,	#
	1100					<	*	%	@
	1101					(	)	_	'
	1110					+	;	>	=
	1111					|	¬	?	"

←Bit Positions 0, 1
←Bit Positions 2, 3

		10				11			
		00	01	10	11	00	01	10	11
Bit Positions 4, 5, 6, 7									
	1010								
	1011								
	1100								
	1101								
	1110								
	1111								

←Bit Positions 0, 1
←Bit Positions 2, 3

Figure 13.

(a difference/no difference) .... between a printer with OCR or MICR type faces, and one without.

●●●

no difference

115 This is a byte of S/360 storage:

```
 0       7
|xxxx'xxxx|
```

Each "x" represents a bit position. The byte can contain a maximum of (how many?) .... bit patterns.

●●●

256

116 Each bit pattern in the byte can represent a character of some kind (alphabetic, numeric, special). This is the bit pattern for the letter "A":

```
|1100 0001|
```

and this for the digit "7":

```
|1111 0111|
```

and this for the special character "#"

```
|0111 1011|
```

How many different characters can be represented by one byte of S/360 storage?

●●●

256

117 Actually, in normal usage, there are not 256 different printed characters. Even if the character set includes both upper and lower case alphabetic characters, along with the digits 0-9 and a large variety of special characters, there are many bit patterns within the byte that are not used.

Figure 13 shows you that there are many S/360 bit patterns with no assigned character. If such a bit pattern exists in a data field being printed, what do you think will print for that byte?

●●●

Nothing

118 For a bit pattern to print on a S/360 printer, there are three considerations:

a. There must be a character assigned to the bit pattern.

b. The bit pattern must be present in a byte of the output field in storage.

c. The character must be present in the type set currently installed on the printer.

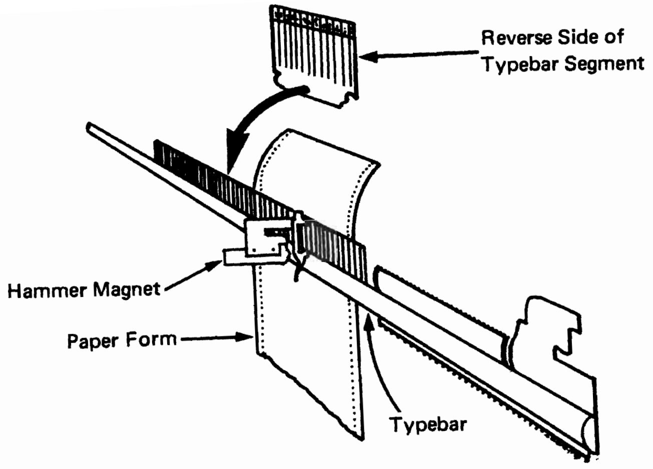
Suppose you want to assign some unused bit patterns to some of the letters of the Greek alphabet. (Greek letters commonly are used in mathematical notation.) If your printer does not have Greek letters in its type set you must ....

●●●

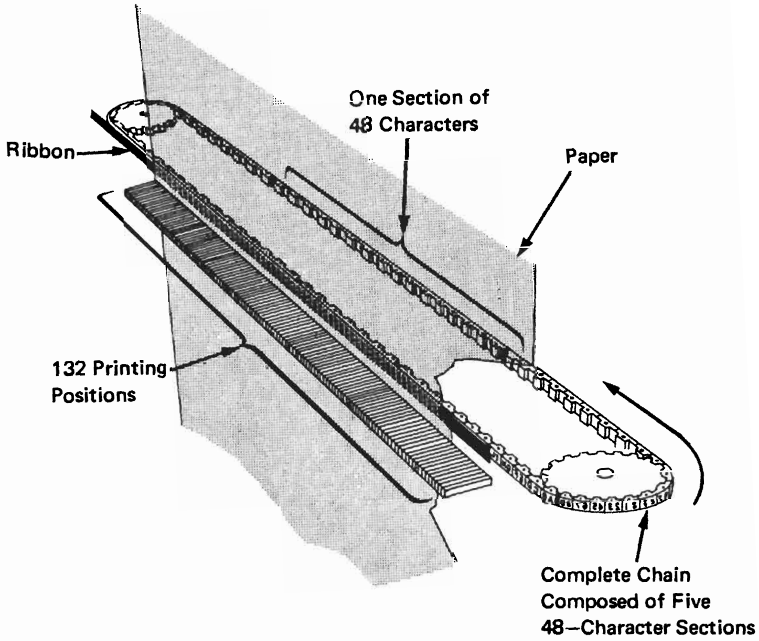
change to a type set that includes Greek letters.

119 The type elements on S/360 printers can contain any characters the user desires. Commonly used characters such as the English alphabet, the digits 0-9, and special characters such as the dollar sign ($), the comma (,), the asterisk (*), and so on, are, of course, standard and readily available. Non-standard characters may have to be made up specially.

The assignment of bit patterns to non-standard characters also is optional with the user; that is, he can designate any bit pattern he desires to represent the special character. However, S/360 printers require a timing relationship between the type element on the printer and the circuits in the computer. Through this timing relationship



1443 PRINTER MECHANISM SCHEMATIC



1403 PRINTING MECHANISM SCHEMATIC

Figure 14.

the computer knows which type faces are in position for printing, and at which printing positions they are. Arbitrary assignment of bit patterns to non-standard characters probably will require that the printing element be made up specially. This can be expensive.

What three conditions must be met for a character to print?

●●●

There must be a bit pattern assigned to the character.
The bit pattern must be present in the output field.
The printer type set must include the character.

120 The type element of the printer is that device that contains all the characters in the type set, and presents them at the printing station so they can be imprinted on the paper.

Figure 14 shows two types of printing mechanisms used by S/360 printers. The printer uses one or the other, depending on the printer in question. For example, the 1443 Printer uses a type ...., the 1403 a type ....

●●●

bar
chain.

121 Both the bar and the chain are simply strings of type faces that are moved along the printing line on the form. The chain runs at high speed in a continuous loop; the bar oscillates back and forth at high speed. The purpose of each is the same: to bring the type faces in the character set into position for printing.

As the type element positions a line of type faces along the printing line, the channel checks each byte whose contents are scheduled to print on the line. If the character in a byte is matched by the type face at that position, printing occurs. If no match is found, the type element positions another type face and again a comparison is made. The entire line will not have been printed until all bytes have been matched by type faces.

Which is true?

a. All characters on a line of printing are printed simultaneously.
b. Some of the characters on a line of printing are printed simultaneously.

●●●

b.

122 For engineering reasons, which we will not discuss, IBM printing chains have 240 type faces, while printing bars have either 120 or 144 type faces.

Assume we want to print a 48-character set composed of 26 alphabetic characters, 10 digits, and 12 special characters. Our printer has a chain. Since the chain has places for 240 type faces, what can we do with the extra ones? (Think for a moment about how printing occurs.)

●●●

Repeat the character set as many times as possible on the type element.

123 A 48-character set can be repeated (how many?) .... times on a chain.

●●●

5

124 Repeating the character set speeds up printing. (Think what would happen if we were ready to print a line and the only character set on the chain was on the non-printing side of the loop.) It simply means that we can get more type faces to printing positions in a given unit of printing time.

Although the type bar has fewer type faces (144 maximum) the principle is the same. The character set is repeated as many times as possible.

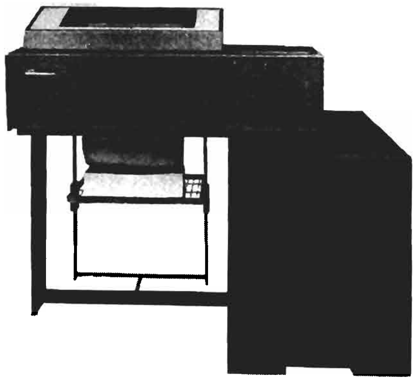
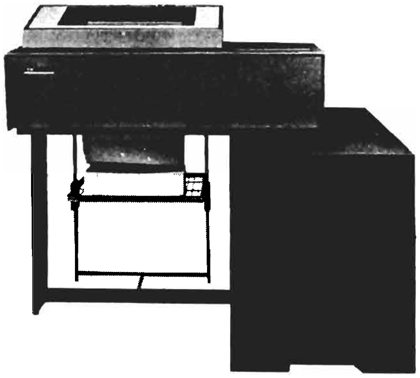
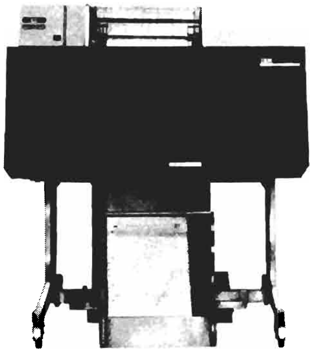
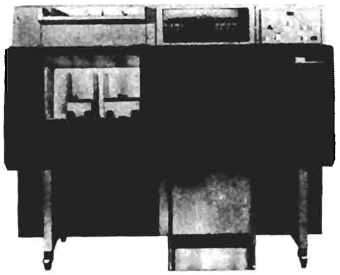
S/360 PRINTERS	
 1443	120 or 144 print positions Speed depends on character set 13/39/52/63 character set gives 600/300/240/200 lines per minute
 1445	113 print positions As 1443 but can print magnetic characters Speed depends on character set 14/42/56 character set gives 525/240/190 lines per minute
 1403	M2 - 132 print positions, 600 lines per minute maximum speed M7 - 120 print positions, 600 lines per minute maximum speed N1 - 132 print positions, 1100 lines per minute maximum speed 48 character set is standard Can use universal character set
 1404	Can print on cards as well as paper Maximum speed - cards - 800 CPM Maximum speed - paper - 600 LPM Optional feature allows reading of 30 columns of card 132 print positions

Figure 15.

The oscillating motion of the bar presents the different type faces at the printing positions. Printing occurs when (your own words) ....

●●●

the contents of the byte to be printed at a given position match the type face currently at that printing position.

125 If printing speed is increased by repeating the character set on the type element, what do you think will happen if we reduce the size of the character set?

●●●

Printing speed will increase.

126 Reducing the size of the character set enables us to get more sets on a given length of chain, or bar. This will increase printing speed for two reasons:

a. Since there are fewer characters to print, the bytes of output data do not have to wait so long for a matching type face to be positioned.

b. Since there are more character sets per type element, a larger number of any one character can be printed at one time.

You can deduce that the principal factor that influences printing speeds for any given printer is ....

●●●

the size of character set being printed.

127 A numeric character set will print (faster/slower) .... than an alphameric set. Why?

●●●

faster

Each printing position can be exposed to all the characters in the set in a shorter period of time.

More of any given character can be printed at one time.

128 As we pointed out earlier, not all characters are printed simultaneously. The contents of a given byte may or may not be matched by the appropriate type face concurrently with other bytes in the data field.

Since many type faces may be positioned at a given spot before a match with the output byte occurs, you might think that these printing techniques would result in slow print speeds. Actually, because of the high speed of the type elements and the very high speeds at which the computer matches output bytes and type faces, very fast printing is achieved.

●●●

129 Figure 15 shows that the 1403 Printer has a maximum speed of .... lines per minute.

●●●

1100

130 If the 1403 were printing 120 characters on each line, at maximum speed, it would print (how many?) .... individual characters in one minute.

●●●

132,000

131 The number of characters printed on each line depends on the printer in question. Figure 15 shows the capacity of each S/360 printer in characters per line.

Match the following:

a.	1403 M2	1.	113 print positions
b.	1403 M7	2.	120 print positions
c.	1403 N1	3.	144 print positions
d.	1404	4.	132 print positions
e.	1443		
f.	1445		

●●●

a. 4
b. 2
c. 4
d. 4
e. 2 or 3
f. 1

132 Earlier, we said that if you want to print characters that are not on the type element of your printer, you must change to an element that has the desired characters. This brings up an important point about S/360 printers.

Different applications may require different character sets. In such a situation it might be desirable to have more than one type element in the installation. The appropriate type element would be installed on the printer for the job being run.

You would not want to have to call an IBM customer engineer every time you wanted to change the type element. With S/360 printers you do not have to. Type elements can be changed by the operator in just a few minutes time.

●●●

133 There is one last consideration about printer type elements we should mention. Earlier, we pointed out that each byte can contain 256 different bit patterns, each of which can represent a character.

Let's assume that we have designed a graphic (the symbol that prints) for each of our 256 characters, that is, we have a 256-character set.

Can we equip a printer to print them all? (Think back to what you learned about type element sizes.)

●●●

No. S/360 printer chains can contain a maximum of 240 characters. Thus we can print only 240 different graphics.

134 What is the principal factor that influences printing speeds of S/360 printers?

●●●

The character set being printed. The more different characters the unit can print, the slower the printing speed.

135 Which of the following can be found on a single printer type chain?

a. Numeric characters
b. Special characters
c. Upper case alphabetic characters
d. Lower case alphabetic characters

●●●

All can be included on a single chain.

136 Who can change the type element (bar or chain) on a S/360 printer and about how long does it take?

●●●

Either the customer engineer or the operator can change the type element in a few minutes time.

137 The bits in each byte of S/360 storage can be arranged in 256 different configurations. This means each byte can represent up to 256 different characters, including upper and lower case alphabetic, numeric, and a large number of special characters. How many of these configurations can be printed at any one time?

●●●

240. This is the maximum number of different characters that can be included on a print chain.

138 How many of the 256 characters that can be represented in a byte can be printed?

●●●

Any character that can be represented in S/360 storage can be printed provided the graphic for that character is present on the type element (bar or chain).

139 Assume a printer is writing one line for every two cards read by a 2540. The program can maintain a constant reading speed of 600 cards per minute. The printed line may contain all alphabetic and numeric characters, plus eight special characters. From Figure 15, select the printer(s) that will permit the S/360 to maintain the 600 card per minute reading speed.

●●●

The 1403 and 1404. Other printers in Figure 15 will not print all the required characters at the desired speed of 300 lines per minute. Note also that the 1404 has card printing capacity. Unless this additional feature was useful on other jobs, the 1403 would be the logical choice.

140 We have talked about how printers print. Now let's take a quick look at another of their capabilities; moving the paper on which they are printing.

There are two basic ways to move the form: spacing and skipping.

Spacing occurs when the forms carriage (the part of the printer that moves paper past the print station) moves the form from one printing line to the next successive printing line.

The carriage automatically spaces one line for each line that is printed. But it is possible to cause the carriage to space more than one line. Additional spacing or space suppression is controlled by the stored program; that is, an .... is executed that tells the printer to space a particular number of lines.

●●●

instruction

141 If you are printing with single spaced lines, is it necessary to signal the carriage to get single spacing?

●●●

No. Single spacing occurs automatically.

142 How can you get additional spaces between lines?

●●●

Execute one or more instructions that cause the carriage to take one or more spaces.

143 By issuing consecutive space instructions you could move the printing form as far as you wish before printing the next line. It is possible to space 1, 2, or 3 lines at a time. The distance spaced would, of course, be the sum of the single, double, or triple spaces called for by the instructions.

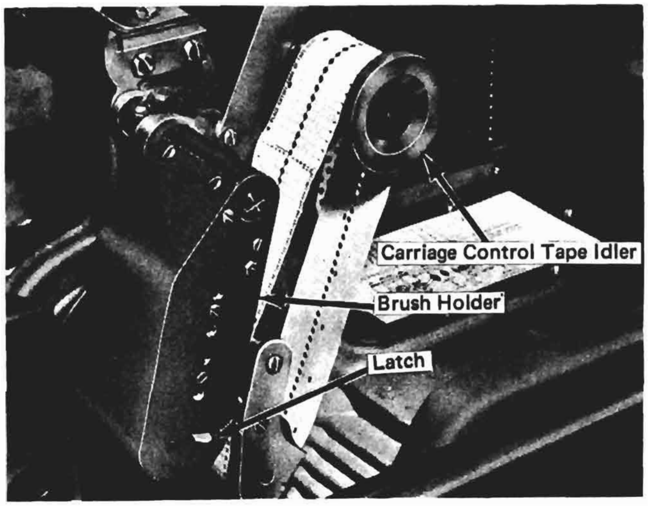
How could you cause the form to space 8 additional times after printing a line?

●●●

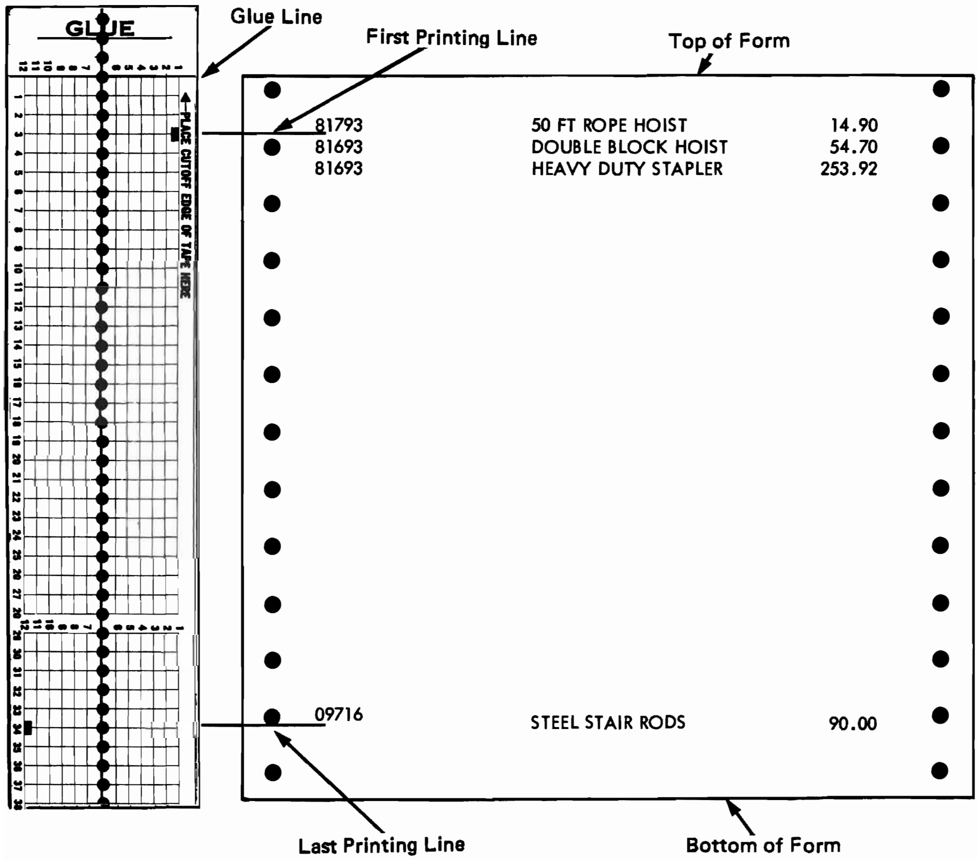
4 double space instructions, or 2 triple and 1 double, or 8 single, etc. Any combination that adds up to 8.

144 Paper forms can be moved by successive space instructions, but it is a rather slow process, and requires all those extra instructions. Paper can be moved faster, for significant distances, by using the alternate method, which is ....

●●●



TAPE—CONTROLLED CARRIAGE



81793	50 FT ROPE HOIST	14.90
81693	DOUBLE BLOCK HOIST	54.70
81693	HEAVY DUTY STAPLER	253.92
09716	STEEL STAIR RODS	90.00

Figure 16.

skipping.

145 Skipping occurs when it is desired to move the form further than normally would be done by spacing.

Skipping is initiated by the stored program; that is, when a skip is desired, the program must contain ....

●●●

an instruction that tells the carriage to skip.

146 Spacing stops automatically after the indicated number of lines has been passed. However, skipping requires a special mechanism to stop the skips. Figure 16 shows this mechanism.

A length of a special paper tape, called the carriage control tape, is formed into a loop and mounted in a special compartment on the carriage. The carriage tape is the same length as, or a multiple of, the length of the form being printed. It travels around its loop as the printing form moves, traveling a distance equal to the distance the form travels.

A 5-inch movement of the form will cause the tape to move ....

●●●

5 inches.

147 There are 12 columns called channels running lengthwise on the carriage tape and there can be one or more holes punched in each channel. Each hole in the tape corresponds to a particular printing line on the form.

If we move the tape to a particular hole the form will have moved to the line ....

●●●

corresponding to that hole.

148 Perhaps you're beginning to see the significance of the tape and the holes in it.

The holes are read by electric brushes. The stored program simply issues an instruction telling the carriage to skip until a specific hole in the tape is read by its brush. The tape and the form move together, at the same speed, until the designated hole is read by its electric brush. The impulse generated .... the forms movement.

●●●

stops

149 A Skip to Channel 8 instruction will move the printing form to what point?

●●●

To the printing line corresponding to the hole in channel 8 of the carriage tape.

150 Skipping is initiated by .... and stopped by ....

●●●

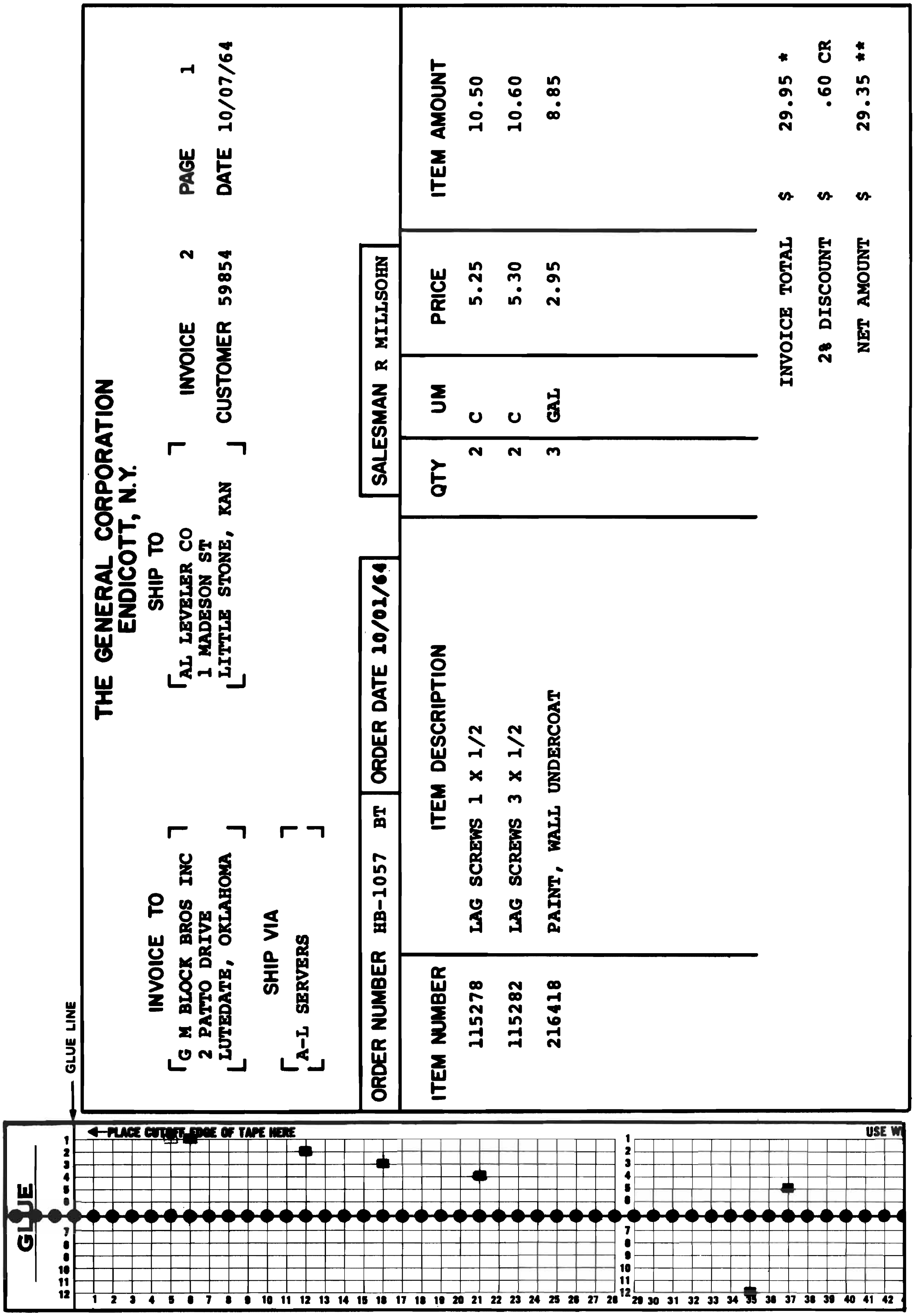
instructions to the printer
holes in the carriage tape, read by electric brushes.

151 It is quite simple to determine where on the form you want your skipping to stop, and to punch a hole in the carriage tape opposite that line. Your instruction to cause skipping must specify ....

●●●

the number of the channel in which the hole is punched.

152 Figure 17 shows a carriage tape and a printed form whose movement it controls. The data items to be printed first on the form are the customer name, the name of the company receiving the merchandise, the invoice number, and the page number.


GLUE LINE
GLUE
PLACE CUTOFF EDGE OF TAPE HERE
USE W

THE GENERAL CORPORATION
ENDICOTT, N.Y.

INVOICE TO
G M BLOCK BROS INC
2 PATTO DRIVE
LUTEDATE, OKLAHOMA

SHIP TO
AL LEVELER CO
1 MADESON ST
LITTLE STONE, KAN

INVOICE 2
PAGE 1
CUSTOMER 59854
DATE 10/07/64

SHIP VIA
A-L SERVERS

ORDER NUMBER HB-1057 BT
ORDER DATE 10/01/64
SALESMAN R MILLSOHN

ITEM NUMBER	ITEM DESCRIPTION	QTY	UM	PRICE	ITEM AMOUNT
115278	LAG SCREWS 1 X 1/2	2	C	5.25	10.50
115282	LAG SCREWS 3 X 1/2	2	C	5.30	10.60
216418	PAINT, WALL UNDERCOAT	3	GAL	2.95	8.85

INVOICE TOTAL	$	29.95 *
2% DISCOUNT	$	.60 CR
NET AMOUNT	$	29.35 **

Figure 17.

A hole in channel .... of the tape stops the form at the line on which these items appear.

●●●

1

153 The form in Figure 17 would (skip/space) .... the next two printing lines. How do you know this?

●●●

space

There are no holes in the tape corresponding to these lines. Since a skip can be stopped only by a hole in the carriage tape, these lines would be bypassed if a skip command were issued. Single, double, or triple spacing normally is used to move the form 1, 2, or 3 lines.

154 The form is stopped at the first printing line for the shipping instructions by ....

●●●

a hole in channel 2 of the tape.

155 Where will the form be positioned when the hole in channel 4 is sensed?

●●●

At the first printing line in the body of the form.

156 As each detail line in the body of the form is printed, the carriage will (skip/space) .... to the next line.

●●●

space

157 If there are more detail lines than one form can hold, the additional lines are printed on the next form. So there must be some way to signal when the last detail printing line has been used. Figure 17 shows that a punch in column .... of the tape corresponds to the last detail line on the form.

●●●

12

158 A hole in column 12 of the carriage tape performs a special function. When read by its electric brush the hole generates a pulse that turns on an indicator inside the computer. This indicator is the signal that the printing line associated with the hole in channel 12 has been reached.

In Figure 17, the hole in channel 12 is associated with which printing line on the form?

●●●

The last detail line.

159 In normal operation, the hole in channel 12 is associated with the last detail printing line on the form. Each time the computer executes a print instruction, it will test this indicator; thus it will know when that point on the form has been reached.

If there are still detail lines to be printed when the last detail printing line is reached, what would you expect the stored program to do?

●●●

Execute an instruction telling the carriage to skip to the next form.

160 The purpose of the hole in channel 12 is ....

●●●

to signal when a particular printing line on the form has been reached.

161 What is the positional relationship between the hole in channel 12 and the printing line it represents?

●●●

The hole is punched in channel 12 exactly opposite the printing line it represents. This is true of every hole in the tape.

162 At the bottom of the form shown in Figure 17 there is space for some total amounts. These would not be printed until all detail lines for a given invoice had been printed. What indicates the first total line? How would this line be reached?

●●●

A hole in channel 5 of the tape.
By a Skip to Channel 5 instruction.

163 You can see that after the last detail line is printed, regardless of where in the body of the form it is, the proper skip instruction will cause the carriage to move the form to the first total line. We can't space to it because we can never be sure just where the last detail line will print; thus we won't know how many spaces we have to take to get to the total line. We have to skip. But we would space from one total line to the next, since they are always the same number of lines apart on every form.

After the last total line has been printed, we are ready to start a new invoice. What instruction would we use to get to the first printing line on the next form?

●●●

A Skip to Channel 1 instruction.

164 A final item of importance about printer carriages is this: they are called dual speed carriages because they skip faster than they space.

Let's assume you are printing only a few lines on a very long form. After the last line is printed on each form you want to move on to the next form as quickly as possible. Although S/360 printers space rapidly, this situation calls for skipping. If you couldn't skip any faster than you could space, there would be no particular advantage in having the flexibility that the carriage and carriage tape combination gives you.

Fortunately, for distances of four or more lines, the carriage can .... faster than it can ....

●●●

skip
space.

165 When it is desirable to move the form more than just a few lines, skipping is called for. Why?

●●●

Skipping is faster than spacing.

166 You can deduce why it is important to move paper as fast as possible. On jobs requiring a high volume of printing, paper movement is at the expense of printing time; that is, no printing can occur while the paper is in motion. Therefore it is desirable to take advantage of every means to speed up forms movement.

●●●

167 Because the tape has 12 channels in it, a large number of different skips is possible for each form. For example, if each channel has but one hole, 12 skips are possible. By punching additional holes in the channels, as many skips as would be required for even extreme circumstances can be had.

●●●

168 Describe a carriage control tape and tell how it controls forms movement.

●●●

The carriage control tape is a loop of paper tape, the length of which is a multiple of the length of the form it will control. Holes can be punched in 12 columnar positions in the tape, with each hole corresponding to a particular line on the form. When mounted, the tape moves in step with the forms, passing a reading station that senses holes in the columns. Forms movement initiated by the program is stopped when the designated hole is read.

●●●

169 Describe the operation of a dual speed carriage and give the advantage of its use.

●●●

The dual speed carriage moves forms at a given speed for spacing and at much higher speeds for skipping.
The overall time required to move forms is reduced, provided there is skipping. This can significantly reduce the time required for a processing run.

170 Insofar as the Electric Credit Sales Company's problem is concerned, a printer might be required regardless of whether or not the computer is centralized. If printed output in any volume is a requirement, the printer is the obvious answer. Very low volumes of printing can be accomplished on a printer keyboard, which we will discuss later, but for larger volumes a fast printer such as we have been discussing here can do the job.

It should be noted that some computer systems are not operated with printers; they use only tape and/or disk units on line. Peripheral operations, such as transferring data from one medium to another are performed on smaller, "satellite" computers.

Writing data on tape or disk for input to the large computer would be done on the smaller computer. Can you guess how printing would be handled?

●●●

Data to be printed would be developed on the larger computer and written on magnetic tape or disk. This would become input to the smaller computer, which would have an on-line printer. Thus, printing would occur on the smaller computer.

171 We have talked about a variety of I/O devices for the S/360. Any of them could be attached to the centralized computer system of the Electric Credit Sales Company. Of course, as we have pointed out, some would be more applicable than others in this particular case. But each has a specific function to perform and many applications exist for each.

There is one thing that all these units have in common, other than that they are all I/O devices. Thinking back to the preceding topic, in which we discussed S/360 channels, can you guess what it is?

●●●

All are slow-speed devices.

172 Slow-speed devices normally are connected to a .... channel on which they operate in .... mode.

●●●

multiplexor
byte

173 Byte mode means the device on the channel may operate (concurrently/ sequentially) .... with other devices on the channel.

●●●

concurrently

174 You recall that any S/360 I/O device can be connected to any channel. Why would paper document I/O devices (OCR reader, MICR reader, OMP reader, paper tape reader, card readers, and printers) normally be connected to a multiplexor channel?

●●●

Their data transmission rates permit them to operate in byte mode. Thus they normally would be connected to the multiplexor channel, where they could operate concurrently with other I/O devices.

175 Name one or more data processing situations or applications that might make use of each of the following I/O devices:

a. Card readers and punches
b. Paper tape readers
c. Printers
d. Optical character reader
e. Magnetic ink character reader
f. Optical mark page reader

●●●

Card readers are used in a high percentage of data processing jobs, such as payrolls, sales analysis, and many different accounting jobs.

Card punches are used whenever punched card output is desired, such as utility billing, subscription renewal, check writing and many others.

Paper tape readers are used in any application in which data recorded in paper tape must be entered into a computer. The tape may be recorded from direct wire transmission, or may have been physically transported to the computer location.

Printers are used for any application requiring sizeable amounts of printed output. Payrolls, sales analysis, and other commercial applications use printers extensively. Almost all data processing jobs require printed output ultimately.

The optical character reader is used to read data from cash register tape. It also can read printed data from many other sources. The processing of meter reading tickets by utility companies can be done on the optical character reader. Also, processing sales checks, merchandise orders, or telephone toll tickets are good applications for OCR.

The magnetic ink character reader is used primarily for processing bank checks, deposit and withdrawal slips, and other banking jobs.

Optical mark page readers are used in test scoring, market analysis, inventory processing, and others in which an item of data can be represented by one or more pencil marks on the page.

System Control Devices

176 The last category of I/O devices we shall discuss in this topic is called "system control". These devices are used primarily by the operator. You probably can guess that the system they control is the .... system.

●●●

computer

Console

177 System control devices enable the operator to communicate with and control the operation of the computer. The basic system control device is the console, a panel of lights, switches and control buttons.

By using the console, the operator can start and stop the system. He can look at the contents of main storage, or the counters and registers in the system.

The START, STOP, and DATA DISPLAY controls would be located on the computer ....

●●●

console.

178 Contents of registers and storage areas can be displayed in the lights on the console. The operator simply dials in the address of the byte(s) whose contents he wants to see and presses the appropriate buttons and keys.

Suppose the computer encounters a condition that interrupts its operation. How could it signal the operator?

●●●

By turning on a console light.

179 Various internal conditions are called to the operator's attention by console lights. Thus the console permits two-way communication between the operator and the computer system.

We would conclude that the operator (can/cannot) .... enter data through the console.

●●●

can

180 It is possible to enter data to be processed through the console, but this is not done except, perhaps, for an occasional corrective item. You can see how slow and inefficient it would be to set up individual digits on console dials and enter them one at a time.

Most information entered through the console is necessary to .... the system.

●●●

control or communicate with

Printer Keyboard

181 Some computer systems are equipped with another system control device called a printer keyboard (sometimes referred to as a console typewriter). This is basically a typewriter with some special keys.

A system with a printer keyboard would be more likely to

a. print short messages to the operator.
b. communicate exclusively via console lights.

●●●

a. print short messages to the operator.

182 Messages can be printed indicating the status of the user's program as well as the conditions inside the computer.

You remember from the topic on Program Execution that a control program called the supervisor program is in storage at all times, along with the user's problem program. Match the following:

a. messages concerning the status of the system.
b. messages concerning the status of the problem program.

1. Problem program
2. Supervisor program

●●●

a. 2
b. 1

183 We said the printer keyboard has some special keys in addition to the usual typewriter keys. These are used by the operator to signal the system to take specific actions.

The printer keyboard facilitates the input and output of control information, but also can be used for data being processed. Again, the volumes of such data would have to be relatively low, since typing (either manual for data entry, or automatic for output) can't compare with reading and writing speeds for other media.

●●●

184 Name two I/O devices used by the operator to communicate with the computer system, and give the characteristics of each.

●●●

The printer-keyboard. It is an auxiliary device consisting of a standard typewriter keyboard with special keys. Messages from the program are typed to the operator; the operator can signal the system to take specific actions. Data can be entered or recorded on the typewriter, but the process is slow and not used for volume data.

The system console. It is a panel of lights, switches, and dials at which certain machine conditions are reflected and through which the operator can, by manipulating various dials and switches, control the computer's operations.

You have completed this section. At this point, you should fill in your notes and take the self-evaluation quiz.

Self Evaluation Quiz
Card and Paper Document Handling, and System Control
I/O Devices

QUESTIONS

1. What are some of the factors to be considered in selecting card and paper document handling I/O devices?

2. In what way do S/360 card readers and punches differ?

3. In what ways do S/360 printers differ?

4. What characteristics do MICR and OCR characters have in common?

5. Identify the sensing techniques used by MICR and OCR readers.

6. What form of data is sensed by the Optical Mark Page Reader?

7. Briefly describe the printing technique used by S/360 printers.

8. What is the major factor that determines printing speeds?

9. Describe how forms movement on S/360 printers is controlled.

10. Describe the channel hookup and other operating conditions for S/360 card and paper document handling I/O devices.

Self Evaluation Quiz
Card And Paper Document Handling, And System Control
I/O Devices

ANSWERS — Frame Reference

1.
   a. How data is recorded at the source. (12,14,19, 20,22,46, 134)
   b. Data transmission requirements.
   c. Maximum input-output speeds to be attained.
   d. Advantages to be gained from reading and punching the same card.
   e. Character set required for printed documents.
   f. Transaction batching possibilities.

2.
   a. Reading and punching speeds. (Figure 2)
   b. Ability to read and punch the same card.
   c. Read/punch pattern (serial vs. parallel)

3. Printing speeds, ability to print magnetic characters, character set size, ability to print on different kinds of documents (cards, paper, etc.) (Figure 3)

4. Both are unique type styles read by special techniques. Both are human- and machine-readable. (53,70)

5. MICR readers sense characters originally printed in magnetic ink, by first magnetizing them and then recognizing the resulting magnetic pattern. OCR readers use a light scanning mechanism to sense the unique outlines of the characters. (55,71)

6. Short straight pencil marks in predetermined positions on 8½" x 11" sheets. (97)

7. A bar or chain containing a number of the character set being printed is passed rapidly along the printing line. The storage bytes to be printed are scanned at electronic speed and printing occurs when the bit pattern in the byte is matched by the type face at the printing position for that byte. (121)

8. The character set to be printed. Generally speaking, the larger the set, the slower the printing speed. (126)

9. The forms carriages on S/360 printers are controlled by loops of paper tape, called carriage control tapes. Each tape has 12 columns extending lengthwise on the tape, in which holes can be punched. Each hole corresponds to a particular printing line on the form, and when sensed by an electric brush, stops the form at that printing line. (168)

10. S/360 card and paper document handling I/O devices are considered slow speed devices and would normally be connected to a multiplexor channel on which they would operate in byte mode. (172)

Data Management

DATA MANAGEMENT

1 In most computer programs, the need exists for moving data between main storage and I/O devices. System/360 handles this through Data Management in the operating system. Data Management also includes:

1. Scheduling and programming the operation of channels.
2. The identifications of data by volume (the device in which data is stored such as a reel of tape, or a disk pack) and by file.
3. The resolution of error conditions that occur during the movement of data.

Data Management is composed of the Input Output Control System, (referred to as IOCS), and a set of standards for working with IOCS.

IOCS and an associated set of standards comprise a system called....................

●●●

Data Management.

2 IOCS is a set of routines (programs) that handle the movement of data between main storage and I/O devices. These routines are supplied by IBM as part of the S/360 programming support.

IOCS is made up of a group of..........(routines/ printed circuits) for moving data.

●●●

routines

3 IOCS routines fall into two categories:

1. Physical IOCS (PIOCS)
2. Logical IOCS (LIOCS)

PIOCS is composed of those I/O routines which supervise the reading and writing of data on I/O devices without regard for its logical content, format or organization.

PIOCS operates independent of and concurrent with its associated problem program.

PIOCS is concerned principally with the.......... (I/O devices/problem program).

●●●

I/O devices

4 Once the data has been brought from the I/O device by PIOCS, the LIOCS routines now operate on this data to make its logical content available to the problem program as required.

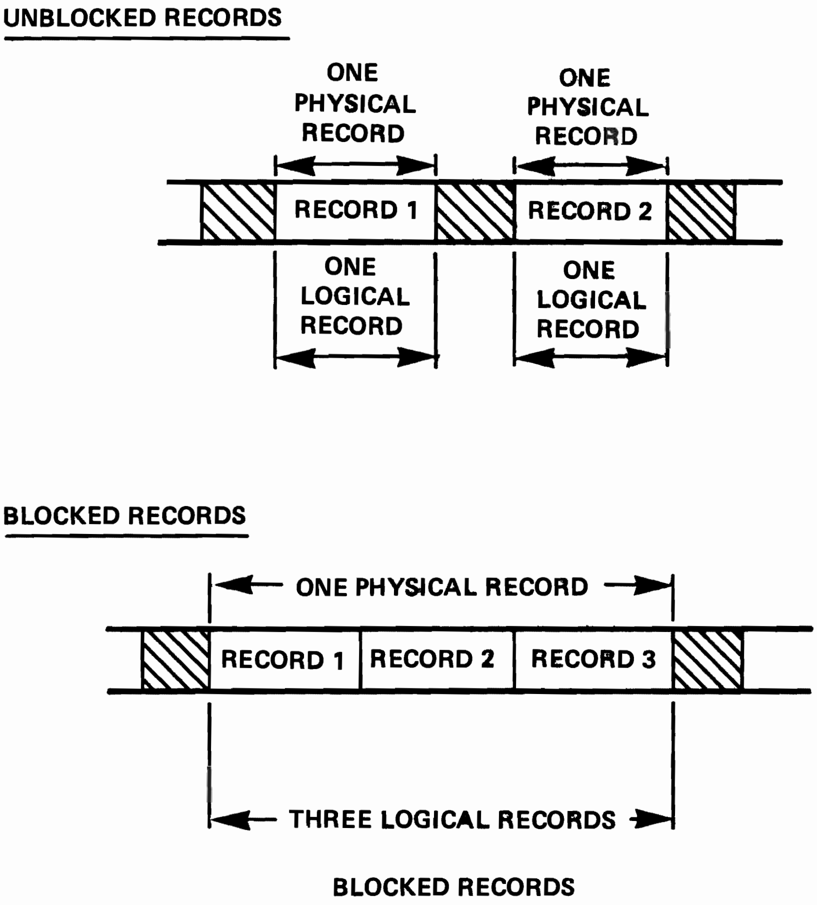
Data Management thus operates through the dynamic interaction of PIOCS, LIOCS and the problem program. The aim and result of Data Management is efficiency in data handling and particularly efficiency in programming effort by the problem programmer.

●●●

5 Data Management facilities are available to the programmer through the use of very simple instructions that he can use in his problem program. For example, at the point in the problem program where the next logical record is needed, the programmer needs only to code "GET" together with the symbolic file name, such as "SYS017". When the program execution reaches this point, LIOCS will be called in and will take care of making the logical record available to the problem program. Other simple instructions of this type are available for other functions of Data Management.

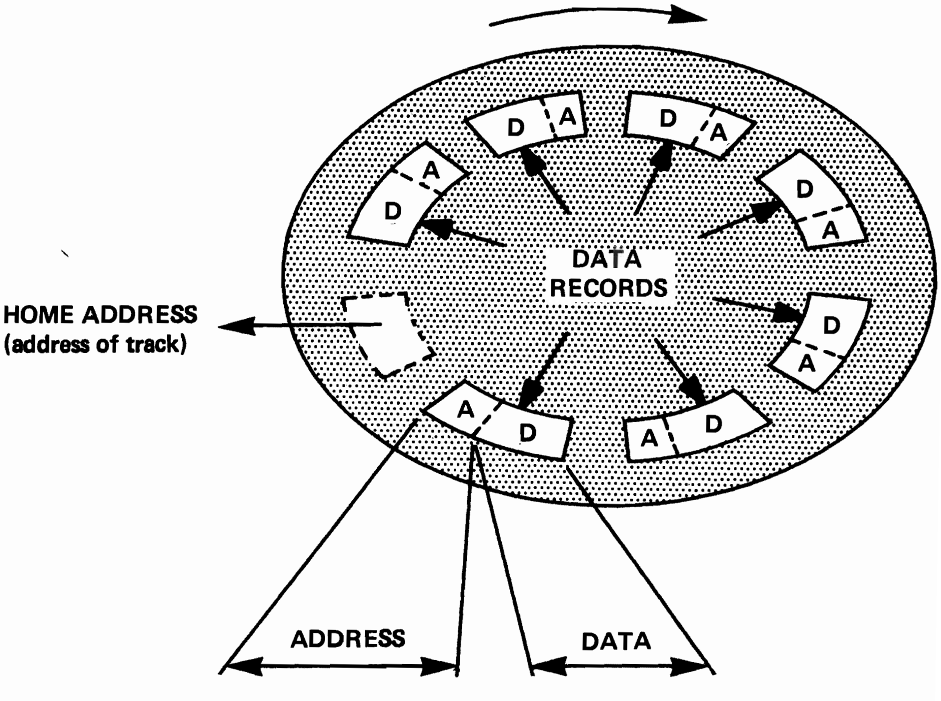
●●●

6 The concept of an Input/Output Control System was developed several years ago when a particularly fruitful study disclosed that, on the average, problem programmers were spending at least 40 per cent of their time coding, testing and debugging input/output routines. By providing the programmer with generalized I/O routines which could be tailored to the specific data format and



BLOCKED RECORDS

Figure 1.



PHYSICAL RECORDS ON DISK

Figure 2.

organization being used by the programmer, the programmer's productivity is almost doubled with little additional training or expense. With the advent of the S/360 with its complex channel operations and device independence, IBM has provided not only the programming efficiency of the earlier IOCS routines, but also hardware efficiency through proper I/O device operation.

Under S/360 Data Management, the programmer is required to code the detail routines of .... (PIOCS/LIOCS/neither PIOCS nor LIOCS).

●●●

neither PIOCS nor LIOCS.

7 The detailed routines of IOCS are IBM supplied to save programmer time and effort. The programmer needs only to define the files used in his problem program to tailor these routines to his specific job. This is a fairly straightforward task and is further simplified through the use of pre-printed IBM forms which list all possible entries needed by the programmer. Filling out one of these forms is much like using a pre-printed order form, where the customer supplies quantity, size, color, etc., for the item he wants.

●●●

8 Data Management standards describe the formats necessary for specifying and identifying data, files and volumes to be handled by the IOCS. These standards specify such things as:

1. Logical record formats.
2. Physical record formats.
3. Organization of data files.
4. Specification of data files.
5. Specification of volume and file labels.

1. Logical record formats. A logical record is the basic unit of information for a data processing program, and is defined by the fields it contains and its logical use rather than its physical form.

The processing program processes the data in the form of a .... (logical/physical) record.

●●●

logical

9 2. Physical record formats. A physical record is a unit of data for a particular medium and file organization. A physical record may contain one or more logical records. Where more than one logical record make up a physical record, the records are said to be blocked. See Figure 1.

In review, you will recall that the grouping of two or more logical records is called "blocking" the records. Conversely, separating the records found in the block is known as "deblocking". Logical IOCS handles these functions under Data Management.

Physical IOCS deals with the reading and writing of.........(logical/physical) records.

●●●

physical

10 Physical records on a disk are diagrammed in Figure 2 and illustrate that a physical record may contain more than just the logical data of a record or records.

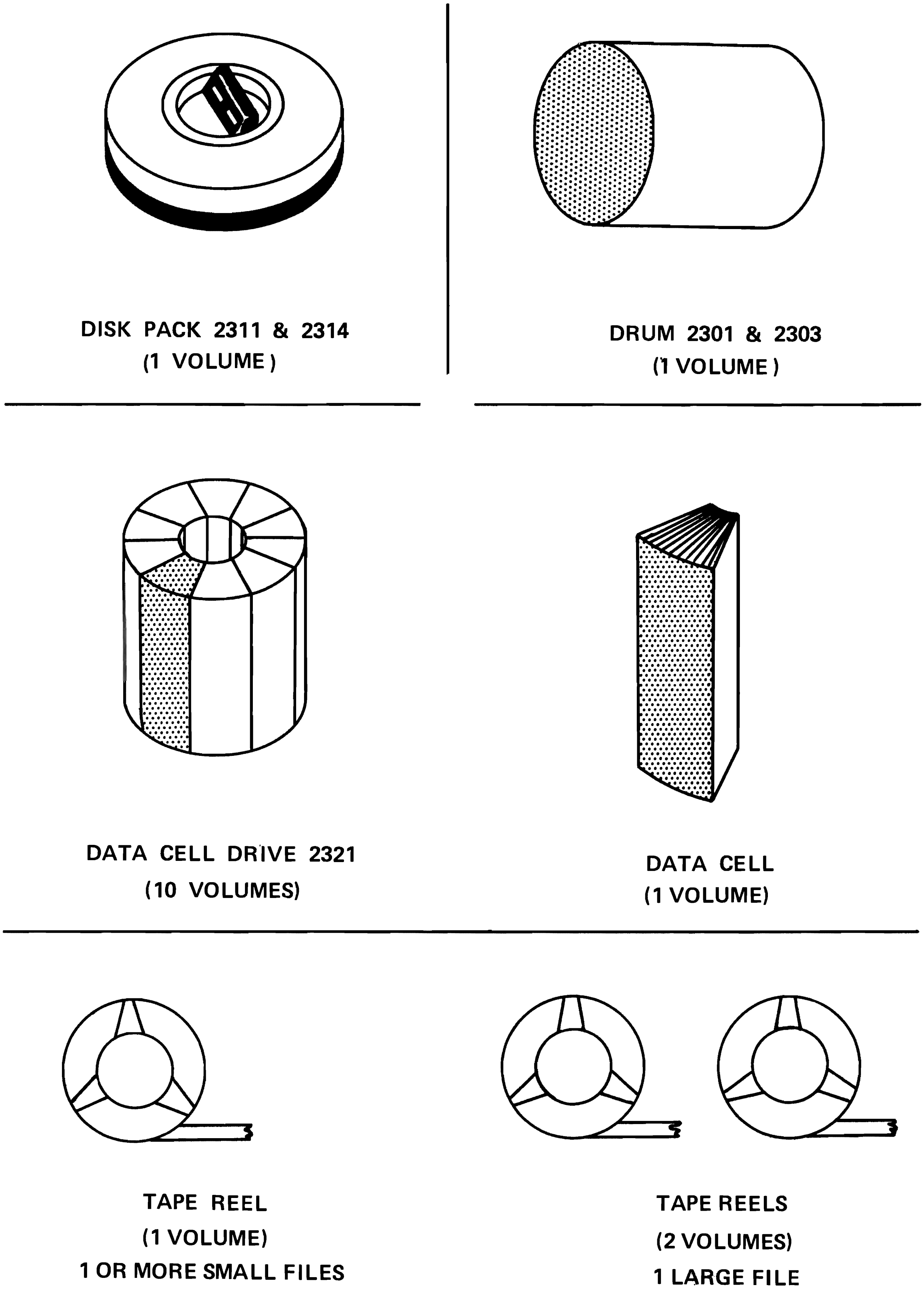
●●●

11 3. Organization of data files. Problem programs do not have identical requirements of the data files used with respect to number of transactions, additions and deletions to the file, response time, etc. Data Management provides several different arrangements of the way records may be stored and retrieved to achieve maximum efficiency with a particular problem program.

The file organization is normally determined by the.......... (application/system) requirements.

●●●

application



DISK PACK 2311 & 2314
(1 VOLUME)

DRUM 2301 & 2303
(1 VOLUME)

DATA CELL DRIVE 2321
(10 VOLUMES)

DATA CELL
(1 VOLUME)

TAPE REEL
(1 VOLUME)
1 OR MORE SMALL FILES

TAPE REELS
(2 VOLUMES)
1 LARGE FILE

S/360 VOLUMES

Figure 3.

12 Define a data file (your own words).

●●●

A data file is a collection of logically related records.

13 4. Specification of data files. The means of specifying the data files has been standardized under Data Management. It includes such items as the type of logical record format in use, access method required, and location and use of the file.

IOCS is tailored to the particular program and file requirements in Data Management file.......... (organization/specification) standards.

●●●

specification

14 5. The specification of volume and file labels. File identification is desirable to prevent destruction of current data in an unexpired file, to assure the processing of the proper data, and similar problems. Data Management provides this facility through the use of volume and file labels.

A label is a preformatted machine readable record used for identification of a volume or file. The labels defined in Data Management are called S/360 Standard Labels.

Files are identified through Data Management by means of..........

●●●

labels.

15 Each volume has a volume label and may contain one or more files. Further, a file may be so large as to require more than one volume to hold all the records. Each file may have one or more labels.

A volume is a..........(device/record).

●●●

device (see Figure 3)

16 Each volume may have ....volume labels and..........file labels.

●●●

one
one or more

17 IOCS can be tailored for a specific problem program and its associated files by the programmer. The programmer does this by defining the files as required by the Data Management specification standards.

●●●

18 Logical IOCS signals physical IOCS whenever an I/O operation is required. PIOCS handles the physical record (scheduling the desired I/O operations for maximum efficiency), and also checks for and handles error conditions that may arise.

PIOCS is a set of Data Management programs dealing with the scheduling of..........

●●●

I/O operations.

19 PIOCS..........(does/does not) interact directly with the problem program.

●●●

does not

20 These programs are called physical IOCS because they deal with physical records.

LIOCS deals with .... records.

●●●

logical

21 Logical IOCS routines are linked directly to the problem program and interface with PIOCS.

LIOCS requests I/O operations from PIOCS and deals with the logical records required by the problem program. It will handle blocking and deblocking of these logical records as necessary.

●●●

22 LIOCS contains routines for determining the End of a File or, when a file extends over more than one volume, the End of the Volume. It will then take the appropriate action.

Automatically switching to the designated alternate I/O device would be the normal manner in which LIOCS would handle an end-of- .... (file/volume).

●●●

volume

23 What is Physical IOCS? (in your own words)

●●●

Physical IOCS is a set of I/O routines which supervise the reading and writing of data without regard for its logical content or format. PIOCS schedules I/O operations and checks for and handles error conditions.

24 What is Logical IOCS? (in your own words)

●●●

LIOCS is a set of routines for handling logical files defined in the user problem program. LIOCS requests I/O operations from PIOCS, handles EOF and EOV conditions, checks and writes labels and blocks and deblocks logical records.

25 A volume may contain one or more files. Also a file may be so large as to require several ....

●●●

volumes.

26 Looking at logical records first, Data Management defines three types. These are:

1. Format F
2. Format V
3. Format U

A format F logical record is a record from a file in which all logical records have a fixed number of characters. These are also called fixed-length records. Format F logical records may be blocked or unblocked.

A format V record is sometimes called a variable-length record.

A format V record has a .... number of characters in each logical record.

●●●

variable

27 In S/360 Data Management, a Format V record has a standard field which indicates the length of the record. Format V records may be blocked.

A physical record that is made up of more than one logical record is said to be ....

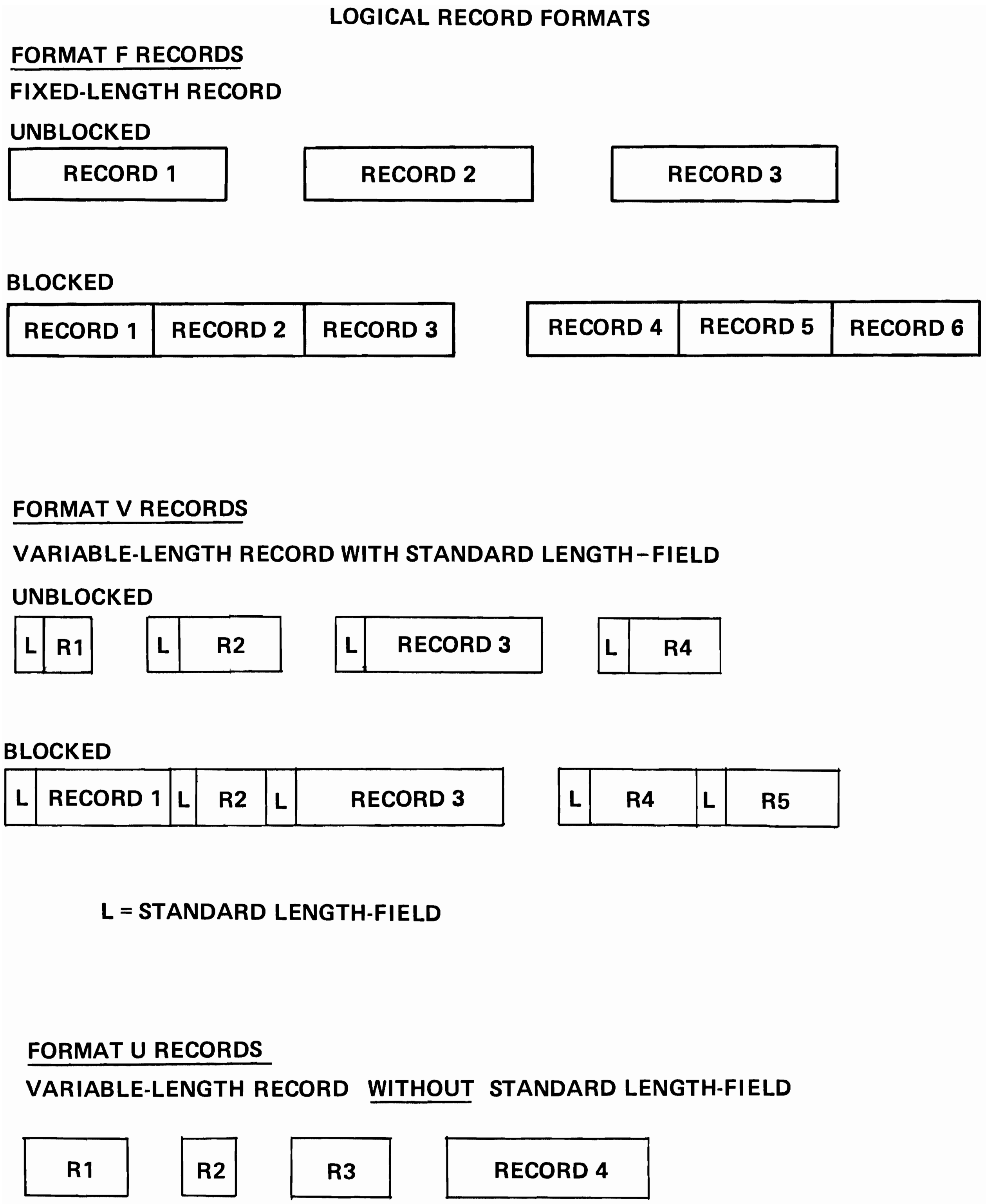
●●●

blocked.

28 A Format U record is a variable-length record that is undefined because it does not contain a standard S/360 field that describes its ....

●●●

length.



FORMAT U RECORDS MAY BE INTERNALLY BLOCKED BUT USER MUST PROGRAM BLOCKING AND DEBLOCKING. THIS IS NOT SUPPORTED FOR FORMAT U RECORDS BY DATA MANAGEMENT.

Figure 4.

29 The Format U record may contain blocked logical records but, because the record is undefined, LIOCS will not perform the blocking or deblocking. This becomes the programmer's responsibility.

●●●

30 Figure 4 illustrates the record formats.

LIOCS .... (does/does not) block and deblock Format F and Format V records.

●●●

does

31 A physical record is the unit of recorded information. It is made up of one or more logical records. If the physical record contains more than one logical record, the records are said to be..........

●●●

blocked.

32 Physical records containing Format F records are always the same length within a file. If the records are unblocked, the physical record will contain one fixed-length record. If the records are blocked, all the physical records will contain the same number of Format F records per block.

With Format F records in the file, the physical records.......... (are/are not) all the same length.

●●●

are

33 When logical records are blocked in a file, LIOCS assembles these records in a reserved portion of main storage. The length of this area is dependent on several factors such as main storage available, actual storage device, etc., and is defined when the file is created.

●●●

34 When working with Format V records, LIOCS will place as many complete Format V records as possible in this area. Thus, the number of records blocked and the length of the blocks will vary from block to block. The standard length-field will be set by LIOCS for each block.

With Format V or Format U records, the physical records.......... (must/may not) be all the same length.

●●●

may not

35 The.......... (exact/maximum) length for each physical record must be specified to assure that the entire record may be stored in main storage.

●●●

maximum

36 In a file containing F type records, the exact length of all the records, both physical and logical, is known when the programmer defines the file.

Think now; in a file containing V-type records or U-type records, the length of the..........(average/longest) record must be specified by the programmer.

●●●

longest (The longest, or maximum length must be be defined, so that enough core storage is set aside to contain the record).

37 Record types may not be mixed in a file.

A file defined for Format V variable-length records,.......... (may/may not) contain Format U variable length records.

●●●

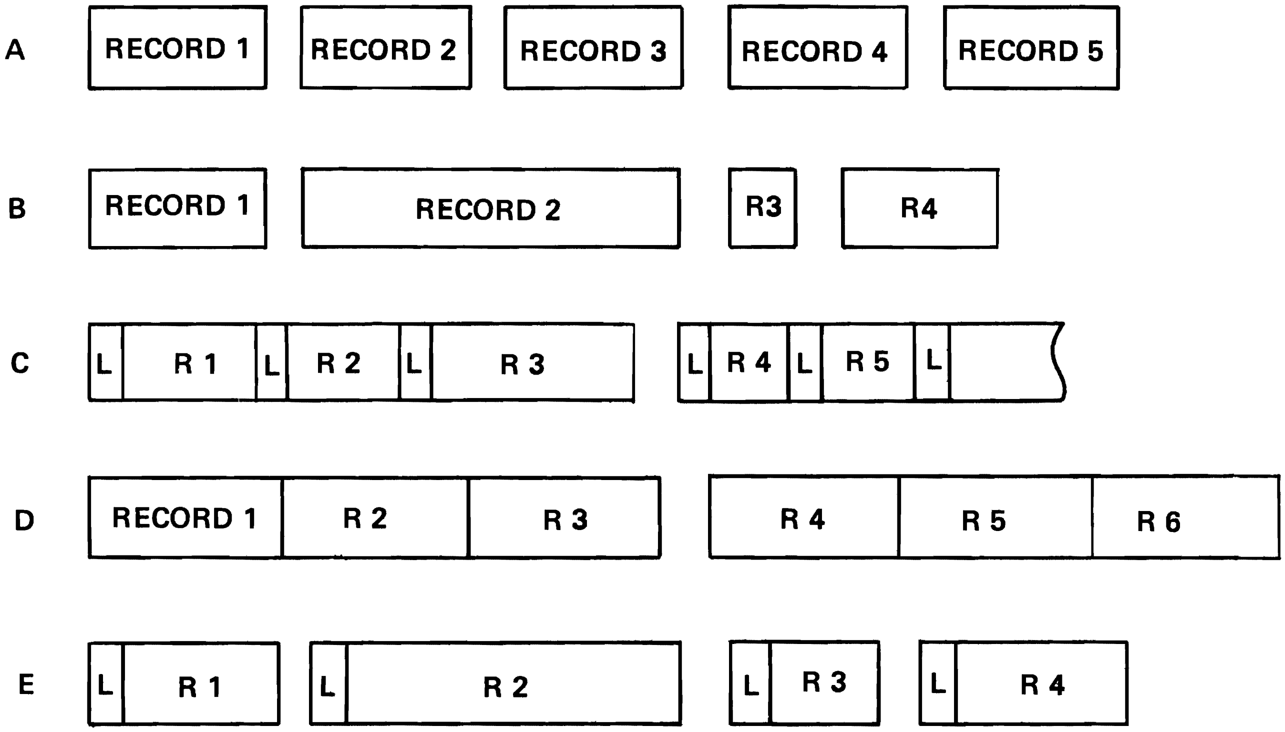

A
RECORD 1
RECORD 2
RECORD 3
RECORD 4
RECORD 5
B
RECORD 1
RECORD 2
R3
R4
C
L
R 1
L
R 2
L
R 3
L
R 4
L
R 5
L
D
RECORD 1
R 2
R 3
R 4
R 5
R 6
E
L
R 1
L
R 2
L
R 3
L
R 4

Figure 5.

may not

38 Match the record formats shown in the illustration (Figure 5) with the proper definition.

1. Format F, unblocked records
2. Format F, blocked records
3. Format V, unblocked records
4. Format V, blocked records
5. Format U records

●●●

1. A
2. D
3. E
4. C
5. B

39 A file, also called data file or data set, is a collection of logically related records. Quite simply, it is a set of data.

A set of records containing details of customer accounts is a..........

●●●

file, data file, data set.

40 A deck of cards containing details of transactions affecting the customer account data set is also a..........

●●●

file, data file, data set.

41 A file to be used in data processing must, of course, be in machine readable format. In addition, it must be organized for ease of use on specific applications and must be labeled for identification.

Data Management provides three major file organizations for use with processing applications. These are:

1. Sequential
2. Random
3. Indexed-Sequential

Arranging the data serially in numeric or alphabetic order is called.......... organization.

●●●

sequential

42 This is the simplest way of organizing a file and the one most commonly used. It is the only practical way to store data on magnetic tape.

Payroll records filed in man number sequence is an example of a.......... file.

●●●

sequential

43 The field in each record which is used when sequencing the records is called the "control field".

The control field in the above payroll file example would contain an employee's..........

●●●

man number.

44 In sequential organization, the records are stored on the file in consecutive storage locations. Blocking the records further improves storage space efficiency. Sequential organization may be used with any storage device.

Data may be stored on..........(tape reels/disk packs/either tape reels or disk packs) in sequential order.

●●●

either tape reels or disk packs

45 Files on magnetic tape may be in sequential order only.

● ● ●

46 Some applications may demand access to a single record on an "as required" basis. To permit this type of operation, the file may be organized in "Random" order. Under random organization, the location or address where the record is to be stored is derived from the control field by means of an "algorithm" which is usually a mathematical formula. This algorithm is designed to uniformly distribute the records being stored throughout the designated storage area, although the records will not necessarily be in sequential order in the file.

Records are retrieved using.......... (the same/a different) algorithm.

● ● ●

the same

47 Since each record carries a unique control field, its address can be rapidly determined and that single record can be retrieved directly. This is called Direct Access and requires a storage device with special characteristics called a D.......... A.......... Storage Device or DASD.

● ● ●

Direct Access

48 A warehouse inventory file is a good example of the type of file which would require direct access.

The file is composed of records of a very large number of items. The stock status must be kept current on all items. However, only relatively few items are active at any one time.

Keeping the file current is best done by creating the file with.......... organization and processing only those items (records) for which there are transactions.

● ● ●

random (direct)

49 A single file may be used by many programs. Sometimes such a file must be processed either sequentially or directly depending on the particular data processing program.

An organization with this characteristic is an indexed-sequential file. Here the records are stored sequentially and indexes are generated for the file.

An indexed-sequential file may be stored on....... .... (tape/disk).

● ● ●

disk

50 Data Management maintains the indexes for the file normally on the same volume.

To do a direct access on an indexed-sequential file, the indexes are examined by Data Management routines to determine in what interval of the file the record is located, then only that interval is examined for the desired record. No special algorithm is required to store or retrieve a record.

An indexed-sequential file.......... (may/may not) be stored on magnetic tape.

● ● ●

may not

51 In order to provide maximum protection for a file, to prevent accidental destruction or erasure, to assure proper updating and maintenance, Data Management provides a means of file identification.

File identification is done by volume and file labels.

A label is a.........

a. a sticky piece of paper stuck on a tape reel with the file name on it.

b. a record identifying a tape reel or a data set.

TAPE FILE

ILLUSTRATING POSITION OF TAPE HEADER AND TRAILER STANDARD LABELS

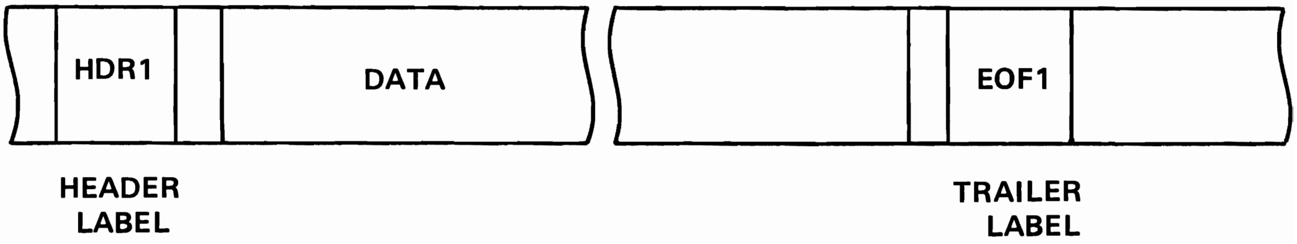


Figure 6.

TAPE FILE

ILLUSTRATING POSITION OF TAPE HEADER AND TRAILER, AND USER HEADER AND USER TRAILER STANDARD LABELS

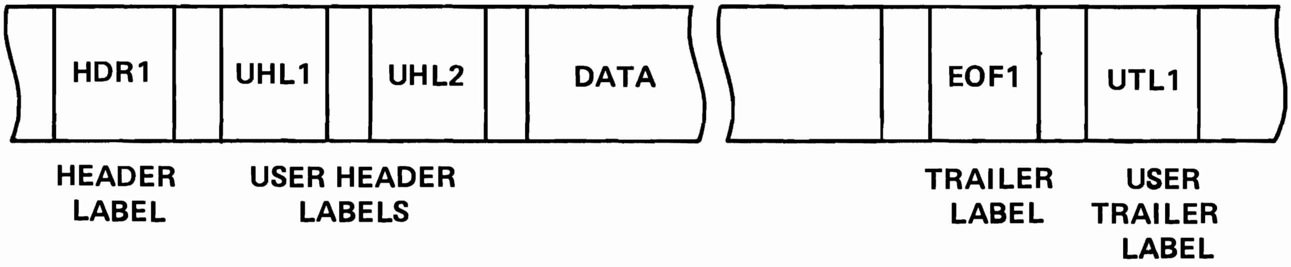


Figure 7.

c. a card slotted into the tape reel identifying the program

●●●

b

52 A label is a machine readable record associated with a specific storage device such as a reel of tape or a disk pack (volume label), or a particular data set (file label).

There are two general categories of labels-volume labels and file labels.

Volume labels identify...........

a. the program being run.
b. the file being used.
c. the tape reel or disk pack.

●●●

c

53 Each device (tape reel, disk pack, drive, etc.) receives a volume label when it is initialized. This volume label is checked but never altered by IOCS.

Labels, when defined by Data Management, are called S/360 standard labels. All others are called non-standard labels.

File labels are records which identify specific ....(data sets/tape reels).

●●●

data sets

54 DASD volumes must always contain S/360 standard labels. Magnetic tape reels may contain standard or non-standard labels.

LIOCS will not process...........(standard/non-standard) labels.

●●●

non-standard

55 The programmer must write his own label-checking and label-writing routines when using files with non-standard labels.

Each tape file with standard labels has at least two labels-a header label and a trailer label. The header label is placed at the...........of the file, and a trailer label is placed at the...........of the file.

●●●

beginning
end

56 These are illustrated in Figure 6.

●●●

57 Standard labels...........(are/are not) processed automatically by LIOCS.

●●●

are

58 In addition to the standard header and trailer labels, a file may have User Header and User Trailer labels. These 80-byte records, containing a standard identifier, are used by the programmer for more detailed identification of the file.

User Labels...........(are/are not) processed by LIOCS.

●●●

are not

59 Since these labels do not contain standard information, they are not processed by LIOCS but only handled and counted.

DASD files do not have trailer standard labels but may have user header and user trailer labels.

●●●

60 User labels are illustrated in Figure 7.

●●●

61 A DASD volume...........(may/may not) contain more than one file.

●●●

may

62 Data Management is composed of.......... and...........

●●●

IOCS routines (programs)
standards.

63 Data Management's main function is ....(your own words).

●●●

moving data between main storage and I/O devices.

You have completed this section. At this point, you should fill in your notes and take the self-evaluation quiz.

Self-Evaluation Quiz
Data Management

QUESTIONS

1. Data Management is composed of:

   a. blocked and unblocked records.
   b. volumes and files.
   c. IOCS routines and standards.
   d. header and trailer labels.
   e. PIOCS and LIOCS

2. A data file contains:

   a. physically related records.
   b. logically related records.
   c. volume labels.
   d. Data Management standards.

3. Which of the following is not defined by data management standards?

   a. Logical and physical record formats.
   b. Data file organization.
   c. Data file contents.
   d. Data file specification.
   e. Volume and file labels.

4. A logical record is:

   a. defined by the count and key fields.
   b. defined by its length.
   c. defined by its physical format.
   d. defined by its content and use.
   e. defined by the file label.

5. A physical record is:

   a. composed of one or more logical records.
   b. defined by the count and key fields.
   c. defined by its content and use.
   d. defined by the file label.
   e. defined by the volume label.

Match the records shown in the illustration with the proper definition below:

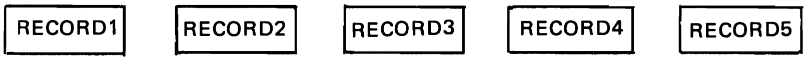
6. Format F, unblocked records

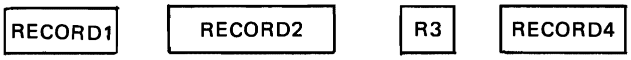
7. Format F, blocked records

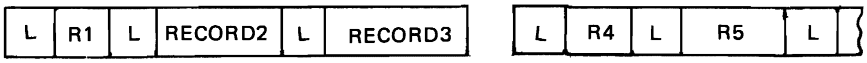
8. Format V, unblocked records

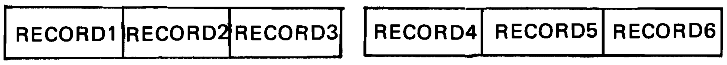
9. Format V, blocked records

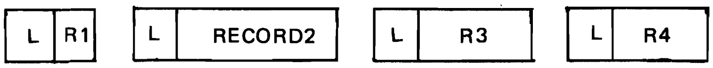
10. Format U records

a. 

b. 

c. 

d. 

e. 

Match the data file organization methods with the appropriate access methods.

11.	Sequential	a.	sequential or direct
12.	Indexed-sequential	b.	direct
13.	Random	c.	sequential

14. Which of the following is the identifying characteristic of a Format V record?

    a. It is fixed in length.
    b. It is variable in length.
    c. It may be blocked.
    d. It contains a standard length field.
    e. It is handled by LIOCS.

Match the following:

15. A label.

16. A volume label.

17. A file label.

18. Tape file labels.

19. Disk labels.

20. User header or trailer label.

a. Required for all volumes and files.
b. Contains non-standard label data.
c. Preformatted record, contains standard label data.
d. Identifies a particular data set.
e. Requires both a header and trailer label.
f. Identifies a particular data cell.

Self-Evaluation Quiz
Data Management

ANSWERS			Frame Reference
1.	c.	IOCS routines and standards	(1)
2.	b.	Logically	(39)
3.	c.	Data file contents	(8-14)
4.	d.	Defined by its content and use.	(8)
5.	a.	Composed of one or more logical records.	(9)
6.	a.		(26)
7.	d.		(26)
8.	e.		(27)
9.	c.		(27)
10.	b.		(28)
11.	c.	Sequential	(41)
12.	a.	Sequential or direct	(49)
13.	b.	Direct	(46)
14.	d.	It contains a standard length field.	(27)
15.	c.	Preformatted record, contains standard label data.	(51)
16.	f.	Identifies a particular data cell.	(53)
17.	d.	Identifies a particular data set.	(53)
18.	e.	Requires both a header and trailer label.	(55)
19.	a.	Required for all volumes and files.	(54)
20.	b.	Contain non-standard label data.	(59)

Magnetic Tape Concepts

MAGNETIC TAPE CONCEPTS

General Considerations

1 A magnetic tape unit provides low cost, efficient, auxiliary storage for programs, intermediate data, and large files of records. Its efficiency lies in its ability to record or read data in a long, continuous string, whose only practical limit is the number of main storage positions that can be reserved for I/O data.

As you may recall from Computing Systems Fundamentals, a tape unit records its data as it passes tape across a read-write head. Recording starts an instant after the tape drive starts moving, and stops an instant before the tape drive stops. This creates a gap in the recording process, at the beginning and end of each string of data. There may be one record or a series of records in each string. In order to distinguish them, each string of data is called a "physical record" or "block". The gap that separates the physical record is called the "interblock gap" (IBG). Each of the data records that make up a string is called a "logical record". The logical record is the basic unit of information used by the computer program. It is defined by the fields it contains, whereas a physical record is defined by the gaps that precede and follow it.

Statements:

1. An area in which no data is recorded, called a gap, separates each .... from the next, on tape.
2. There may be one or more .... in each physical record.
3. There are no .... between logical records in a block.
4. A logical record is defined by the .... that it contains.

●●●

1. physical record, or block
2. logical records
3. gaps
4. fields

2 The number of logical records in each physical record is called the "blocking factor". It is determined by the programmer, from the number of bytes of storage available for I/O and from processing considerations. Ignoring the latter for a moment, suppose that 80-byte card records are being processed, and the programmer sets aside a 400-byte output area for writing on tape. What would the blocking factor be?

●●●

5 (400 = 80 x 5)

3 With blocking, the efficiency in the use of magnetic tape increases as the volume of the file increases. For example, one tape unit records data with a density of 800 bytes per inch. The length of an interblock gap is .6 inch. How much data could be recorded in the space used for each gap?

●●●

480 bytes

4 If we were recording 100 byte records unblocked (one logical record per physical record), we could put about 39,000 of them on a full reel of tape. If we used a blocking factor of 50, we could record 209,000 of them on the same reel. This increase in efficiency is due to a reduction in the number of ....

●●●

interblock gaps.

5 Another point about efficiency can be shown by considering tape-passing time – the time required to process a reel of tape.

Suppose that we have sixty thousand 900-byte records on a reel of tape, and that the tape unit takes 8 milliseconds to pass an interblock gap.

With unblocked records, we would have 60,000 gaps. With a blocking factor of 3, we would have 20,000 gaps. By avoiding having to pass 40,000

ADDITIONS, CHANGES, AND DELETIONS

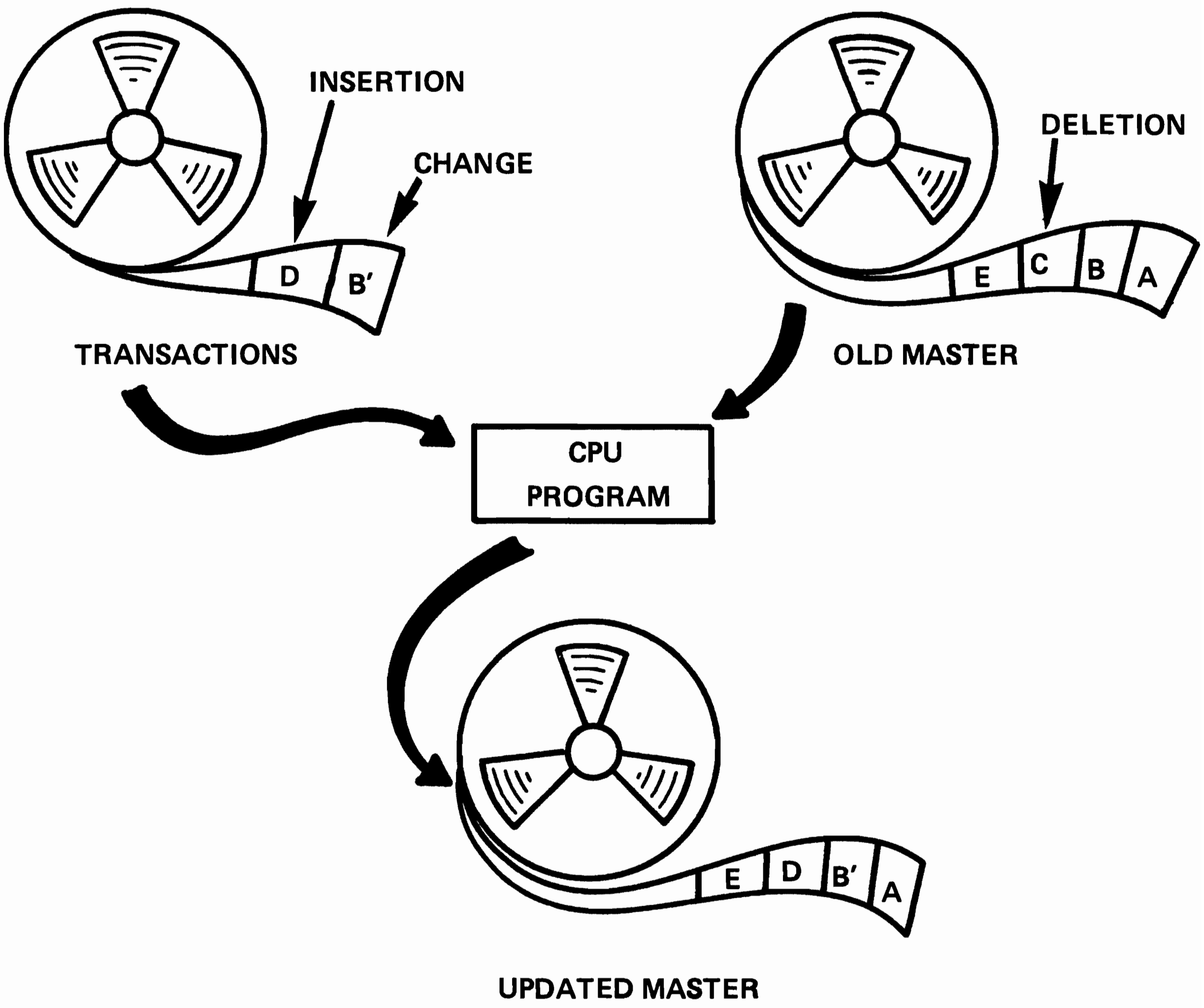


Figure 1.

gaps, we would save 40,000 x .008 sec. = 320 sec. = 5 minutes, 20 seconds.

When handling large files of records on tape, blocking increases efficiency by .... the number of records per reel and by .... the tape passing time.

●●●

increasing
decreasing

6 Decreasing tape passing time, of course, decreases the overall processing time for a job - an important consideration. When, then, would we not block records?

IOCS, as you may recall, is responsible for blocking and deblocking records. This takes time. In cases where there is little processing to be done on the data from each record, the CPU may be idle while waiting for IOCS to examine a block and deliver a given logical record to a work area in storage. This may even happen when alternate I/O areas are used to keep records coming, in as continuous a stream as possible: processing may be so brief that the net running time, with unblocked records, is less.

To block or not to block? Here is a summary of the considerations:

1. Blocking increases the .... by reducing the number of interblock gaps.

2. Blocking decreases the .... since fewer interblock gaps must pass the read-write head.

3. Blocking may not be an advantage, if .... is brief, compared to the time taken by .... to block or deblock the records.

●●●

number of records on a reel of tape
tape-passing time
processing, IOCS

Organization of Data

7 Data is arranged sequentially on tape. When a program is written out on tape, its instructions are in the same sequence as they were in core storage. When intermediate results of calculations are recorded, one work area after another is written out.

When card records (or records created from cards) are written on tape, they are sequenced according to the contents of one or more control fields. For example, records of subscribers, which are to be printed from a tape, would be written on the tape in alphabetic order, based on the subscriber name field. The same records, if used for billing, would be written on the tape in account number sequence.

Sequential organization is the familiar arrangement that card files and other types of files have often used in the past. A difference exists, however, when it is necessary to add or delete a record on tape: To maximize efficiency, records are written consecutively. There is no space left for insertions, and deletions must not leave gaps.

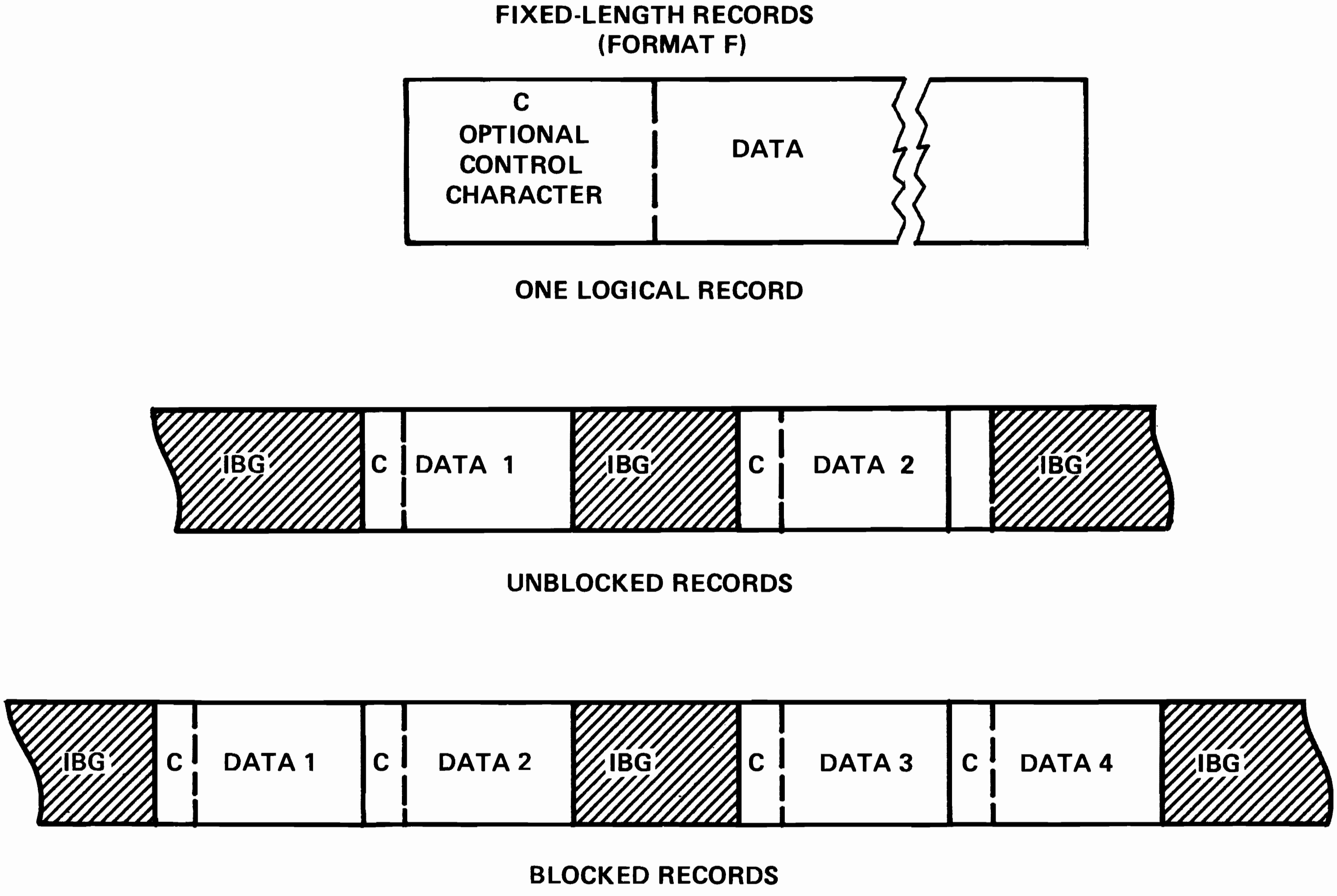
The procedure is shown in Figure 1. When there are insertions or deletions to a master tape, an .... must be created.

●●●

updated master

8 This brings us to another point about the efficient use of tape, aside from its value for large files of records: Tape is best used, if there is a lot of activity against the file. That is, since a new file must be written if there is any change in any record, the process increases in efficiency with a higher percentage of changes.

To use absurd cases as examples, if we had a file of 100,000 records, and our total processing run involved the insertion of one record, a card file would be a far more efficient method. If every one of those records had to be changed, however, tape would be the best medium for the job.



FIXED-LENGTH RECORDS
(FORMAT F)

ONE LOGICAL RECORD

UNBLOCKED RECORDS

BLOCKED RECORDS

Figure 2.

Which of the following files would most likely be recorded on magnetic tape?

a. 100,000 magazine subscriptions, of which approximately 2,000 are updated for renewal or deletion each week.

b. 100,000 credit accounts, in a large department store, of which approximately 50,000 must be updated each month.

● ● ●

b.

9 It is somewhat of a disadvantage to have to batch transactions, so that they represent a high percentage of activity against a master file: the master file is not up to date until the batch is processed, and a separate run is required to sort the transactions into the same sequence as the master file records. The programming required to process a sequential file, however, is straightforward, compared to that for other types of file organization.

List the advantages and disadvantages of sequential file organization on tape.

● ● ●

Advantages:
1. Efficient use of storage medium.
2. Efficiency increases with activity against the file.
3. Straightforward programming.

Disadvantages:
1. Transactions against the file must be batched and sorted into its sequence.
2. A new file must be created whenever there are insertions or deletions, no matter how few.

Formats

10 In order for IOCS to handle the reading and writing of records on tape, a number of facts about the file must be specified. These include:

a. Block size - the length of a block (physical record) in bytes.
b. Record size - the length of a logical record in bytes.
c. Whether the records are fixed length (F) variable length(V), or undefined (U), as well as whether they are blocked or unblocked.

The simplest case is for fixed-length records, schematicized in Figure 2. IBG stands for interblock gap; data stands for the data record (of whatever length). The only item that is new is the control character. This is a special character which may be placed (by the programmer) in the first byte of each logical record. It is used for carriage control, in a tape-to-printer operation, or to control stacker selection, in tape-to-card punch.

1. What is the blocking factor of the blocked records in Figure 2?
2. What is the "blocking factor" of unblocked records?

● ● ●

1. 2
2. 1 (one logical record per physical record)

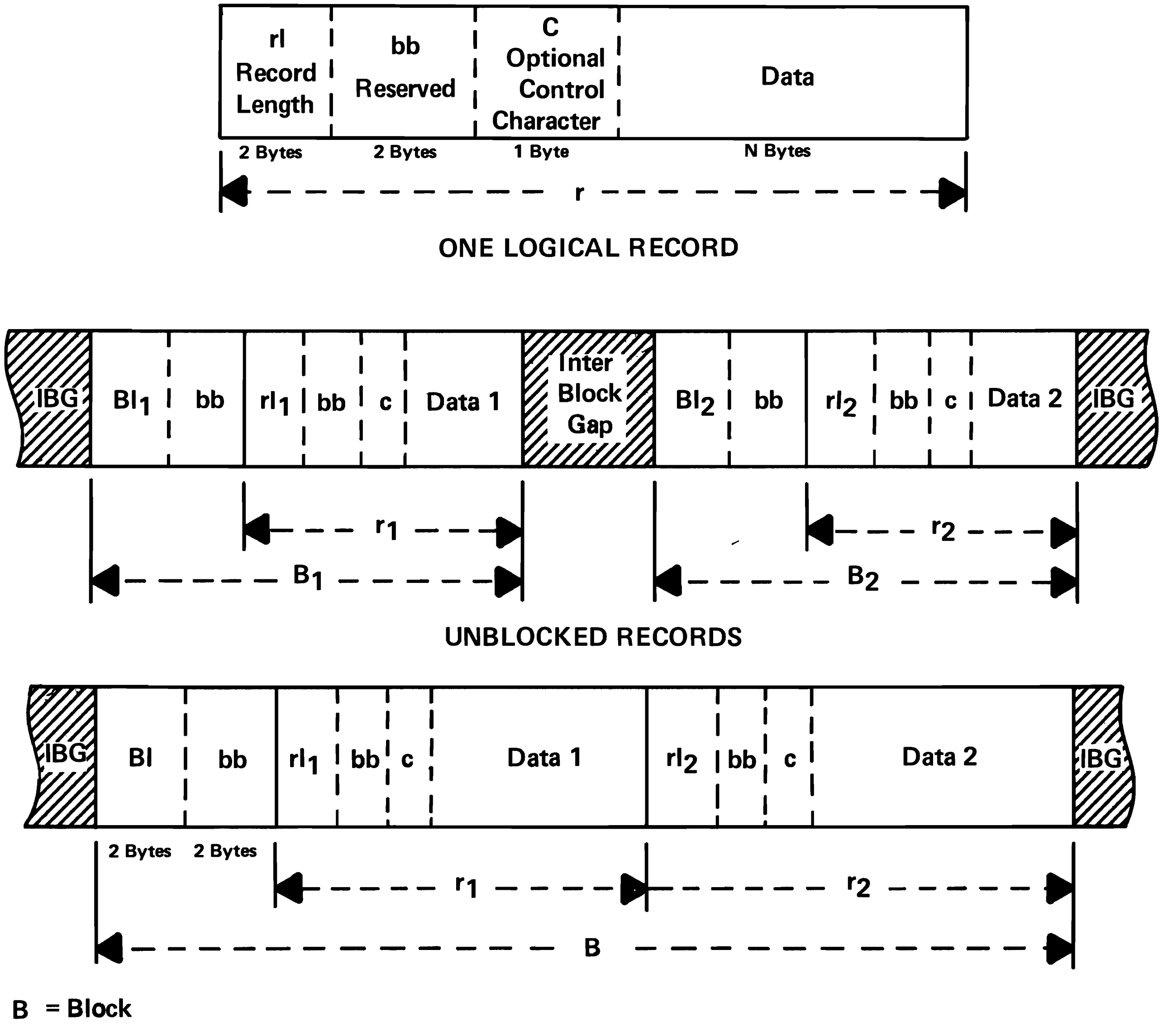
11 If the records are unblocked, IOCS will cause them to be read into a specified input area (or out of a specified area) one at a time, on command.

If they are blocked, IOCS will cause an entire block to be read in, or written out, each time. If the programmer has specified a work area separate from the input area, IOCS will move the first logical record of the block into it. If he has not, it will establish the symbolic name of the input area as the address of the first logical record in the block.

As each command to get a record is issued, IOCS directs the program to the next logical record in the block. This is called "deblocking". After the last record has been processed, and when the next command to get a record is issued, IOCS causes the next block of records to be read into storage.

VARIABLE - LENGTH RECORDS

(FORMAT V)



ONE LOGICAL RECORD

UNBLOCKED RECORDS

BLOCKED RECORDS

Figure 3.

IOCS makes it seem as if the program is causing records to be read in one at a time:

a. if the records are unblocked.
b. if the records are blocked.

●●●

Either a. or b. is true

12 While the records in most files are fixed length, some must be variable. Examples include the transactions against a checking account in a given accounting period, the parts lists for assembled items, the names of assemblies in which each part is used, and so on.

A variable-length record is not infinitely variable: It may be as small as its control field (no data to go with it, but still in the file) or as large as some pre-specified fixed number of bytes (the largest expected record length).

IOCS must be told how large the maximum number of bytes in a record or block will be, so that it does not exceed this maximum when bringing data into storage or writing data from it. It also uses record length and block length when deblocking records.

Thus the first information in each block must be a numerical field giving the length of the block, and the first information in each record must be ....

●●●

a numerical field giving the length of that record.

13 Figure 3 shows blocked and unblocked variable length records (format V). First consider the sample logical record shown at the top. The entire record, shown by r, contains a two-byte record length field, two bytes (bb) reserved for use by the system, a one-byte optional control character, and the data record itself.

If the length of r is 105 bytes:

a. What numerical value is recorded in the rl field?
b. What is the length of the data record?

●●●

a. 105 (It gives the length of the logical record.)
b. 100 bytes (Five bytes are used for the rl, bb, and c fields.)

14 Now look at the blocked records at the bottom of Figure 3. At the beginning of the block, a two-byte field gives the length of the block (the <u>real</u> length, not the maximum referred to previously). Then there are two bytes reserved for use by the system, and the rest of the block is divided into logical records r_1, r_2).

1. Does each of the logical records in the block have the same format as the example shown at the top of Figure 3?
2. How many more bytes does a block require, over and above those used for logical records?

●●●

1. Yes (record length, reserved space, optional control character, and data).
2. 4 (two for the block length field and two for reserved space).

15 Since at least 480 bytes are saved by the elimination of an interblock gap, the bytes used to specify block length and record length are (considerable/negligible) ....

●●●

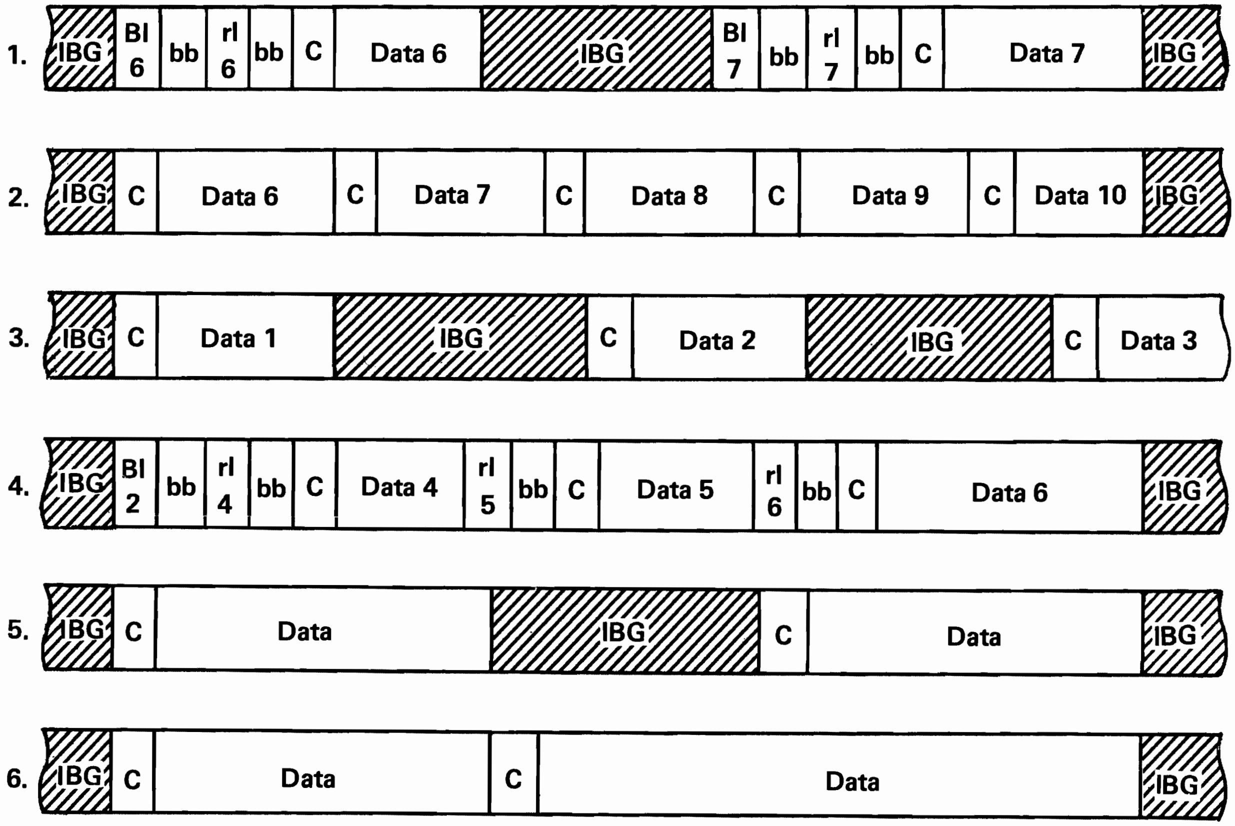
negligible.

16 Now consider the unblocked records, illustrated in the middle of Figure 3. In addition to the two-byte field for record length and the two reserved fields, there is a two-byte .... field, for each unblocked logical record.

●●●

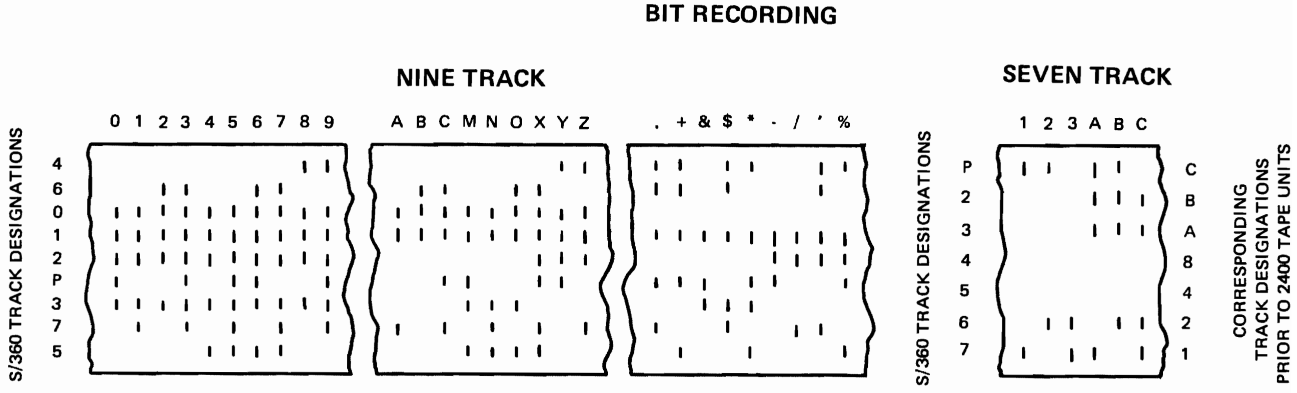
block length (Bl_1 for record r_1, Bl_2 for record r_2, etc.)

Record Formats



1. IBG Bl 6 bb rl 6 bb C Data 6 IBG Bl 7 bb rl 7 bb C Data 7 IBG
2. IBG C Data 6 C Data 7 C Data 8 C Data 9 C Data 10 IBG
3. IBG C Data 1 IBG C Data 2 IBG C Data 3
4. IBG Bl 2 bb rl 4 bb C Data 4 rl 5 bb C Data 5 rl 6 bb C Data 6 IBG
5. IBG C Data IBG C Data IBG
6. IBG C Data C Data IBG

Figure 4.



Vertical Lines Represent One-Bits

ONE-HALF INCH TAPE

Figure 5.

17 Why block lengths for unblocked records? If IOCS had to include routines for handling unblocked records, as well as routines for handling blocked ones, it would have to occupy considerable storage space. In order to maximize its efficiency, System/360 IOCS uses the same routines to handle both cases.

IOCS treats unblocked records as if they were blocked, with a blocking factor of ....

●●●

1

18 Indicate, for each of the first four tape illustrations in Figure 4, whether the format is fixed or variable length, with blocked or unblocked records.

●●●

1. Variable length, unblocked
2. Fixed length, blocked
3. Fixed length, unblocked
4. Variable length, blocked

19 What is the blocking factor in 2? In 4?

●●●

2. 5 (the illustration shows a block containing records 6-10).
4. 3 (the illustration shows block No. 2, containing records 4-6).

20 Illustrations 5 and 6, in Figure 4, show unblocked and blocked records with an undefined (U) format. Note that the logical records are not shown as Data 1, Data 2, etc.

An example of U-type record files is an existing file which was created and used in pre-S/360 systems such as the 1400 and the 7000 series.

Which is blocked, and which is unblocked in Figure 4?

●●●

6. is blocked and 5. is unblocked

21 Actually, undefined records are variable-length records whose maximum size has been specified for the system, but which do not carry the standard two-byte length fields and reserved spaces. They can contain the optional control characters, but they usually do not. If a programmer exercises that much control over the format of an originally undefined record, he usually turns it into a standard variable-length record, with the length fields and reserved spaces specified.

IOCS treats an undefined physical record as if it contains one logical record. If blocking exists, it is up to the problem program to identify the component records and deblock them.

From the foregoing, do you think that it is likely or unlikely that a programmer would specify blocked, undefined records to be written on tape?

●●●

unlikely (They may as well be unblocked records, since IOCS treats them that way.)

Magnetic Tape Recording

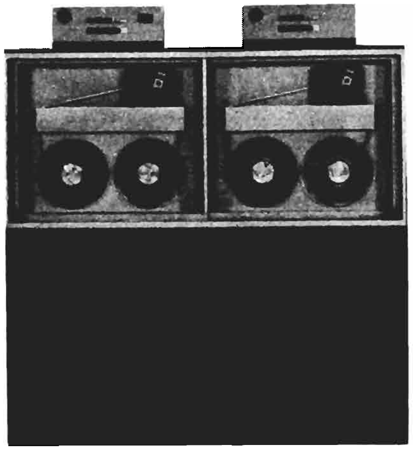
22 Magnetic tape is a flexible plastic tape coated on one side with a magnetizable iron compound. Different System/360 tape drives record data in either 7 or 9 tracks.

The two recording methods are shown in Figure 5. The characters above each of the tape sections are coded by the bit patterns shown below them, in the nine-track and seven-track formats. The characters to the left of each tape section give the track designations. Tracks designated as "P" or "C" are non-data tracks used for checking data and correcting errors.

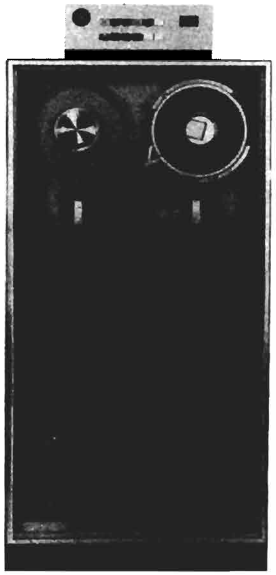
Regardless of the number of tracks used, each character of data is recorded (across/along) .... the tape.

●●●

MAGNETIC TAPE UNITS



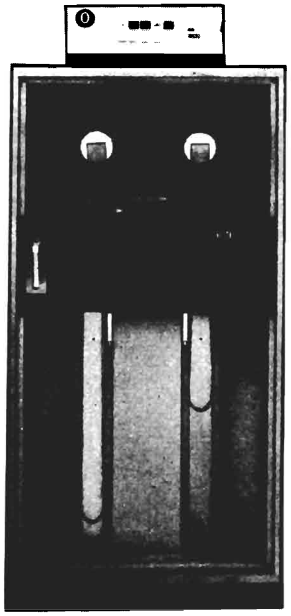
2415



2420

Characteristics	2415 tape units		2420 Tape Unit
	Model 1-3 Model 4-6	Model 1-3	Model 7
Number of Tracks	9-Track	7-Track	9—Track
Density (Bytes per inch)	800 1600	a. 800 b. 556 c. 200	1600
Data Rate (Bytes/sec)	15,000 30,000	a. 15,000 b. 10,425 c. 3,750	320,000
Tape Speed (In/Sec)	18.75	18.75	200
Interblock Gap (Inches)	0.6	.75	0.6

Characteristics	2401 Tape Units					
	Model 1 Model 4	Model 1	Model 2 Model 5	Model 2	Model 3 Model 6	Model 3
Number of Tracks	9-Track	7-Track	9-Track	7-Track	9-Track	7-Track
Density (Bytes per inch)	800 1600	a. 800 b. 556 c. 200	800 1600	a. 800 b. 556 c. 200	800 1600	a. 800 b. 556 c. 200
Data Rate (Bytes/Sec)	30,000 60,000	a. 30,000 b. 20,850 c. 7,500	60,000 120,000	a. 60,000 b. 41,700 c. 15,000	90,000 180,000	a. 90,000 b. 62,500 c. 22,500
Tape Speed (In/Sec)	37.5	37.5	75.0	75.0	112.5	112.5
Interblock Gap (Inches)	0.6	.75	0.6	.75	0.6	.75



2401

Figure 6.

across

23 Each 8-bit byte of data on nine-track tape is associated with (how many?)........... parity (P) (checking) bit(s).

●●●

1

24 Seven-track tape is a special case. Users of tape systems prior to System/360 have libraries of seven-track tape that may number in the thousands of reels. In order for them to use System/360, a conversion feature is made available. The converted tape unit reads <u>and</u> writes in the binary coded decimal format (shown to the right of the seven-track tape segment), or it can read seven-track tape and <u>another</u> tape unit can be used to write the results out on nine-track tape.

Which of the foregoing would an installation use, in each of the following cases?

a. Seven-track master files are updated on System/360, using transaction tapes produced on prior equipment, still installed at branch locations.
b. Seven-track tapes are processed and new tape files are created, to convert to nine-track tape units.

●●●

a. Read and write seven-track tape.
b. Read seven-track tape and write nine-track tape.

25 A byte on nine-track tape represents any of the types of data that a byte can represent in storage. These include..........

●●●

an 8-bit binary field.
any one of 256 EBCDIC characters.
two hexadecimal digits.
two decimal digits (packed format).
one signed decimal digit (zoned format).

26 Another way to put this is to say that these magnetic tape units are "code insensitive". Byte-for-byte they will accept and record anything that is sent to them from main storage.

If an output record contains alphanumeric data and packed numeric fields, could it be recorded, as such, on tape?

●●●

yes

27 This code insensitivity is extremely important for the versatility of magnetic tape as a medium. By contrast, a printer will respond, appropriately, only to EBCDIC coding.

●●●

<u>Tape Units</u>

28 Tape units currently available with System/360 are shown and their characteristics are tabulated, in Figure 6. Notice that one tape unit contains more than one tape drive. It is the..........

●●●

2415

29 The 2415 comes with 2, 4, or 6 independently operating tape drives, under the control of one, single channel tape control unit. Since it will be connected to a single selector channel, which of the following would be possible?

a. One of the tape units could be writing tape, while another is reading tape and still another is rewinding.
b. One of the tape units could be reading <u>or</u> writing tape while another is rewinding and still another is being loaded with a new reel.

●●●

b. Recall that only one input or output operation can occur on a selector channel at one time.

30 By contrast, the 2401 tape unit, which does not have its own control unit, can be connected either to a single channel control unit or to a two-channel control unit. Which do you think would be used for simultaneous read-while-writing operation of two tape drives?

●●●

The two-channel control unit.

31 Up to eight 2401 tape units can be connected to either the single channel, or two-channel control unit, and 8 control units can be connected to a single selector channel. What is the maximum number of tape units that can be connected to a selector channel?

●●●

64

32 The 2420 Tape Unit model 7 has a completely new type of tape transport which minimizes mechanical delay and wear. It uses IBM series 500, Dynexcel, or heavy duty half-inch magnetic tape. It features automatic threading and uses a wraparound cartridge for maximum tape protection.

As with the 2401 Tape Drive, up to...........2420 model 7's can be connected to a single selector channel, through a single control unit.

●●●

8

33 The tables in Figure 6 give the number of tracks, density, data rate, etc. for all models of all tape units that are currently used with System/360. The diagonal lines in some of the columns separate the corresponding data for different models.

For example, the 2401 Model 1, operating in 9-track mode, records data at a density of 800 bytes per inch; the model 4, operating in the same mode, records at...........bytes per inch.

●●●

1600

34 Letters a.,b., and c., in one row of a column give data that corresponds to the same lettered entries in another row of that column. Thus, the 2401 model 1 can record in any one of three densities: Opposite a., in the Density row, we see 800 (bytes per inch). In the next row down we see that the corresponding data transmission rate is 30,000 bytes per second.

On the same model, what data rate will result from a recording density of 200 bytes per inch?

●●●

7500 bytes per second

35 Now find the model with each of the following:

a. The fastest 7-track data rate.
b. The fastest 9-track data rate.

●●●

a. 2401 model 3 (90,000 bytes per sec.)
b. 2420 model 7 (320,000 bytes per sec.)

36 The overlapping data rates, from one model tape unit to another, show that data rate depends on:

a. Recording density
b. Tape speed
c. Neither
d. Both

●●●

d. Data rate depends on both recording density and tape speed.

37 One additional feature of all System/360 Tape drives is their ability to read backwards. Having written a file (or part of a file) of unblocked, fixed-length records, it is possible to start reading them immediately, in reverse order. This is of particular importance in a sorting operation, where the work tapes have to be read many times. The tape unit can simply read backward over a tape that has just been written.

The read backward feature saves (read/rewind) ....time.

●●●

rewind

You have completed this unit. Fill in your notebook, before taking the Self-Evaluation Quiz.

Self-Evaluation Quiz
MAGNETIC TAPE CONCEPTS

QUESTIONS

1. Why is magnetic tape an efficient storage medium?

2. What is the effect of blocking on:

   a. The number of records per reel.
   b. Tape passing time. Why?
   c. Main storage requirements.

3. When may processing blocked records not be faster than processing unblocked records?

4. Define:

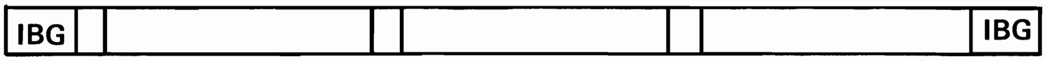
   a. A physical record.
   b. A logical record.

5. What is the "blocking factor"?

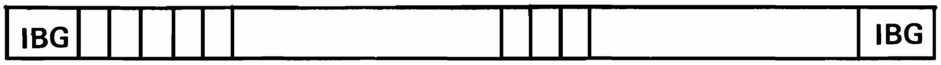
6. What are the advantages and disadvantages of sequential organization?

7. For each of the following formats, the data records, if sequential, are in the order Data 1, Data 2, Data 3, etc. Show what each field would be, using Bl for block length, rl for record length, bb for reserved space, and C for control character.

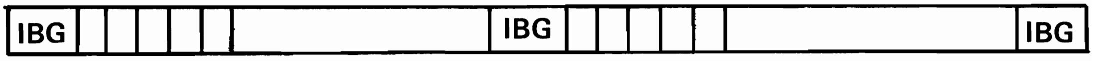
   a. Fixed-length, blocked records:



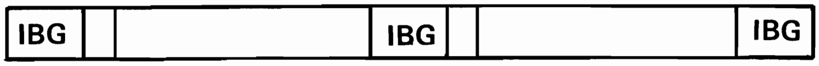
   b. Variable-length, blocked records:



   c. Variable-length, unblocked records:



   d. Undefined, unblocked records:



8. Why do variable length, unblocked records need block length fields?

9. What must the programmer do, if he wishes to process undefined, blocked records?

10. What is the reason for having seven-track tape reading and writing facilities available on System/360 tape units?

11. What distinguishes the tape reel used in the 2420 model 7 from the tape reel in other magnetic tape units?

12. Upon what does the data rate of a magnetic tape unit depend?

1. Data can be recorded on tape in a long, continuous string. (1)

2. a. Blocking increases the number of records that can be stored on a reel. (4)

   b. Blocking decreases the tape passing time, because fewer interblock gaps must pass the read-write head. (5)

   c. Blocking increases the size of main storage areas that must be reserved for I/O. (1)

3. When processing time is short, compared to deblocking time. (6)

4. a. A physical record is the continuous string of data recorded between two interblock gaps. (1)

   b. A logical record is the basic unit of information used by the computer program. It is defined by the fields it contains.

5. The blocking factor is the number of logical records in each physical record. (2)

6. Advantages: (9)

   a. Efficient use of the storage medium.
   b. Efficiency increases with activity against the file.
   c. Straightforward programming.

   Disadvantages:

   a. Transactions must be batched and sorted in the file's sequence.
   b. A new file must be created whenever there are insertions or deletions.

7. a. Fixed-length, blocked records: (17)

IBG	C	Data 1	C	Data 2	C	Data 3	IBG

   b. Variable-length, blocked records:

IBG	Bl_1	bb	rl_1	bb	C	Data 1	rl_2	bb	C	Data 2	IBG

   c. Variable-length, unblocked records:

IBG	Bl_1	bb	rl_1	bb	C	Data 1	IBG	Bl_2	bb	rl_2	bb	C	Data 2	IBG

d. Undefined, unblocked records:

IBG	C	Data	IBG	C	Data	IBG

8. So that IOCS can maximize its efficiency by handling them as blocked records, with a blocking factor of 1. (16)

9. He must write his own record identification and blocking/ deblocking routines. (20)

10. Seven-track tape allows tapes created on prior computer systems to be processed by System/360. (23)

11. It is a wraparound reel that provides maximum tape protection. (32)

12. Recording density and tape speed. (35)

Direct Access Storage Devices

DIRECT ACCESS STORAGE DEVICES

General Considerations

1 Direct access storage (disks, drums, data cell) is high capacity auxiliary storage, with a wide range of data rates. It is used for storing programs, intermediate data, and files of records. Its chief feature, as its name implies, is its ability to locate any record directly, without having to read preceding records. This feature greatly facilitates many types of data processing jobs; without it, some jobs could not be performed at all.

For example, an airline needs to keep track of available seating space, at all times, in order to maximize the efficiency of its reservation system. If a passenger cancels his trip, or stops over somewhere unexpectedly, the system must be able to sell that space to someone else.

The status of every space, on every flight, can be recorded on a direct access storage device (DASD). Notice of cancellation can be sent in from one remote terminal and, an instant later, a request for the space can be confirmed via another terminal, possibly in another city. The file continuously reflects the flow of requests, cancellations, "holds" (while space is sought on connecting flights), and so on.

Processing of inquiries about status requires access to the records. It would not be feasible with a serial (sequential) device, such as a magnetic tape drive or a card reader.

Another type of DASD application is one in which the processing of a transaction against one file requires the accessing of a different file. For example, change in rate of pay data, processed against an employee master file, requires accessing a tax table record. All the files required for the job can be stored on the same DASD. This is another situation that would not be feasible with a serial device.

DASD allows routines and tables of data to be stored, and accessed when required, thus providing for more flexible and efficient use of main storage.

For multiprogramming, or a time sharing system, they are essential: Main storage could not hold all of the various users' programs at the same time, and magnetic tape drives could not access them rapidly enough.

Even some types of batch processing jobs can be made more efficient with DASD. For example, records of sales are cumulatively updated, on a direct access basis, as they are reported throughout the week. They are held as intermediate data. Then a commission run is performed, with the records processed in sequence, by employee number.

Unlike records stored in a sequential device, records in a DASD can be processed either directly, or ....

●●●

sequentially.

2 Along with the unique tasks that they perform, DASD generally increase processing efficiency:

a. With the libraries of processor programs on DASD (always available to the system), the length of time required for a job is reduced by the reduction of setup time.

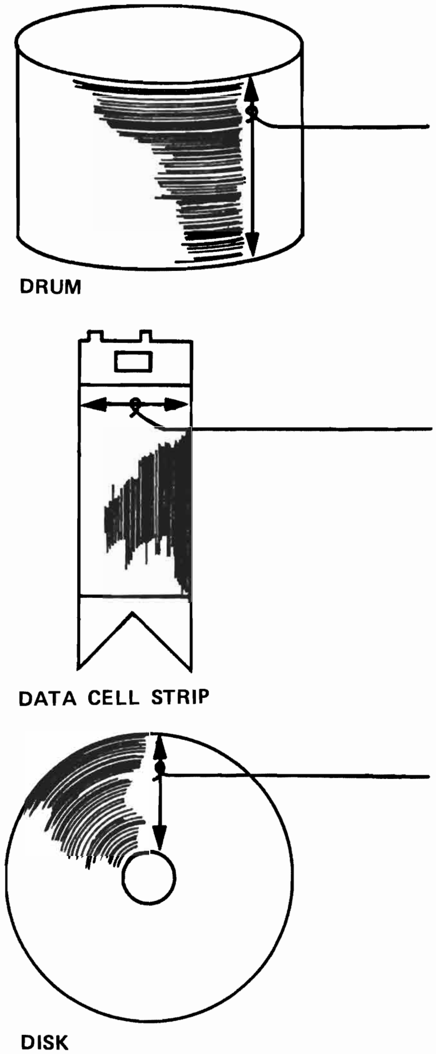
b. With the data files on line, presorting is reduced or eliminated.

c. Peak loads can be reduced by processing more often, with small batches, or by performing some jobs inline.

List the advantages of DASD, over sequential devices, and the ways in which they increase processing efficiency.

●●●

SYSTEM/360
DASD TRACK ARRANGEMENT



2301 : 200 Tracks
20483 Maximum Data Bytes
Per Track

2303 : 800 Tracks
4892 Maximum Data Bytes
Per Track

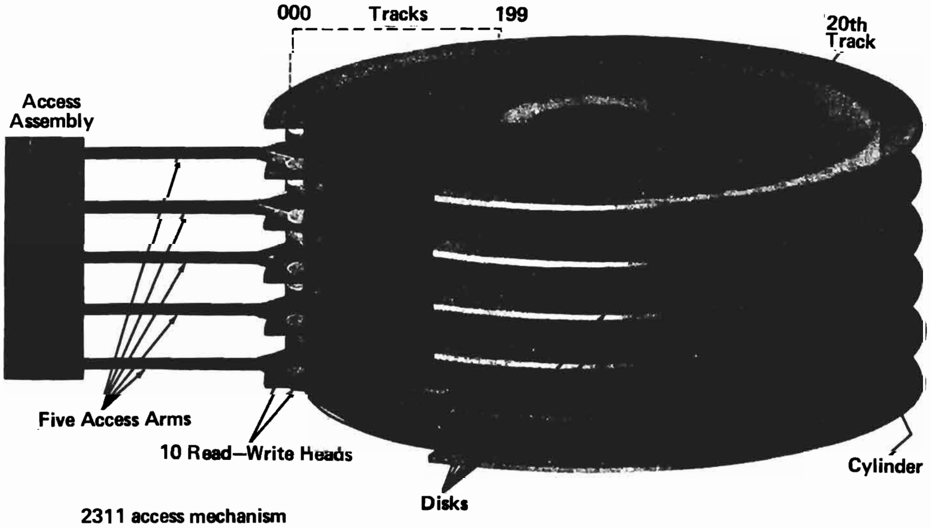
2321 : 100 Tracks
2000 Maximum Data Bytes
Per Track

2302 : 500 Tracks Per Surface
4984 Maximum Data Bytes
Per Track

2311 : 200 Tracks Per Surface
3625 Maximum Data Bytes
Per Track

2314 : 200 Tracks Per Surface
7294 Maximum Data Bytes
Per Track

Figure 1.



2311 access mechanism

Figure 2.

1. They can access any record, without extensive searching of other records. This allows:

   a. Inline processing of transactions.
   b. Accessing records from one file during the processing of transactions against another file, with both files on the same device.
   c. Processing a file of records either directly, as transactions are reported, or sequentially in a batch.

2. They reduce setup time by holding processing programs, and accessing them when needed.

3. They decrease the length of time required for jobs by reducing setup time and reducing, or eliminating, presorting.

Recording on DASD

3 DASD recording is done on magnetizable surfaces:

a. The outside surface of a polished cylindrical metal drum.
b. Both the upper and lower surfaces of a disk.
c. The coated side of a plastic data cell strip.

For reading or recording, the data cell strip is picked out of a sub cell (a holder for 10 strips) and wrapped around a drum, with its coated side out.

Recording on a data cell strip is very similar to recording on a (disk/magnetic drum)..........

●●●

magnetic drum (When a strip is wrapped around the drum, it is like the surface of a magnetic drum.)

4 On all three types of DASD, the basic recording unit is the track – the area of the magnetic medium that passes under one read/write head.

Figure 1 shows the track designations for the three types of DASD. A track forms:

a. A band around the surface of a drum.
b. A straight line along a data cell strip.
c. A circle around the axis of a disk.

Read/write heads are attached either to fixed or movable mountings. The mountings on magnetic drums are fixed – each read/write head is always positioned at the same track.

How many read/write heads must the 2303 magmetic drum have?

●●●

800 (One read/write head for each track.)

5 The read/write heads for disk drives are mounted on access arms that move them from track to track, across the disks. Disks are held, separated from one another, in a vertical stack, and the access arms move in and out between them. Each access arm holds two read/write heads, for the disk surfaces directly above and below it.

Figure 2 shows this access assembly for the 2311 Disk Storage Drive. The upper surface of the top disk and the lower surface of the bottom disk are not used for recording. They help protect the other surfaces.

Since the read/write heads are positioned one above another, if the top head is positioned at the 20th track in from the edge, on the underside of the top disk, what is the position of every other read/write head, with respect to the other recording surfaces?

●●●

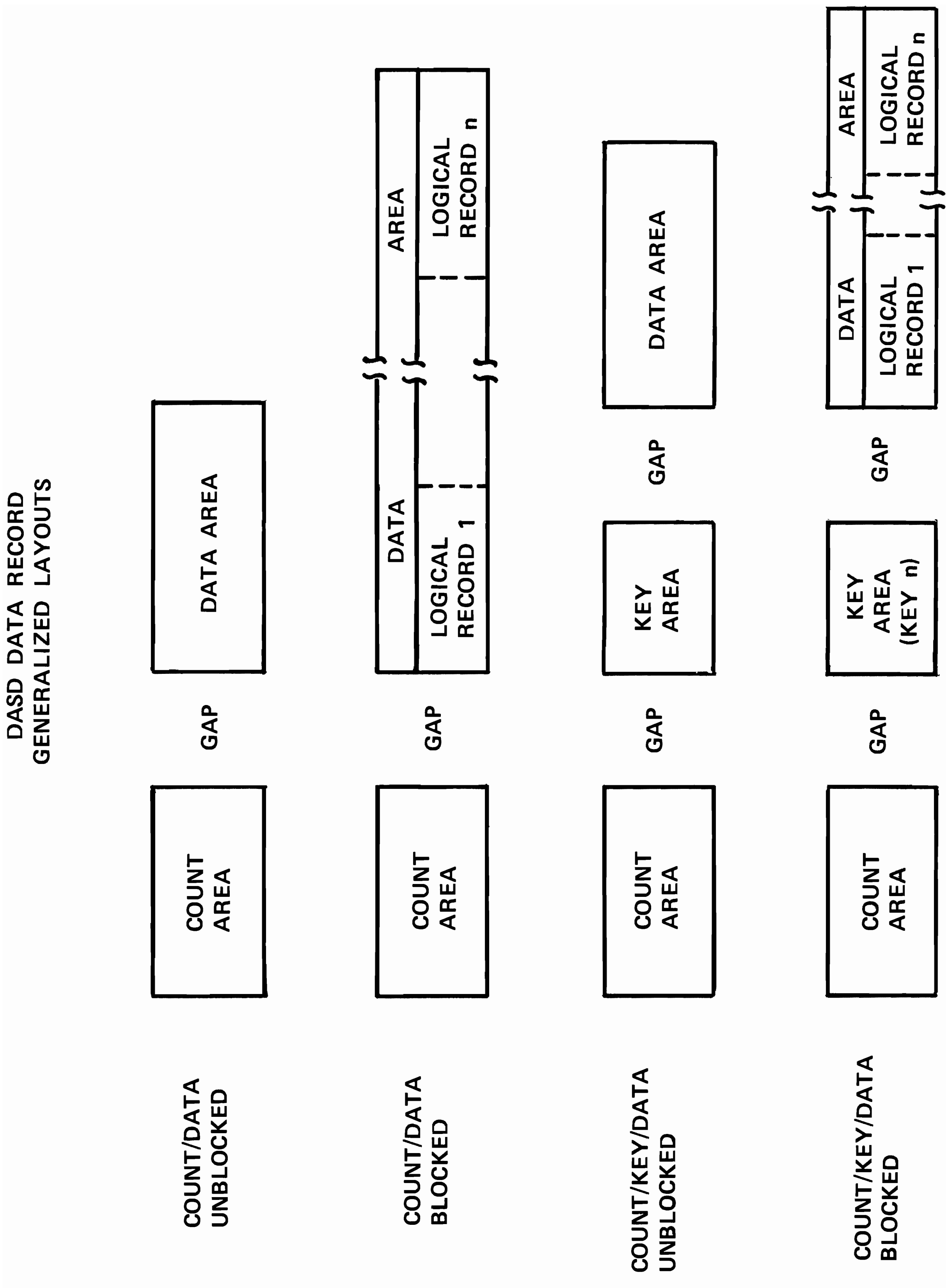

DASD DATA RECORD
GENERALIZED LAYOUTS

Figure 3.

Each read/write head is at the 20th track in from the edge, on its associated recording surface.

6 The tracks that are available for reading or recording, at one position of the access mechanism, are said to make up a cylinder. How many tracks are there in one cylinder of the 2311, as illustrated in Figure 2?

●●●

10 (One on the underside of the top disk, one on the upper surface of the bottom disk, and one on each surface in between.)

7 The track numbers show the tracks that are addressable by the program. Actually, there are three more tracks, on each surface, called "alternates". One of these is used by the IOCS, if one of the addressable tracks becomes defective.

a. The programmer can address (how many?) .... tracks on one surface of a 2311 disk.

b. The access mechanism can stop at (how many?) .... positions as it moves the read/write heads across the disks.

●●●

a. 200 (000 to 199, inclusive)
b. 203 (000 to 202, inclusive)

8 There are (how many?) .... addressable cylinders of tracks in the 2311.

●●●

200 (one for each addressable position of the access mechanism)

9 Every track on a drum is accessed, at all times, by the fixed read/write heads. How many cylinders of tracks does a drum have?

●●●

one (it contains all the tracks)

10 To be accurate, a drum always has only one physical cylinder comprising all of the tracks, but it may be divided into multiple cylinders, electronically, for addressing purposes. For example, the 2303 has one 800-track cylinder, but the addressing system divides it into (how many?) .... cylinders with 10 tracks in each.

●●●

80

11 A 2321 data cell strip is read by 20 read/write heads that can be shifted, as a group, into five access positions. How many cylinders does it have?

●●●

5 (Each cylinder is defined by an access position)

12 How many tracks are there in each cylinder of the 2321 data cell?

●●●

20 (one for each read/write head)

Track Format

13 There are two basic data record layouts, called count/data and count/key/data, used in System/360. Figure 3 shows them, in combination with the additional option of blocking. Note that the various types of areas are separated from each other by gaps in the recording process, as on magnetic tape.

Each count area and key area (if present) is used by the system to identify the following physical record. Which is true?

a. If records are unblocked, the count and key areas relate to the following logical record.

b. If records are blocked, the count and key areas relate to the entire block that follows them.

●●●

Both a. and b. are true. Recall that a physical record contains one unblocked record or one block of records.

14 Note that a given data record contains:

a. A count area.
b. A key area (if used).
c. A data area.

1. A data record contains (how many?) .... physical record(s).

2. A data record contains (one/one or more) .... logical record(s).

●●●

one
one or more (depending on whether they are unblocked or blocked)

15 The count area is made up of:

1. An address (called the "record ID") showing the device address of the physical record. This includes:

   a. A two-byte cylinder number.
   b. A two-byte track number.
   c. A one-byte record number (record 1, 2, etc., on that track).

2. A one-byte key length field. This field is 0, if the count/data recording method is used (no key area).

3. A two-byte data length field.

Considering the parts of the address separately, for a moment, there are five count area elements listed above. If each logical record on a track is 100 bytes long, and the five count area elements contain the following information:

1.	118
2.	10
3.	3
4.	6
5.	400

a. Which recording method (count/data or count/key/data) is being used?

b. What is the device address of the data record to which the count area refers?

c. Are the records blocked? If so, what is the blocking factor?

●●●

a. Count/key/data, because the key length field value is 6.
b. 118th cylinder, 10th track, 3rd physical record on the track.
c. Yes, and the blocking factor is 4, because the data length field value is 400.

16 The key area is a copy of the control field which is an organizing factor for the data record. Examples are, part number (for an inventory record), employee number(for a payroll record), customer number (for an accounts receivable record), and so on. If the count/key/data method is used for recording unblocked records, each key area contains the key for the logical record (in the data area) that follows it. If the records are blocked, the key area contains the key for the last logical record in the data area (the highest key).

DASD TRACK LAYOUT

SYSTEM/360, MODEL 30 AND UP

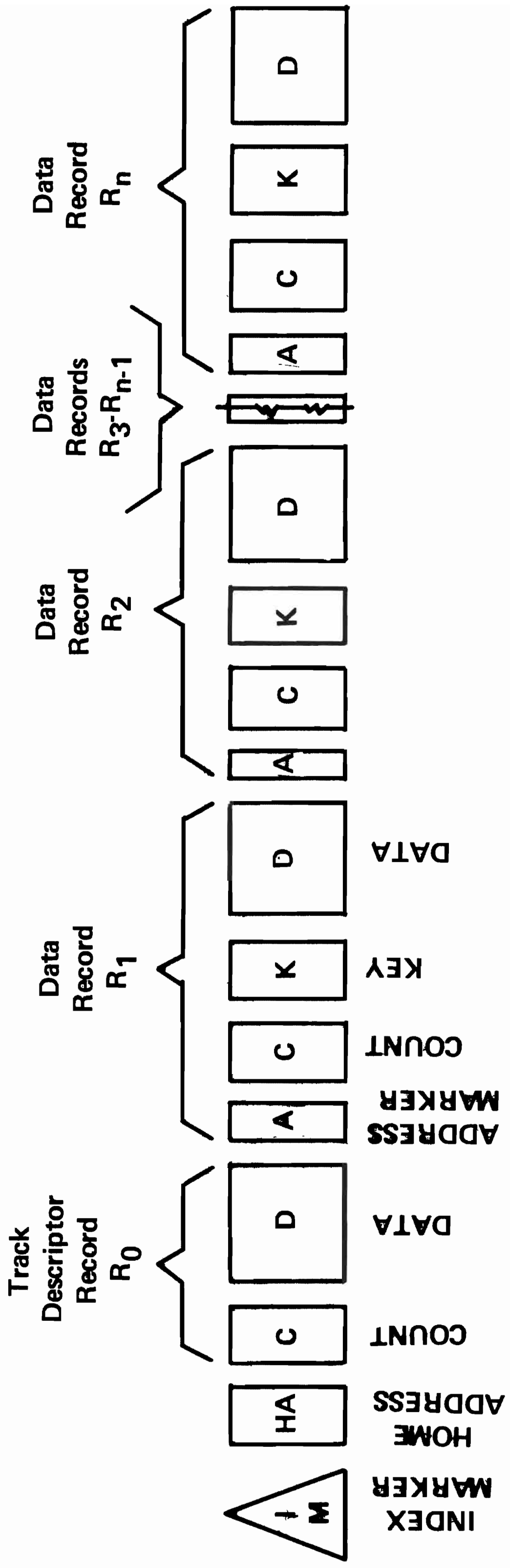


Figure 4.

One count/key/data illustration in Figure 3 shows a block containing n records. The key area contains the key for ....

●●●

record n (the last record in the block)

17 If records are unblocked, the key field identifies the record that follows it, so it need not also be recorded as a control field (imbedded key) in that record.

Each record in a block must contain an imbedded key. Why?

●●●

Without an imbedded key, there would be no way to identify each record. The key area only identifies the last one.

18 Match:

1. Count/data,unblocked.
2. Count/data, blocked.
3. Count/key/data, unblocked.
4. Count/key/data, blocked.

a. Must have imbedded key.
b. May have imbedded key.

●●●

1. a (Since there is no key area, key must be imbedded)
2. a. (Since there is no key area, key must be imbedded)
3. b.
4. a.

19 The data area contains the logical record, or block of records, as arranged by the programmer. When the DASD finds a desired data record, either the key area and data area, or the data area alone is read into main storage.

The selection made is a function of the Data Management system. This means that it (is/is not) .... determined by the programmer.

●●●

is not

20 We have concerned ourselves with data records – areas on the track that are important to the programmer. Let's briefly consider the non-data areas that are important to the system. Figure 4 shows a simplified track layout (Model 20 systems store dissimilar, but analogous, information).

Refer to Figure 4 after reading each of the following definitions:

a. Index Marker – a physical (not recorded) indication,of the beginning of one track, that is used to synchronize activity at the beginning of all the other tracks.

b. Home Address – the contents of the first seven bytes of every track. It shows the track condition (operative or defective) and, if operative, how the track is being used (primary track for data storage or alternate track). It also contains the physical location (cylinder number and head number) of the track.

c. Track Descriptor Record – the first data record on the track, always in count/data form. On an operative track, the count area contains the "record ID" (address) of the last physical record on the track. The data area shows the number of bytes that are available for recording, from the last physical record to the end of the track.

On an inoperative track, the count area holds the address of the alternate track to which the data records have been moved.

d. Address Marker – a special recorded signal which means that a count area is next on the track.

Maximum Bytes per Record Formatted without Keys						Records per Track	Maximum Bytes per Record Formatted with Keys					
2311	2314	2302	2303	2301	2321		2311	2314	2302	2303	2301	2321
3625	7294	4984	4892	20483	2000	1	3605	7249	4964	4854	20430	1984
1740	3520	2403	2392	10175	935	2	1720	3476	2383	2354	10122	920
1131	2298	1570	1558	6739	592	3	1111	2254	1550	1520	6686	576
830	1693	1158	1142	5021	422	4	811	1649	1139	1104	4968	406
651	1332	912	892	3990	320	5	632	1288	893	854	3937	305
532	1092	749	725	3303	253	6	512	1049	730	687	3250	238
447	921	634	606	2812	205	7	428	877	614	568	2759	190
384	793	546	517	2444	169	8	364	750	527	479	2391	154
334	694	479	447	2157	142	9	315	650	460	409	2104	126
295	615	425	392	1928	119	10	275	571	406	354	1875	103
263	550	381	346	1741	101	11	244	506	362	308	1688	85
236	496	344	308	1585	86	12	217	452	325	270	1532	70
213	450	313	276	1452	73	13	194	407	294	238	1399	58
193	411	286	249	1339	62	14	174	368	267	211	1286	47
177	377	264	225	1241	53	15	158	333	245	187	1188	38
162	347	244	204	1155	44	16	143	304	224	166	1102	29
149	321	225	186	1079	37	17	130	277	206	148	1026	21
138	298	209	169	1012	30	18	119	254	190	131	959	15
127	276	196	155	952	24	19	108	233	176	117	899	9
118	258	183	142	897	20	20	99	215	163	104	844	
109	241	171	130	848	15	21	90	198	152	92	795	
102	226	161	119	804	10	22	82	183	142	81	751	
95	211	151	109	763	6	23	76	168	132	71	710	
88	199	143	100	726		24	69	156	123	62	673	
82	187	135	92	691		25	63	144	116	54	638	
77	176	127	84	659		26	58	133	108	46	606	
72	166	121	77	630		27	53	123	102	39	577	
67	157	114	70	603		28	48	114	95	32	550	
63	148	108	64	577		29	44	105	89	26	524	
59	139	102	58	554		30	40	96	83	20	501	

Track Capacity Table

Figure 5.

Questions:

1. What signals the storage device control unit that the read/write heads are at the beginning of the tracks?

2. What signals the beginning of a data record, as opposed to the beginning of a logical record?

3. What does the control unit use to make sure that it has accessed the desired track?

4. What shows the control unit where to start writing on a track, or, that the track is already too full to accept another physical record?

●●●

1. Index marker.
2. Address marker.
3. Home address.
4. Count area and data area of the track descriptor record.

Blocking

21 Records are blocked on DASD for the same basic reasons as on magnetic tape – to save storage space and read-in (or write-out) time. Track length must be considered, however, when deciding what the blocking factor should be.

You will not calculate track and cylinder capacity, in this course. In a later course, you will probably use a table like the one in Figure 5. It accounts for the bytes used up by Home Address, Track Descriptor Record, Address Markers, Count Fields, Gaps, and so on, so that you don't have to. It shows the number of physical records (blocks or not), of a given length, that can be recorded on one track in a given device.

For example, suppose that we have a file of 14,000 records, each 300-bytes long and each requiring a 12-byte key area. Looking forward to expansion, we will reserve extra storage space: We want our current file to occupy only 90% of it. This is a "load factor" of 0.9.

We want to find out how many tracks will be required, on a 2311 Disk Storage Unit.

14,000 records ÷ 0.9 load factor = 15,556.
We need space for 15,556 records. Each of our records, formatted with a 12-byte key, requires 312 bytes.

From Figure 5, we find that the maximum bytes per record, closest to, and higher than, 312, is 315. This allows 9 records per track.

15,556 ÷ 9= 1,728
(remainder of 4)

What if we try to find a blocking factor of 4?

15,556 ÷ 4 = 3,889 physical records

Each four-record block requires one 12-byte key, so the bytes per physical record are 1212. The closest to this, on the 2311, is 1720, allowing 2 records per track.

3,889 ÷ 2= 1,945.

Interestingly enough, if we use a blocking factor of 2, we need about 1556 tracks.

It is clear that, as the blocking factor is increased, DASD track utilization (increases/varies) .... in efficiency.

●●●

varies (In this example, it first increased then decreased.)

22 The foregoing is not a disadvantage of blocking: It is something to be aware of. The main disadvantage occurs when records are not processed consecutively. It takes no longer to access a block than it does a single record, but it takes longer to transmit the block to and from storage. Non-consecutive processing of blocked records takes (more/less)...........time than processing of unblocked records.

●●●

FIXED-LENGTH RECORDS
(FORMAT F)

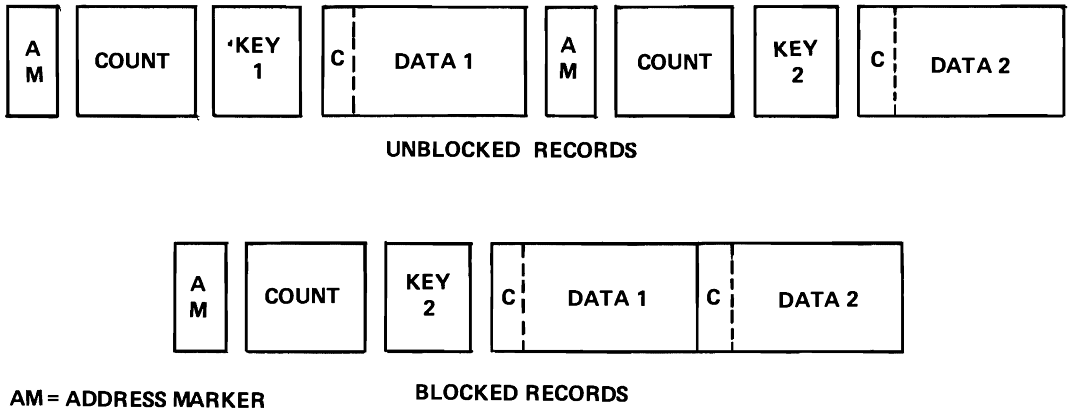


Figure 6.

VARIABLE - LENGTH RECORDS
(FORMAT V)

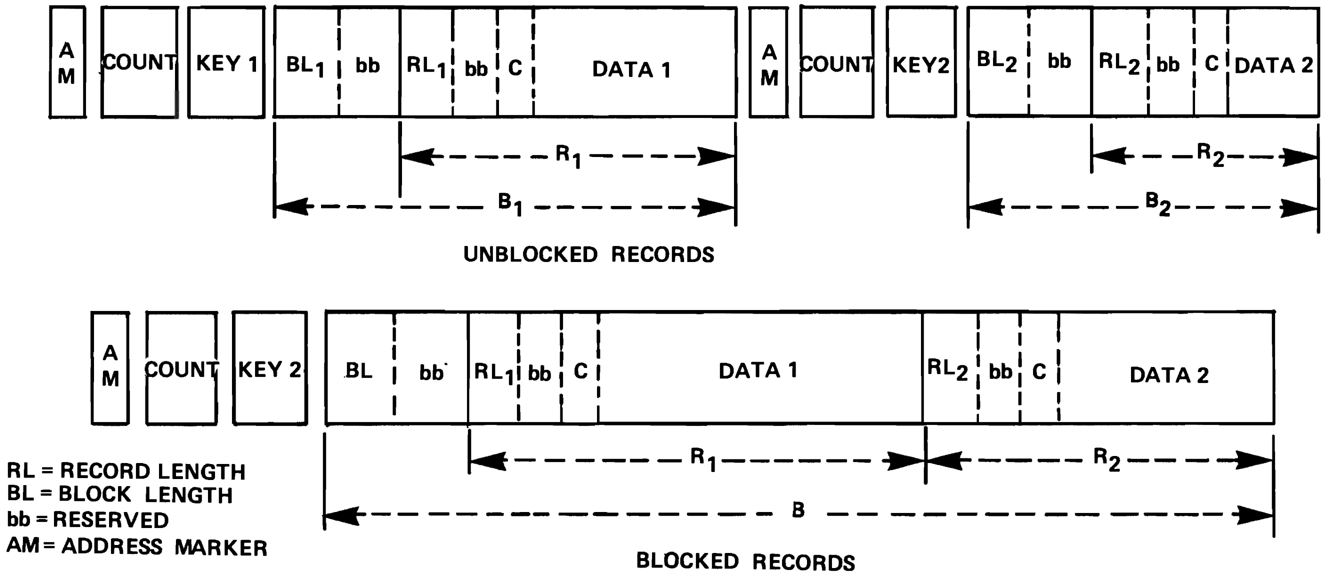


Figure 7.

UNDEFINED RECORDS
(FORMAT U)

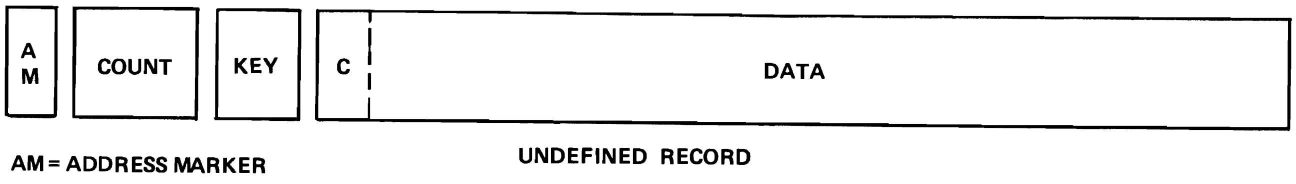


Figure 8.

more

23 As on magnetic tape, records on DASD may be fixed length (F), variable length (V), or undefined (U).

Fixed-length records are shown in Figure 6. Unlike the fixed-length records on magnetic tape, each logical record or block is associated with a .... area that tells the length of the data area.

●●●

count

24 Like magnetic tape records, each fixed-length record may contain an .... that is used for printer carriage control or card punch stacker selection.

●●●

optional control character

25 Variable-length records are shown in Figure 7. Locate the corresponding parts of each record, as you read the following information:

a. At the beginning of each data area, a two-byte block length field tells the length of the physical record. This is followed by two bytes reserved for IOCS.

b. At the beginning of each logical record, a two-byte record length field tells the length of the logical record (including the RL field's own length),two reserved bytes, and an optional control character.

If the total of the data lengths, in 10 variable-length records, is 1000 bytes, how much storage would they take up as one block? Assume no optional control characters.

●●●

1044 bytes	Block length field =	2
	Reserved =	2
	Record length fields=10x2=	20
	Reserved = 10 x 2 =	20
	Data =	1000
		1044

26 Undefined record format is shown in Figure 8. An undefined record is an unblocked variable-length record which does not carry .... and .... fields, where required for the V format.

●●●

block length (BL)
record length (RL)

File Organization

27 The three types of file organization used with System/360 DASD are sequential, indexed sequential, and direct.

Sequential files are produced by loading sequenced records into successively higher track addresses. They are processed like sequential card or tape files, e.g., transaction records, in the same sequence, are processed against the related records on the DASD. Such processing does not use the direct accessing ability of this type of storage.

Generally, sequential processing is most efficient when:

a. The file is large.
b. Percentage of activity is high (a large proportion of the DASD file is affected).
c. The file is completely "static" (no changes) or extremely "volatile" (a very high percentage of additions or deletions). This, as you may recall, is because any addition or deletion to a sequential file requires that it be ....

●●●

rewritten (since it must be rewritten, there should be many changes).

28 Which of the following would most likely be stored sequentially on a DASD?

a. A large customer account file, containing lists of purchases and payments.

b. A file listing the occupants of nearly-filled graveyards.

● ● ●

a.

29 In direct organization, each record is assigned a unique storage address, not usually in key sequence, but based on the numerical value of the key field. Depending on the operating system used, this address may be constructed to represent:

a. The Record ID.
b. The track address and record key.

One useful method of construction is called the "Division/Remainder Randomizing Technique".

The Record ID (cylinder number, head number, and record number within a track) can be produced as follows:

1. Determine the number of storage locations desired.

   a. A load factor of .8 to .85 is usual.

2. Find the prime number (a number which has no integral divisor except itself and 1) closest to, but less than, the desired number of locations.

3. Divide the key field, from a record, by the prime number.

   a. Ignore the quotient.

4. Use the remainder as a relative address (starting from the first location in the storage available for the file) to calculate the Record ID.

Example:

1. 8000 parts records ÷ .8 loading factor = 10,000. 10,000 record locations are needed.

2. 9,973 is the prime number nearest to, but less than, 10,000.

3. Record key field 25463514 ÷ 9,973 gives a remainder of 2445.

4. Assume you have found, from the Reference Card, that 12 records can be stored on a track:

   a. 2445 ÷ 12 gives a quotient of 203 and a remainder of 9.

5. The quotient, 203, is turned into cylinder and head designations. Assume the device is a 2311, which has heads numbered 0-9.

   a. Cylinder number is 20.
   b. Head number is 3.

6. Add 1 to the remainder, to find the record position on the track, since R_0 (the track designator record) is never used as a data record.

   a. Record number is 10.

The Record ID has been specified.

Now let's follow the steps for track addressing, using the same example:

1. 10,000 locations ÷ 12 locations/track = 834 tracks to be reserved.

2. The prime number nearest to, but less than, 834 is 829. Now to find a relative track address for the record with the key 25463514, what do we divide by what?

● ● ●

Divide 25463514 by 829.

30 The remainder, our relative track address, is 779. If this is divided by the number of tracks in a cylinder, the quotient will be the cylinder number and the remainder will be the ....

●●●

track number.

31 A 2311 has 10 tracks per cylinder. The track address of the record in our example would be:

a. Cylinder ....
b. Track ....

●●●

cylinder 77
track 9

32 The record would be recorded on track 9, in the first available space. When needed later, the record would be accessed by:

a. Calculating the same track number from the key field.

b. Searching that track until a matching (count area/key area) .... is located.

●●●

key area

33 By contrast, the Record ID method, shown first, produces an address which can only be located on a "preformatted" track. Here's how it's done:

a. Before file loading begins, a utility program puts specified count, key, and data areas, on all the tracks used for the file. These are "dummy records", with a valid count area, but with key and data areas filled with some specified character(zeros, perhaps).

b. When the randomizing program produces the first record ID, the specified track is accessed and searched for the matching ID. When it is found, the associated key and data areas are tested to make sure that they contain only the "fill character", in byte after byte. If they do, the key and data are "overlayed", on the dummy record.

When needed later, the record would be accessed by recalculating the .... from the key field, and accessing it, on the addressed track.

●●●

Record ID

Unused Areas and Synonyms

34 The foregoing are only two of many algorithms for calculating random addresses. But any algorithm will produce:

a. Unused storage space – addresses that never occur as the result of calculations based on existing keys.

b. Synonyms – Identical addresses produced from different keys. The first such address is called the "home address"; the record whose key produced it is stored there and is called the "home record". When a synonym is produced it is "chained" to another location.

For example, suppose that a home record is written out as record 10 on a given track, and a synonym is produced shortly thereafter. If the data area for record 3 (on the track with the next higher address) is the first one available, the synonym record is written there, and its Record ID(called a link address) is stored at the home address.

When processing, if the synonym record is accessed, the home record is read first. When its key is found not to match, the link address is used to access and read the synonym record.

True or False Statements:

1. The method of handling synonyms, just described, would tend to fill unused storage addresses.

2. A lower load factor would tend to reduce synonyms, because a wider range of possible storage addresses would be available.

3. Randomizing to track addresses (instead of to record addresses), then searching the track for the desired key, produces fewer synonyms. This is because a track, as a randomizing address, can store many physical records, while a record ID can store only one.

4. A home record is written sooner than a synonym and read into storage sooner, as well.

● ● ●

true
true
true
true

35 A sequential file must be rewritten whenever there are any deletions or additions, because it does not contain unused addresses, in anticipation of new records, or as a result of deleted ones.

A direct file organization contains unused addresses, because of its load factor. Additions are stored as home records or synonyms, and deletions merely create more unused addresses.

Which one:

a. uses storage most efficiently.
b. can update a record without rewriting the file.

● ● ●

sequential organization
direct <u>or</u> sequential organization (on DASD)

36 A sequential file is almost completely independent of its storage device: Its records are simply written and rewritten, consecutively, with no concern for the specified address occupied by any one record.

When a direct file organization needs to be rewritten – or even reorganized – because additions have created an unwieldy number of synonyms, the dependence on specific device addresses is apparent. A new algorithm may have to be developed, and the file recreated, in order to regain its initial efficiency in locating records.

How does the efficiency of a direct file organization compare to a sequential one, when processing many unsequenced transactions against a large, volatile file?

● ● ●

A direct organization is much more efficient than a sequential organization, in that case.

37 Which data record layout would be used with sequential organization, and which with direct?

● ● ●

a. Sequential – count/data. Recall that the records are loaded and processed consecutively. There is no search of a track by key.
b. Direct – count/key/data, with or without embedded keys, and count/data.

38 List advantages and disadvantages of sequential and direct organization on DASD.

●●●

Sequential organization:

Advantages –
a. Efficient use of storage.
b. Good for large sequenced files with a high percentage of activity.
c. Almost complete independence of storage device.

Disadvantages –
a. Must be rewritten, if there are any additions or deletions.
b. Not feasible, if unsequenced transactions must be processed.

Direct organization:

Advantages –
a. Can be the most efficient system for processing a large volume of unsequenced transactions against a volatile file.

Disadvantages –
a. Relatively inefficient use of storage.
b. Dependence on track addresses of a given DASD.

Indexed Sequential Organization

39 The indexed sequential method of file organization has some of the strengths of sequential and direct organization, and lacks some of the weaknesses.

It is a straightforward system: Records are stored sequentially by key, on one track after another; an index is constructed, at the same time, showing the highest key in each track.

A batch of transactions can be processed against the entire file, almost as efficiently as in any sequential processing system. In addition, sequential processing can start at a given record, without reading preceding records, and run to the end of the file.

Individual, unsequenced records can be processed, by using the index to locate the track on which a record is stored, accessing that track directly, and searching for the key.

As with sequential or direct organization, indexed sequential does not require the file to be rewritten when records are updated. It also does not require rewriting, when records are added or deleted. In this latter respect, it is like a .... file organization.

●●●

direct

40 A deletion is handled by "tagging" the record (e.g., by fillingthe first byte of the data area with 1-bits). An addition is inserted in sequence where it belongs. Since the tracks (called "prime" tracks) are filled at the outset, this causes the last record on the affected track to be bumped onto a track in an "overflow area". A note that this has happened is made in the index, so that the overflow record can be accessed on the overflow track that contains it.

As transactions are processed against the file a point will be reached where reorgnization is required: A number of records will have been tagged as ...., and the .... area will have been filled with ....

●●●

deletions
overflow
additions

41 By the use of a program supplied with the system, reorganization is easily accomplished. Records are simply read out sequentially and written back. Overflow records are put into the prime tracks, in sequence, where they belong. Tagged records are dropped. At the end of the process, a new index has been constructed and the overflow area is clear of records.

Since storage is reserved in advance for additions, and deletions effectively cause gaps, indexed sequential organization is less efficient than ....

File reorganization, with indexed sequential, is simpler than with .... organization.

●●●

sequential
direct

42 The movement of the access arm from one cylinder to another, called a "seek", takes a small amount of time. Switching from one track to another, within a cylinder, is practically instantaneous. If possible, the prime and overflow tracks for a given quantity of records, and the index tracks that reference them, should be on (the same/different) .... cylinders.

●●●

the same

43 Although electronic switching from one track to another is extremely fast, a wait is incurred while the track rotates to bring the desired record under a read/write head. This is called 'rotational delay". It intervenes between finding the track address for a desired record, via the index, and accessing the record itself.

As the number of unsequenced transactions and additions increases, the efficiency of indexed sequential organization (increases/decreases) .... This is because each new track that must be accessed involves another ....

●●●

decreases
rotational delay

44 Match the items below:

a. Sequential
b. Direct
c. Indexed Sequential

1. A record is accessed on the basis of an arithmetic calculation that derives its address.

2. A record is accessed on the basis of a stored list of record addresses.

3. A record is accessed on the basis of examination of all preceding records.

●●●

a. 3
b. 1
c. 2

45 Review:

List (a) the advantages of, and (b) the constraints on each type of DASD file organization.

●●●

Sequential:

a. Efficient use of storage; programming to access records is straightforward; is practically device independent.

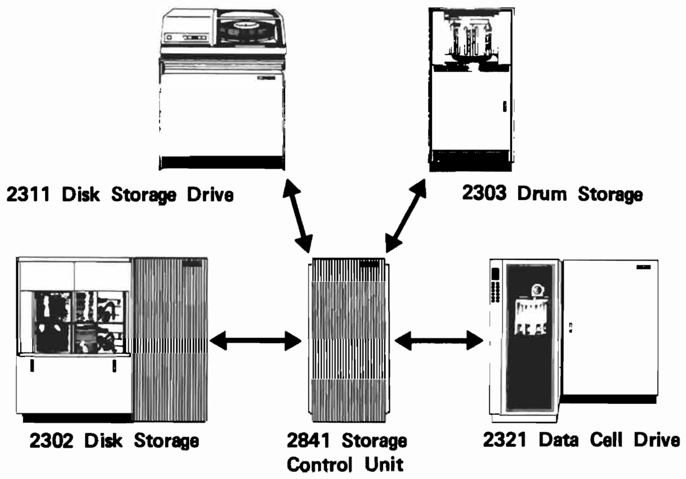
b. Transactions must be batched and in file sequence; additions and deletions require rewriting the file.

Direct:

a. Can efficiently handle large numbers of unsequenced transactions against a volatile file.

b. Storage efficiency is reduced by gaps, processing efficiency is reduced by synonyms, and file may be difficult to reorganize.

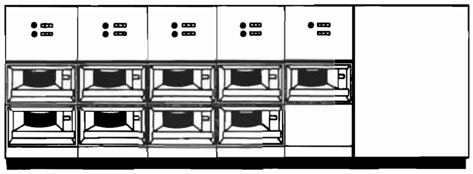
SYSTEM / 360
DIRECT ACCESS STORAGE CONTROL



2302, 2303, 2311, 2314, 2321

ALIKE IN:

Data Recording
Checking
Formatting
Program Control



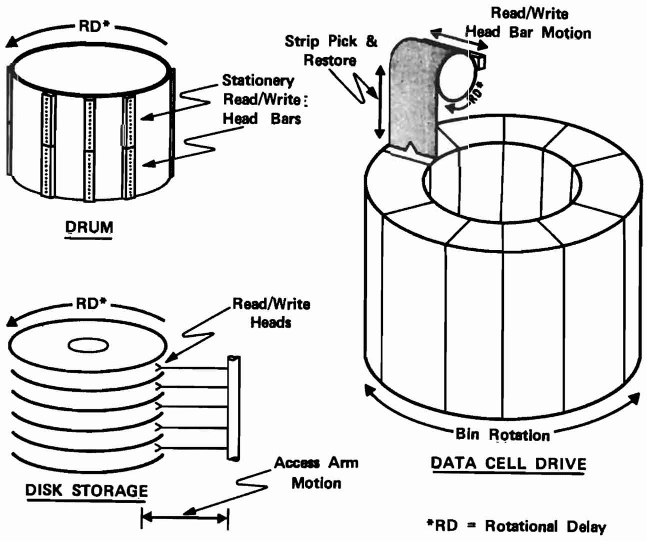
2314 Direct Access Storage Facility (includes control)

UNLIKE IN:

Physical Appearance
Capacity
Speed
Price

Figure 9.

DASD ACCESS MOTIONS



*RD = Rotational Delay

Figure 10.

Indexed Sequential:

a. Either sequenced or unsequenced transactions may be routinely processed; programming is relatively straightforward; file may be reorganized easily.

b. Unsequenced transactions tend to reduce processing efficiency, and a volatile file (additions and deletions) may have to be reorganized frequently.

Devices

46 Figure 9 shows the range of types of System/360 DASD. It also illustrates a remarkable example of the standard interface concept; Four types of devices, quite different in speed and capacity, can be connected to a channel via the ....

●●●

2841 Storage Control

47 In fact, the 2841 can control up to eight access mechanisms. These can be in the same type, or combinations, of devices.

The 2311, 2321, and 2303 have one access mechanism each. It (is/is not) .... possible to have a disk storage drive, a drum, and a data cell on line, on the same channel.

●●●

is

48 The storage units in three of these devices are removable and interchangeable with units of the same type:

a. 2311 holds one disk pack.
b. 2314 holds eight disk packs and one spare, on line.
c. 2321 holds ten data cells on line.

Extra units may be taken from a library and plugged into the device that recorded their contents, or another device, as required.

Which of the following is true?

a. These devices provide practically unlimited off line storage.

b. Files produced at one location can be processed at another location.

●●●

both a. and b. are true.

Mode of Operation

49 Access motion (seek) time is the time required to position the read/write heads at the desired cylinder. The relationship between device rotation and accessing is shown in Figure 10.

The data cell has the longest seek time, the disk drive is intermediate, and the drum has (little/none) ....

●●●

none (since the read/write heads do not move, there is no seek time involved in accessing a record).

50 The second component of the time required to access a record is the rotational delay, signified by RD in Figure 10. For the data cell, this is the time during which:

a. The bin is rotating so that the desired data cell strip can be picked out of a subcell.

b. The data cell strip is being wrapped around the drum.

●●●

b.

2321 ACCESS MOTIONS

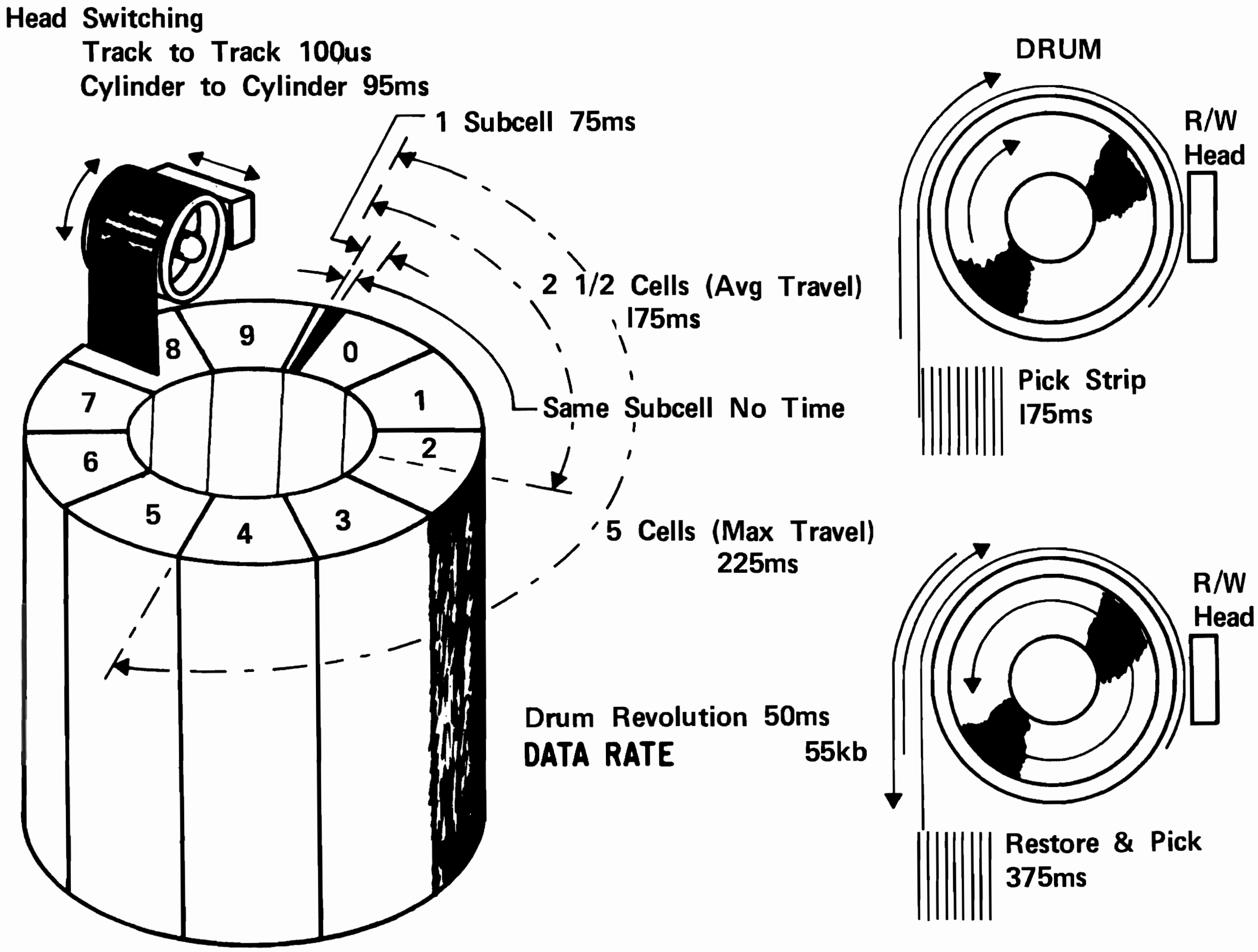


Figure 11.

51 The tracks rotate under the read/write heads continuously (except, in the data cell, just after one strip has been replaced in its subcell and before another is picked). A record search can begin at any time. The record that takes longest to access is the one that has just passed under the read/write head, when the search is started. The record that takes the least time to access is the one that comes under the read/write head immediately after the search begins.

The average time would represent a record (¼/½/¾) .... of the way around the track.

●●●

½ (some records will be more than halfway around the track, when the search begins, and some will be less).

52 Assuming that a drum revolves once in 17.2 milliseconds (ms.), how long, on the average, will it take to access a record?

●●●

8.6 ms (there is no seek time, only rotational delay).

53 Figure 11 shows the operation of the data cell in greater detail. Note that the entire unit, called the bin, comprises ten cells. Each of these cells holds 20 subcells, and each subcell holds 10 strips.

The bin can rotate in either direction. Consequently, when the seek for a given strip is initiated, and a certain cell must be located, the maximum distance that the bin rotates is (all the way/halfway) .... around.

●●●

halfway

54 Note that the average time to reach a given cell is a little more than half the time it takes to rotate 5 cells. The average is ....

●●●

175 ms

55 When the appropriate subcell has stopped under the pickup mechanism, the strip containing the desired record must be wrapped around the drum. Assuming that there is not another strip already on the drum, ... ms are required to pick the desired strip out of the subcell.

●●●

175 ms

56 Usually there will be another strip on the drum, from another subcell or the same subcell. That strip must be restored, before the desired strip can be picked. How long does it take to restore a strip?

●●●

200 ms

57 Once the strip is wrapped around the drum, the desired cylinder must be accessed, then the track, and finally the record. In terms of cell number, subcell number, etc., list the components of a data cell record address.

●●●

cell
subcell
strip
cylinder
track
record

SYSTEM / 360 Direct Access Storage Device Characteristics

Direct Access Storage Device	2311 Disk Storage Drive			2302 Disk Storage		2321 Data Cell	2314 DAS	2301 Drum	2303 Drum
Model	1	11	12	3	4	1	1	1	1
Models on which available	30 & up	20	20	30 & up	30 & up	30 & up	30 & up	50 & up	40 & up
Removable Media/type	Disk Pack	Disk Pack	Disk Pack	No	No	Data Cell	Disk Pack	No	No
No. of Units per Control Unit	8	2	2	4	2	8	8	4	2
No. of Access Mechanism Per Unit	1	1	1	2	4	1	8	1	1
No. of Heads per Mechanism	10	10	10	46	46	20	20	200	800
No. of Cylinders per Unit	200 (a)	200 (a)	100 (d)	500	1000	10,000	200(a)	- - -	80 (f)
No. of Tracks per Cylinder	10	10	10	45(b)	45(b)	20	20	- - -	10(g)
No. of Data Tracks per Unit	2,000	2,000	1,000	22,500	45,000	200,000	32,000	200	800
Maximum Data Capacity per Unit 8-bit mode (thousands)	7,250	5,400	2,700	112,100	224,200	400,000 (c)	233,408	4,090	3,910
Maximum Data Capacity per Cylinder 8-bit mode	36,250	27,000	27,000	224,280	448,560	40,000	145,880	- - -	48,920
Maximum Data Capacity per Track 8-bit mode	3,625	2,700 (e)	2,700 (e)	4,984	4,984	2,000	7,294	20,486	4,892
Transfer Rate (KB) 8-bit mode	156	156	156	156	156	55	312	1,200	312.5
Rotational Period (MS) (Average Delay)	25 (12.5)	25 (12.5)	25 (12.5)	34 (17)	34 (17)	50 (25)	25 (12.5)	17.2 (8.6)	17.2 8.6
Seek Times (MS) Minimum	25	25	25	50	50	175	25	- - -	- - -
Average	75	75	60	165	165	372	75	- - -	- - -
Maximum	135	135	90	180	180	600	135	- - -	- - -
Control Unit	2841	2020	2020	2841	2841	2841	included	2820	2841

FOOTNOTES

(a) 203 cylinders are accessible, but only 200 are available with programming systems.

(b) 46 tracks are accessible, but only 45 are available under programming systems.

(c) Represents 10 removable and interchangeable cells of 40,000,000 bytes each.

(d) 103 cylinders are accessible, but only 100 are available with programming systems.

(e) Ten 270-byte sectors per track.

(f) Electronically subdivided into 80 cylinders of 10 tracks each for addressing.

Figure 12.

58 Figure 12 lists various characteristics of System/360 DASD.

For the 2321 Data Cell, the maximum capacity per unit is ....

●●●

400,000,000 bytes (based on 10 cells, with 40,000,000 bytes per cell).

59 The transfer rate, in thousands of bytes per second (KB), varies widely. The slowest is ...., and the fastest is ....

●●●

55 KB, for the 2321
1200 KB, for the 2301 Drum

60 The data cell is used for very large files of records, when fast access is not a consideration. The drum is often used to store programs, in very large installations. Its (high/low) .... rotational delay and (fast/slow) .... transmission rate make it ideal for that purpose. It is almost like having the programs always available in main storage.

●●●

low
fast

Figure 12 is reproduced in your notes, for future reference.

You have completed this section. Fill in your notebook before taking the Self-Evaluation Quiz.

Self-Evaluation Quiz
Direct Access Storage Devices

QUESTIONS

1. What are the advantages of DASD over serial devices?

2. What is the basic unit for recording on DASD?

3. Define a cylinder.

4. Describe

   a. Sequential Organization
   b. Direct Organization
   c. Indexed Sequential Organization

5. What are two types of addresses that can be produced using the Division/Remainder Randomizing Technique?

6. What is a synonym?

7. List the advantages of and constraints on, the use of each type of file organization.

8. What are the two basic data record layouts on DASD?

9. Which layout would be used with:

   a. Sequential organization
   b. Direct organization

10. Define:

    a. Count area
    b. Key area
    c. Data area

11. To what do the count and key areas relate, if records are (a) unblocked, or (b) blocked?

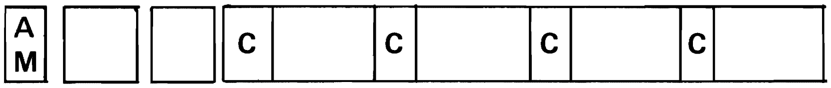
12. What does the key area for blocked records contain?

13. What is (a) an advantage, and (b) a disadvantage of blocking?

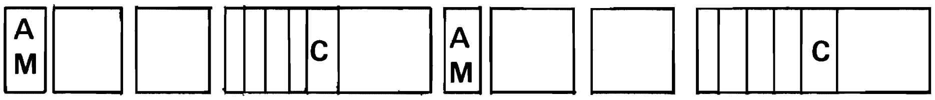
14. For each of the following, identify the format and fill in the blocks.

Use BL for block length
RL for record length
bb for reserved space
Data 1, Data 2, Data 3, etc., for logical records
Key 1, Key 2, Key 3, etc., for their respective keys.

a.



b.



15. Define seek time.

Self-Evaluation Quiz
Direct Access Storage Devices

ANSWERS — Frame Reference

1. a. They can access any record, without extensive searching of other records. (2)

   b. Transactions can be processed inline.

   c. Multiple files can be stored on the same device and accessed during the same job.

   d. Transactions can be processed either directly, as they are reported, or sequentially, in a batch.

   e. They reduce setup time by holding processing programs and accessing them when needed.

   f. They reduce or eliminate presorting of files.

2. The track – the area on the recording medium that passes under one read/write head. (4)

3. A cylinder is made up of the tracks that are available for reading or recording, with one positioning of the access mechanism. (6)

4. a. Sequential organization is one in which records are sequenced, then stored in successively higher track locations. (27)

   b. Direct organization is one in which each record is assigned a unique storage address, not usually in key sequence, but based on the numerical value of the key field. (29)

   c. Indexed sequential organization is one in which records are stored in sequence by key, in successively higher tracks, and an index is constructed, showing the highest key in each track. (39)

5. a. Record ID (29)
   b. Track address

6. A synonym is a record address, produced from a key by a randomizing technique, which is identical to an address produced from a different key. (34)

7. a. Sequential (45)

      1. Advantages – uses storage efficiently; programming to access records is straightforward; is practically device independent.
      2. Constraints – transactions must be batched in file sequence; additions or deletions require rewriting the file.

ANSWERS Frame Reference

b. Direct

1. Advantages – can efficiently handle large numbers of unsequenced transactions against a volatile file (large number of additions and deletions).
2. Constraints – storage efficiency is reduced by gaps, processing efficiency is reduced by synonyms, and file may be difficult to reorganize.

c. Indexed Sequential

1. Advantages – either sequenced or unsequenced transactions may be routinely processed; programming is relatively straightforward; file may be reorganized easily.
2. Constraints – unsequenced transactions tend to reduce processing efficiency, and a volatile file (additions and deletions) may have to be reorganized frequently.

8. a. Count/data (13)
b. Count/key/data

9. a. Count/data (37)
b. Count/key/data or count/data

10. a. The count area carries the record address (cylinder No., track No., and record position), the key length, and the length of the data area. (15)

b. The key area is a copy of the control field for the record. (16)

c. The data area contains the logical record, or block of records, as arranged by the programmer. (19)

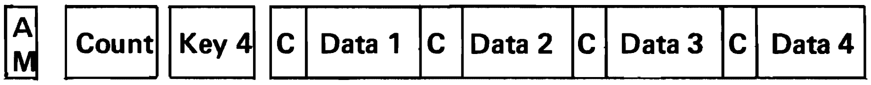
11. a. To the following logical record. (13)
b. To the entire block

12. The key for the last logical record in the block. (16)

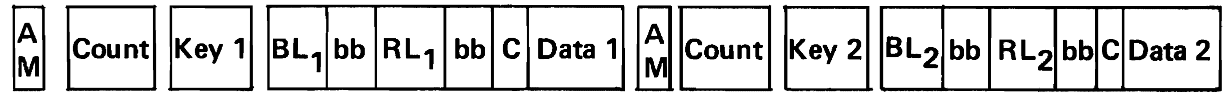
13. a. Saving in time and storage space. (22)

b. Non-consecutive processing of blocked records may take more time.

14. a. Fixed length, blocked records (23)



b. Variable length, unblocked records (25)



15. Seek time is the time required to position the read/write heads at the desired cylinder. (49)

Programming Systems

HIGH-LEVEL LANGUAGE

ADD BONUS TO GROSS

ASSEMBLER LANGUAGE

```
L          6, BONUS
A          6, GROSS
ST         6, GROSS
```

MACHINE LANGUAGE (IN HEX)

```
5860C050
5A60C054
5060C054
```

Figure 1.

PROGRAMMING SYSTEMS

1 This topic will give you an introduction to the programming support for the IBM System/360. When you have completed the topic you will not be able to write a program in symbolic language, but you will know why symbolic languages exist and what their advantages are. You will know the differences among the various languages, and what kinds of jobs each is most fitted for.

If you are familiar with symbolic language concepts, including compilation, debugging, the relationship between source and object programs, and the advantages of using symbolic languages, you may skip to frame 18.

First, let's review some basic language concepts and the vocabulary associated with them.

● ● ●

2 A stored program can be run only when it is in the computer's storage unit. To be executable it must be in a form that the computer can understand and act on; that is, it must be in machine language. As you know, each computer has a language of its own. This language is designed by the engineers who designed the computer, and it is the only one the computer will react to. You, as a human being, can carry out commands only if they are given to you in a language you understand. In the same way, the computer system must receive its commands in a language it can understand. So we can say that any program in computer system storage, in executable form (ready to be run) is in..........

● ● ●

machine language.

3 You know that main storage is made up of cores, and that information in cores, whether instructions or data, is represented by one or more binary digits (0's and 1's). Therefore, machine language is made up of binary digits.

We know that alphabetic, numeric, and special characters can be represented in storage by combinations of binary digits. We can write computer programs using such characters. In written form they may appear as meaningless combinations of letters, digits, and special characters, but when punched into a card and entered into storage (where they are represented as combinations of 0's and 1's) they constitute a program in machine language.

Machine language, whether written as letters and digits and special characters, or sitting in storage as binary digits, is not easily understood. To simplify the job of writing computer instructions we need something more easily read and written. This is the first reason for programming support.

Programming support, in the form of various symbolic languages, provides an alternative to writing programs in machine language. Programs in symbolic language are (easier/more difficult) .... to understand.

● ● ●

easier

4 Programs in symbolic language must be translated into machine language. This translation process, you recall, is called compilation.

A program written in the symbolic language FORTRAN must undergo .... before it can be executed by the computer.

● ● ●

compilation (translation)

5 Translation is not the only thing that happens when we compile a program. We will consider some of the other things later in the course.

The program we write is called the source program; the program that results from the translation run is called the object program.

Compilation converts .... programs into .... programs.

●●●

source (or written)
object (or machine language)

6 The time required to compile source programs into object programs can be called .... time.

●●●

compilation

7 The question naturally arises, why are we prepared to devote expensive computer time to compiling symbolic language programs into machine language? Why not write in machine language? We already know one important reason. It is that ....

●●●

programs written in symbolic language are easier to understand.

8 Easier, in this context, means more comprehensible than a string of 1's and 0's (machine language), or unpronounceable combinations of letters, digits, and special characters (the characters represented by the 1's and 0's).

Difficulty in working with machine language would mean that you would be more prone to make errors in your programming.

And, of course, if you have difficulty for any reason, you probably will take longer to write your program.

So now you know three reasons why it's to your advantage to use a programming language in writing your program. Programming languages, as opposed to machine languages, are:

a. ....
b. ....
c. ....

●●●

a. more easily understood
b. less prone to error
c. more quickly written

9 Let's review the two main categories of programming languages that we're talking about.

To write a program in machine language (it is often called absolute language, by the way) you must write either one string of 1's and 0's (or one string of letters, digits, and special characters that would be represented in storage by 1's and 0's) for each step the computer is to take. Each such combination would be an instruction in machine or absolute language. Several such instructions might be required just to add two numbers together, or to do similar operations.

There are programming languages that, although they use meaningful expressions to represent machine operations and the data the machine operates on, still require you to write one instruction for each step the machine takes. They are more easily read and understood than machine language instructions, but you must write just as many of them.

Such languages, you will remember, are called .... for .... languages.

●●●

one/one

10 If you have to write an instruction for each step you want the computer to take, it may seem that a one-for-one language has little to offer over machine language. Actually it has much to offer, not the least of which is ease of comprehension by humans. (You'll see some examples, with comparisons, later.)

One-for-one symbolic languages, since they are closest to absolute language, are at the lowest level of programming language, and in fact, often are called "low-level" languages.

Would you expect a high-level language to require one source program instruction for each object program instruction?

●●●

No. Normally, the high-level languages require fewer source program instructions than are developed in the resultant object programs.

11 Which is correct?

a. High-level languages are written in 0's and 1's.
b. High-level languages can be executed immediately, without compilation.
c. High-level languages normally contain fewer instructions than will be developed in the resultant object programs.

●●●

c.

12 If you selected either (a) or (b) in answer to the last frame, you have forgotten two basic facts we discussed at the beginning of this topic. All programming languages require compilation, and the whole idea of using programming languages is to get away from writing programs in absolute language. Although programming languages differ in many ways, they have these two things in common.

●●●

13 Here is an example of coding in a one-for-one language. Two sums are added by this sequence.

```
L       REG1,SUM1
A       REG1,SUM2
ST      REG1,SUM3
```

The first instruction loads (L) into a register (REG1) an item of data (SUM1).

The second instruction adds (A) into the same register another item of data (SUM2).

The third instruction stores (ST) the result by moving the contents of the register (REG1) to the storage location called (SUM3).

A high-level language might accomplish the same thing with this statement:

```
ADD BONUS TO GROSS
```

This statement and the one-for-one instructions shown above would be compiled into essentially the same set of machine instructions.

Figure 1 shows what this operation looks like as written in a high-level language, as written in a low-level language (not the one shown previously in this frame) and as it looks in S/360 machine language. (Incidentally, S/360 uses the hexadecimal number system to represent characters in storage.)

Which is most like English? Which is the least like English?

●●●

high-level language
machine language

14 What will happen to the statement "ADD BONUS TO GROSS" when it is compiled? What will the result be?

●●●

It will be translated into machine language.
Three machine language instructions.

15 Now, what about the words BONUS and GROSS? Obviously, they represent data items somewhere in storage, probably there as the result of reading in an employee's payroll record. But where in storage are they? If you were writing a program in absolute language you would have to decide the exact addresses in which these items were to be held, where they were to be added together, and where the result was to be stored. You would have to use, in your instructions, the storage addresses of the data and the register. What's more, you would have to remember these storage addresses, together with perhaps hundreds of others, all the while you were writing your program, and use the right ones in every instruction in which those data items were involved. This would not be easy, and you would be likely to make many mistakes.

Here both the one-for-one and the high-level languages help you, because both allow you to refer to data items by names which you make up yourself. You never have to worry about an address, but only remember the name, or "label", you have assigned to the data item that will be at that address.

So we have another example of how programming languages are advantageous; they save you worrying about ....

●●●

addresses of data in storage.

16 You may wonder how this is possible. It's really very simple. One of the main jobs of the compiler program is to assign actual storage addresses to the symbolic names (like BONUS and GROSS) that you will use in your program. Where, in your source program, you used BONUS, the object program will have an actual machine address. Further, all instructions that refer to BONUS will have that same address in their operands.

At the end of the compilation, a list of the symbolic names you used, and the actual addresses assigned by the compiler, will be printed out for you.

Compiler programs assign (symbolic/actual) .... addresses to the symbolic names used in the source program.

●●●

actual

17 Which of the following, if any, justify the use of symbolic languages in writing computer programs?

a. Programs are easier to write.
b. Programs do not have to be compiled.
c. The programmer is less likely to commit errors in coding.
d. Programs are more understandable in that they resemble English or mathematical notation.
e. Programs do not have to be documented.
f. The programmer does not have to cope directly with machine addresses.
g. A single source statement can result in many instructions in the object program.

●●●

a, c, d, f, g

18 In the next series of frames we will discuss some additional advantages of using programming support. If you are familiar with the concepts of language compatibility, debugging compilation diagnostics, and documentation, you may skip to frame 51.

Suppose that due to an increased work load, or because you want to start doing some new jobs that will require special computer features your present system doesn't have, you need to get a new system.

The relationship between your new and old system(s) could be:

a. The new system is completely different and can't accept the machine language programs used by the old system.
b. The new system will accept and run the machine language programs used by the old system, but has special capabilities and features the old programs don't take advantage of.

In either case, if your programs were written in a programming language designed for your old system only, you would have to reprogram all your current jobs to get them on the new system and have them utilize all its features.

On the other hand, if your original programs had been written in one of the high-level languages you might be able to take those programs, in their source language form, and by compiling them again, using a compiler designed for the new system, get executable object programs in a minimum of time. The compiler would make use of the new system's special features, and, in the case of a totally different kind of computer, develop an object program in the new machine's language.

This means that the same source language program can be compiled by two different compiler programs, designed for two entirely different computers, and come up with an object program for each of them. This facilitates the transfer of running jobs from one computer system to another.

Allowing for recompilation, programming languages will help you when your system changes in any way. How?

●●●

They enable you to adapt existing programs to the new system easier and quicker.

19 It is only fair to note that some modification of the source program probably would be necessary. For example, you might have to rewrite that part of the source program that describes the machine you will be using, and the special features you want to take advantage of. You might have to rewrite a few of the major statements. But this would take much less time than rewriting the entire program.

The feature of programming languages that allows a given source program to be compiled into the machine languages of different computers is called compatibility. The source language can be thought of as a kind of common language for the programmers, translatable into the machine language of any computer so long as there is a translator (a compiler program) for that computer.

What is meant by programming language compatibility? (your own words)

●●●

The translatability of source language programs into machine languages of different computers, via the appropriate compiler programs.

20 You can see that compatibility is a desirable feature of programming languages. If a programming language is compatible with several different computers it means that there is a compiler program for each. You have only to write your program in the programming language, and you can compile it for the system you intend to use. If you change systems you recompile your source programs for the new system. A minimum reprogramming effort is required.

A program written in COBOL (a high-level programming language) can be compiled for both the IBM S/360 Model 30 and the Model 40. The COBOL language, then, is .... with both these computers.

●●●

compatible

21 Some languages, such as COBOL and FORTRAN, may be used to write source programs for execution on computers made by different manufacturers. Having learned such a language, if you find yourself in an installation where the system(s) are made by a company other than the one whose machines you're familiar with, you wouldn't necessarily have to learn another language; you might be able to use the high-level language(s) you already know. Compatibility that extends across manufacturing lines might be called m.......... c..........

●●●

manufacturer compatibility.

22 In addition to compatibility, there is another advantage offered by programming languages. It is the division of the language into sections called subsets. Subsets enable you to write programs using only those parts of the language needed for your application. You need not learn the rules for coding in the other sections of the language.

For example, suppose a given language has a subset that lets you write programs to handle engineering or scientific problems; this subset lets you use the floating-point arithmetic features of your computer. But for commercial applications, the floating-point features aren't needed. To write commercial programs, then, you need to learn the rules for which subset?

●●●

the commercial subset.

23 Other programmers in your installation, however, may need to know the mathematical and scientific features of a language. They would concentrate on those subsets, rather than the commercial. The language would still remain a sort of common denominator for all programmers, but probably nobody would have to learn it in its entirety. Having mastered those sections they require, programmers start writing programs. As the need arises, they can expand their knowledge of the language by learning new ....

●●●

subsets.

24 Give one of the advantages of having a programming language divided into subsets.

●●●

Programmers can become productive more quickly because they can bypass learning the subsets they don't need immediately.

25 You will remember that one of the elements of problem solving is documentation. Documentation is all the recorded information about the problem and its method of solution. This recorded information must be collected, organized and stored.

Anything that provides written evidence in readable form of how the problem was solved would be useful as documentation. The less documentation you have to prepare manually, the faster and simpler the documenting job. Fortunately, most compilers provide several very good items of documentation.

If you get printed material from the compiler that is useful in showing how the problem was solved, you can say that your programming support gives ....

●●●

good documentation.

26 One item of documentation produced by the compiler is the <u>source listing</u>. This is a neatly arranged printout of your source program, showing all the statements you used. Of course, you could use the coding forms on which you write your source program, but the compiler listing is neat and possibly more legible than your printing.

Source listings are produced during ....

●●●

compilation.

27 If you ever have to make changes to a program now in operation, you will find the source listing essential to your understanding the program.

Which would be more meaningful: a source language program using English-like words and phrases, or an object program printed in binary or other graphic notation?

●●●

```
* LOC    OBJECT CODE         ADDR1  ADDR2  STMT         SOURCE STATEMENT                         COMMENTS

0005B4   4199  0004                 00004  100+IJFVOUB  LA   IJFVRC,4(IJFVRC)                    INCREASE BLOCKSIZE BY FOUR
0005B8   4560  F184                 006DC  101+         BAL  IJFVRA,IJFVEX1
0005BC   1B99                              102+         SR   IJFVRC,IJFVRC                       ZERO BLOCKSIZE
0005BE   47F0  F03C                 00594  103+         B    IJFVEX3                             GO TO EXIT
0005C2   906E  F318                 00870  104+IJFVGET  STM  IJFVRA,IJFVRH,IJFVSAV               SAVE USER REGISTERS
0005C6   9889  1048                 00048  105+         LM   IJFVRB,IJFVRC,IJFVDB3               PICK UP CURRENT I/O AND BLCKSZ
0005CA   9180  1015          00015         106+         TM   IJFVSWI,IJFVOPN                     TEST IF FIRST TIME
0005CE   4780  F08E                 005E6  107+         BZ   IJFVGTR                             YES READ A BLOCK
0005D2   4560  F16A                 006C2  108+         BAL  IJFVRA,IJFVDBL                      GO TO DEBLOCKING ROUTINE
0005D6   5590  1058                 00058  109+         CL   IJFVRC,IJFVDB6                      TEST IF END OF BLOCK
0005DA   47A0  F08E                 005E6  110+         BC   10,IJFVGTR                          YES GO TO READ NEW BLOCK
0005DE   4570  F0F8                 00650  111+IJFVSTW  BAL  IJFVRV,IJFVWRK                      GO TO WORKAREA SUBROUTINE
0005E2   47F0  F03C                 00594  112+         B    IJFVEX3                             GO TO EXIT
0005E6   4890  105C                 0005C  113+IJFVGTR  LH   IJFVRC,IJFVBZ1                      LOAD BLOCKSIZE TO REGISTER
0005EA   4560  F184                 006DC  114+         BAL  IJFVRA,IJFVEX1                      GO TO ROUTINE TO READ BLOCK
0005EE   1B9A                              115+IJFVGTB  SR   IJFVRC,IJFVRD                       SUBTR. RES. COUNT FROM BLOCKZ
0005F0   D201  1054 8000     00054  00000  116+         MVC  IJFVDB1(2),0(IJFVRB)                MOVE BLKSIZE IN RECORD READ
0005F6   4990  1054                 00054  117+         CH   IJFVRC,IJFVDB1                      AND COMPARE IT WITH ACTUAL BLC
0005FA   41E0  1064                 00064  118+         LA   IJFVRH,IJFVWLR                      LOAD WLR-ADDR TO REGISTER
0005FE   4770  F208                 00760  119+         BNE  IJFVER1                             IF COMPARE NE GO TO ERR.EXIT
```

* LOC - Storage addresses of machine language instructions.
OBJECT CODE - Machine language instructions in hexadecimal notation.
ADDR1 and ADDR2 - Actual storage addresses of instruction operands.
STMT - Sequential number assigned to source program statement by compiler.
SOURCE STATEMENT - Source program statements, including labels, operation codes, and programmer comments.

Figure 2.

The source language program using English-like words and phrases would be far more useful in learning about and understanding the program.

28 Figure 2 shows a portion of the source listing from an Assembler Language compile run. What are some of the kinds of information it contains?

●●●

Machine language instructions developed by the compiler from source language statements.
Actual addresses assigned to object program instructions.
All source program statements.
Programmer's comments from source program sheets.
Statement numbers of source program statements.

29 You can see that in addition to printing the program that you wrote and compiled, the compiler gives you some additional information: a listing of the object program that it developed from your source program.

●●●

30 In addition to the source listing, you get another valuable listing from the compile run: the cross-reference listing. It shows all the references made in your source program to any given symbolic name or word, such as BONUS or GROSS. This is of great value in grasping the logic of the problem solution as it is reflected by the source listing.

Figure 3 shows a portion of the cross-reference listing developed during the compile run of the Assembler Language program shown in Figure 2. The symbolic term CCOM3 is referred to (how many?) .... times in the source program.

●●●

4. Each hexadecimal number opposite CCOM3 in the cross-reference listing indicates the address of one object program instruction that refers to the data called CCOM3 in the source program.

31 Name two kinds of documentation developed during compile runs.

●●●

source listing
cross-reference listing

32 Documentation provided by the compiler, plus other documentation developed during the problem solving process, provides a permanent record of great value to all interested persons. It forms a communications link between the specialist and non-specialist, the programmer and the executive. For example, anyone could check a program to be sure that a vital step such as ADD BONUS TO GROSS had been included.

What is the importance of good documentation? (Your own words.) ....

●●●

It provides a clear and comprehensive record of everything involved in the problem solution.

33 Name one important source of good documentation.

●●●

The compile run.

34 We have named only part of the documentation you will get by using programming languages. Other items will be discussed later.

It should be noted that because of the nature of the languages, some source listings are more easily read by the non-specialist than others. COBOL statements, for example, are written in English-like words and phrases; whereas FORTRAN statements more closely resemble mathematical notation. A non-programmer probably would find the .... source listing easier to read and understand.

●●●

TEST RUN

POST ASSEMBLY DATA

REFERENCES TO DEFINED SYMBOLS

```
 4    2A3    DDED      128,   12E
 4    29F    DFIC
 4    29B    DFIT
 2    100    L004      151,   2AB
 6    102    L010
 6    128    L080      200
 6    134    L100      186
 6    162    L130      124
 6    174    L152      1F6
 6    18A    L170      114,   1A6
 6    1AA    L230      196
 4    1F6    M120
 2    16O    M230
 6    1FA    M240      1BC
 2    288    CCOM3     102,   18A,    19A,   1E4
 3    242    CCOM5     1E4
 3    285    CFIC3
 3    282    CFIT3
 3    27F    CGRO3
 3    28A    CHOS3     108,   190,    1AO,   1EA
 4    245    CHOS5     1EA
 1    150    CTLST     14B
 3    28D    CTOD3     10E,   16E,    18A,   190,   1C6,   1CC,   1F0,  1FA
 4    249    CTOD5     1F0
 4    1A7    DNETP
 1    20C    MINUS     16E,   1CC
16    24F    NAME3
20    224    NAME5     1D2
 3    290    NETP3     118,   11E,   162,    168,   1B6,   1C0
 3    27C    YFIC3
 4    278    YFIT3
 4    274    YGRO3
 2    204    CCWORK    102,   118,   128,    134,   162,   174,   174,  19A,  1B0
 4    270    DEPTN3    1DE
 4    23E    DEPTN5    1DE
 4    297    DGROSS
 3    294    DHOURS
 1    2AB    ENDDNP    155
 3    206    HSWORK    108,   11E,   12E,    13A,   168,   17A,   17A,  1A0,  1AA
 1    24D    INITA3    1D2
 1    24E    INITB3
 6    26A    MANN03    1D8
 6    238    MANN05    1D8
69    24D    MASTOU
80    224    PNCHOU
 7    20D    PRINT4    134
 8    214    PRINT5    13A
 8    210    PRINT6    140
```

Figure 3.

35 There is another aspect to the actual scripting of a program that is of interest, and here we find programming languages differing considerably. Some languages, particularly those at the one-for-one level, require the source statements to be written according to a rigidly prescribed format. For example, each part of source language statements must be placed in a particular place on the form, only one statement can be written on each line, a specified number of blanks must be left between certain parts of each statement, and so on.

By contrast, high-level languages are more "free-form"; that is, they permit the programmer more freedom in writing his statements. The formats are much less rigid, and the source program reads more like a narrative. This tends to produce fewer coding errors and makes it easier to amend and insert statements.

Of the three languages listed below, which is the most "free-form"?

a. Assembler Language (one-for-one)
b. COBOL (high-level)
c. S/360 machine language (absolute)

●●●

b. COBOL and other high-level languages are, generally speaking, more free-form than others.

36 What is meant by "free-form" coding?

●●●

Writing source language statements in a form resembling a narrative, rather than in accordance with rigidly prescribed format rules.

37 It should be noted that there are some exceptions to this rule. Some high-level languages do require coding in a rather definitely specified format. But in general, high-level languages are more free-form than low-level languages.

●●●

38 Free-form coding usually results in (more/fewer) .... errors and makes amendments to and insertion of source statements (easier/more difficult) ....

●●●

fewer
easier

39 The compiler program, as we know, converts source programs to object programs. In so doing, it allocates storage for I/O areas, assigns storage locations to data referred to in the source statements by symbolic names, creates machine language instructions from the source language and assigns storage addresses to them, and builds into the machine instructions the addresses of the data they will process.

In addition the compiler program produces a source listing and a cross-reference listing. They are of value in testing your program and as documentation for future reference.

You can see that compilers are pretty versatile programs.

Here's something else the compiler will do for you. It will check your source program statements for errors.

This doesn't mean that the compiler will compensate for all your mistakes. It won't recognize your logical errors. If your program, for example, adds two amounts together, and does it correctly so far as language rules and use of computer hardware are concerned, the compiler can only assume that you intended it so.

But coding errors such as inconsistencies in the spelling of symbolic names, incorrect operation codes, unreferenced symbolic names, incomplete branches, and many others can be recognized by the compiler as actual or possible errors.

```
IJS001I      E - LITERAL EXCEEDS 120 CHARACTERS
IJS0031      C - LITERAL IMPROPERLY CONTINUED OR CONTINUATION QUOTE IS MISSING
IJS011I      E - XXX IS UNDEFINED
IJS0251      C - XXX IS AN INVALID LIBRARY NAME OR NOT FOUND ON LIBRARY
IJS031I      W - VALUE-NUMBER OF INTEGERS IN LITERAL XXX AND DATA ENTRY DISAGREE
IJS1851      W - LABEL RECORD IS OMITTED:   LABELS ASSUMED STANDARD
IJS2281      E - SYNTAX PERMITS ONLY ONE XXX IN SOURCE PROGRAM
IJS2291      E - WORKING STORAGE SECTION OUT OF SEQUENCE
IJS2311      E - ENVIRONMENT DIVISION MISSING
IJS2331      C - REPORT SECTION REQUIRES F LEVEL COMPILER
IJS2341      W - WORD SECTION MISSING
IJS2351      W - PERIOD MUST FOLLOW WORD SECTION
IJS2381      W - XXX IS AN INVALID SECTION NAME OR INVALID/MISPLACED LEVEL INDICATOR
```

Figure 4.

What is a logical way for the compiler to notify you of actual and suspected errors in your source program?

●●●

List them.

40 The process of checking the source program for errors is called error diagnosis. Error diagnosis is a function of the .... program, which prints out an .... listing.

●●●

compiler
error

41 Figure 4 shows part of a list of errors discovered during a COBOL compile run. They are not particularly meaningful to you now, but the list emphasizes the number and types of errors you can make. For example, the first two items indicate that the coder violated certain rules about the allowable size of literals used by the program and how literals must be represented on the coding sheet.

The eighth entry on the error listing indicates that a rule concerning the .... of source program sections was broken.

●●●

sequence

42 If a compile run gives you an error listing, would you

a. load the object program into storage and try it out?
b. rewrite the source program in another language?
c. correct the errors in the original source program and recompile it?

●●●

c. is the sensible choice. You probably had good reason for selecting the particular programming language in which you wrote; there would be nothing gained in switching to another. Also, there is no benefit in testing a program until all errors are out of the source program. Machine time is expensive. Continue to recompile your program until the compiler can detect no more errors.

43 Let's assume you have corrected all the mistakes discovered by the compiler error diagnosis, and now you are ready to start testing. If you have used a high-level language you may not know precisely which instructions were developed in your object program. You might conclude that testing will be doubly difficult because you don't have this intimate knowledge of the object program.

Let's consider as an example the statement we've been using: ADD BONUS TO GROSS. The actual operation might be done in storage or in an addressable register apart from storage; the form of the data might have to be changed from binary to decimal, or vice versa; the data may or may not have decimal or binary points that have to be aligned.

We've already seen that the single statement ADD BONUS TO GROSS can be developed into (how many?) .... individual machine instructions. (See Figure 1 if you can't answer this.)

●●●

3

44 If you wrote in a low-level language you would have to write each of these individual instructions in one-for-one form. If data conversion and/or decimal alignment of the data is necessary, you would have additional instructions to write. You would, of course, have an intimate knowledge of all the instructions in your program.

By writing in a high-level language, you can avoid having to write all these individual instructions. The compiler examines your source statement, ADD BONUS TO GROSS, and develops all the instructions necessary to locate the data and add it, converting it from one form to another as required, aligning decimal points, if any, and selecting either a register or a storage location in which to add. You have to supply only the source statement.

The question is, does this lack of familiarity on your part with the details of your object program make testing more difficult?

●●●

The answer is no, and we will see why this is so.

45 High-level language compilers have libraries of routines to do addition, subtraction, data movement in storage, and so on. Depending on what the source statement indicates is to be done, the compiler will select from its library the appropriate routine and insert it into the object program. These routines are already debugged (tested and known to be error free) and a program made up of them should be (slower/faster) .... to test.

●●●

faster

46 Pre-tested routines in an object program makes testing easier. Suppose, in the ADD BONUS TO GROSS operation,it is necessary to convert one of the factors from decimal to binary. If you wrote the individual steps for the conversion routine you could easily violate factor size restrictions, or make other mistakes. This obviously would take some time to test and clear up.

By using a programming language you could, in effect, assign to the compiler the job of analyzing the source statement to see if conversion actually was necessary, and if it was, the compiler would pull out of its library the routine to do the conversion. You would know that the routine was already ....

●●●

tested (or debugged).

47 Pre-tested routines in an object program make .... easier.

●●●

debugging/testing

48 The source of the pre-tested routines incorporated into your object program is ....

●●●

the compiler program library,

49 Which of the following is/are true? In addition to analyzing your source program for errors, the compiler

a. indicates where you have made logical errors.

b. automatically corrects logical errors.

c. incorporates pre-tested routines that simplify testing.

●●●

c

50 If you selected either (a) or (b) in answer to the last frame, you are expecting too much of your compiler. It can't tell whether or not the sequence in which you've written your instructions or the operations you've called for in your program will give you the results you want. It can tell only if you've followed the language rules in writing your program, and if you're using the hardware correctly. If the logic of your problem solution is faulty, no compiler can know that.

●●●

51 Now we will consider the use of automatic testing programs to simplify and speed up the debugging of object programs. If you are familiar with the concept of automatic testing, including storage and register displays, patching during the tests, and end-of-job dumps, you may skip to frame 65.

Even though you wrote in a high-level language and your object program contains many pre-tested routines, you still must test the program. Here again the use of programming support pays dividends.

Some programming languages have associated with them a special program used to test object programs. This program is loaded into storage together with the object program being tested. During the test run the computer sometimes is controlled by the object program and sometimes by the .... program.

●●●

test

52 When the object program is in control, it will be processing your test data. But you can arrange in advance for the test program to assume control of the computer from time to time and perform certain functions that are very helpful in finding errors in your program.

For example, suppose that at a certain point in the execution of your program you would like to know what data is in certain storage areas or registers. The test program will display this information for you if you tell it at what point to do so. At the proper time during the test run it assumes control of the computer, prints out the storage and register contents you asked for, and turns control of the computer back to the object program.

The display feature of the test program will print out information from what locations in the computer?

●●●

storage and/or registers

53 The feature of the test program that prints out storage and register contents, at predesignated points in the test run, is the .... feature.

●●●

display

54 The display feature prints out storage and register contents as directed during the test run. Suppose you locate an incorrect instruction and wish to exchange it for a correct one. You can deduce that this feature of the test program is called the .... feature.

●●●

exchange

55 The exchange feature permits you to load a program and then exchange one or more instructions that you know are incorrect for ones that are correct. All you need to supply the test program are the replacement instructions and the storage addresses of the instructions being replaced. Before the test starts, the test program will make the necessary replacements, and then turn control over to the object program.

Without the exchange feature, how could you substitute one instruction for another? (Think about how you get object programs from source programs.)

●●●

The easiest way would be to make the change in the source program and then recompile. But this would mean recompiling each time you located a few incorrect instructions. Compile runs take time, and computer time is expensive. Better to list your errors and exchange wrong instructions via the test program until your object program is clean. Then all corrections can be incorporated in one final recompilation.

56 What does the exchange feature of the test program permit you to do during a test run?

●●●

Substitute one instruction for another without recompiling.

57 Suppose that after a test run you find you need some additional instructions. By supplying the needed instructions in machine language, and the point at which they are to be executed, the test program will cause them to be executed at the right time.

In the interest of time and economy, to get additional instructions into your object program, would you

a. compile them into the object program as the need arises?

b. add them to the object program at each test, via the test program, and compile them into the object program only when that program is completely tested?

●●●

b

58 The test program will, in addition to displaying storage and register contents at specified points, let you "patch" your program by deleting and adding instructions at test time. Suppose you wanted to delete some instructions you found you didn't need. Can you suggest a way the test program might accomplish this, so you wouldn't have to recompile your program to get rid of the unwanted instructions?

●●●

Give the test program the locations of the beginning and ending instructions to be deleted. During the test run it will cause the object program to branch around those instructions.

59 Name three ways you can use the test program to patch your object program during a test run.

●●●

Exchange instructions, add instructions, delete instructions

60 Test programs have an additional feature that is helpful in testing: they give you a complete print out of all storage locations and all registers when the test run ends.

It is important to note that there are two ways a program run can end: normal and abnormal. The test program will give you a printout (called a dump) for each.

Programs being tested have an unpleasant way of suddenly stopping before they have satisfactorily processed all the test data. This sudden stop is due, of course, to an error of some kind. The error may cause the program to go into an endless loop, to destroy some of its own instructions by storing data (inadvertently, of course) in their locations, or to develop an interrupt condition because certain arithmetical factors are too large or too small. Such an unexpected stop is called a "hang up."

Would you expect a normal or abnormal end of job dump when a hang-up occurs?

●●●

abnormal

61 When a hang-up occurs, the test program takes over and gives you a print out of all registers and storage locations, plus certain other information such as PSW contents. This print out is of great value in analyzing the operation of your program, because it reflects the machine conditions at the moment the hang-up occurred.

An abnormal dump occurs when ....

●●●

the program being tested halts prematurely due to an error condition.

62 Suppose your program runs smoothly through the test and develops some output data. You must check this data to be sure it is correct; very often it isn't. To assist you in checking, the test program gives you a normal EOJ dump. This dump includes register contents, PSW contents, and other valuable information.

The only difference between the two EOJ dumps is that the abnormal one prints all of storage, including the supervisor; the normal one does not dump the supervisor.

Describe briefly the two types of machine dumps and the conditions under which they occur.

●●●

The abnormal EOJ dump is a printout of all machine storage, registers, PSWs, and other locations, that occurs when the program being tested hangs up, that is, encounters an error that interrupts its progress. The normal EOJ dump is a similar print out occurring when the object program processes all test data and reaches the end of the run without a hang up. Abnormal EOJ dump prints the supervisor in addition to storage; normal EOJ dump does not print the supervisor.

63 What is the advantage of using automatic test programs to debug object programs?

●●●

Debugging time is shortened and made more efficient.

64 Describe the services provided by automatic test programs.

●●●

Patching the object program by exchanging, adding, and deleting instructions during the test, without the necessity for recompiling each time an error is discovered.

Displaying storage and register contents at specified points in the execution of the object program.

Dumping storage and register contents when either normal or abnormal end-of-job is reached.

65 Now we're going to talk about the ability of programming support to simplify input/output operations through the use of input/output control systems, and the concept of checkpoint records to permit restarting programs without going back to the beginning. If you are familiar with these concepts you may skip to frame 99.

●●●

66 Programming languages simplify programming by

a. allowing you to code in English-like words and phrases,

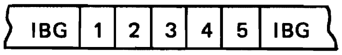
b. converting your source programs to machine language via a compiler program,

c. performing error diagnostics on your source program during compilation, and listing the errors, and

d. printing your source program and a cross-reference listing of symbolic names and references made to them in your program.

Hėre are some additional advantages. You have studied the concept of blocked records. To review briefly, logical records are blocked (grouped together to form one continuous physical record) on certain input/output media such as magnetic tape, disks, and drums. This is done to speed up I/O operations and to save·space on the medium in question.

This magnetic tape has (blocked/unblocked) .... records written on it.



●●●

blocked

67 The blocking factor is ....

●●●

68 Assume we are using two I/O areas for our input file. This means that every time a new physical record is read, the I/O area will contain (how many) .... logical records to be processed.

●●●

5

69 Our program must process the first record in I/O area number 1, then the second record in that area, and so on until the fifth record has been processed. Then the program must switch to the second I/O area and process records 1 through 5. While the records in the second area are being processed, the first area will be filled with more records from the input tape.

I/O AREA NO. 1	I/O AREA NO. 2
1	1
2	2
3	3
4	4
5	5

After processing all the records in area 2, the program switches back to area 1 and repeats the cycle.

The records in the input areas occupy 10 different record areas. Thus the actual addresses of the data fields in each record will be (different from/ the same as) .... the addresses of the other nine records.

●●●

different from

70 So we must process 10 different records in 10 different locations. The question is this: must the program have a separate set of processing instructions for each record (10 in all) or can it have a single set that has its data addresses modified for processing each successive record? Considering that storage space sometimes is scarce, which would you think most efficient?

●●●

A single set of processing instructions is used. Each instruction in the set is modified before processing of the next successive record is begun.

71 Remember that records in a data set normally have the same format; that is, fields containing similar data items are in the same relative positions within the records. For example, in a file of payroll records, if man number is in the first field in one record, it will be the first field in all records; if pay rate is the second field, it will be the second field in all records, and so on.

How many records in a file will have sales amount as the fifth data field, if all records in the file have the same format?

●●●

all

72 If records are of the same length, and similar data items are in the same relative positions within the records, we can see that similar data items are always a fixed number of bytes apart in storage.

I/O AREA NO. 1		I/O AREA NO. 2	
1	hrs. wkd	1	hrs. wkd
2	hrs. wkd	2	hrs. wkd
3	hrs. wkd	3	hrs. wkd
4	hrs. wkd	4	hrs. wkd
5	hrs. wkd	5	hrs. wkd

If 40 bytes separate hours worked in records 1 and 2 (I/O area no. 1) how many bytes separate hours worked in records 2 and 3? 4 and 5? Similar data items in I/O area no. 2 are separated by how many bytes?

●●●

All are 40 bytes apart.

73 The processing instructions, as originally compiled, will address the data fields of the first record in the first I/O area. After the first record is processed, when it is desired to process the second record, all that must be done is to add the necessary increment (number of bytes that separate similar data items in the records) to the data addresses of all instructions that work directly with those data items.

After modification, the program will work with which data record?

●●●

The next successive one in the I/O area.

74 Since I/O areas normally are consecutive in storage, the same modifying increment will alter the instructions so they address the proper data fields in the next consecutive record area. This will continue until all ten records have been processed. But what must happen when the tenth record has been processed?

●●●

Processing instructions must be altered so they again address the data fields in the first record area; that is, they must be returned to their original form.

75 Now, what has all this got to do with programming support, and programming languages? Just this: instead of having to write all these modifying routines yourself, all you do is write one instruction that says something like this: GET RECORD. Then you write your processing instructions as though there were only one record to be processed, using the symbolic names you made up for the data items in the record. After the last processing instruction, you branch back to the GET instruction. Here is an example.

```
START     GET       RECORD
           •
           •
   processing instructions
           •
           •
           BR      START
```

Each time this set of instructions is executed, the program will go where for its next instruction?

●●●

Back to the instruction labeled START.

76 The GET instruction will call into operation a modifying routine that will add the necessary increment to all data addresses in the instructions that directly address the records. Thus the instructions will, when next executed, work with the (which one) .... record in the block.

●●●

next consecutive

77 Where do these modifying routines come from? They are selected by the compiler from its library of routines and built into your object program at compile time.

So an advantage of using programming support is (your own words) ....

●●●

Blocked records can be processed without the necessity for writing separate processing routines for each record in the block.

78 How is this accomplished?

●●●

By successively modifying a single set of processing instructions so they work with each record in the block, until all are processed.

79 The process of isolating the record that is to be processed next is called "de-blocking". It doesn't necessarily mean, however, that the record is physically removed from the block (although this can be done if the record is moved to a work area for processing). Rather, it means "making the record available for processing". The expression "making it available" can mean modifying the processing instructions so they will address the record, or simply moving it to a work area. In either case, the record is de-blocked, in that it is isolated and designated as the one to be processed next.

Making a blocked record available for processing is called ....

●●●

de-blocking.

80 If each record is de-blocked by moving it to a work area, would instruction modification of processing instructions be required? Why?

●●●

No. Each record, as it was processed, would occupy the same storage locations in the work area as the preceding one, so the processing instructions would never need to be modified.

81 So, regardless of whether or not instruction modification is necessary, a programming language enables you to write only one set of processing instructions for blocked records. Of course, you would tell the compiler at compile time whether you planned to process in the input area, requiring instruction modification routines, or in a work area. In either case, the compiler builds into your program the appropriate routines.

Another advantage of this approach is that when all the records in an I/O area have been processed, the GET instruction does more than just modify processing instructions so they will address the first record in the alternate I/O area. It also initiates a read operation on the appropriate channel. This will enter fresh data into the just processed area, while processing continues in the alternate one.

The GET instruction .... records (makes them available for processing) and initiates .... operations on the channel to fill one area while the other is being processed.

●●●

de-blocks
read

82 For output records an instruction called PUT is used. This instruction moves processed records into the output I/O areas, each record being placed in the next successive space in the area, until the block is filled. What kind of operation would be necessary when an output block was full?

●●●

A write operation.

83 As one block is written out, processed records are being placed where?

●●●

In the alternate output area.

84 So another advantage of programming support is that input records automatically are de-blocked for processing and output records are automatically .... for writing out.

●●●

blocked.

85 When are the blocking and de-blocking routines incorporated into your program?

●●●

During the compile run.

86 The blocking and de-blocking instructions are part of a routine called LIOCS (for Logical Input Output Control System). This routine, as its name implies, works with the logical records in the input and output files, after they have been read in and before they have been written out.

The actual read and write operations, you will recall, are performed by routines in the supervisor program. This set of routines, in contrast to logical input/output control system, is called the .... system (PIOCS).

●●●

physical input/output control

87 If you have questions about the concept and functioning of the PIOCS, you should review topic 3 in this course.

●●●

88 Now we'll consider one last feature of programming support before looking at the individual languages available for use with S/360. If you are familiar with the concept of taking checkpoints during processing runs, you may skip to frame 99.

●●●

89 Even at electronic speeds, computer runs take time. It isn't unusual for a computer to run for hours to process extensive files of data. This brings up the problem of what to do if the run is interrupted. Of course, the job can always be run again. But suppose you're only thirty minutes or so from the end of a four-hour run. If something happens and you have to begin again at the beginning, you'll be repeating three and a half hours of computing time. This would be expensive in time and money. Waiting jobs would be set back, schedules disrupted.

The answer lies in the checkpoint, a periodic writing out, during the run, of everything necessary to restart the run at the last point at which a checkpoint was taken.

Checkpoint lets you .... a run without going back to the beginning.

●●●

restart

90 Routines to write checkpoint records are part of the programming support for the S/360. They are called into storage by the supervisor program each time the problem program reaches a point at which a checkpoint is to be taken.

This means that the programmer decides where and when in his program the checkpoints are to be taken. He writes an instruction at these points in his program. During execution, when these points are reached, the .... routine will be called into storage by the ....

●●●

checkpoint
supervisor

91 The checkpoint routine takes control of the computer and writes, on a designated output device, the entire contents of storage and of all registers, the positions of all I/O files, the status of all indicators in the system, and all other information necessary to restart the program at that point in its operations. After the checkpoint record is written, control will return to the .... program.

●●●

problem

92 The problem program will continue its operations until one of three things occurs:

1. Another checkpoint is reached in the problem program.
2. The end of the job is reached.
3. An error or other difficulty stops the run.

Which of these three calls for a restart?

●●●

An error or other difficulty that stops the run.

93 Of course, if the end of the job is reached, successfully, the checkpoint records are

never used. But they provide valuable insurance that you won't have to repeat long runs unnecessarily.

The purpose of the checkpoint record is, of course, to allow restoration of storage, registers, I/O devices, and machine indicators, to the conditions in which they were at the time of the checkpoint. When this is done, the problem program can resume processing at that point, instead of having to go back to the beginning.

How many checkpoints can be taken in one computer run?

●●●

As many as the programmer decides are necessary.

94 Checkpoints often are taken every hour (the time is kept on an internal clock called the interval timer) or each time a reel of tape has been completely written or read. Checkpoint records usually are written on magnetic tape or disk, to facilitate the restart operation.

To restart a run, the operator enters certain information on a punched card. The supervisor program, acting on this restart information, calls into storage a restart program and turns control over to it. By reading in the last checkpoint record the restart program can restore storage, registers, indicators, I/O devices, and all other machine components to the status in which they were at the time of checkpoint.

The routine that is read in when processing operations are to be resumed is called the .... routine.

●●●

restart

95 What must the restart routine do to ready the machine for restart and how is this accomplished?

●●●

It must restore storage, registers, machine indicators, I/O devices, and all other machine components to the status in which they were when the checkpoint was taken. It accomplishes this by reading in the last checkpoint record from the checkpoint device.

96 After the restart routine has readied the machine for restart, control is returned to the problem program, which resumes its processing.

Where is the checkpoint written? Name two devices often used for checkpoint records.

●●●

Checkpoints are written on an available output device. Magnetic tape and disks often are used for checkpoints.

97 Describe a checkpoint operation and tell why checkpoints are taken.

●●●

The problem program periodically signals the supervisor to call a special routine called the checkpoint routine, which writes the entire contents of storage, plus all register contents, on a designated output device (usually tape or disks). Also written is all information reflecting conditions in the computer system at the point where the checkpoint is taken. The problem program then resumes normal operation.

Checkpoints are taken during lengthy computer runs so that the job can be restarted, in case it is interrupted for any reason, without going back to the original beginning. It is necessary only to return the system to the condition it was in at the last checkpoint.

98 How is the computer system returned to the status in which it was at the last checkpoint?

●●●

A special routine is called by the supervisor. This routine (called the restart routine) locates the last checkpoint record on the checkpoint device, and uses it to return the system to checkpoint status. This includes repositioning all I/O devices, restoring storage and register contents, and resetting all machine indicators.

99 Now it's time to consider the individual programming languages that are available for use with S/360. We will look briefly at each, considering only the characteristics and capabilities of the language. We won't try to teach you how to code programs; this will come later in the appropriate language coding classes.

If you are familiar with the general capabilities and characteristics of the Report Program Generator (abbreviated RPG) language, and can:

a. classify RPG as high- or low-level,
b. identify the format of RPG source statements,
c. identify the applications areas (commercial and/or scientific) for which RPG is intended, and
d. describe RPG coding as free form, partially restricted, or rigidly prescribed,

you may skip to frame 125.

● ● ●

100 A programming language is one part of a programming system. Programming systems may be thought of as composed of two major components: a symbolic language, and a compiler for that language.

You already know the function of the language. It lets you write computer instructions that are more meaningful to humans than the actual computer language. Thus, programs in symbolic languages are less liable to error and are easier to write.

The function of the compiler, as we have seen, is to translate the source language statements into machine, or absolute language.

Name the two principal components of a programming system.

● ● ●

a symbolic langauge
a compiler

101 Wide use is made of programming systems. It's probably true that no one writes programs in machine language any more. This means that languages and compilers get a lot of use. Some installations don't use operating systems, but more and more are doing so.

No matter what computer and programming systems you are using, there are some things you must do to use the programming systems. They are:

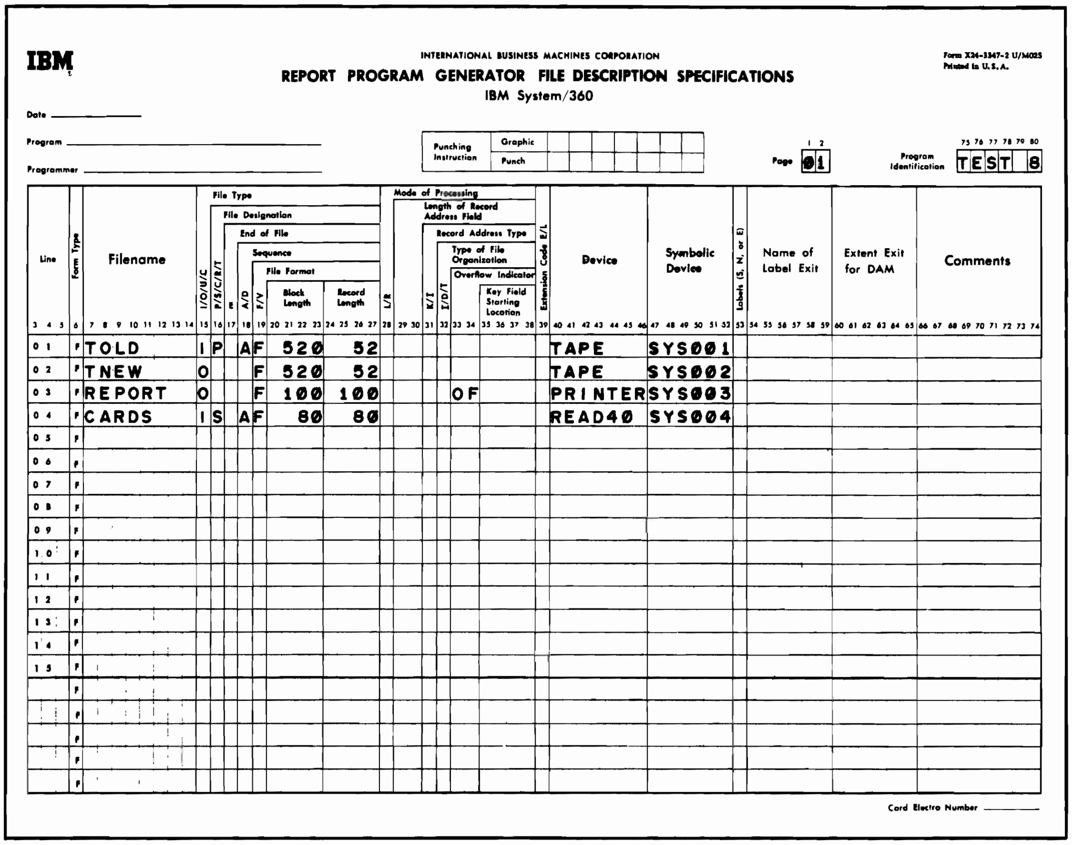
a. Describe your input data and files in the format required by the language you are using.
b. Do the same for your output data and files.
c. State the processing steps required to develop the desired output from the input.
d. Describe the computer system to be used.

As we consider the characteristics of S/360 programming languages we also will see briefly how these four requirements are met for each of the languages.

To use any programming system you must

a. describe the .... data and files,
b. describe the .... data and files,
c. state the required .... steps, and
d. describe the ....

● ● ●



IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

REPORT PROGRAM GENERATOR FILE DESCRIPTION SPECIFICATIONS

IBM System/360

Form X24-3347-2 U/M025
Printed in U.S.A.

Date

Program

Programmer

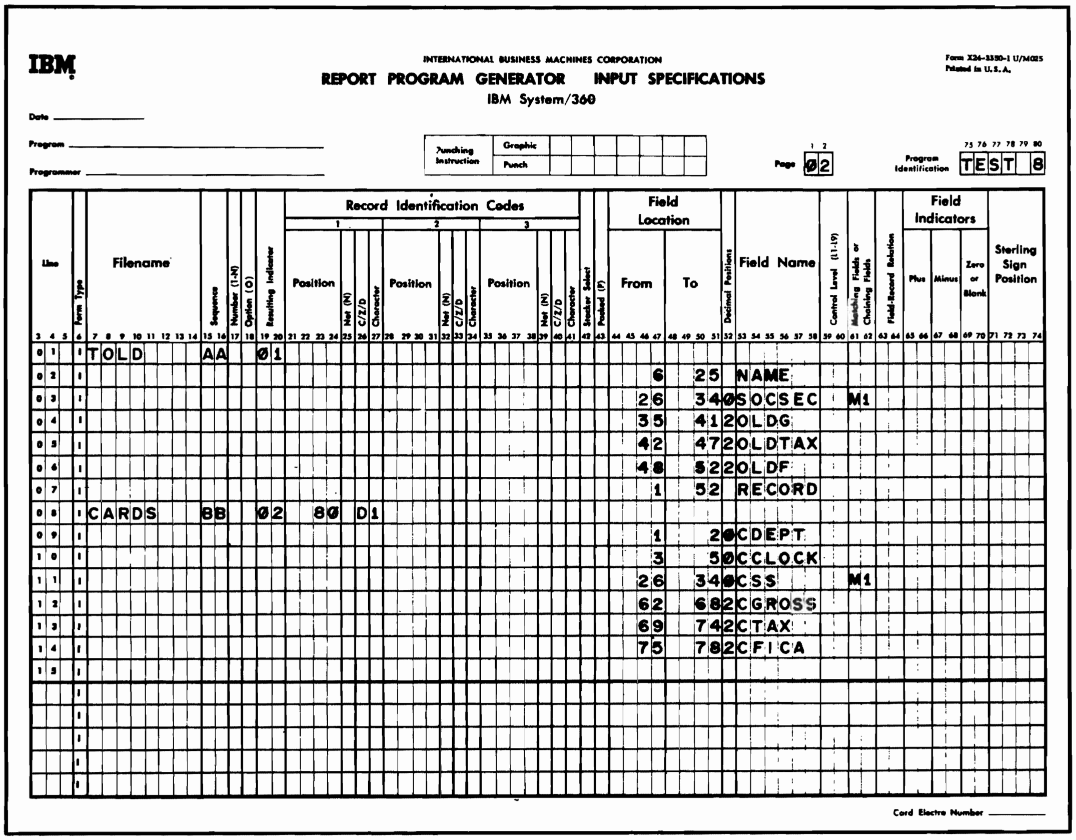
Punching Instruction	Graphic	
	Punch	

Page Ø1 Program Identification TEST 8

Line	Form Type	Filename	File Type I/O/U/C	File Designation P/S/C/R/T	End of File E	Sequence A/D	File Format F/V	Block Length	Record Length	L/R	Mode of Processing	K/I	I/D/T	Overflow Indicator	Key Field Starting Location	Extension Code E/L	Device	Symbolic Device	Labels (S, N, or E)	Name of Label Exit	Extent Exit for DAM	Comments
01	F	TOLD	I	P		A	F	520	52								TAPE	SYSØØ1				
02	F	TNEW	O				F	520	52								TAPE	SYSØØ2				
03	F	REPORT	O				F	1ØØ	1ØØ					OF			PRINTER	SYSØØ3				
04	F	CARDS	I	S		A	F	8Ø	8Ø								READ4Ø	SYSØØ4				
05	F																					
06	F																					
07	F																					
08	F																					
09	F																					
10	F																					
11	F																					
12	F																					
13	F																					
14	F																					
15	F																					

Card Electro Number

Figure 5a.



IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

REPORT PROGRAM GENERATOR INPUT SPECIFICATIONS

IBM System/360

Form X24-3350-1 U/M025
Printed in U.S.A.

Date

Program

Programmer

Punching Instruction	Graphic	
	Punch	

Page Ø2 Program Identification TEST 8

Line	Form Type	Filename	Sequence	Number (1-N)	Option (O)	Resulting Indicator	Record Identification Codes 1 Position	Not (N)	C/Z/D	Character	2 Position	Not (N)	C/Z/D	Character	3 Position	Not (N)	C/Z/D	Character	Stacker Select	Packed (P)	Field Location From	To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field-Record Relation	Field Indicators Plus	Minus	Zero or Blank	Sterling Sign Position
01	I	TOLD	AA			Ø1																									
02	I																				6	25		NAME							
03	I																				26	34	Ø	SOCSEC		M1					
04	I																				35	41	2	OLDG							
05	I																				42	47	2	OLDTAX							
06	I																				48	52	2	OLDF							
07	I																				1	52		RECORD							
08	I	CARDS	BB			Ø2	8Ø		D	1																					
09	I																				1	2	Ø	CDEPT							
10	I																				3	5	Ø	CCLOCK							
11	I																				26	34	Ø	CSS		M1					
12	I																				62	68	2	CGROSS							
13	I																				69	74	2	CTAX							
14	I																				75	78	2	CFICA							
15	I																														

Card Electro Number

Figure 5b.

input
output
processing
computer system

102 We want to consider some of the characteristics and capabilities of the S/360 symbolic languages. RPG is an abbreviation for ....

●●●

Report Program Generator

103 The name of this language implies that it produces programs whose main output is ....

●●●

printed, in the form of reports.

104 Modern business relies heavily on reports of all kinds to provide the information needed to make business decisions. RPG was developed to meet the specific need for a language and compiler that would quickly develop programs to print reports.

But it would be a mistake to assume that this is all RPG can do. As we will see, it has other very important capabilities that make it useful as an all around programming language.

A programming language, to be of greatest use to the programmer, should be able to produce programs for (nearly all/a few of) .... the jobs in an installation.

●●●

nearly all

105 A programming system that develops programs to prepare business reports would be used principally in the (scientific/commercial) .... areas of data processing.

●●●

commercial

106 Figure 5a shows one of several special forms used by the RPG programmer. On these forms he records information required by RPG. The form shown in Figure 5a is used to record ....

●●●

file description specifications

107 File description specifications are simply descriptions of the files (both input and output) that will be processed by the object program. As we pointed out previously, describing the input is one of the things you must do to use any programming language.

The RPG programmer uses symbolic names to identify the files, along with special codes (numeric and alphabetic) that provide other information about how records are arranged on the file, record length, and so on. Both English and symbolic names are given to the I/O devices for the files.

Figure 5b shows a similar sheet on which the specifications for .... are recorded.

●●●

the input

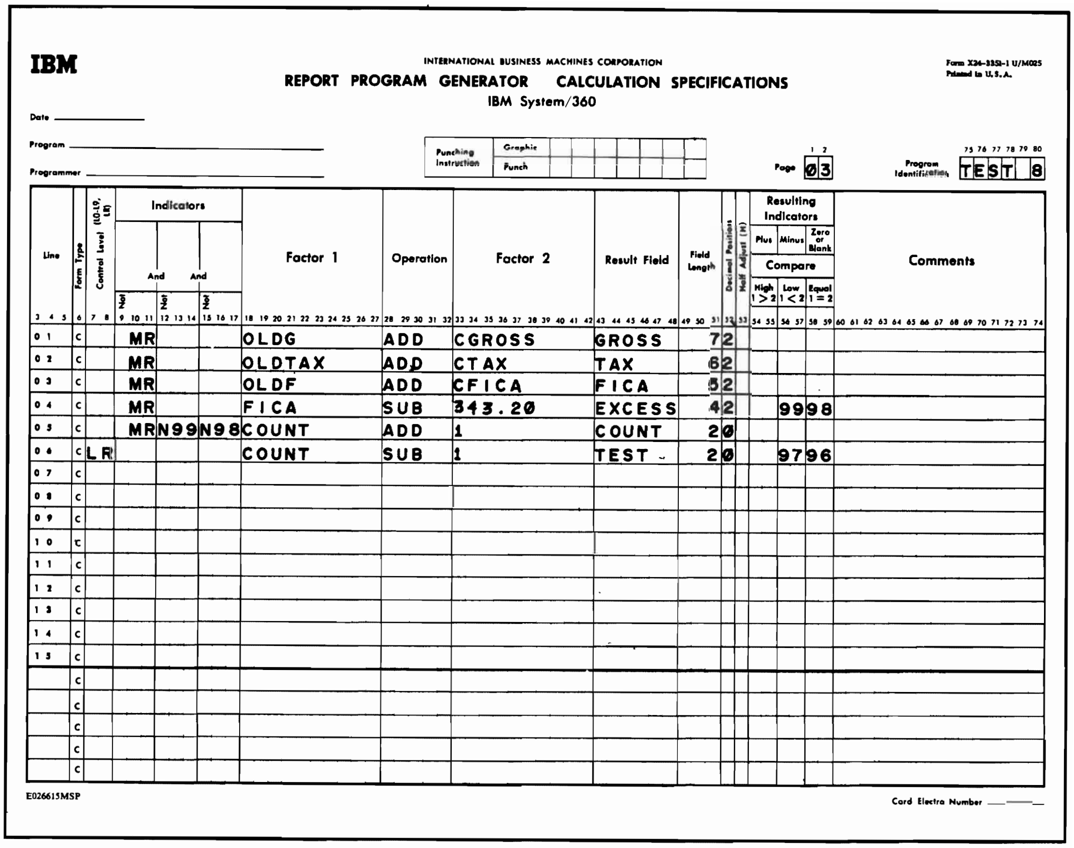
108 Input specifications describe the data in the files. Symbolic names are used to describe both the files and the data fields in the records. What other information is provided about the data fields? (See Figure 5b.)

●●●

The positions of the data fields within the records. (Field Locations)

109 The form shown in Figure 6a is used to record what information?

●●●



IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

REPORT PROGRAM GENERATOR CALCULATION SPECIFICATIONS

IBM System/360

Form X24-3351-1 U/M025
Printed in U.S.A.

Date

Program

Programmer

Punching Instruction: Graphic / Punch

Page Ø3

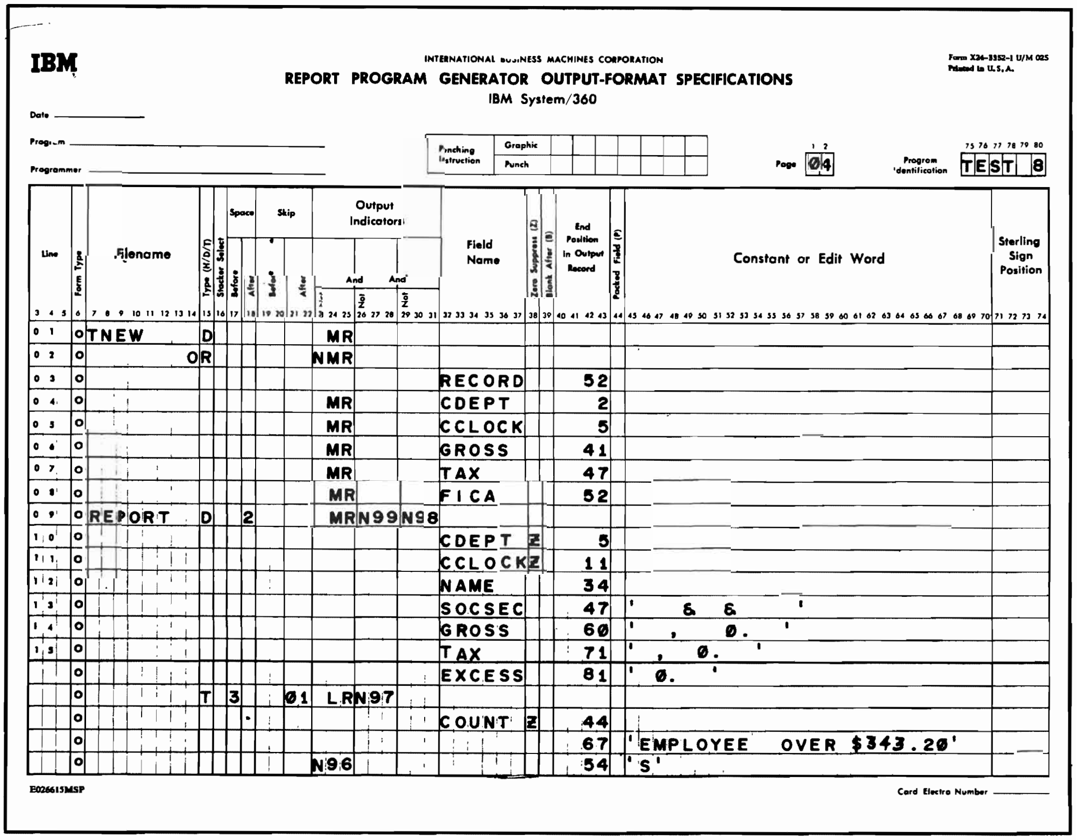
Program Identification TEST 8

Line	Form Type	Control Level (L0-L9, LR)	Indicators (Not)	And (Not)	And (Not)	Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators: Plus / Compare: High 1 > 2	Minus / Low 1 < 2	Zero or Blank / Equal 1 = 2	Comments
3 4 5	6	7 8	9 10 11	12 13 14	15 16 17	18–27	28–32	33–42	43–48	49 50 51	52	53	54 55	56 57	58 59	60–74
0 1	C		MR			OLDG	ADD	CGROSS	GROSS	7	2					
0 2	C		MR			OLDTAX	ADD	CTAX	TAX	6	2					
0 3	C		MR			OLDF	ADD	CFICA	FICA	5	2					
0 4	C		MR			FICA	SUB	343.2Ø	EXCESS	4	2			99	98	
0 5	C		MR	N99	N98	COUNT	ADD	1	COUNT	2	Ø					
0 6	C	LR				COUNT	SUB	1	TEST	2	Ø			97	96	
0 7	C															
0 8	C															
0 9	C															
1 0	C															
1 1	C															
1 2	C															
1 3	C															
1 4	C															
1 5	C															
	C															
	C															
	C															
	C															
	C															

E026615MSP

Card Electro Number

Figure 6a.



IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

REPORT PROGRAM GENERATOR OUTPUT-FORMAT SPECIFICATIONS

IBM System/360

Form X24-3352-1 U/M 025
Printed in U.S.A.

Date

Program

Programmer

Punching Instruction: Graphic / Punch

Page Ø4

Program Identification TEST 8

Line	Form Type	Filename	Type (H/D/T)	Stacker Select	Space Before	Space After	Skip Before	Skip After	Output Indicators (Not)	And (Not)	And (Not)	Field Name	Zero Suppress (Z)	Blank After (B)	End Position in Output Record	Packed Field (P)	Constant or Edit Word	Sterling Sign Position
3 4 5	6	7–14	15	16	17	18	19 20	21 22	23 24 25	26 27 28	29 30 31	32–37	38	39	40–43	44	45–70	71–74
0 1	O	TNEW	D						MR									
0 2	O	O	R						NMR									
0 3	O											RECORD			52			
0 4	O								MR			CDEPT			2			
0 5	O								MR			CCLOCK			5			
0 6	O								MR			GROSS			41			
0 7	O								MR			TAX			47			
0 8	O								MR			FICA			52			
0 9	O	REPORT	D			2			MR	N99	N98							
1 0	O											CDEPT	Z		5			
1 1	O											CCLOCK	Z		11			
1 2	O											NAME			34			
1 3	O											SOCSEC			47		' & & '	
1 4	O											GROSS			6Ø		' , Ø. '	
1 5	O											TAX			71		' , Ø. '	
	O											EXCESS			81		' Ø. '	
	O		T		3		Ø1		LR	N97								
	O											COUNT	Z		44			
	O														67		'EMPLOYEE OVER $343.2Ø'	
	O								N96						54		'S'	

E026615MSP

Card Electro Number

Figure 6b.

The calculations necessary to process the input data.

110 Which of these are used on the calculations specifications sheet?

a. Mnemonic operation codes.
b. Actual storage addresses.
c. Symbolic data names.

●●●

a, c

111 A fourth form is shown in Figure 6b. It is used to record the specifications for the ....

●●●

output.

112 The same approach is used on this form as with the others. Each allows you to use (symbolic names/actual addresses) .... to describe data and files, and to state processing steps.

●●●

symbolic names

113 It should be noted that a partial description of the computer to be used is given by the File Description Specifications form (Figure 5a.). Describing the computer is one of the four things you must do to use any programming system. The entries on the File Description Specifications sheet, however, do not completely describe the computer. The remainder of the description is given to the operating system at execute time. We will talk more about this later, when we consider the characteristics of operating systems.

So we see that on these special forms the programmer provides the four major categories of information that are required by all programming systems. Name these four categories.

●●●

A description of the input.
A description of the output.
The processing steps required.
A description of the computer to be used.

114 Some high-level languages permit the programmer to use English-like words and phrases to state the required processing steps and describe the data. Here is an example, from another high-level language:

IF SHIPPING WEIGHT IS LESS THAN 10 GO TO PARCEL POST ROUTINE

Compared to the above example, would you say RPG source language resembles English?

●●●

No. RPG source language statements are composed of mnemonic operation codes and symbolic names for data files, data fields, and storage locations.

115 Even though RPG source statements are less like English than certain other languages, it still is considered to be a high-level language. This is because the specifications sheets enable you to think in terms of the problem and its solution, rather than of the computer features you must use to get the job done. For example, you don't have to specify the exact registers, indexing procedures, switches, I/O operations, and other considerations when you write in RPG. The compiler builds a program that does all these things for you. RPG can be thought of as (machine/problem) .... oriented.

●●●

problem

116 A language that is machine-oriented allows the programmer to specify exactly which machine features he wants to use, at the particular points in the program where he wants to use them.

A language that is problem-oriented leaves it to the compiler to select the machine features to be used and determine when to use them.

Naturally, the compiler may use the machine differently from the way the programmer would use it, and admittedly the object program from a problem-oriented language compiler may not use all the machine features as efficiently as a program from a machine-oriented language compiler. This is because the machine oriented language lets the programmer specify the machine features to be used and when and how.

Which of the following would be most likely to produce programs that make the most efficient use of the computer?

a. Assembler Language (a one-for-one type language).
b. COBOL (a high-level language using English-like words and phrases).
c. FORTRAN (a high-level language resembling mathematical notation).

●●●

Assembler Language. One-for-one languages require the user to write one source language instruction for every machine language instruction. Thus, he will write the instructions to direct the use of every standard computer feature to be used in executing the object program. High-level languages leave the selection of machine features, and the development of the instructions that activate them, to the compiler.

117 Some machines have non-standard features, installed bv special request for certain out-of-the-ordinary operations. Would a problem-oriented language be likely to make use of such features? Why? Does this include RPG?

●●●

No. Compilers are written to assume that only certain standard features are available on the object computer and have no provisions for use of non-standard features. This would include RPG.

118 Earlier, we said that some languages are free-form, that is, they allow the programmer to write his source statements in narrative style, rather than according to a definite format.

The statement IF SHIPPING WEIGHT IS LESS THAN 10 GO TO PARCEL POST ROUTINE tends to be more of a narrative than, for example, a series of instructions like these:

```
COMP        SHPWGT,10
BL          PARPST
MPY         WGT,FRTRATE
```

The last group of instructions would be considered to have a rigidly prescribed format. How would you rate RPG - as free-form or as rigidly prescribed? (Refer to Figures 5a, 5b, 6a, 6b.)

●●●

RPG format is rigidly prescribed. Every specifications sheet entry must be made in a certain way at a certain place on the form.

119 From the list below, select the characteristics that best describe RPG.

1. High-level language
2. Low-level language
3. Source program statements composed of mnemonic operation codes, and symbolic terms representing storage addresses of instructions and data
4. Source program statements resemble English
5. Source program statements resemble mathematical notation
6. Source language format is free-form
7. Source language format is partially restricted
8. Source language format is rigidly prescribed
9. Designed for commercial applications with high report preparation requirements
10. Designed for commercial applications in general
11. Designed for scientific applications in general
12. Designed for both scientific and commercial applications
13. Designed to permit maximum use of machine facilities

14. Designed to permit utilization of unusual machine features

●●●

1, 3, 8, 9

120 Earlier, we said that RPG had other valuable capabilities than its ability to prepare report programs. Here is an example. A program is needed to read in a file of records that periodically require updating, that is, new information is added or old information is deleted to keep these records current. The information needed to make these changes is in another file of records called a transaction file. The program must read in the change records, compare them to the master records, and when a match occurs, apply the change information to the master record.

This type of operation is called file maintenance. Reports may or may not be produced during such a run. RPG, however, can produce programs that not only would prepare the reports, but would perform the maintenance functions as well.

●●●

121 Here's another. In preparing reports, one normally thinks of just taking the information from the input file and printing it out on the report form. However, sometimes the information would be more easily assimilated if it were summarized into totals within various categories. You can deduce that RPG has the ability to .... information from the input files and write it out.

●●●

summarize

122 In addition to summarizing information from the input files, RPG object programs can perform .... runs that update master files with information from transaction files.

●●●

file maintenance

123 We know,of course, that RPG programs in addition to doing file maintenance and summarization of input data, can prepare reports from input data. It's important to note that it can do all these things simultaneously. This demonstrates that RPG is not just limited to programs that write reports. It is an excellent language for all around use in a computer installation.

●●●

124 Which of the choices below most accurately describes RPG?

RPG object programs will:

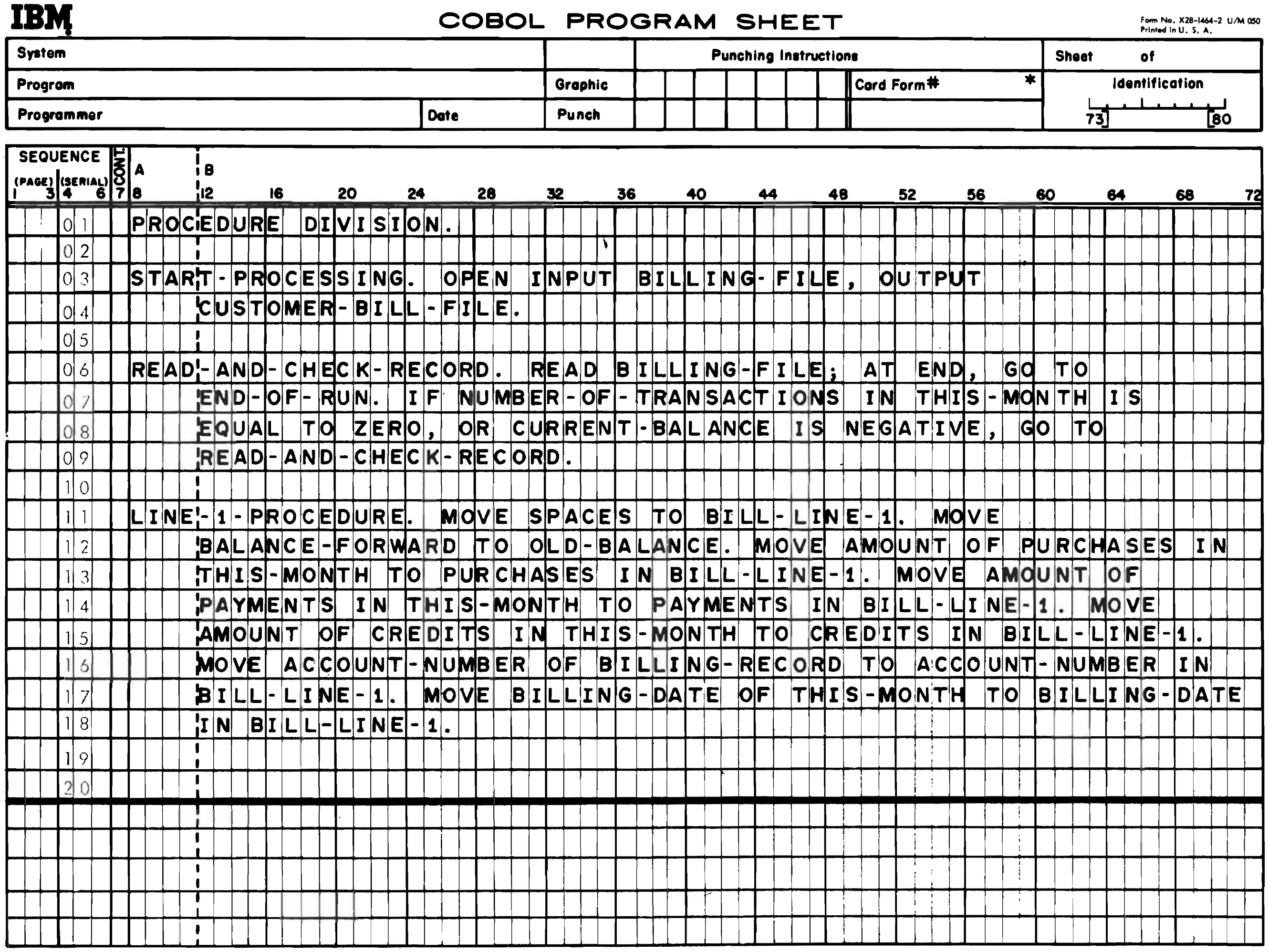
a. perform file maintenance functions.
b. extract and summarize specific information from input files and write the summarized information on a new output file.
c. print detailed reports containing processed information from input files, independently or in conjunction with file maintenance and/or summary file preparation.
d. do only (a) and (c) above.
e. do neither (a), nor (c) above.

●●●

a, b, c

125 A programming language currently in use is COBOL. If you are familiar with the general capabilities and characteristics of COBOL, and can:

a. classify COBOL as high-or low-level,
b. identify the four categories of information the user must supply,
c. give the application area (commercial and/or scientific) for which COBOL was designed,
d. describe COBOL as free-form, partially restricted, or rigidly prescribed, and
e. identify COBOL coding as more closely resembling English or mathematical notation,



IBM COBOL PROGRAM SHEET

Form No. X28-1464-2 U/M 050
Printed in U. S. A.

System		Punching Instructions	Sheet of
Program	Graphic	Card Form# *	Identification
Programmer Date	Punch		73 80

SEQUENCE (PAGE) 1 3 (SERIAL) 4 6 CONT. 7 A 8 B 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72

```
   01  PROCEDURE DIVISION.
   02
   03  START-PROCESSING. OPEN INPUT BILLING-FILE, OUTPUT
   04      CUSTOMER-BILL-FILE.
   05
   06  READ-AND-CHECK-RECORD. READ BILLING-FILE; AT END, GO TO
   07      END-OF-RUN. IF NUMBER-OF-TRANSACTIONS IN THIS-MONTH IS
   08      EQUAL TO ZERO, OR CURRENT-BALANCE IS NEGATIVE, GO TO
   09      READ-AND-CHECK-RECORD.
   10
   11  LINE-1-PROCEDURE. MOVE SPACES TO BILL-LINE-1. MOVE
   12      BALANCE-FORWARD TO OLD-BALANCE. MOVE AMOUNT OF PURCHASES IN
   13      THIS-MONTH TO PURCHASES IN BILL-LINE-1. MOVE AMOUNT OF
   14      PAYMENTS IN THIS-MONTH TO PAYMENTS IN BILL-LINE-1. MOVE
   15      AMOUNT OF CREDITS IN THIS-MONTH TO CREDITS IN BILL-LINE-1.
   16      MOVE ACCOUNT-NUMBER OF BILLING-RECORD TO ACCOUNT-NUMBER IN
   17      BILL-LINE-1. MOVE BILLING-DATE OF THIS-MONTH TO BILLING-DATE
   18      IN BILL-LINE-1.
   19
   20
```

* A standard card form, IBM electro C61897, is available for punching source statements from this form.

Figure 7.

you may skip to frame 138.

●●●

126 COBOL stands for Common Business Oriented Language. It would be more widely used in (commercial/scientific) .... applications.

●●●

commercial

127 An example of COBOL coding is shown in Figure 7. The statements in the example specify some of the procedures the computer is to go through to process the data.

COBOL source language statements more closely resemble (English/mathematical) .... notation.

●●●

English

128 Programs written in COBOL are divided into four parts called divisions. The statements in Figure 7 tell the computer some specific procedures to go through in processing data for a typical commercial problem: computing and printing customer bills.

Procedure statements are part of the Procedure division of COBOL. Since procedure statements tell the computer what to do, you can deduce that the Procedure division relates to which of these programmer activities?

a. Describing the input data.
b. Describing the output data.
c. Stating the required processing steps.
d. Describing the computer to be used.

●●●

c.

129 COBOL coding is not completely free-form, but the format is not rigidly prescribed as in RPG and some other languages. Except for a few specific limitations on margins and indentations, COBOL allows the coder to write pretty much where he pleases on each line. (Of course, the language rules must be obeyed at all times.)

The COBOL coding format is:

a. free-form.
b. partially restricted.
c. rigidly prescribed.

●●●

b.

130 Complete the following sentence.

COBOL is a .... level language employing .... like statements. It features a .... coding format and is used primarily for .... applications.

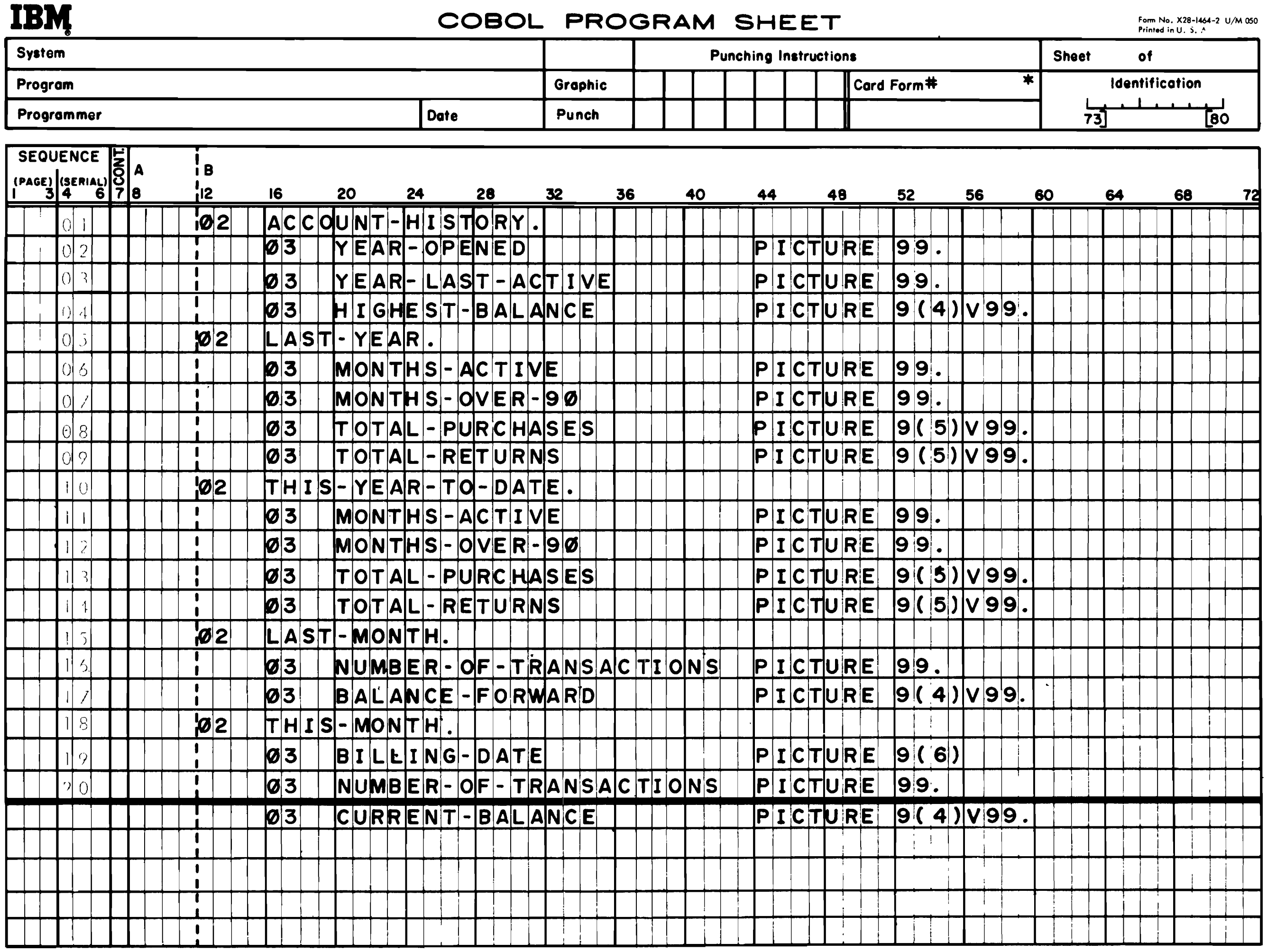
●●●

high
English
partially restricted
commercial

131 As a high-level language, does COBOL permit the programmer to take maximum advantage of all machine features, including special or unusual ones?

●●●

No. Like RPG, COBOL is problem oriented. Choice of machine features to do the processing is left to the compiler. This allows the programmer to think in terms of the job, rather than the computer system.



IBM COBOL PROGRAM SHEET

Form No. X28-1464-2 U/M 050 Printed in U. S. A

System		Punching Instructions	Sheet of
Program	Graphic	Card Form*	Identification 73 80
Programmer Date	Punch		

```
SEQUENCE  CONT. A   B
(PAGE)(SERIAL)
1  3 4  6  7   8   12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
    01     Ø2  ACCOUNT-HISTORY.
    02         Ø3  YEAR-OPENED             PICTURE 99.
    03         Ø3  YEAR-LAST-ACTIVE        PICTURE 99.
    04         Ø3  HIGHEST-BALANCE         PICTURE 9(4)V99.
    05     Ø2  LAST-YEAR.
    06         Ø3  MONTHS-ACTIVE           PICTURE 99.
    07         Ø3  MONTHS-OVER-9Ø          PICTURE 99.
    08         Ø3  TOTAL-PURCHASES         PICTURE 9(5)V99.
    09         Ø3  TOTAL-RETURNS           PICTURE 9(5)V99.
    10     Ø2  THIS-YEAR-TO-DATE.
    11         Ø3  MONTHS-ACTIVE           PICTURE 99.
    12         Ø3  MONTHS-OVER-9Ø          PICTURE 99.
    13         Ø3  TOTAL-PURCHASES         PICTURE 9(5)V99.
    14         Ø3  TOTAL-RETURNS           PICTURE 9(5)V99.
    15     Ø2  LAST-MONTH.
    16         Ø3  NUMBER-OF-TRANSACTIONS  PICTURE 99.
    17         Ø3  BALANCE-FORWARD         PICTURE 9(4)V99.
    18     Ø2  THIS-MONTH.
    19         Ø3  BILLING-DATE            PICTURE 9(6)
    20         Ø3  NUMBER-OF-TRANSACTIONS  PICTURE 99.
               Ø3  CURRENT-BALANCE         PICTURE 9(4)V99.
```

*A standard card form, IBM electro C61897, is available for punching source statements from this form.

Figure 8.

132 We said that there are four things you must do to use a programming system. One is to state the processing steps, which in COBOL is done in the procedure division. A COBOL division is simply a collection of one kind of information that must be recorded for the compiler. The processing steps are one such kind of information; another is the description of the input and output data. Which of these COBOL divisions would be concerned with descriptions of input and output data?

a. Procedure division
b. Data division
c. Identification division
d. Environment division

●●●

b.

133 Data is described in the COBOL Data division. Figure 8 shows a portion of a Data division. English-like names are used for individual data fields in the input/output records, and a "picture" is given of each field. Which of the following is the data field name and which is the "picture"?

9(5)V99
TOTAL-PURCHASES

●●●

TOTAL-PURCHASES is the name of the field; 9(5)V99 is the "picture".

134 The picture clause is a coded representation of the size of the data field and the nature of the data it contains. A picture clause is written by the programmer for each data name to be addressed. This is part of the description of input/output data in the COBOL .... division.

●●●

Data

135 We said that in addition to stating the processing steps (the Procedure division of COBOL) the programmer must describe the computer to be used. This is the third kind of information he must record for the compiler: the fourth is to identify himself and the program he is writing. Match the two items in the first list below with the appropriate division names in the second list.

a. Description of computer
b. Identification of programmer and program

1. Environment division
2. Data division
3. Identification division
4. Procedure division

●●●

a. 1
b. 3

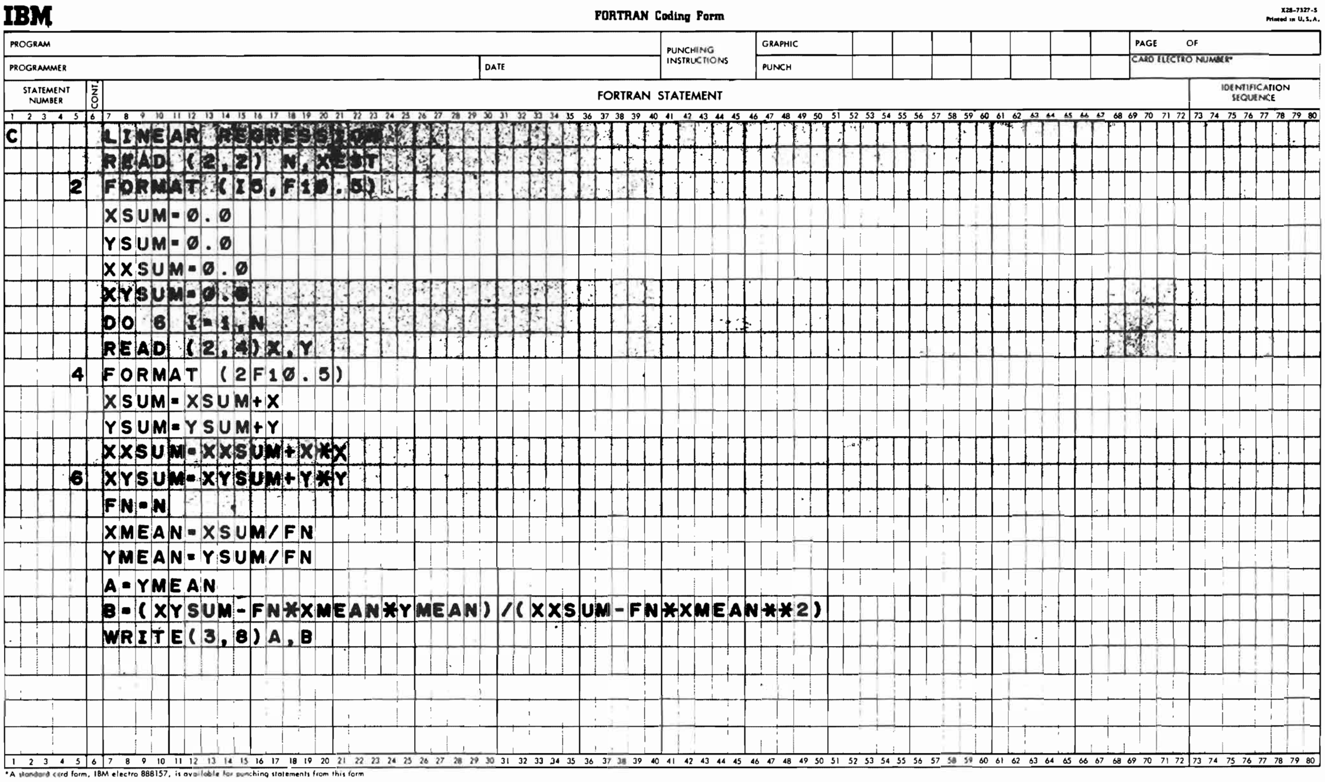
136 We noted in a previous frame that COBOL is manufacturer compatible. This means that a source program in COBOL can be compiled and run on any computer, regardless of the manufacturer, providing there is a compiler for the system to be used. In fact, as we pointed out, the letters CO come from the word .... which implies a more or less universal language.

●●●

common

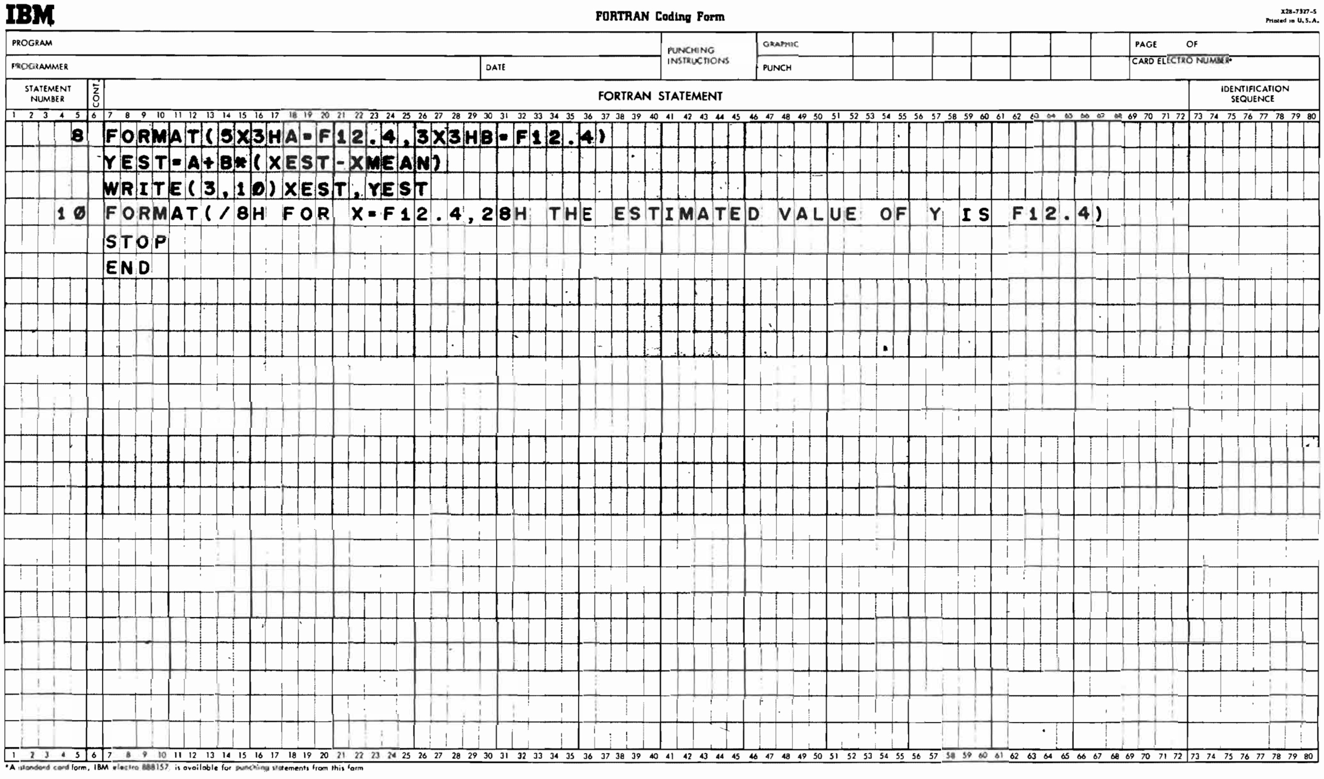
137 From the list below, select the characteristics that best describe COBOL.

1. High-level language
2. Low-level language
3. Source program statements composed of mnemonic operation codes, and symbolic terms representing storage addresses of instructions and data
4. Source program statements resemble English
5. Source program statements resemble mathematical notation
6. Source language format is free form
7. Source language format is partially restricted



IBM FORTRAN Coding Form

```
C     LINEAR REGRESSION
      READ (2,2) N,XEST
    2 FORMAT (I5,F1Ø.5)
      XSUM=Ø.Ø
      YSUM=Ø.Ø
      XXSUM=Ø.Ø
      XYSUM=Ø.Ø
      DO 6 I=1,N
      READ (2,4)X,Y
    4 FORMAT  (2F1Ø.5)
      XSUM=XSUM+X
      YSUM=YSUM+Y
      XXSUM=XXSUM+X*X
    6 XYSUM=XYSUM+Y*Y
      FN=N
      XMEAN=XSUM/FN
      YMEAN=YSUM/FN
      A=YMEAN
      B=(XYSUM-FN*XMEAN*YMEAN)/(XXSUM-FN*XMEAN**2)
      WRITE(3,8)A,B
```



IBM FORTRAN Coding Form

```
    8 FORMAT(5X3HA=F12.4,3X3HB=F12.4)
      YEST=A+B*(XEST-XMEAN)
      WRITE(3,1Ø)XEST,YEST
   1Ø FORMAT(/8H FOR X=F12.4,28H THE ESTIMATED VALUE OF Y IS F12.4)
      STOP
      END
```

Figure 9.

8. Source language format is rigidly prescribed
9. Designed for commercial applications with high report preparation requirements
10. Designed for commercial applications in general
11. Designed for scientific applications in general
12. Designed for both scientific and commercial applications
13. Designed to permit maximum use of machine facilities
14. Designed to permit utilization of unusual machine features

●●●

1, 4, 7, 10

138 FORTRAN is an acronym formed from the words FORmula TRANslation. It is the name of a programming language widely used in coding scientific and engineering problems for computers. If you are familiar with the general characteristics and capabilities of FORTRAN and can:

1. classify FORTRAN as high- or low-level,
2. identify the coding format as rigidly prescribed, partially restricted, or free form,
3. describe the appearance of source program statements as close to English or to mathematical notation, and
4. select the applications area (commercial or scientific) for which FORTRAN was designed,

you may skip to frame 153.

●●●

139 A programming language that is designed to simplify the writing of programs involving many complex formulas would be more likely to be widely used in (commercial/scientific) .... applications.

●●●

scientific

140 FORTRAN was designed primarily for scientific computing. This is not to say that only scientific work involves formulas. Commercial jobs often feature fairly complex calculations that are prescribed by formula. But normally we think of scientific work as being more coricerned with formula-prescribed calculations.

If there is a basic distinction between scientific and commercial computer work it is this:

1. Commercial jobs have high volumes of input and output data with relatively small amounts of calculating to be done for each record.
2. Scientific jobs have low volumes of input and output data requiring extensive, sometimes very extensive, calculations to be performed on each record.

A payroll program probably would have (low/high) .... I/O volumes and be programmed in (FORTRAN/COBOL).......... An electric power network loading program probably would have (low/high) .... I/O volumes and be programmed in (FORTRAN/COBOL) ....

●●●

high
COBOL
low
FORTRAN

141 An example of FORTRAN coding is shown in Figure 9. Do the individual statements more closely resemble English, or mathematical expressions?

●●●

mathematical expressions

142 Each individual FORTRAN statement must be written on a separate line of the coding form, but within the statements themselves the programmer has a certain amount of freedom to write as he pleases. He may, for example, start the statement in column 7 or in any column to the right of 7. Spacing between characters of the statement is sometimes prescribed by FORTRAN, and sometimes is left to the programmer's discretion.

FORTRAN source statements could be classified as (free form/partially restricted/rigidly specified) ....

● ● ●

partially restricted.

143 You can see that free form coding would be the least demanding, and rigidly specified coding the most demanding, of the programmer. FORTRAN falls somewhere in between.

We know that to use any language the programmer must describe to the compiler the input and output data formats. Which statement(s) in the coding in Figure 9 do you think do this?

● ● ●

The format statements.

144 Format statements describe the data by giving the number of fields in the input record, the size of each field, the nature of the data in the field, and the location of decimal points, if any. For example, in Figure 9 the statement "2 FORMAT (I 5,F10.5)" says that each record contains two fields, one of which is a five-place integer (I5), and the other a ten-place fixed-point number with five decimal places (F10.5).

Read and write statements in FORTRAN always refer to a format statement that denotes the appearance of data being read or written. Read and write statements also refer by number to the input/output device to be used in the operation. The second statement in Figure 9 is a read statement. In this read statement, the format statement number and the input device number are in parentheses. What are these numbers?

● ● ●

format statement no. 2
input device no. 2

145 In addition to statements that describe the data, we require statements that tell the computer to do things. Take the statement XSUM = XSUM + X. This tells the computer to add an amount called "X" to an amount called "XSUM" and store the result in the location where XSUM was stored.

What do you think A = YMEAN causes the computer to do?

● ● ●

Data in storage location called YMEAN is moved to the location called A.

146 FORTRAN uses special symbols to indicate arithmetic operations the computer is to perform. The plus symbol (+) is used for addition, and the minus symbol (-) for subtraction. Multiplication is represented by the (*). Which symbol used in Figure 9 do you think indicates a divide operation?

● ● ●

The slash (/)

147 Statements such as A = YMEAN, XSUM = XSUM + X, and XXSUM = XXSUM + X * X:

a. describe data
b. state processing steps

● ● ●

b.

148 It is possible to loop, branch, set switches, and perform other programming functions in FORTRAN, as in other languages. For example, the DO statement tells the computer to execute a certain group of instructions a given number of times. Thus, the DO statement generates the instructions to perform a ....

●●●

loop.

149 One or more absolute language instructions are developed from each source program statement in FORTRAN. Is FORTRAN a low-level language?

●●●

No. Low-level languages usually are one-for-one.

150 FORTRAN, then, is a high-level language. High-level languages let the programmer think in terms of the (computer system/problem to be programmed) ....

●●●

problem to be programmed.

151 Problem-oriented languages such as FORTRAN, for the most part, allow the user to take advantage of unusual and special features on the computer system. True or false?

●●●

False. Problem-oriented languages depend on the compiler to build the processing routines in the object program. The compiler usually is not designed to include instructions that will operate special and unusual features.

152 FORTRAN has one other important feature. Like COBOL, it is manufacturer compatible. This means that programs written in FORTRAN can be compiled and run on computers made by different manufacturers, with only minor changes in the source program. Of course, there must be a compiler for the system on which the object program is to run.

From the list below, select the characteristics that best describe FORTRAN.

1. High-level language
2. Low-level language
3. Source program statements composed of mnemonic operation codes, and symbolic terms representing storage addresses of instructions and data
4. Source program statements resemble English
5. Source program statements resemble mathematical notation
6. Source language format is free form
7. Source language format is partially restricted
8. Source language format is rigidly restricted
9. Designed for commercial applications with high report preparation requirements
10. Designed for commercial applications in general
11. Designed for scientific applications in general
12. Designed for both scientific and commercial applications
13. Designed to permit maximum use of machine facilities
14. Designed to permit utilization of unusual machine features

●●●

1, 5, 7, 11

153 PL/I is the name of a language developed for the IBM System/360. If you are already familiar with the characteristics and capabilities of PL/I and can

a. identify PL/I as high- or low-level,
b. describe the PL/I coding format as free-form, partially restricted, or rigidly prescribed,
c. describe PL/I source statements as resembling English or mathematical notation, and,
d. identify the application area(s) (scientific and/or commercial) for which the language was designed,

you may skip to frame 163.

●●●

```
UPDATE: PROCEDURE;

          DECLARE PAY_# DECIMAL FIXED (7),
                  LOAN_# DECIMAL FIXED (7),
                  PRINCIPAL DECIMAL FIXED (8,2),
                  BALANCE DECIMAL FIXED (8,2),
                  PAYMENT DECIMAL FIXED (6,2),
                  REFUND DECIMAL FIXED (6,2),
                  INTEREST DECIMAL FIXED (5,2),
                  RATE DECIMAL FIXED (3,3),
                  MASTER FILE INPUT,
                  NEW_MASTER FILE OUTPUT,
                  INPUT FILE INPUT,
                  OUTPUT FILE OUTPUT;
          ON ENDFILE (INPUT) GO TO MASTER_FILE;

NEW_RECORD: GET FILE (INPUT) LIST (PAY_#,PAYMENT);
MASTER_FILE: GET FILE (MASTER) LIST (LOAN_#,PRINCIPAL, RATE);
          INTEREST=PRINCIPAL *RATE/12;
          IF LOAN_#¬=PAY_#
            THEN DO;
                  PRINCIPAL=PRINCIPAL + INTEREST;
                  PUT FILE (NEW_MASTER) LIST (LOAN_#,PRINCIPAL,RATE);
                  GO TO MASTER_FILE;
                  END;
            ELSE IF PAYMENT<=PRINCIPAL + INTEREST
                  THEN BALANCE = PRINCIPAL + INTEREST–PAYMENT;
                  ELSE DO;
                    BALANCE = 0;
                    REFUND = PAYMENT–PRINCIPAL + INTEREST;
                    PUT FILE (OUTPUT) LIST (LOAN_#, 'REFUND:   ', REFUND);
                    END;
          PUT FILE (OUTPUT) LIST (LOAN_#,PRINCIPAL, INTEREST,PAYMENT,
            BALANCE);
          IF BALANCE = 0
            THEN GO TO NEW_RECORD;
            ELSE PUT FILE (NEW MASTER) LIST (LOAN_#,BALANCE,RATE);
          GO TO NEW_RECORD;
          END UPDATE;
```

Figure 10.

154 In the past, the development of programming languages has been influenced by the kind of problems that would be coded in the languages. As we have seen, RPG was developed for jobs with heavy report preparation requirements. COBOL came from the need for a common language for commercial applications, while FORTRAN evolved from the needs of the scientific community for a language geared to mathematical and engineering work.

Today the line between commercial and scientific computing is not so distinct as it once was. Scientific approaches to commercial problems are common, while actual scientific jobs are finding a need for the high volume input/output techniques of the commercial world. Few computers are used exclusively for one kind of work - they are much more likely to be shared by users from both accounting and engineering, for example. Most installations, however, still use different languages for different kinds of jobs. This means that in a given installation there may be source programs written in any one of several languages. Due to the trend towards specialization, some programmers concentrate on commercial languages, others on scientific. Communication among programmers is handicapped in that they don't speak the same language to the computer.

In an effort to overcome this problem and at the same time to develop a language that would let the programmer exploit the full power of S/360, IBM developed PL/I. It is a high-level language that can be used with equal ease and efficiency by scientific and commercial programmers.

Why was PL/I developed? (Your own words.)

●●●

To make available a single high-level language that is equally useful in scientific and commercial work, and to permit the programmer to use the S/360 with maximum efficiency.

155 PL/I shows the influences of both COBOL and FORTRAN. Below are two statements from a PL/I program. Which resembles COBOL and which FORTRAN?

a. IF PAYROLL.NAME = PAY_RECORD.NAME THEN DO;

b. INTEREST = PRINCIPAL * RATE / 12;

●●●

a. COBOL
b. FORTRAN

156 This doesn't mean that PL/I is simply a hybrid, made of equal parts of other languages. It does mean that the best features of some existing languages have been incorporated, along with some new and powerful language concepts.

Figure 10 shows an example of PL/I coding. Although individual statements are shown written on separate lines on the form, this is not required. Statements can be written one after the other on the same line, and continued on successive lines as required. Spacing between characters and phrases is generally left to the coder's discretion. The only consideration is that each statement must be terminated by a semicolon (;).

A statement terminated by the correct character (can/cannot) .... be followed immediately on the line by another statement.

●●●

can

157 How would you classify the coding format of PL/I?

a. Rigidly prescribed.
b. Free-form.
c. Partially restricted.

●●●

Compared to other existing languages, PL/I is free-form. Of course, the language rules for syntax and punctuation must be followed, but otherwise the programmer is given wide latitude in the placement of source statements on the coding form.

158 Figure 10 contains these entries:

DECLARE PAY_#DECIMAL FIXED (7),
LOAN_ #DECIMAL FIXED (7),

IF LOAN_#¬= PAY_ # THEN DO; PRINCIPAL = PRINCIPAL + INTEREST;

Which do you think defines data and which states processing steps.

•••

The DECLARE entry describes two data fields fields called PAY_# and LOAN_# as being decimal integers of seven places each.
The IF entry prescribes processing steps, based on the results of a comparison between two fields called LOAN_# and PAY_# .

159 Like other languages, PL/I has facilities for describing input/output data, stating processing steps, reading and writing records, etc. Would you assume that PL/I allows the programmer to set switches, perform loops, branch, count, and do table lookup?

•••

It does.

160 For what computer system was PL/I developed?

•••

The IBM System/360.

161 Each PL/I statement generates one or more absolute language instructions. Would you classify PL/I as a high-level language?

•••

yes

162 We mentioned previously that high-level languages, in general, do not permit the maximum use of special and non-standard machine features. This is true of PL/I, although to a lesser degree than of some other languages. PL/I does, however, generate highly efficient programs for S/360 since it was designed for that system. Maximum efficiency in object programs, though, is still obtainable only through the use of one-for-one languages.

From the list below, select those characteristics that best describe PL/I.

1. High-level language
2. Low-level language
3. Source program statements composed of mnemonic operation codes, and symbolic terms representing storage addresses of instructions and data
4. Some source program statements resemble English
5. Some source program statements resemble mathematical notation
6. Source language format is free-form
7. Source language format is partially restricted
8. Source language format is rigidly restricted
9. Designed for commercial applications with high report preparation requirements
10. Designed for commercial applications in general
11. Designed for scientific applications in general
12. Designed for both scientific and commercial applications
13. Designed to permit maximum use of machine facilities
14. Designed to permit utilization of unusual machine features

•••

1, 4, 5, 6, 12

163 The programming languages we have been considering are all high-level; that is, their source statements resemble English or mathematical notation and result in one or more machine language statements in the object program. In general, they tend to be free-form (although some are not) and they permit the programmer to think in terms of his problem rather than of the computer itself. Thus they are said to be problem oriented.

Assembler Language is another language for the S/360. We shall consider it briefly. If you are familiar with the characteristics and capabilities of Assembler Language, and can:

1. classify Assembler Language as high- or low-level,
2. identify the coding format as rigidly prescribed, partially restricted, or free-form,
3. describe the appearance of source program statements, and
4. identify the application area(s) for which Assembler Language was designed,

you may skip to frame 184.

●●●

164 Assembler Language is a one-for-one programming language. This means that each source language statement results in (how many?) .... machine language statements.

●●●

one

165

OP CODE	FIRST OPERAND	SECOND OPERAND

Machine language instructions, as shown above, typically consist of an operation code that tells the computer what to do, and one or more operands that specify the storage or register addresses of data to be processed.

This is an instruction in Assembler Language:

AP	SUM, TOTAL

Does its format more closely resemble high-level language formats or machine language format? Why?

●●●

It resembles machine language format, because it has an operation code (ADD) and two operands (SUM and TOTAL) representing storage addresses of data to be processed.

166 The operands in Assembler Language instructions can refer either to storage addresses or to registers. Channel Commands, which are just a special form of instruction, activate and control the I/O units. Thus, Assembler Language instructions allow the programmer more direct control over the different parts of the computer than do the high-level languages. (You remember that high-level languages depend on the compiler to select the exact machine instructions and, in some cases, the machine components, to be used for any given series of steps.)

On this basis, we can say that Assembler Language is (problem/machine) .... oriented.

●●●

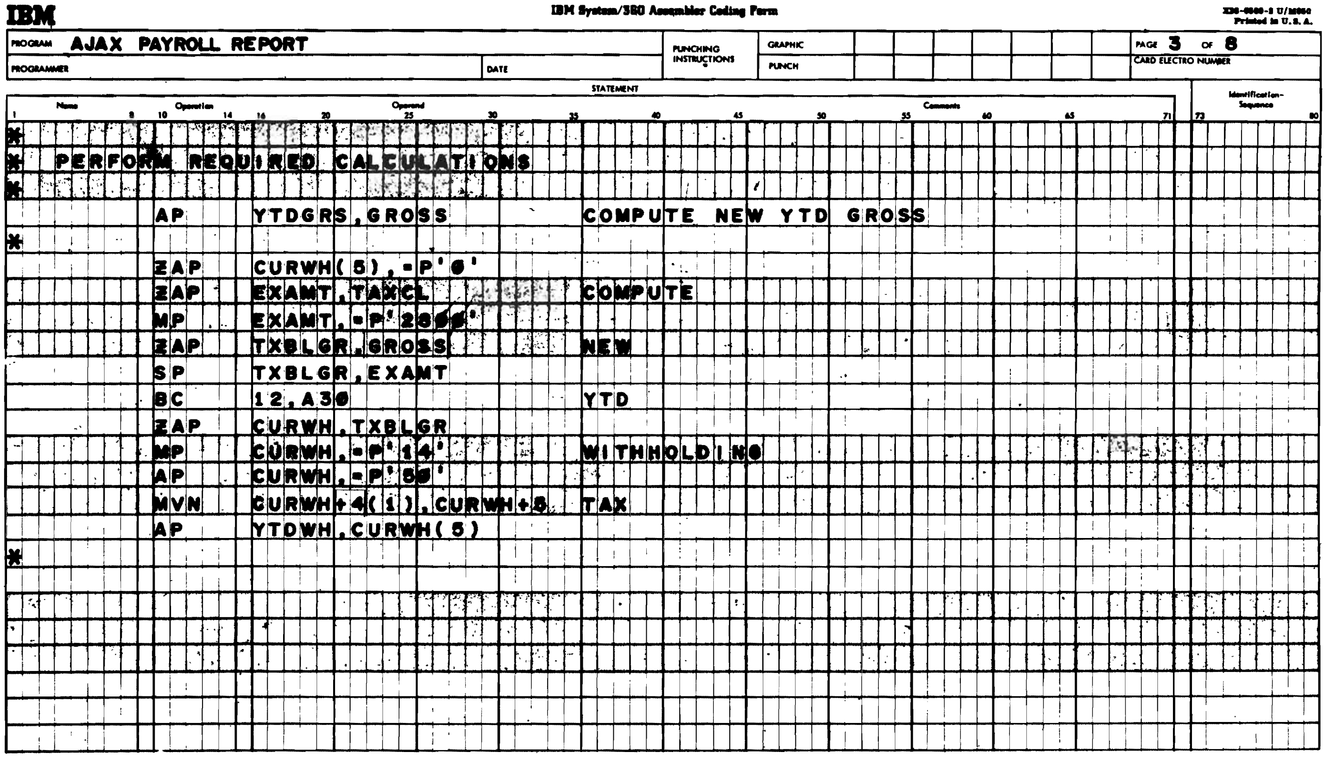
machine

167 A machine-oriented language requires the programmer to think more in terms of the (computer system/problem to be programmed) .... A high-level language permits the programmer to think more in terms of the (computer system/problem to be programmed).........

●●●

computer system.
problem to be programmed.

168 When you, as a programmer, have the power to select the exact machine instructions and components to be used by the object program,you often can write programs of maximum efficiency. However, since a one-for-one language employs one source statement for each instruction developed in the object problem, it may take you (more/less)



IBM

IBM System/360 Assembler Coding Form

[illegible]
Printed in U.S.A.

PROGRAM AJAX PAYROLL REPORT		PUNCHING INSTRUCTIONS	GRAPHIC		PAGE 3 OF 8
PROGRAMMER	DATE		PUNCH		CARD ELECTRO NUMBER

```
*
*  PERFORM REQUIRED CALCULATIONS
*
         AP    YTDGRS,GROSS        COMPUTE NEW YTD GROSS
*
         ZAP   CURWH(5),=P'0'
         ZAP   EXAMT,TAXCL         COMPUTE
         MP    EXAMT,=P'2500'
         ZAP   TXBLGR,GROSS        NEW
         SP    TXBLGR,EXAMT
         BC    12,A30              YTD
         ZAP   CURWH,TXBLGR
         MP    CURWH,=P'14'        WITHHOLDING
         AP    CURWH,=P'50'
         MVN   CURWH+4(1),CURWH+5  TAX
         AP    YTDWH,CURWH(5)
*
```

Figure 11

.... time to write the program in Assembler Language than it would if you wrote in, say, COBOL or FORTRAN.

●●●

more

169 Thus, the choice of a programming language can be governed, at least in part, by the degree of efficiency you want in the object program compared to the time that will be required to write the program.

●●●

170 The necessity for thinking in terms of the machine is not without its advantages, however. Suppose your object machine has some special features that are not standard on the S/360. Could you take advantage of these special features by programming in Assembler Language? Why?

●●●

Yes, because Assembler Language provides a source statement for each instruction in S/360 machine language. You have only to write the appropriate source statements and the special features can be utilized. High-level language compilers do not provide this, since they are written for standard configurations and generally do not have the ability to compile statements that activate special machine features.

171 Figure 11 shows a few instructions from a program coded in Assembler Language. As we previously pointed out, each instruction consists of an operation code and one or more operands. The operation codes are mnemonic; that is, they indicate by their spelling or general construction the nature of the operation involved.

AR is the operation code for Add Register (when each operand is in a separate register.)
SR is the operation code for Subtract Register.
M is the operation code for ....
D is the operation code for ....

●●●

multiply.
divide.

172 In addition to mnemonic operation codes, Assembler Language permits the use of symbolic names for the storage addresses of data to be processed. These names usually are constructed to indicate the nature of the data whose storage addresses they represent. For example, GROSS could be used to indicate an employee's total earnings, NETPY to indicate his take-home pay. You could use TXCL to indicate the employee's ....

●●●

tax class.

173 Note that these abbreviations do not have to be English words. Any combination of letters, numbers and special characters may be used (with a few exceptions) that is meaningful to the programmer. They could be entirely unintelligible to someone else.

For example, the symbolic name TNPRDLY could represent "tons produced daily". It is not an English word, yet is easier to remember than the actual address where the data would be stored.

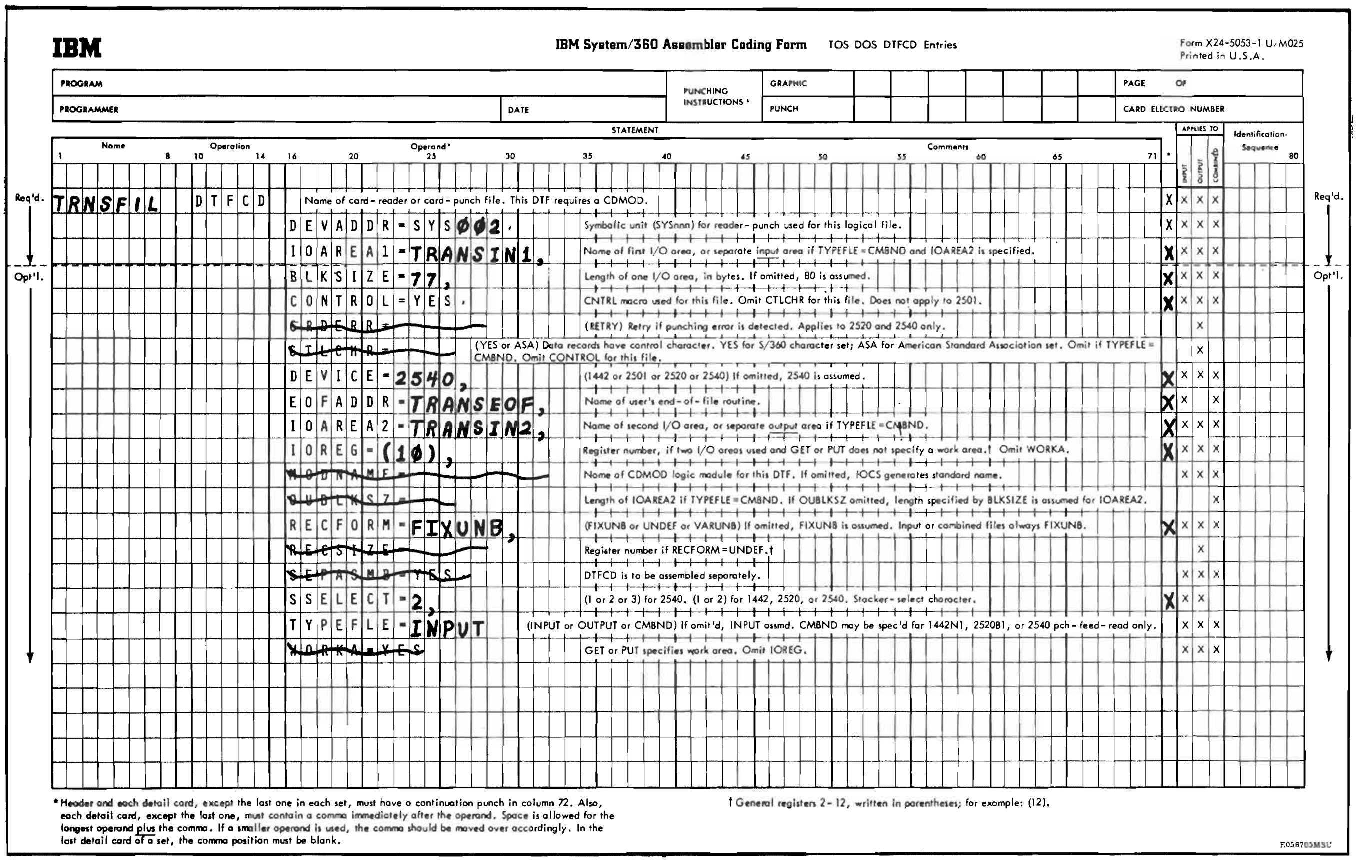
In a payroll program TXBLGR, while not an English word, might be recognized as meaning ....

●●●

taxable gross.

174 Which of these three statements best describes Assembler Language?

1. Source program statements resemble mathematical notation.
2. Source program statements composed of mnemonic operation codes, and symbolic



IBM IBM System/360 Assembler Coding Form TOS DOS DTFCD Entries

Form X24-5053-1 U/M025
Printed in U.S.A.

PROGRAM | PUNCHING INSTRUCTIONS * | GRAPHIC | PAGE OF
PROGRAMMER | DATE | PUNCH | CARD ELECTRO NUMBER

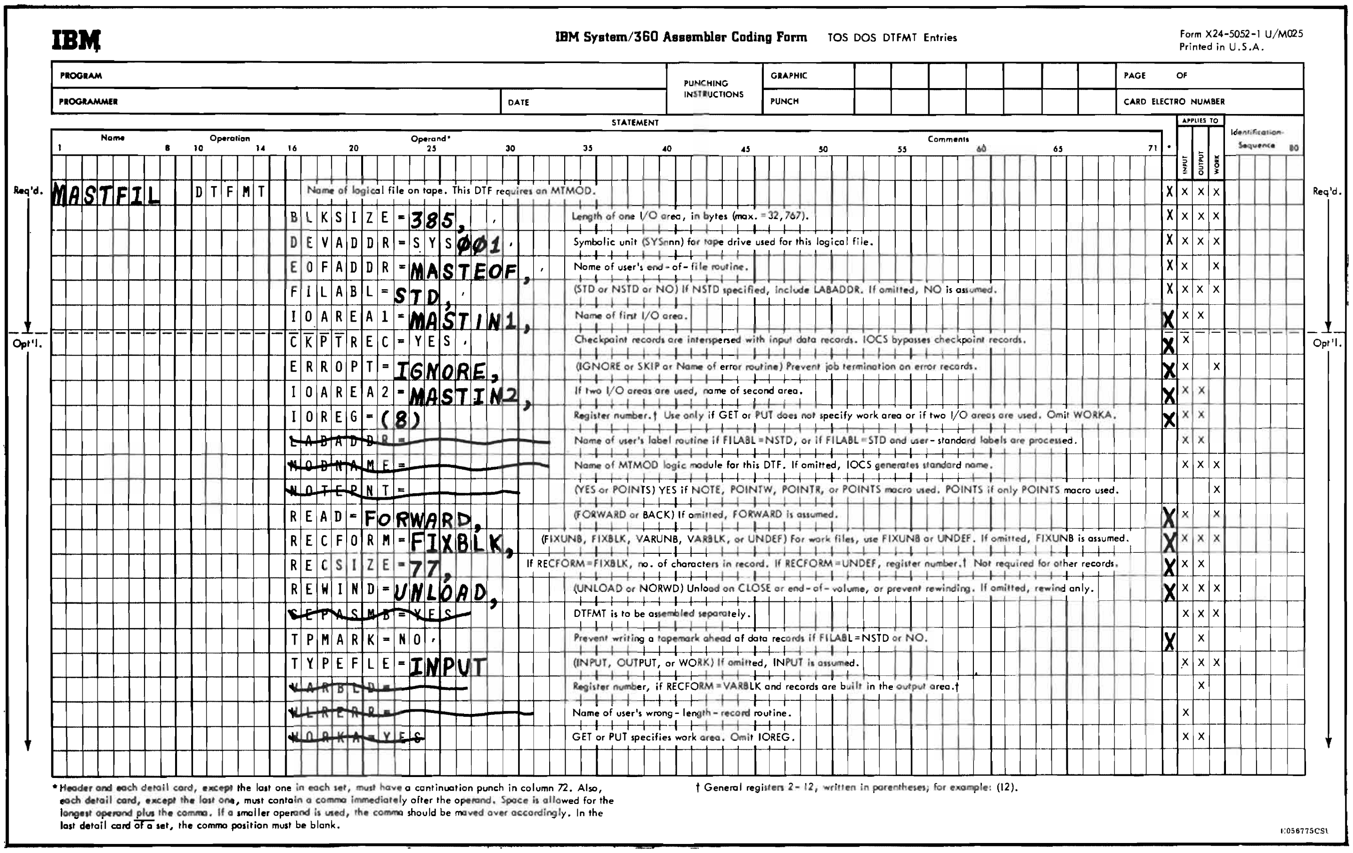
STATEMENT

	Name	Operation	Operand*	Comments	*	Input	Output	Combined	Identification-Sequence
Req'd.	TRNSFIL	DTFCD		Name of card-reader or card-punch file. This DTF requires a CDMOD.	X	X	X	X	
			DEVADDR=SYSØØ2,	Symbolic unit (SYSnnn) for reader-punch used for this logical file.	X	X	X	X	
			IOAREA1=TRANSIN1,	Name of first I/O area, or separate input area if TYPEFLE=CMBND and IOAREA2 is specified.	X	X	X	X	
Opt'l.			BLKSIZE=77,	Length of one I/O area, in bytes. If omitted, 80 is assumed.	X	X	X	X	
			CONTROL=YES,	CNTRL macro used for this file. Omit CTLCHR for this file. Does not apply to 2501.	X	X	X	X	
			~~CRDERR=~~	(RETRY) Retry if punching error is detected. Applies to 2520 and 2540 only.			X		
			~~CTLCHR=~~	(YES or ASA) Data records have control character. YES for S/360 character set; ASA for American Standard Association set. Omit if TYPEFLE=CMBND. Omit CONTROL for this file.			X		
			DEVICE=2540,	(1442 or 2501 or 2520 or 2540) If omitted, 2540 is assumed.	X	X	X	X	
			EOFADDR=TRANSEOF,	Name of user's end-of-file routine.	X	X		X	
			IOAREA2=TRANSIN2,	Name of second I/O area, or separate output area if TYPEFLE=CMBND.	X	X	X	X	
			IOREG=(1Ø),	Register number, if two I/O areas used and GET or PUT does not specify a work area.† Omit WORKA.	X	X	X	X	
			~~MODNAME=~~	Name of CDMOD logic module for this DTF. If omitted, IOCS generates standard name.		X	X	X	
			~~OUBLKSZ=~~	Length of IOAREA2 if TYPEFLE=CMBND. If OUBLKSZ omitted, length specified by BLKSIZE is assumed for IOAREA2.				X	
			RECFORM=FIXUNB,	(FIXUNB or UNDEF or VARUNB) If omitted, FIXUNB is assumed. Input or combined files always FIXUNB.	X	X	X	X	
			~~RECSIZE=~~	Register number if RECFORM=UNDEF.†			X		
			~~SEPASMB=YES~~	DTFCD is to be assembled separately.		X	X	X	
			SSELECT=2,	(1 or 2 or 3) for 2540. (1 or 2) for 1442, 2520, or 2540. Stacker-select character.	X	X	X		
			TYPEFLE=INPUT	(INPUT or OUTPUT or CMBND) If omit'd, INPUT assmd. CMBND may be spec'd for 1442N1, 2520B1, or 2540 pch-feed-read only.		X	X	X	
			~~WORKA=YES~~	GET or PUT specifies work area. Omit IOREG.		X	X	X	

*Header and each detail card, except the last one in each set, must have a continuation punch in column 72. Also, each detail card, except the last one, must contain a comma immediately after the operand. Space is allowed for the longest operand plus the comma. If a smaller operand is used, the comma should be moved over accordingly. In the last detail card of a set, the comma position must be blank.

† General registers 2-12, written in parentheses; for example: (12).

E056705MSC



IBM IBM System/360 Assembler Coding Form TOS DOS DTFMT Entries

Form X24-5052-1 U/M025
Printed in U.S.A.

PROGRAM | PUNCHING INSTRUCTIONS | GRAPHIC | PAGE OF
PROGRAMMER | DATE | PUNCH | CARD ELECTRO NUMBER

STATEMENT

	Name	Operation	Operand*	Comments	*	Input	Output	Work	Identification-Sequence
Req'd.	MASTFIL	DTFMT		Name of logical file on tape. This DTF requires an MTMOD.	X	X	X	X	
			BLKSIZE=385,	Length of one I/O area, in bytes (max. = 32,767).	X	X	X	X	
			DEVADDR=SYSØØ1,	Symbolic unit (SYSnnn) for tape drive used for this logical file.	X	X	X	X	
			EOFADDR=MASTEOF,	Name of user's end-of-file routine.	X	X		X	
			FILABL=STD,	(STD or NSTD or NO) If NSTD specified, include LABADDR. If omitted, NO is assumed.	X	X	X	X	
			IOAREA1=MASTIN1,	Name of first I/O area.	X	X	X		
Opt'l.			CKPTREC=YES,	Checkpoint records are interspersed with input data records. IOCS bypasses checkpoint records.	X	X			
			ERROPT=IGNORE,	(IGNORE or SKIP or Name of error routine) Prevent job termination on error records.	X	X		X	
			IOAREA2=MASTIN2,	If two I/O areas are used, name of second area.	X	X	X		
			IOREG=(8)	Register number.† Use only if GET or PUT does not specify work area or if two I/O areas are used. Omit WORKA.	X	X	X		
			~~LABADDR=~~	Name of user's label routine if FILABL=NSTD, or if FILABL=STD and user-standard labels are processed.		X	X		
			~~MODNAME=~~	Name of MTMOD logic module for this DTF. If omitted, IOCS generates standard name.		X	X	X	
			~~NOTEPNT=~~	(YES or POINTS) YES if NOTE, POINTW, POINTR, or POINTS macro used. POINTS if only POINTS macro used.				X	
			READ=FORWARD,	(FORWARD or BACK) If omitted, FORWARD is assumed.	X	X		X	
			RECFORM=FIXBLK,	(FIXUNB, FIXBLK, VARUNB, VARBLK, or UNDEF) For work files, use FIXUNB or UNDEF. If omitted, FIXUNB is assumed.	X	X	X	X	
			RECSIZE=77,	If RECFORM=FIXBLK, no. of characters in record. If RECFORM=UNDEF, register number.† Not required for other records.	X	X	X		
			REWIND=UNLOAD,	(UNLOAD or NORWD) Unload on CLOSE or end-of-volume, or prevent rewinding. If omitted, rewind only.	X	X	X	X	
			~~SEPASMB=YES~~	DTFMT is to be assembled separately.		X	X	X	
			TPMARK=NO,	Prevent writing a tapemark ahead of data records if FILABL=NSTD or NO.	X		X		
			TYPEFLE=INPUT	(INPUT, OUTPUT, or WORK) If omitted, INPUT is assumed.		X	X	X	
			~~VARBLD=~~	Register number, if RECFORM=VARBLK and records are built in the output area.†			X		
			~~WLRERR=~~	Name of user's wrong-length-record routine.		X			
			~~WORKA=YES~~	GET or PUT specifies work area. Omit IOREG.		X	X		

*Header and each detail card, except the last one in each set, must have a continuation punch in column 72. Also, each detail card, except the last one, must contain a comma immediately after the operand. Space is allowed for the longest operand plus the comma. If a smaller operand is used, the comma should be moved over accordingly. In the last detail card of a set, the comma position must be blank.

† General registers 2-12, written in parentheses; for example: (12).

E056775CS1

Figure 12.

terms representing storage addresses of instructions and data.
3. Source program statements resemble English.

●●●

2.

175 In the example in Figure 11 we see that each component of an Assembler Language instruction is located in specified columns on the coding form. The operation (op code) is left-justified in columns 10-14, the operand(s) start(s) in column 16. If the instruction has a label, or name, so it can be referred to by other instructions, the label is in columns ....

●●●

1-8.

176 From Figure 11 you can tell that, unlike COBOL and PL/I, (how many?) .... source statements in Assembler Language can be written on a single line of the coding form.

●●●

one

177 Assembler Language, then, is not free-form; that is, its source language format does not give the programmer any latitude in the placement of source statements on the coding form.

Which best describes Assembler Language?

1. Source language format is free-form.
2. Source language format is partially restricted.
3. Source language format is rigidly prescribed.

●●●

3.

178 You will remember that to use any programming system you must describe the input and output data, as well as state the processing steps. In COBOL, this is done in the Data Division; in FORTRAN with the format statement. In Assembler Language, it is done with the DTF (Define The File) statement.

Figure 12 shows the entries in typical DTF statements for a master file called MASTFIL and a transaction file called TRNSFIL. The DTF statement for the master file is DTFMT, which means Define The File for Magnetic Tape. DTFCD defines a card file. Note the use of pre-printed forms for DTF statements; also that if all possible entries are not required to describe a particular file the unneeded ones are crossed off.

Refer to Figure 12. The master file is an (input/output) .... file.

●●●

input (see the last entry in MASTFIL DTF).

179 The master file is recorded on (cards/disk/tape) ....

●●●

tape.

180 How many characters are in one block (group of data records all of which are read consecutively during a single read operation) in the master file?

●●●

385 (see second entry in MASTFIL DTF's).

181 You can see that a DTF statement provides a complete description of the data and the file that contains it. There must be one DTF statement for each file involved in the run.

The initials DTF stand for ....

●●●

define the file.

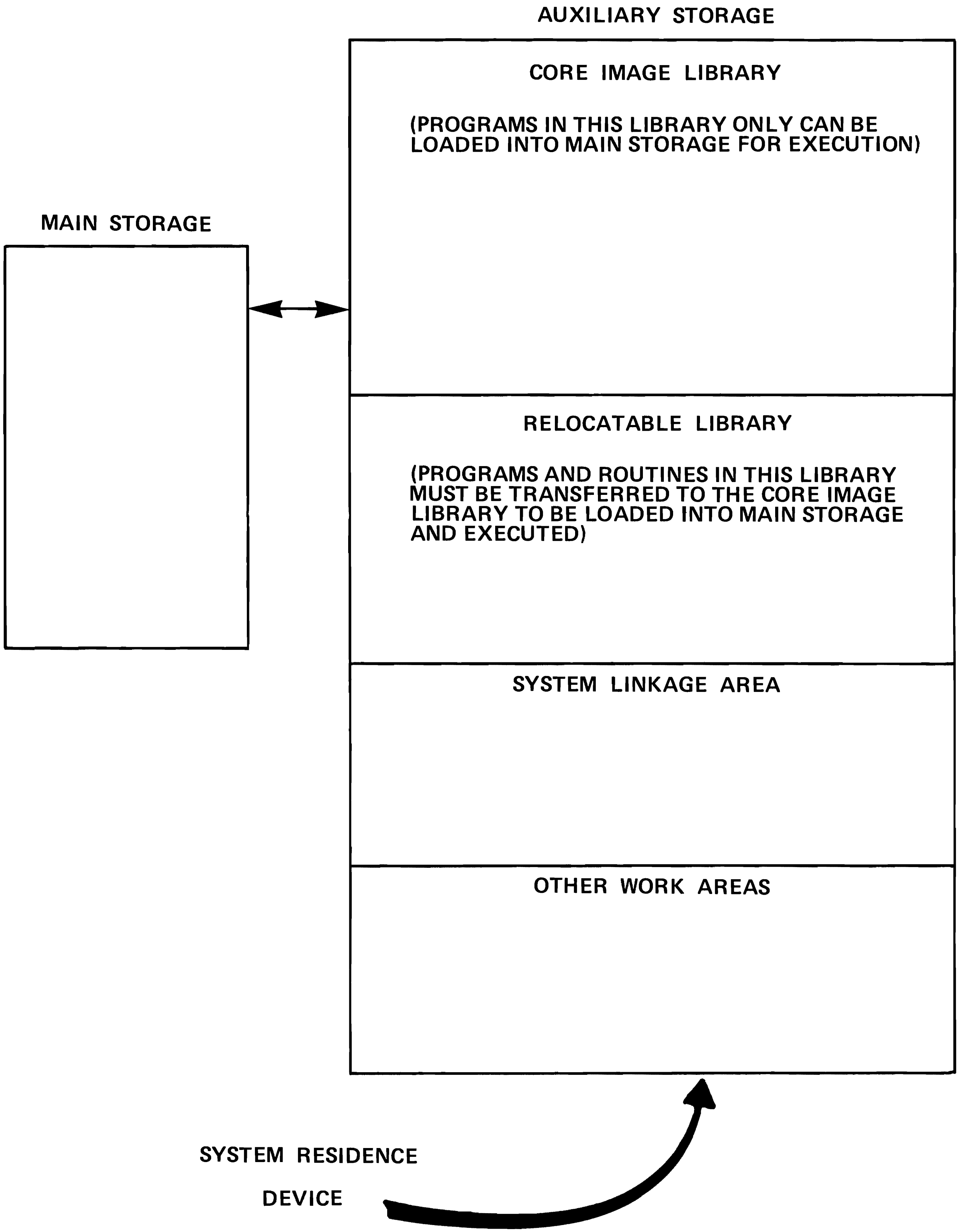

AUXILIARY STORAGE
CORE IMAGE LIBRARY
(PROGRAMS IN THIS LIBRARY ONLY CAN BE LOADED INTO MAIN STORAGE FOR EXECUTION)
MAIN STORAGE
RELOCATABLE LIBRARY
(PROGRAMS AND ROUTINES IN THIS LIBRARY MUST BE TRANSFERRED TO THE CORE IMAGE LIBRARY TO BE LOADED INTO MAIN STORAGE AND EXECUTED)
SYSTEM LINKAGE AREA
OTHER WORK AREAS
SYSTEM RESIDENCE
DEVICE

Figure 13.

182 Assembler Language was designed primarily for commercial applications, but can be used for any kind of job. Seldom would a scientific programmer choose Assembler Language over FORTRAN for his jobs, but it is not impossible. Most usage of Assembler Language is in the commercial area.

From the list below, select the characteristics that best describe Assembler Language.

1. High-level language
2. Low-level language
3. Source program statements composed of mnemonic operation codes, and symbolic terms representing storage addresses of instructions and data
4. Source program statements resemble English
5. Source program statements resemble mathematical notation
6. Source language format is free-form
7. Source language format is partially restricted
8. Source language format is rigidly prescribed
9. Designed for commerçial applications with high report preparation requirements
10. Designed for commercial applications in general
11. Designed for scientific applications in general
12. Designed for both scientific and commercial applications
13. Designed to permit maximum use of machine facilities
14. Designed to permit utilization of unusual machine features

●●●

2, 3, 8, 10, 13, 14

183 Associate each of the items in the first list below with one of those in the second list.

a. One source program statement is written for each statement developed in the object program.

b. One or more object program statements is developed from each source program statement.

c. Source program statements more closely resemble English or mathematical notation, depending on the language involved.

d. Source program statements may be completely free-form, or may follow a certain amount of coding conventions, depending on the language involved.

e. Source program statement format is rigidly prescribed.

f. Problem oriented.

g. Machine oriented.

1. High-level symbolic language.
2. Low-level symbolic language.

●●●

a.	2
b.	1
c.	1
d.	1
e.	2
f.	1
g.	2

184 In sections 1, 2, and 3 of this course you were introduced to the concept of a control program which is in storage at all times when the computer is operating, and whose duties include initiating and controlling I/O operations, handling interrupts, and facilitating job transition.

In this section we shall examine the concept of the operating system , of which the control program is one part. If you are familiar with the elements of S/360 operating systems, including such programs as the linkage editor, the initial program loader, the supervisor, the librarian, and the job control program, you may skip the rest of this section.

●●●

185 Fig. 13 shows a schematic drawing of the main storage of a S/360, plus an auxiliary storage unit. This auxiliary storage unit is called the..........

●●●

system residence device.

186 The system residence device holds a file consisting of programs that together make up the operating system for the computer. The programs in the operating system are called into main storage to be executed, as they are needed.

Can you think of a reason why the entire operating system is not kept in main storage?

●●●

Main storage is more expensive than auxiliary storage. It is more economical to use an auxiliary storage device and load the desired program into main storage when it is needed.

187 The system residence device usually is a magnetic disk unit, although other devices such as card readers and magnetic tape units are used on those systems that do not have disk files. What advantage does a disk file have over cards and tape?

●●●

Disks permit direct access to the information they contain. Cards and tape permit sequential access only. This means that the desired program from the operating system can be located in the system residence device and loaded into storage in a minimum of time.

188 The auxiliary storage device on which the operating system is kept is called..........

●●●

the system residence device.

189 We will assume that the system residence device is a magnetic disk file. Fig. 13 is a drawing of a disk file, showing the general areas into which it is divided for use as a system residence device. From which area of the file can programs be loaded into main storage for execution?

●●●

The core image library.

190 Only programs from the core image library can be loaded directly into main storage for execution. What must be done to a program in the relocatable library before it can be loaded into main storage? (See Fig. 13)

●●●

It must be transferred to the core image library.

191 The words "core image" indicate the status of programs in this library. All program elements (instructions, constants, storage and work areas, etc.) have been assigned the actual main storage addresses they will occupy. Thus they are said to be in "core image", in that their positioning in main storage has been established.

You can deduce that programs in the relocatable library (have/have not)..........been assigned the final main storage addresses they will occupy.

●●●

have not

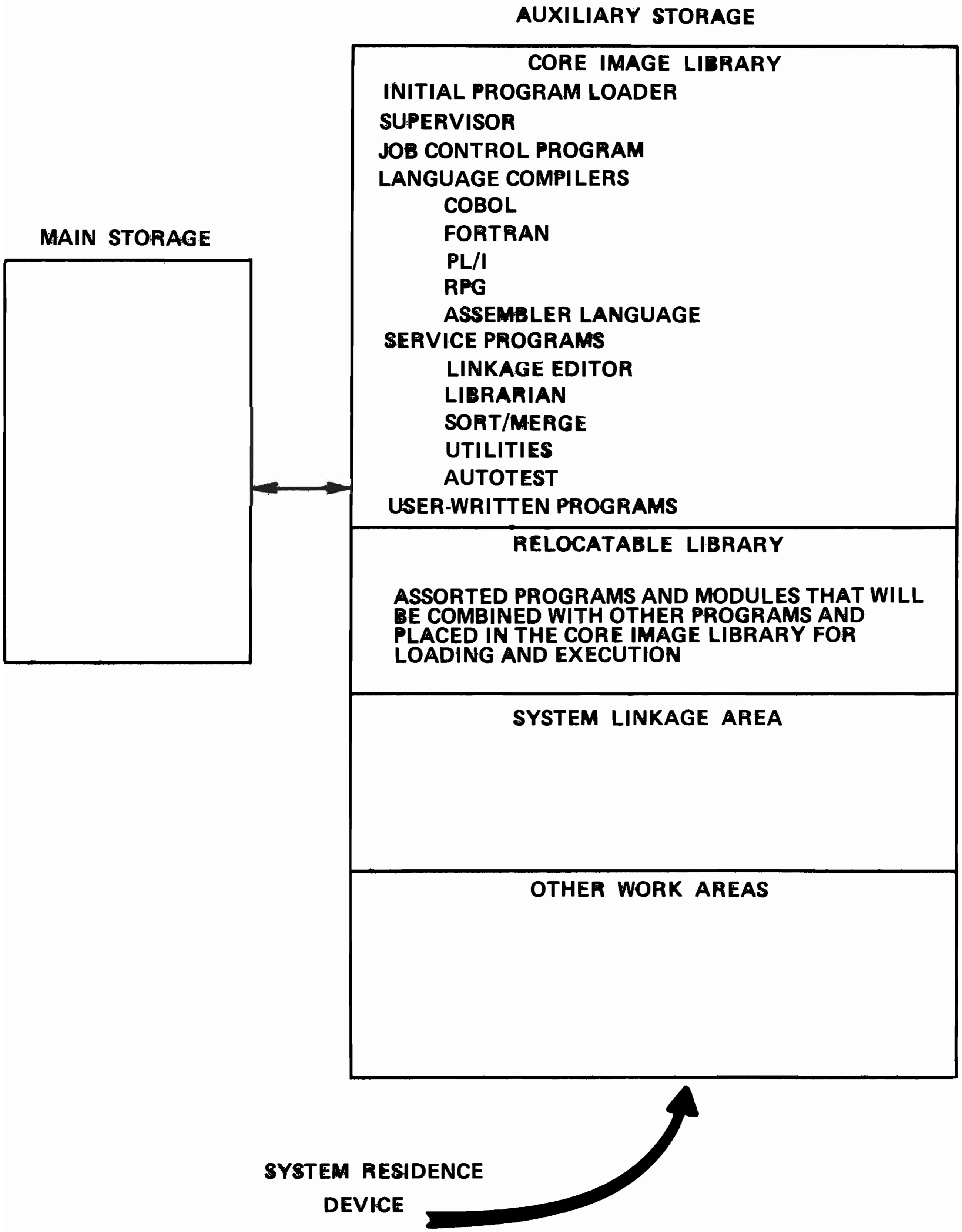
192 One of the things that must be done to programs being transferred from the relocatable library to the core image library is..........

●●●

to assign the main storage addresses the program actually will use when loaded for execution.

193 A little later you will see how this system provides more flexibility and efficiency in the use of an operating system.

Previously we said that when a program in the



ASSORTED PROGRAMS AND MODULES THAT WILL BE COMBINED WITH OTHER PROGRAMS AND PLACED IN THE CORE IMAGE LIBRARY FOR LOADING AND EXECUTION

Figure 14.

operatıng system is wanted,it must be located in the system residence device and then read into main storage. But this means that there must be a program already in storage that will find the desired program in the system residence device and load it into main storage.

Which of these types of programs do you think would do this?

a. compilers
b. control programs
c. user-written problem programs

●●●

control programs

194 You will remember that the S/360 must have a control program in main storage at all times while the computer is operating. The control program supervises the loading and execution of all other programs in the operating system.

Fig. 14 shows a system residence device with a set of operating system programs in the core image library. Which of these programs do you think performs the supervisory tasks of a control program?

●●●

The supervisor.

195 The supervisor is the S/360 control program. It is the one part of every S/360 operating system that is in storage at all times while the computer is operating.

The programs in main storage while you are compiling a FORTRAN source program are ....

●●●

the FORTRAN compiler and the supervisor.

196 The programs in main storage while you are computing your payroll are.............

●●●

the payroll program and the supervisor.

197 Where are the language compilers and the user-written programs kept when not in main storage?

●●●

In the core image library of the system residence device.

198 What other programs are kept in the system residence device? (See Fig. 14)

●●●

All other programs in the operating system, including the initial program loader, the job control program, and the service programs.

199 So we see that all the programs the computer will execute are stored on the system residence device. What is this collection of programs called?

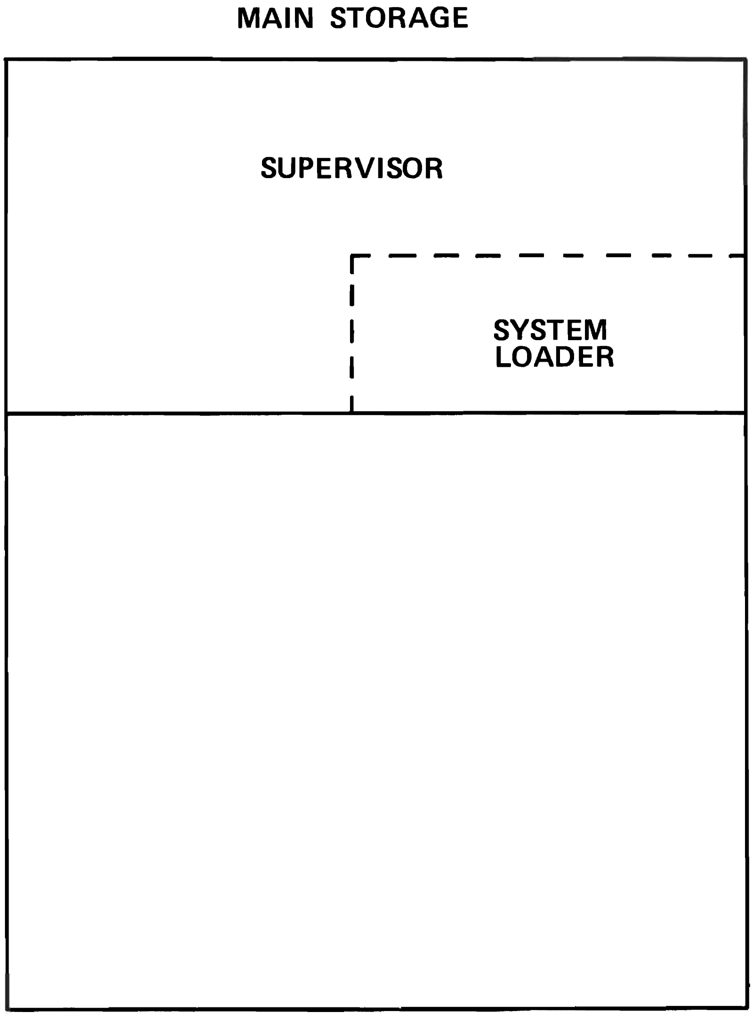
●●●

The operating system.

200 The operating system may be defined as an integrated set of programs designed to improve the total operating efficiency of a computer operation.

To demonstrate the functioning of an operating system, let's consider a typical operating situation. Assume you have written a program in COBOL. You want to compile and test it, and put it in the system residence device with other user-written programs.

Remembering what we said about control programs and the S/360, what is the first thing you



NOTE: SUPERVISOR OCCUPIES LOWER PORTION OF MAIN STORAGE

Figure 15.

must do?

●●●

Load the supervisor program into main storage.

201 You know that for a computer to do anything it must execute one or more instructions. But until the supervisor is loaded there are no instructions in storage to be executed. The question then arises: how can we cause the computer to seek out the supervisor in the system residence device and load it?

The answer lies in the first program shown in the core image library in Fig. 14. It is the..........

●●●

initial program loader.

202 The initial program loader, hereafter called the IPL program, is a very small program that can be loaded into main storage only through manual operations at the console. By setting certain controls, and then pressing the console load key, the S/360 will cause the IPL program to be transferred to predetermined main storage locations. Note that this is a machine function, not a programmed function. No instructions are executed during the process of loading the IPL.

Once the IPL program is loaded, control of the computer system passes to it; that is, the computer will start executing IPL instructions. The IPL program will then locate and load another program from the system residence device. Which program do you think will load?

●●●

the supervisor.

203 How is the IPL program loaded?

●●●

By manual operation at the console.

204 How is the supervisor loaded?

●●●

By the IPL program.

205 After the supervisor is loaded, control of the computer passes to it. This means that the computer will be executing (IPL/supervisor)..........instructions.

●●●

supervisor

206 Fig. 15 shows main storage with the supervisor loaded. The supervisor actually consists of many separate routines, each of which performs a specific function. One such routine is shown in Fig. 15. It is the

....

●●●

system loader.

207 The system loader, as part of the supervisor, is in storage all the time. It is the routine the supervisor uses to load all other programs in the operating system.

Match each program in the left-hand list below with the appropriate load program in the right hand list.

a.	COBOL compiler	1. IPL
b.	supervisor	2. system loader
c.	user-written payroll program	

●●●

a.	2
b.	1
c.	2

208 Immediately after the supervisor is loaded, it calls on the system loader (which as we noted is part of the supervisor) to load the job

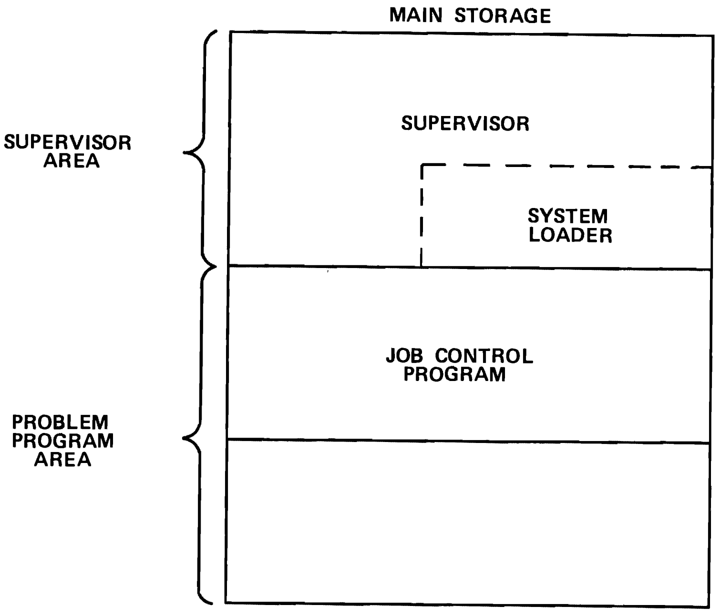
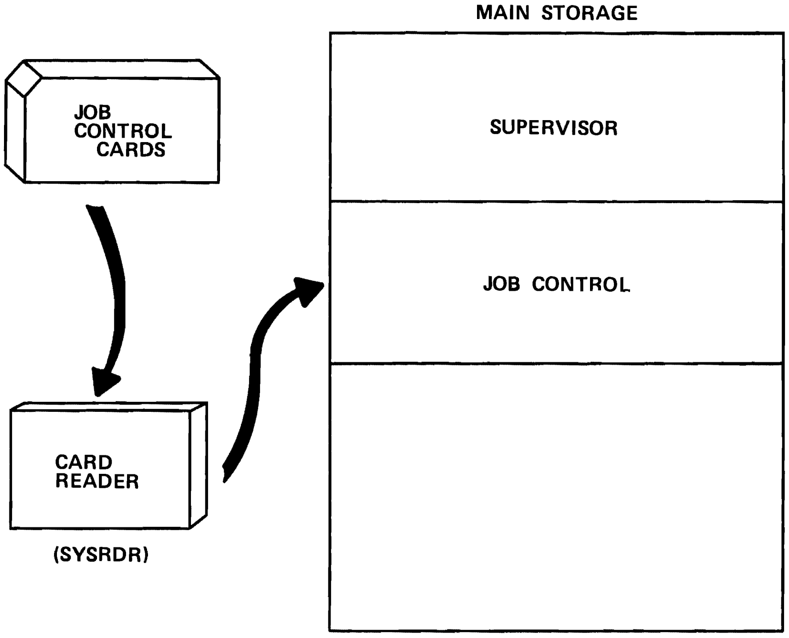


Figure 16.



A SET OF JOB CONTROL CARDS IS
REQUIRED FOR EACH JOB. THESE
CARDS IDENTIFY THE PROGRAM
TO BE LOADED AND PROVIDE OTHER
INFORMATION NEEDED TO PREPARE
FOR THE RUN.

Figure 17.

control program. Figure 16 shows main storage with the supervisor and the job control program both loaded. The job control program occupies part of the (problem program/supervisor) .... area of storage.

●●●

problem program

209 The routine that loads the job control program into main storage is the (IPL/system loader) ....

●●●

system loader. (The IPL program loads only the supervisor. All other programs will be loaded by the system loader.)

210 A job is the basic independent unit of work performed by the operating system. A job can be a compilation run, a problem program such as payroll or inventory file maintenance, a sort-merge program - in short, any complete operation involving input, processing, and output.

The job control program is the communication link between the operator and the other parts of the operating system, including the supervisor.

To call for a program to do a specific job the operator must communicate with the .... program.

●●●

job control

211 Assuming that we have a COBOL source program to compile, how do we communicate this information to the job control program?

Figure 17 provides the answer. It is ....

●●●

a set of job control cards that identify the program to be run and provide other information needed to prepare for the run.

212 Remember that the job control program, immediately after being loaded by the system loader, will assume control of the computer system. Which of the following is the logical thing for the job control program to do first?

a. Load the desired program from system residence
b. Read the job control cards
c. Turn control of the computer over to the supervisor

●●●

b.

213 The job control program can't do anything until it knows what you want it to do. Therefore it must first read the job control cards. Job control cards are read by a card reader that is called the system reader, abbreviated SYSRDR. Any card reader attached to the computer can be designated as SYSRDR.

What program causes SYSRDR to operate?

●●●

The job control program.

214 What information enters storage when SYSRDR operates?

●●●

The identification of the job to be run and other information necessary to prepare for the job.

215 Let's briefly review the sequence of operation so far.

a. The operator loads the IPL program by manually operating the system console.
b. The IPL program assumes control of the computer and loads the supervisor, which is the S/360 control program.
c. The supervisor assumes control of the computer and uses its system loader routine to load the job control program.

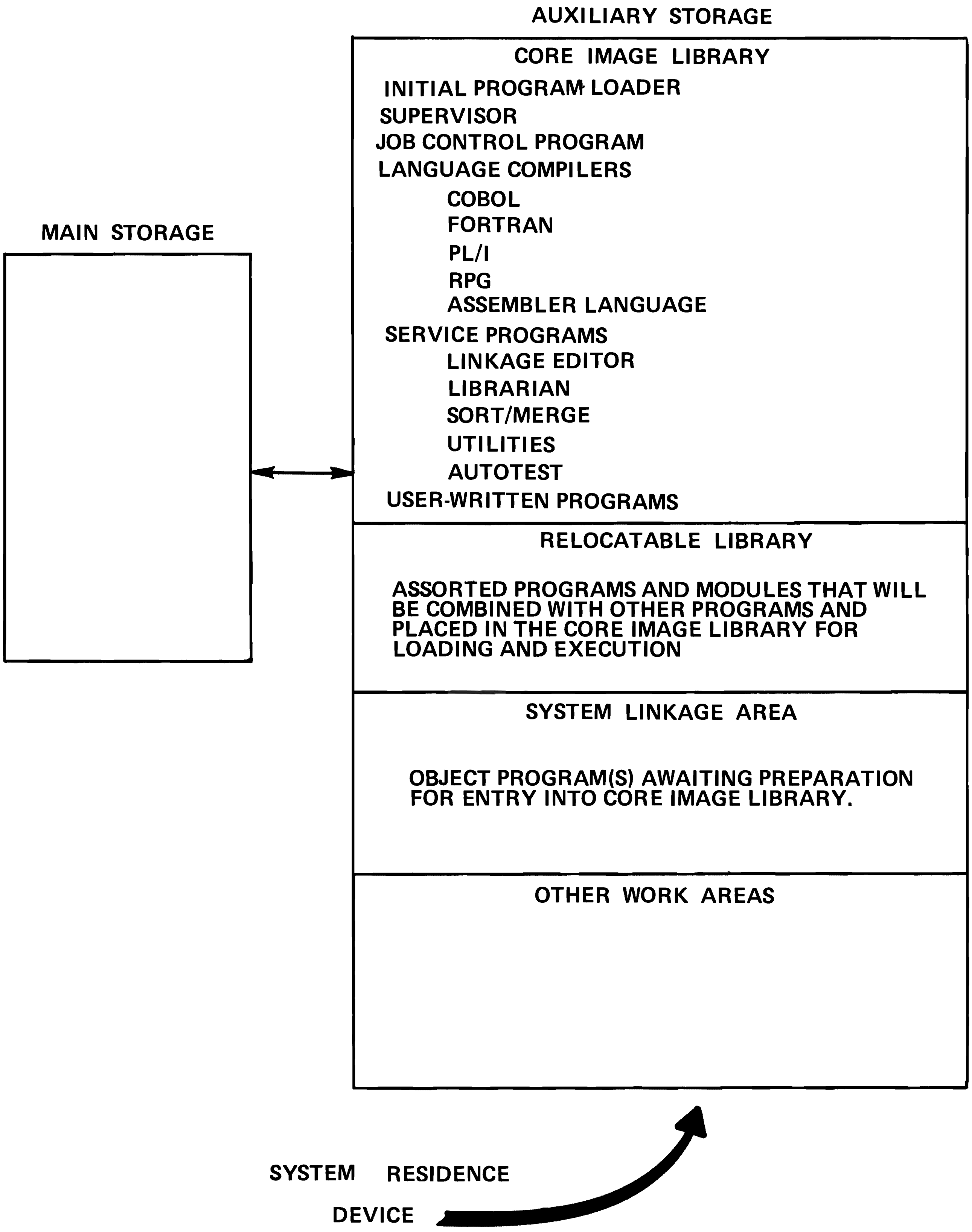

AUXILIARY STORAGE
CORE IMAGE LIBRARY
INITIAL PROGRAM LOADER
SUPERVISOR
JOB CONTROL PROGRAM
LANGUAGE COMPILERS
COBOL
FORTRAN
PL/I
RPG
ASSEMBLER LANGUAGE
SERVICE PROGRAMS
LINKAGE EDITOR
LIBRARIAN
SORT/MERGE
UTILITIES
AUTOTEST
USER-WRITTEN PROGRAMS
RELOCATABLE LIBRARY
ASSORTED PROGRAMS AND MODULES THAT WILL BE COMBINED WITH OTHER PROGRAMS AND PLACED IN THE CORE IMAGE LIBRARY FOR LOADING AND EXECUTION
SYSTEM LINKAGE AREA
OBJECT PROGRAM(S) AWAITING PREPARATION FOR ENTRY INTO CORE IMAGE LIBRARY.
OTHER WORK AREAS
MAIN STORAGE
SYSTEM RESIDENCE
DEVICE

Figure 18.

d. The job control program assumes control and reads the job control cards, which are in a card reader called SYSRDR.

●●●

216 A little later in the course we will see why we call I/O devices by symbolic names such as SYSRDR. We also will see what other kinds of information are contained in the job control cards. For the present we need only know that the information they contain is necessary to prepare the computer and the operating system for the job to be run.

●●●

217 The first job control card read will contain the name of the job program to be run. (In our example, the COBOL compiler.) The job control program simply stores this name in a location available to the supervisor, and continues reading and processing the remaining job control cards. After all remaining job control cards have been read and processed, the job control program passes control back to the supervisor.

Remembering the function of one of the supervisor routines (See Figure 15), what would you expect the supervisor to do first?

●●●

Call on the system loader to locate the desired program in the system residence device and load it into storage.

218 Earlier, we said that the job control program, while in storage, occupies part of the problem program area. Now that the job control program has completed all its tasks for this job, would you expect the incoming job program to:

a. enter that part of storage not occupied by supervisor or job control?
b. overlay the supervisor and job control programs?
c. overlay the supervisor only?
d. overlay job control only?

●●●

d. overlav job control

219 Here you see one of the advantages of an operating system. Both the user-written programs and the IBM-supplied programs in the operating system occupy the same areas in storage. As each program is loaded, it overlays the previous one. Thus the amount of main storage required by a computer using an operating system is (greater/ less) .... than that required if no operating system is used.

●●●

less

220 So now we have the job program loaded and control of the system passes to it. Assuming that the input units have been readied with the correct input data, and all output units are ready to receive data, the job is off and running.

We assumed the job to be a COBOL compile run. What job program will be in storage?

●●●

The COBOL compiler.

221 What will be the input for this job?

●●●

The source program written in COBOL.

222 The output from any compilation run is, of course, an object program in machine language. There are two options available for writing out the object program. It can be written directly on the system residence device, or it can be written on some other medium such as cards. There are advantages to each. For our example we will assume the object program is written directly on the system residence device.

Figure 18 shows the object program in the system residence device. In what part of the file was it written?

●●●

The system linkage area.

223 The system linkage area is essentially a work area in which object programs are prepared for entry into the core image library.

You will remember that one of the things that must be done to a program before it enters the core image library is ....

●●●

to assign the final main storage addresses to the elements of the program.

224 This is not to say that object programs in the system linkage area have no main storage addresses assigned. Normally, they are assigned according to an address in a base register, as specified by the programmer in the source program. It is possible, under certain circumstances for this address to be in conflict with the supervisor, which occupies the (lower/higher) .... addressses in main storage.

●●●

lower

225 Some indication must be made of the lowest main storage address the object program can use. This address will be (lower/higher) .... than the highest address used by the supervisor.

●●●

higher

226 There is a service program in the operating system that performs the address assignment function.

Immediately after the compilation run is completed, the COBOL compiler turns control over to the supervisor. As before, the supervisor calls on the system loader to load the job control program, and the job control program reads the next set of job control cards in SYSRDR.

One of the job control cards will specify the next job to be done. That job is to edit the object program in the system linkage area. The service program that does this can be identified by its name. It is (see Figure 18) the ....

●●●

linkage editor.

227 After the job control program has processed all the job control cards for the linkage edit job, it returns control to the .... which in turn calls on the .... to load the .... program.

●●●

supervisor
system loader
linkage editor

228 The linkage editor program checks with the supervisor to see how much storage the supervisor uses; then it tags the object program with a "start" address. This address will be the location of the first instruction in the object program.

An object program in the (core image library/system linkage area) .... is edited by the (linkage editor/supervisor) .... The operation involves tagging the object program with its starting address so it will occupy addresses (higher/lower).......... than those used by the supervisor.

●●●

system linkage area
linkage editor
higher

229 The linkage editor can perform other valuable services for the user. For example, you may want to include a common routine in several of your job programs. You can compile the common routine separately and use a librarian routine (see Figure 18) to catalog it into the relocatable library.

By appropriate coding in your source language programs, and use of certain job control cards, you can cause the linkage editor to add the common routine from the relocatable library to your object program in the system linkage area. Following this the entire program can be cataloged into the core image library.

Match the following:

A. Core image library
B. System linkage area
C. Relocatable library
D. Linkage editor

1. An area in the system residence device containing programs ready to be loaded.
2. A program that prepares object programs for entry into the core image library.
3. An area in system residence holding various routines and programs to be combined with other programs during link editing.
4. An area in which programs are edited for entry into the core image library.

●●●

A	1
B	4
C	3
D	2

230 Here is another service the linkage editor will perform. If you are coding a large program in Assembler Language, you may find it convenient to assign parts of the job to different programmers. These individual sections of the source program can be compiled (in the same run, or separately) and, if properly cross-referenced, can be link edited into one single program. The linkage editor will resolve all cross-references in each section, so that the result is a single unified program ready to be cataloged into the core image library.

Would a source program that was not written in sections require the services of the linkage editor? Why?

●●●

Yes. As previously pointed out, source programs can be compiled with addresses that conflict with the supervisor addresses. The linkage editor must indicate a start address that will not encroach on any of the area used by the supervisor.

231 Name three functions performed by the linkage editor to prepare programs for cataloging into the core image library.

●●●

Assigning final main storage addresses
Adding common routines from the relocatable library
Combining separately coded sections

232 We mentioned previously that a librarian routine would be used to catalog programs into the relocatable library. This also is true of the core image library. To enter a program into either of these libraries,it is necessary to call the appropriate librarian routine, via job control cards, and specify the cataloging job to be done.

Look at Figure 18. The librarian is a set of (compiler/service/user-written) .... programs.

●●●

service

233 The librarian is a program composed of special routines that maintain, service, and organize the operating system libraries. Librarian routines enable you to add, delete, and rename programs in the libraries. For example, if you wanted to print out a program from one of the libraries, you probably would use a:

a. catalog routine.
b. display routine.
c. condense routine.

●●●

b.

234 The display routine prints out programs in the libraries. The condense routine is used to reposition library programs to minimize vacancies between them.

You would expect to use the condense routine after using a (catalog/delete) .... routine.

●●●

delete. Deletions leave vacant areas in the library. Condensing repositions the remaining programs to eliminate the areas left vacant by the deletions.

235 For each library there is a list of the programs and their library addresses. Such a list is called a <u>directory</u>. All programs requiring access to other programs in either of the libraries must consult the appropriate directory to find the address and length of the desired program.

When a program is cataloged, what two things occur?

●●●

The program is written in the library.
Its name and address are placed in the directory for that library.

236 Match the following items:

a. Enter a program in a library and its name and address in the library directory.
b. Change the identification of a program in a library.
c. Print out a library program.
d. Remove a program from a library.
e. Eliminate vacancies between library programs.

1. Delete
2. Condense
3. Catalog
4. Display
5. Rename

●●●

a. - 3
b. - 5
c. - 4
d. - 1
e. - 2

237 Now that our program is compiled and placed in the core image library, it must be tested. We already have discussed the capabilities of the Autotest program. Some further notes on testing are in order.

For programs written in Assembler Language, the use of autotest can expedite the testing process. This is because, in a one-for-one language, the user can select the exact points in the object program at which a patch, a display, or a panel operation would be helpful. The ability to patch the object program eliminates the need for recompiling until all errors have been located.

In higher level languages, of course, there is no clear relationship between source statements and machine language. A great deal of analysis is required of the object program to determine exactly where a display, a panel, or a patch would be of benefit. Because of this, high-level language programs usually are tested by working with the source statements. After a test run, if the results obtained are incorrect, corrections are made to the source program and it is again compiled, link edited, placed into core image library, and retested. This process continues until the object program gives correct results.

Match the following:

a. FORTRAN
b. COBOL
c. Assembler Language
d. PL/I

1. Use autotest and recompile only when all bugs have been removed from the object program.
2. Change source statements after each test and recompile the program.

●●●

a. - 2
b. - 2
c. - 1
d. - 2

238 You will recall that after compilation of our COBOL source program, control returned to the supervisor, which called the job control program. Job control then read the next set of job cards, calling for the linkage editor. Job control then returned control to the supervisor, which loaded the linkage editor. After link editing the COBOL object program, to which program would the linkage editor return control? Why?

●●●

The supervisor, because it must again call in job control to read the next set of job cards, specifying the next job to be done.

239 The next job was a catalog operation, involving a catalog routine from the librarian. This operation placed the object program in the core image library. When finished, the catalog routine would return control to the ...., which would ....

●●●

supervisor
call the job control program again.

240 How would you inform the operating system that you wanted to execute a particular program, say autotest, in system residence?

●●●

Punch in the necessary job control cards and place them in the card reader (SYSRDR).

241 How would you inform the operating system that you wanted to execute a payroll program that you had compiled, link edited and placed in the core image library?

●●●

Punch the necessary job control cards and place them in SYSRDR.

242 You can see that the operating system always returns to SYSRDR (via the job control program) to find out what it is to do next. Therefore, so long as there are sets of job control cards in SYSRDR, the computer will move smoothly from one job to the next without interruption. This is called a "stacked job" operation. The only requirement is that the operator must keep the I/O units ready for each successive run.

Which is more efficient?

a. Wait until the job control program is ready to read and then put your next set of job cards in SYSRDR.
b. Always keep at least one set of job cards in SYSRDR.

●●●

b. This will avoid delays in reading job control cards, and will speed up transition from one job to another.

243 Several times in the preceding frame we have made reference to "other" job control cards. The first job card in a set, of course, tells which operating system program is to be executed next. Let's look briefly at some other types of job control cards, specifically the assign card.

●●●

244 We have been referring to the card reader that reads job control cards as SYSRDR. This is in keeping with a basic approach to I/O unit designation that is required of all S/360 programmers. It is this: all source program references to I/O units are made by symbolic names.

In an Assembler Language program would you identify your printer by the actual machine address at which it is connected, say 00E, or as SYSLST? Why?

●●●

SYSLST, because it is a symbolic name.

245 Even after the programs are compiled, they still include symbolic names of the I/O devices they will use. Compilation does not substitute the actual machine addresses of the I/O units.

But, of course, the supervisor and the channel must know the actual addresses, in hexadecimal, of the devices. That is, the commands that activate the devices must use machine addresses.

Assume a S/360 has these devices, with the respective symbolic names.

SYSRDR - a card reader connected at machine address	00C
SYSPCH - a card punch connected at machine address	00D
SYSLST - a printer connected at machine address	00E
SYSLOG - a console printer-keyboard connected at machine address	01F
SYSRES - a disk file containing the operating system connected at machine address	190
SYSIPT - a card reader connected at machine address	00C
SYS000 - a tape unit connected at machine address	180
SYS001 - a tape unit connected at machine address	181
SYS002 - a tape unit connected at machine address	182
SYS003 - a tape unit connected at machine address	183

To read in the job control program from the system residence device the channel would use which of these?

a. SYSRES
b. 00C
c. 180
d. SYSRDR

●●●

None. It would use the machine address of the system residence device, in this case, 190.

246 The question is then, how does the supervisor associate the symbolic names in the problem programs with actual machine addresses of I/O units?

The answer is the Physical Unit Block (abbreviated PUB) table. The PUB table is part of the supervisor. It is a list of the symbolic names of the I/O devices on the system, and the machine addresses of these units. Each unit is associated in the table with a specific symbolic name.

In the PUB table below, the symbolic name SYSIPT is assigned to the device at machine address ....

SYSRES	190
SYSIPT	00C
SYSPCH	00D
SYSRDR	00C
SYSLST	00E
SYSLOG	01F
SYS000	180
SYS001	181

●●●

00C

247 You may have noticed that two symbolic names (SYSIPT and SYSRDR) are assigned to the same device in the table. This is quite legitimate. A given device may be referred to by more than one symbolic name. So long as the name is matched with the device address in the PUB table, no error will occur.

When an object program being executed notifies the supervisor that it wants to read or write data, it will provide the supervisor with the symbolic name of the I/O device involved. Where can the supervisor find the machine address of that I/O device?

●●●

In the PUB table.

248 The supervisor simply locates the symbolic name in the PUB table, obtains the machine address associated with the symbolic name, and sends that machine address to the channel via the Start I/O instruction.

The channel requires the (symbolic/machine) .... address of the I/O device in order to initiate the operation.

●●●

machine

249 Suppose you have three card readers connected to the multiplexor channel of your system. They have symbolic names SYSRDR, SYSIPT, and SYS006. Machine addresses are, respectively, 00A, 00B, and 00C. You have a program that uses SYS006. Just before starting your run you find that a card reader at machine address 00C is inoperative. Would you

a. recode your program, using one of the other symbolic names, then recompile and make the run using the alternate card reader?
b. change the PUB table so SYS006 is assigned to one of the other readers; then make your run with the data cards in that reader?
c. physically disconnect the inoperative reader and connect a working one in its place?

●●●

You could do any of the three, but only (b) could be done efficiently and quickly.

250 By the simple expedient of punching a job control card called an assign card, you can associate any symbolic name with any machine address, in the PUB table. Your object program still refers to the unit by the same symbolic name. But when the supervisor locates that name in the PUB table it will find the address of one of the alternate card readers. (The address you put in the PUB table via the assign card.)

Which unit will the channel cause to operate?

a. The I/O unit originally assigned to SYS006.
b. The I/O unit currently assigned to SYS006.

●●●

b.

251 Assign cards are job control cards. They will be processed by the .... program.

●●●

job control

252 The assign card contains a symbolic name used by one or more of your object programs, and the machine address you want assigned to that name.

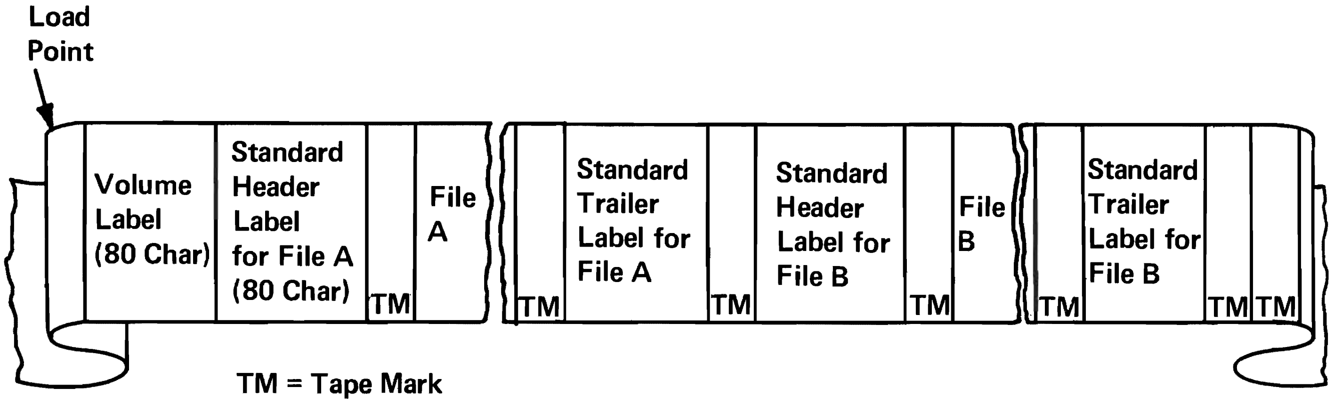
The job control program will enter this information where?

●●●

In the PUB (physical unit block) table.

253 The ability to change the assignments of I/O devices to specific symbolic names affords great flexibility in computer operations, and improves operating efficiency. For example, assume you want to run a program requiring two card readers on a system having only one card reader. By making a preliminary run to transfer one file of card data to magnetic tape, you could run the job. What changes would you have to make in the PUB table?

●●●


Load Point
Volume Label (80 Char)
Standard Header Label for File A (80 Char)
TM
File A
TM
Standard Trailer Label for File A
TM
Standard Header Label for File B
TM
File B
TM
Standard Trailer Label for File B
TM
TM
TM = Tape Mark

Tape File with Standard Labels

Figure 19.

Change the assignment of one symbolic name used by your program from a card reader to a tape unit.

254 Describe how you would designate an alternate device for input/output.

●●●

Punch an assign card with the appropriate symbolic name and the machine address of the device to be associated with that name. Include the assign card with the other job cards for the run.

255 Where is the list of symbolic device names and associated device addresses maintained, and what is it called?

●●●

In storage, as part of the supervisor. It is called the PUB (for physical unit block) table.

256 You have learned about two kinds of job control cards: the job card itself that specifies the program to be executed, and the assign card that associates symbolic names of I/O devices with actual machine addresses of the devices.

Another type of job control card is the <u>label</u> card. Figure 19 shows a file of data on magnetic tape. The first and second records in the file are .... records.

●●●

label

257 Label records are recorded at the beginning and end of each file of data on magnetic tape and disk. They contain information about the file. Typical label information is the file name, serial number, date of origin, date of expiration, and number of blocks of data in the file. You would expect the label records in the beginning of the file to be processed (after/before) .... the data records in the file.

●●●

before

258 The basic purpose of label records is to identify the file. Thus it is necessary to check them to assure that the file being processed is the correct one, and this checking must be done before the data records in the file are processed.

The supervisor will call special routines to process the label records. However, these routines must be told what the contents of each label should be.

Can you suggest a way to provide the label processing routines with this information?

●●●

Punch a job control card with the necessary information and read it in with the other job control cards for the run.

259 The job control program will read the label cards, and store the information they contain for later use by the label processing routines. These routines will check the information in the actual file labels with that supplied by the label cards. If there is a discrepancy, would you expect the operating system to:

a. continue processing the data in the file?
b. print a message to the operator and enter the wait state?

●●●

b.

260 Name three kinds of job control cards and state the purpose of each.

●●●

Job card. Specifies the next program to be executed.
Assign card. Associates symbolic names used in object programs with actual I/O device addresses on the machine.
Label card. Contains information against which

the contents of file labels will be checked.

261 Match the functions in the first list below with the operating system control and service programs in the second column.

a. Assigns I/O devices for each problem program and requests the supervisor to initiate problem program execution. Acts on information supplied by job cards in the input job stream.
b. Remains in storage while problem programs are being executed. Loads problem programs and other operating system programs.
c. Is loaded as result of manual action by the operator. Loads the supervisor.
d. Transforms all object programs into loadable form, assigning storage addresses and combining separately coded sections of the program into a unified whole.
e. A group of programs used for maintaining, servicing, and organizing the system libraries.

1. Linkage Editor
2. Initial Program Loader
3. Supervisor
4. Librarian
5. Job Control Program

●●●

a. 5
b. 3
c. 2
d. 1
e. 4

262 The steps below reflect the operations that occur when a computer program is written, compiled, and placed in the operating system library. Sequence these steps by numbering them in the order in which they occur.

a. Job control program reads job cards, makes I/O device assignments, tells supervisor which compiler program to call.
b. Console operator manually loads IPL program.
c. Source program statements are punched into cards.
d. Compiler translates source program into object program and puts it (object program) out. Signals supervisor when finished.
e. Programmer codes source program.
f. Supervisor calls job control, which reads job cards stating that linkage editor is next program to be executed. Supervisor calls linkage editor.
g. IPL loads supervisor.
h. Linkage editor program edits object program, assigning storage addresses and linking all individual sections together.
i. Supervisor calls compiler program.
j. Supervisor uses its system loader to load job control.
k. Supervisor, job control, etc., repeat their operations to call librarian. Librarian catalogs object program into library of programs ready for execution (the core image library).

●●●

1. e
2. c
3. b
4. g
5. j
6. a
7. i
8. d
9. f
10. h
11. k

263 Which of the following most accurately describes an operating system?

a. A program that translates source language programs into object programs.
b. A control program that remains in storage at all times.
c. An integrated set of programs designed to improve the total operating efficiency of a computer operation.

S/360 OPERATING SYSTEMS SUMMARY

	BPS	BOS	TOS	DOS	OS
Minimum Storage Requirements	8K	8K	16K	16K	64K
System Residence Device	Cards or Tape	Disk	Tape	Disk	Disk
Languages Supported	Assembler Language RPG FORTRAN	Assembler Language RPG	Assembler Language RPG COBOL FORTRAN PL/I	Assembler Language RPG COBOL FORTRAN PL/I	Assembler Language RPG COBOL FORTRAN PL/I
I/O Devices Supported by All Systems	Card Reader, Card Punch, Card Reader - Punch, Printer, Magnetic Tape, Printer - Keyboard Paper Tape Reader, Optical Character Reader				
Additional I/O Device Support by System		Magnetic Disk		Magnetic Disk Data Cell Remote Processing I/O Devices Data Collection System Data Communication System Display Station	Magnetic Disk Data Cell Remote Processing I/O Devices Data Collection System Data Communication System Display Station Magnetic Drum
Other Capabilities	Limited Remote Processing	Limited Remote Processing	Storage Protection Multiprogramming Interval Timer Tape Switching Universal Character Set	Remote Processing Multiprogramming -- Storage Protection Interval Timer Device Independence -- Universal Character Set	Remote Processing Multiprogramming Multiprocessing Storage Protection Interval Timer Device Independence Dynamic Storage Allocation Universal Character Set

Figure 20.

d. A program called into operation by the control program, to search out and load the next problem program.

●●●

c.

264 There are several operating systems available to S/360 users. All employ the operational principles we have discussed. S/360 operating systems offer a wide range of capabilities, since each is tailored to the specific needs of a particular group of users, as reflected in the system configurations and languages they select.

Figure 20 lists these operating systems, the languages and I/O devices they support, and the other capabilities of each. The system offering the maximum in I/O device support and other capabilities is ....

●●●

OS (Operating System).

265 S/360 Operating System (OS) is the most powerful system available. It requires (more/less) .... storage than the less powerful systems.

●●●

more

266 In general, the more powerful the operating system, the more main storage required for it.

A user whose S/360 has no disk units and small main storage, and whose language requirements can be met by Assembler Language and/or RPG would probably use ....

●●●

BPS (Basic Programming Support).

267 A user having a small S/360 with disk units, and whose language requirements can be met by Assembler Language and/or RPG would probably use ....

●●●

BOS (Basic Operating System).

268 A system without disk files, but having need for support of all languages plus multiprogramming, storage protection, and the interval timer would select ....

●●●

TOS (Tape Operating System).

q**269** Both DOS (Disk Operating System) and OS use magnetic disks for system residence. Give two major differences between these systems.

●●●

OS requires more minimum storage than does DOS.
OS supports more I/O devices than DOS.
OS offers additional capabilities, such as multiprocessing.

270 Match each operating system with its description.

a. A disk resident system offering the maximum in operating system capacity.
b. A disk resident system supporting Assembler Language and RPG, with sufficient capacity to satisfy requirements of small installations.
c. A disk resident system for users requiring more advanced operational support than that offered by (b) above. Supports all languages as well as remote processing, storage protection, multiprogramming, and the interval timer.
d. A minimum system resident in cards or tape, for use in installations having no disk files. Supports Assembler Language and RPG.
e. A tape resident system offering the

capabilities of (c) above, except for remote processing, for installations having no disk files.

1. BPS
2. BOS
3. TOS
4. DOS
5. OS

●●●

a. 5
b. 2
c. 4
d. 1
e. 3

271 One other type of program should be mentioned before we go on to the next topic. In a previous frame we mentioned the possibility of transferring data from cards to magnetic tape, without doing any processing. You will find that in most computer installations there is a substantial need for this kind of data transfer, without special processing.

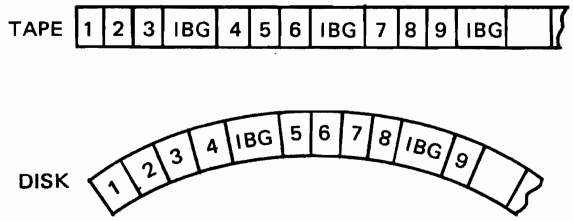
Which of the following involve a data transfer from one medium to another, without special processing?

a. Source program compilation.
b. Printing data written on magnetic tape during a previous run.
c. Computing a payroll and writing out a payroll register.
d. Writing data from a disk file onto magnetic tape.

●●●

b, d

272 IBM provides as part of each operating system, a set of utility programs whose basic purpose is to transfer data from one medium to another. In most cases, these programs permit a certain amount of manipulation of the input data during the transfer. The following shows the kind of manipulation that is possible. What has been done to these records as they were transferred from a tape file to a disk file?



●●●

The blocking factor has been changed from 3 (input) to 4 (output).

273 These records were transferred from one medium to another. What was done to them during the transfer?

INPUT

Emp. Name	Emp. No.	Dept. No.	Shift No.	Hrs. Wkd.	Hourly Rate	Tax Class

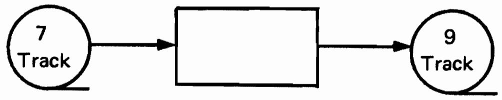
OUTPUT

Emp. No.	Emp. Name	Tax Class	Hourly Rate	Dept. No.	Hours Worked

●●●

The data items were rearranged and one was dropped from the record.

274 This represents a utility run on a S/360. What is being done to the data?



●●●

The coding of the data is being changed from 7-track to 9-track.

275 For what purpose are utility programs used, and what data operations do they permit?

●●●

To transfer data from one medium to another. They permit the deletion of specified items from input records, changing data coding, rearranging of data items selected for output, and reblocking of output records.

You have completed this section. At this point you should fill in your notes and take the self-evaluation quiz.

Self Evaluation Quiz
Programming Systems

QUESTIONS

1. Name five programming languages available to users of System/360.

2. Describe a checkpoint operation and tell why checkpoints are taken.

3. What is the auxiliary storage device, on which an operating system is stored, called?

4. How is the Initial Program Loader (IPL) program loaded into storage?

5. How is the Supervisor loaded into main storage?

6. What is the function of the Job Control program?

7. From what source are programs loaded into main storage to be executed?

8. What is the function of the Linkage Editor program?

9. What is the principal function of a utility program?

10. Name three capabilities of automatic testing programs.

Self-Evaluation Ouiz
Programming Systems

ANSWERS

Frame Reference

1. Report Program Generator (RPG), COBOL, FORTRAN, PL/I, Assembler Language. (99,125,138, 153,163)

2. Checkpoints are taken to enable the operator to restart the run without going back to the beginning, in case the run is interrupted for any reason. At appropriate points in his program the programmer writes instructions for checkpoints to be taken. At these points the Supervisor takes over, and calls in the checkpoint routine. The checkpoint routine writes on a designated output device all information that would be necessary to restart the system, including storage and register contents, positions of all I-O devices, and status of all machine indicators. If the run is prematurely halted, a restart routine, called for by the operator, is entered from the system residence device, and positions the system as it was when the last checkpoint was taken. The run resumes at that point. (97)

3. The auxiliary storage device holding the operating system is called the system residence device. (188)

4. The IPL program is loaded by manual operation at the console. (202)

5. The supervisor is loaded into main storage by the IPL program. (204)

6. The job control program reads cards, called job cards, prepared by the operator, from a card reader designated as SYSRDR. Job cards contain the information concerning each successive job to be run and the parameters applying to it. The job control program acts on this information in preparing for execution of the job. (213,261)

7. Programs to be executed are loaded from the core image library of the system residence device. (190)

8. The linkage editor prepares programs for entry into the core image library by: (231,261)

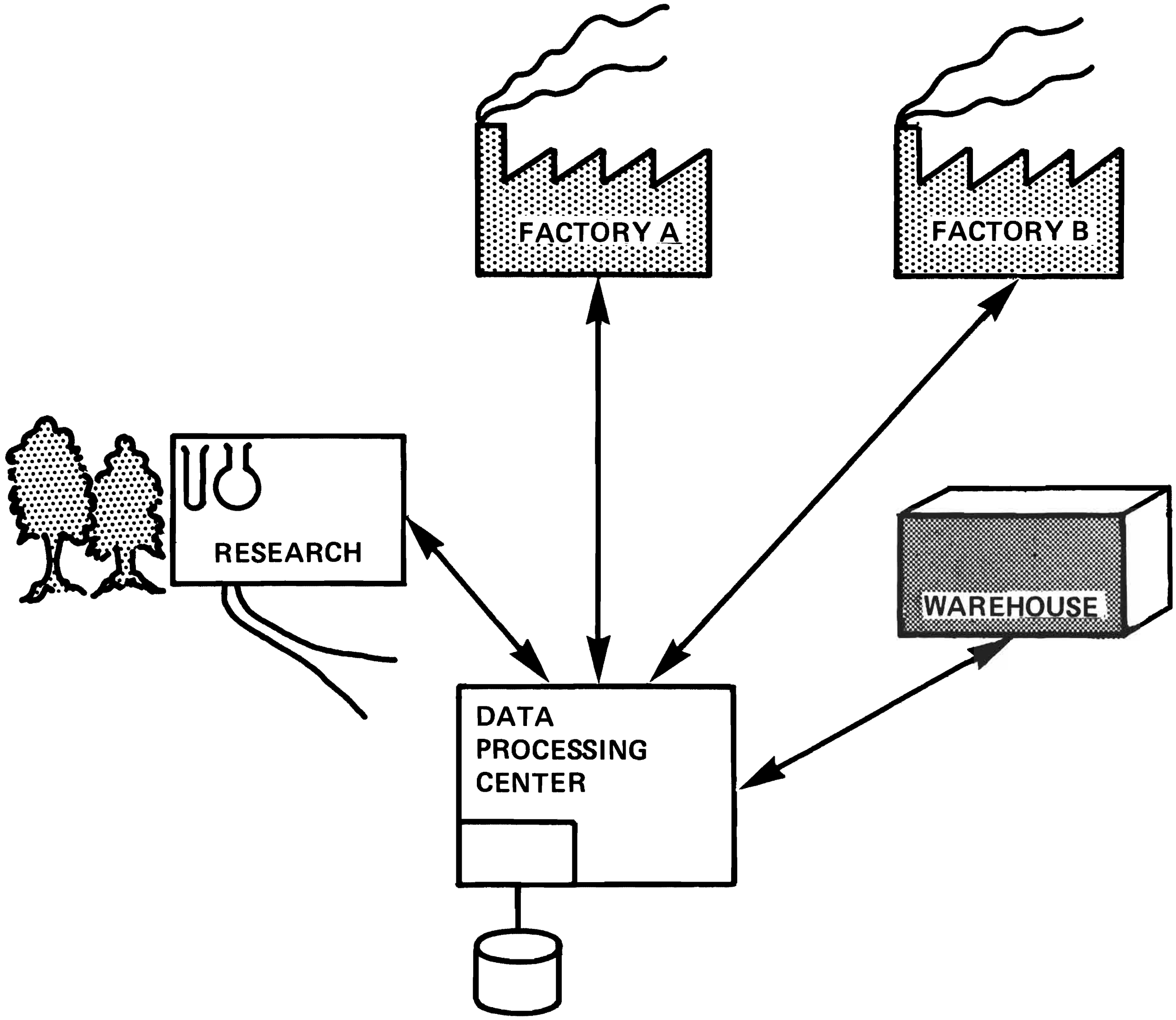
   a. assigning final main storage addresses to data and instructions,
   b. incorporating subroutines from the relocatable library, as requested by the programmer, and
   c. combining separately written sections of the program into a single, unified program.

9. The principle function of a utility program is to transfer data from one medium to another. (272)

10. Automatic testing programs speed and simplify testing by: (64)

    a. displaying data in registers and storage locations at specified points in the program being tested.
    b. patching the program under test by adding, deleting, or substituting instructions, and
    c. providing normal and abnormal end-of-job dumps.

Teleprocessing



INFORMATION NETWORK

Figure 1.

TELEPROCESSING

1 Teleprocessing (TP) is a word coined from a combination of the words telecommunications and data processing. An IBM TP system uses transmission facilities to permit the processing of data at a point remote from the origin of the data.

This characteristic is particularly useful in applications where the coordination of several functions, all remote from each other, is under the control of a data processing center. An example of this is shown in Figure 1.

Here the data processing function is coordinating the activities of two factories, so that, for example, the number of headlamps produced in A is related appropriately to the number of car bodies in factory B. It controls stock levels in the warehouse so that these levels are replenished from the factory when required. The results of research will be applied in the factories. Data from new factory processes need to be communicated to research.

Some of this data can travel by conventional means; other data must be communicated immediately.

Hence, teleprocessing - the combination of telecommunications and data processing.

● ● ●

2 Where this communication must be immediate and where data communication by conventional means such as the mails, by courier, etc., is too slow, teleprocessing provides a significant advantage through remote data processing. System performance can be improved by increasing the speed of data movement. Data may be captured and introduced to the system at its source. This results in improved timeliness of reports.

Teleprocessing thus involves entering data into a computer or receiving data from a computer. Direct communication between computers is a special case of teleprocessing and is called multiprocessing.

A card read at one remote terminal.......... (may/may not) be directly printed out at another terminal.

● ● ●

may not

3 Other advantages result from being able to capture data at its source and enter it into the system through a TP network. These advantages relate to time, effort, and expense.

1. Because data can be entered into the system at its source and in relatively raw condition, the time required to reflect the effect of that data on the system is reduced.
2. The effort to enter the data into the system is reduced. Simplicity of preparation of data for entry into the system is usually designed into the TP operation.
3. The expense is reduced, again by accepting entry of relatively raw data, the expense of converting and communicating the data by other methods are significantly reduced.

List three advantages related to capturing and entering source data in a TP network.

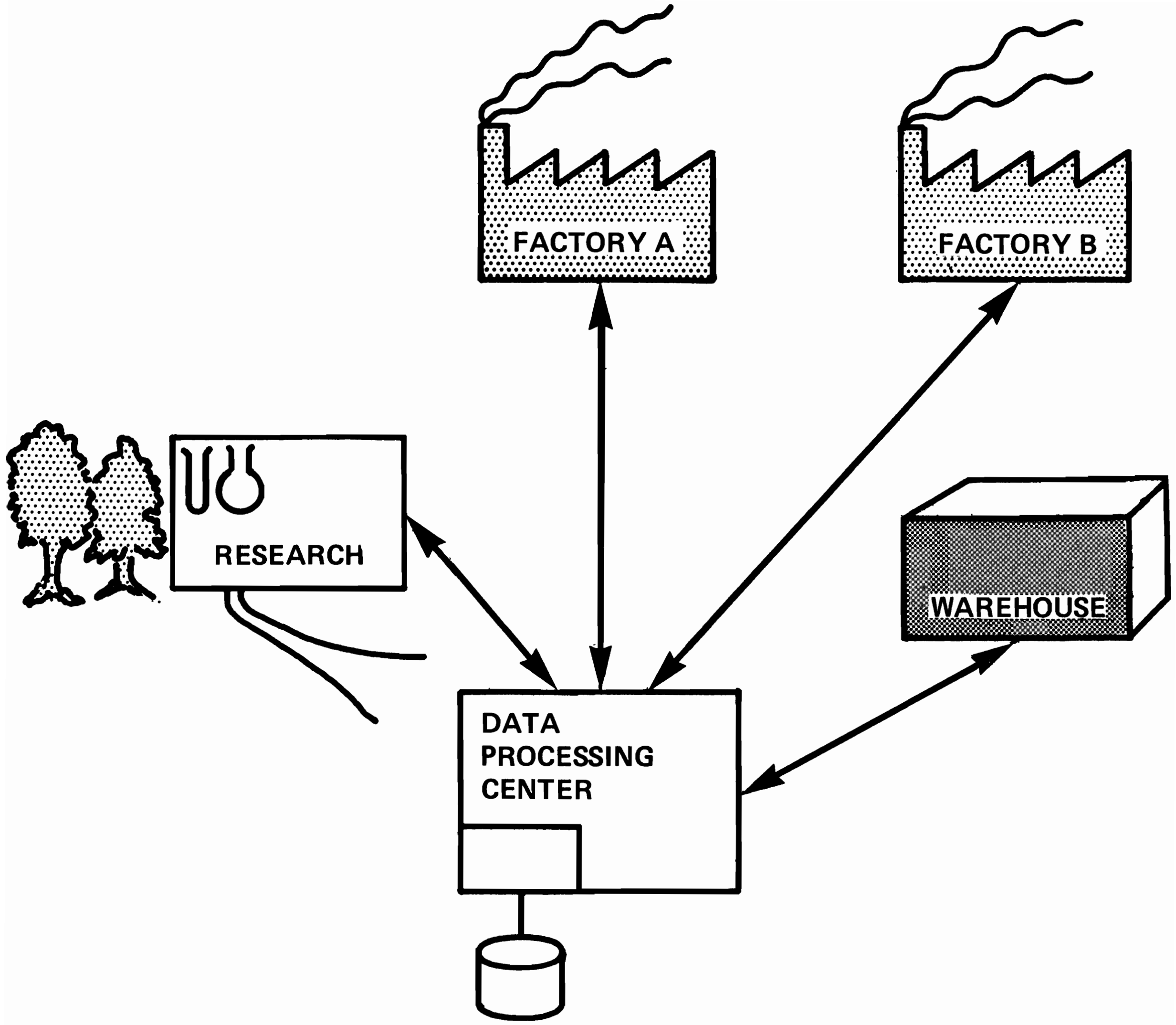
a. ....
b. ....
c. ....

● ● ●

a. time
b. effort
c. expense

4 The three major types of teleprocessing are:

1. Data Transmission
2. Data Collection
3. Data Communication



INFORMATION NETWORK

Figure 2.

DATA TRANSMISSION is characterized by high speed and high volume. Magnetic tape-to-computer communications are examples of data transmission.

In our example in Figure 2, machine scheduling data might be transmitted from the data processing center to the factory by data transmission. New circuit diagrams resulting from a design modification to some electrical apparatus could be sent from the computing center to the engineers in the research department, by data..........

●●●

transmission.

5 DATA COLLECTION implies several scattered points feeding information to one central point. It is associated with data being available in small volumes at each terminal. If at several points in a production process there are monitoring devices - temperature gauges, switches, etc. - these can be inspected at discrete intervals by the program.

Because the amount of data from any one terminal may be small, you would expect data collection speeds to be..........

●●●

low.

6 In the example quoted, data is continuously available at the monitoring point; the program inspects each point at intervals. However, this is not a usual characteristic of data collection.

A common example of data collection uses a Badge Reader as an input terminal. Here a factory worker may signal the beginning or end of a task by setting up data such as task number, start or stop, etc., through the use of prepunched cards. The computer is then signaled and the data entered into tne systeın when the worker inserts his coded badge into the Badge Reader. The system stores the data entered for immediate or subsequent processing in a Cost Accounting program or similar application. Several of these stations are normally used in this type of system.

What are three characteristics of data collection?

●●●

Several input points
limited amounts of data
slow-speed devices

7 Finally, let's consider DATA COMMUNICATIONS. This may have elements of data transmission and data collection. As well as data being entered from remote devices, inquiries may be made at the terminal of some centrally held data set. Responses to the inquiries will be sent over the network. The characteristics of the system may change during the course of the day. For example, in a banking system, exception reports highlighting accounts which appear to be overdrawn may be transmitted in the middle of the day so that individual branch managers can take appropriate action that day. The volume of data being transmitted at this time makes the operation more akin to data transmission; yet at other times, details of over-the-counter transactions are entered at each terminal and the system is predominantly a data collection system.

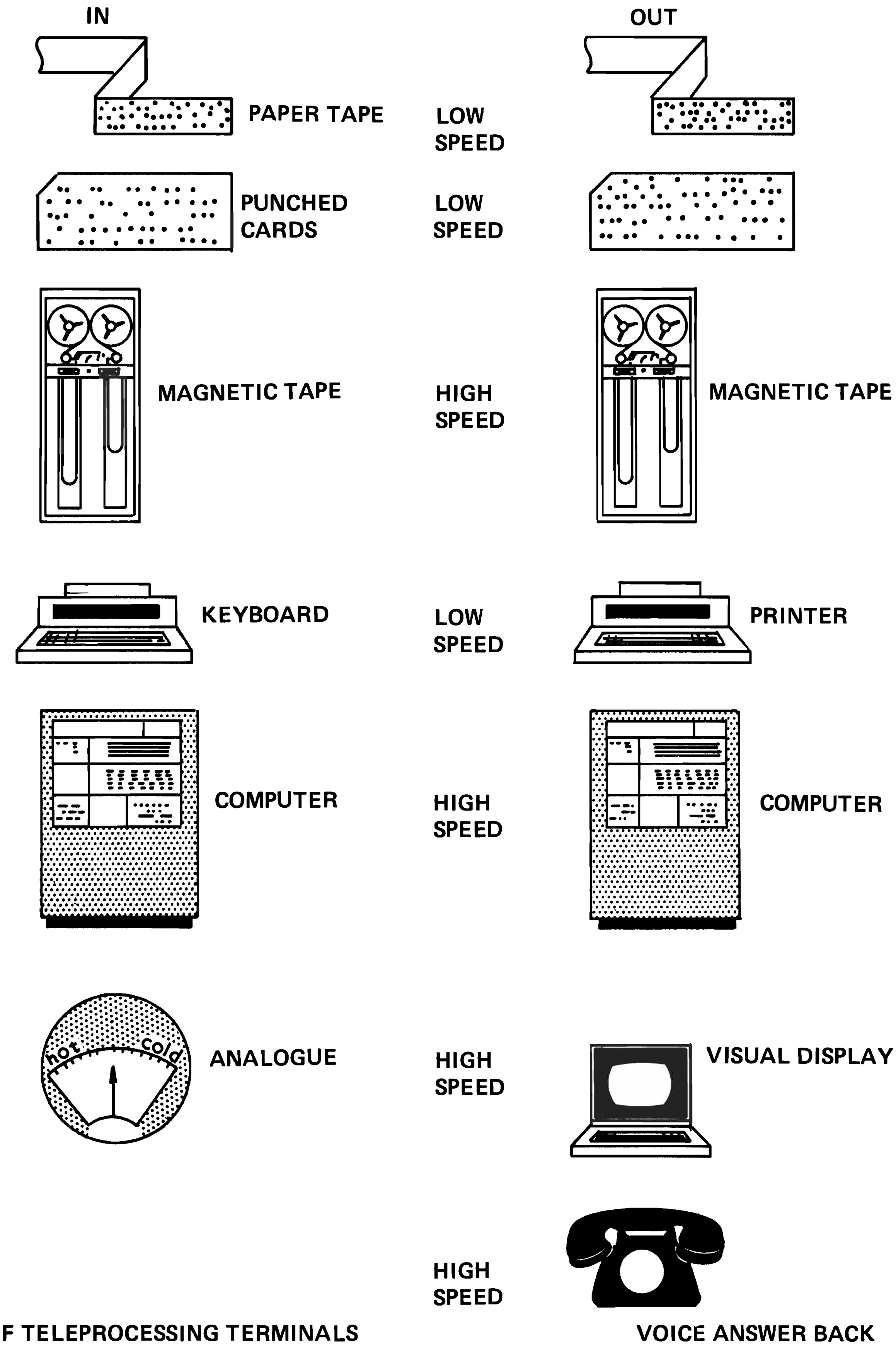
In addition, audio-response facilities may be provided in which the customer of the bank could dial from his home phone, a code number followed by his account number. The system audio-response unit would "speak" into the phone line and the customer would hear the amount of his balance.

A data communications system normally involves two-way communication and several terminals. It will probably embrace the characteristics of data collection and data transmission.

Teleprocessing is a system which permits remote processing of data by using..........facilities.

●●●

transmission



TYPES OF TELEPROCESSING TERMINALS

Figure 3.

8 List the three major types of teleprocessing.

●●●

1. Data Transmission
2. Data Collection
3. Data Communication

9 How does data communication differ from the other two types of TP?

●●●

Two-way communication is involved.

10 Every TP system has three major components. These are:

1. terminals
2. lines
3. controls

In TP, an I/O device at the end of a transmission facility is called a terminal.

Examples of teleprocessing terminals are shown in Figure 3. Relative speed of data transfer is also given for each of the devices. In addition to I/O devices, a computer may serve as terminal in a TP system.

In all cases, the terminals serve as input to or output from a..........

●●●

computer.

11 Which of the following are examples of TP terminals?

a. Tape drive
b. Visual display
c. Railroad station
d. Card punch
e. Soldering lug

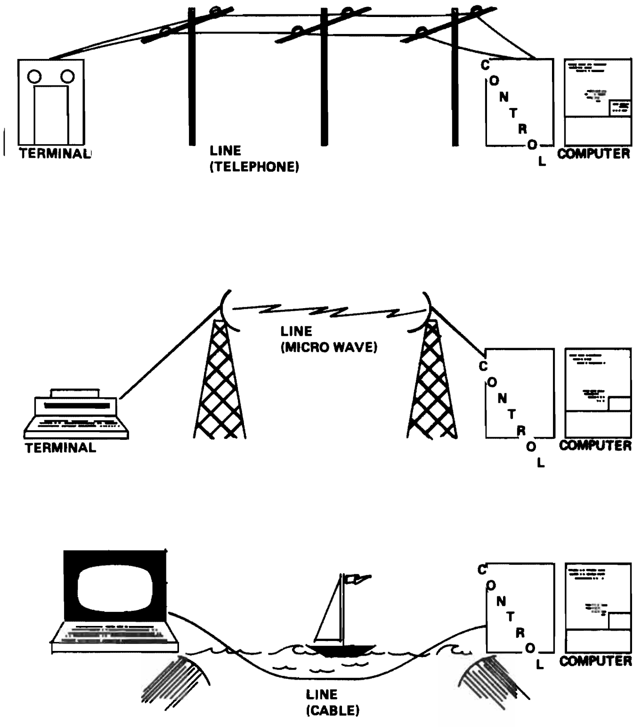
●●●

a.
b.
d.

12 The data to and from terminals are carried by "lines". A line is the transmission medium for carrying data signals. Lines may be such things as telephone lines, cables, micro wave links, Telstar and similar media.

The third major element of a TP system is called a "Control". A control refers to the interconnection between the computer and the line. The interconnection may be made directly into a channel but usually requires a control unit.

The figure below gives some examples of the three elements of a TP system.



EXAMPLES OF TP SYSTEMS SHOWING VARIOUS TYPES OF TERMINALS, LINES AND CONTROLS.

The three major elements of a TP system are a.........., ..., and a..........

●●●

terminal
line
control

Self-Evaluation Quiz
Teleprocessing

QUESTIONS

1. Which of the following is not true about TP?

   a. TP uses transmission facilities.
   b. TP permits processing data remotely.
   c. TP is a word derived from telecommunications and processing.
   d. TP involves direct communication between terminals.
   e. TP is useful in the coordination of remote functions under the control of a data processing center.

Select the description which best characterizes each of the following:

2. Data Communication

3. Data Transmission

4. Data Collection

   a. Many slow speed terminals.
   b. Remote data processing.
   c. High speed high volume.
   d. Analog input devices.
   e. Inquiry and answer back.

5. Which of the following is not always required in a TP system?

   a. A line.
   b. A terminal.
   c. A control
   d. A computer

Complete the following statements.

6. TP presents an advantage in entering data into a processing system by reducing the following:

   a. ____________
   b. ____________
   c. ____________

7. System performance is improved by increasing ______________________ ____.

8. Select the most correct answer. Terminals are:

   a. Keyboard-printers.
   b. Badge readers.
   c. Audio response units.
   d. All of the above.
   e. None of the above.

9. Which of the following is not used as a TP line?

   a. A telephone cable.
   b. The shortest distance between two points.
   c. A microwave relay link.
   d. A telephone network.

10. Name the three elements of a TP system.

    a. ________________
    b. ________________
    c. ________________

Self-Evaluation Quiz
Teleprocessing

ANSWERS — Frame Reference

1. d. Teleprocessing does not permit direct communication between terminals. (2)

2. b. Remote data processing. (7)
   e. Inquiry and answerback.

3. b. Remote data processing. (4)
   c. High speed, high volume.

4. a. Many slow speed terminals. (5)
   b. Remote data processing.
   d. Analog input devices.

5. c. A control (12)

6. a. Time
   b. Cost
   c. Effort (3)

7. Increasing the speed of data movement (2)

8. d. All of the above. (10)

9. b. A line is an electrical connection, not a theoretical concept. (12)

10. a. line (12)
    b. terminal
    c. control

Multiprogramming and Multiprocessing

MULTIPROGRAMMING

1 The use of an operating system to increase the production time of a computer has been covered in previous portions of this course. Other techniques have been developed to further increase the usage of the processing time available on the computer. One of these is "Multiprogramming".

Multiprogramming is defined as the concurrent execution of two or more programs simultaneously residing in the core storage of a computer.

Execution is defined as concurrent (or parallel) because of two significant characteristics of the computer.

First, the CPU executes only one instruction at a time regardless of the number of programs residing in main storage. Simultaneous execution of two programs cannot occur with one CPU.

Second, channel programs, which perform the actual I/O operations, are executed independently of the program using the CPU. In addition, pending channel programs are queued (or scheduled) when the channels are busy. While these channel operations are occurring, the CPU is available for further execution of instructions. These instructions may be from the same or another program.

Thus programs may overlap and run concurrently although execution of instructions cannot occur simultaneously.

A job in which the I/O time exceeds the processing time is said to be "I/O bound". Another type of job is one in which the processing time exceeds the I/O time. Such a job is said to be " .... ".

●●●

process bound

2 In addition to these two job types, a job in which the I/O time and the processing time are equal is called a "balanced" job; and finally, a job in which the I/O time exceeds the processing time during part of the job and the processing time exceeds the I/O time during part of the job is called a "complex" job.

Match the following job types and definitions:

1. I/O bound
2. Process bound
3. Balanced
4. Complex

a. I/O and processing are equal.
b. I/O exceeds processing.
c. I/O exceeds processing part of the time and processing exceeds I/O part of the time.
d. Processing exceeds I/O.

●●●

1. b
2. d
3. a
4. c

3 In multiprogramming, the number of programs existing in core storage at any given time is .... (one/more than one).

●●●

more than one.

4 In multiprogramming, jobs share a CPU for execution. Since a CPU can only perform one operation at any one time, multiprogramming jobs must run .... (concurrently/simultaneously).

●●●

concurrently

5 When more than one independent program is stored in core storage, core storage must be divided, or allocated, into an area for each program. These areas are called partitions.

These partitions are normally fixed in size and location at the time the operating system is generated and are thus not under operating system control. Only under the highest level of operating systems (OS) is dynamic control of the size and location of the partitions to meet program requirements possible. This is called multi-programming with a variable number of tasks (MVT).

DOS multiprogramming stores programs in.......... (fixed/variable) partitions.

●●●

fixed

6 Partitions have two major characteristics, priority and protection.

Protection, more formally storage protection, is a feature required to assure that a program in one partition will not accidentally write over and destroy the instructions and data of another program in a different partition. Thus, each partition is "locked" and instructions must have the proper <u>key</u> to operate on data in that partition. Protection is implemented in 2K sections of main storage.

The storage protect special feature.......... (is/ is not) required for multiprogramming.

●●●

is

7 Priority is normally determined by locating the programs in specific partitions. The high priority partitions are called "foreground" partitions or areas. The lowest priority partition is called the background area.

All foreground partitions must have.......... (identical/different) keys for storage protection.

●●●

different

8 Remember, all programs running under multiprogramming have .... (the same/ an individual) CPU.

●●●

the same

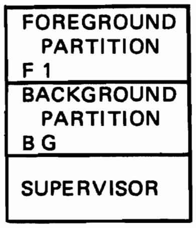
9 In addition, they also share the same supervisor. The priority assignments thus determine which partition will have control of the CPU at any given time.

A foreground partition has a .... (higher/lower/ equal) priority compared to the background partition.

●●●

higher

10 Assume that main storage has been partitioned as shown in the following diagram.



Let us call the foreground partition F1 and the background partition BG for short.

Which has the lower priority?

●●●

BG

11 Let us examine what types of programs might be executed in these partitions. Remember, a program in which the processing time exceeds the I/O time is said to be .... and one in which the I/O time exceeds the process time is said to be ....

●●●

process bound
I/O bound.

12 The foreground program, because of its higher priority, will have first call on the CPU and the supervisor for its execution. Rank the program types in the order of least use of CPU for its execution.

a. I/O bound
b. Complex
c. Process bound
d. Balanced

●●●

1. a
2. b, d
3. c

13 In addition, you will recall that when an I/O operation is taking place during the execution of a program, the .... is operating but the .... may be in the wait state.

●●●

channel
CPU

14 Once the channel program has been started, the CPU is free to continue execution of instructions. If there is no data pending the completion of the channel program (the I/O operation), the CPU goes into the wait state.

Under multiprogramming, this wait time may be used to execute instructions of a lower priority program.

An I/O bound program would spend most of its execution time running channel programs while the CPU sits idly by waiting for a free channel.

A high I/O requirement is a desirable characteristic for an F1 program.

If the supervisor can detect when the F1 program is not using the CPU, it can assign the system to execute the BG program during F1 wait time.

A foreground program characteristic would be one that is .... (I/O bound/processor bound).

●●●

I/O bound

15 One class of programs which meets this requirement very well are TP programs. The system is normally available "on-demand" when running TP. Thus, while waiting for TP input in the foreground area, a different problem can be running in the background area.

A data collection system would be a good program to run in the .... (foreground/background) area.

●●●

foreground

16 A second characteristic of a foreground program is low processing time. A very good example of this is the group of file-to-file utility programs. Here data from one file is placed in another file with only minor rearranging of the data permitted. Thus the processing is almost nonexistent compared to the I/O requirements.

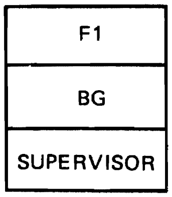
On the other hand, a scientific program would have almost no I/O requirements and would surely be run in a .... (foreground/background) partition.

●●●

background

17 Normally, standard data processing programs such as sales analysis or accounts receivable are processed in the background area.

Assume we have a payroll program and a voice-response program to run our computer partitioned as shown.



Which program would be placed in the F1 partition..........? In the BG partition..........?

●●●

voice-response
payroll

18 These programs share the same .... and ....

●●●

supervisor
CPU

19 Normally in such a system, the BG program would be running. Assume that a request of the voice-response program was made by dialing a phone to the proper number and code. This would initiate an external interrupt which would be interpreted by the supervisor as a request for service by the F1 program.

Since the CPU can only execute one program at a time, execution of the BG program must be suspended temporarily. Remember, the CPU contains the general purpose registers which contain program data. This data must be saved, then restored when execution of the suspended program is resumed. The logical status of the program is found in the current ....

●●●

PSW

20 The PSW must also be saved. This information, together with the program name, is stored in the proper "savearea".

The operation of storing this information prior to switching control of the CPU to another program, is done by the supervisor.

The information contained in the registers and the PSW is sufficient to permit resumption of the execution of program without error at a later time. The program name is stored for program identification in case of an intervening system malfunction.

When the interrupt is being serviced, the supervisor stores the ...., ..., and .... from the BG area and turns control over to the F1 program.

●●●

registers (general and floating-point)
PSW
program name

21 Execution of the F1 program will begin, and in our case, initiate an I/O channel program to accept an account number. The CPU is now in the wait state and free to return to the BG program.

When we return to the BG program, we can expect to store the ...., ..., and .... belonging to the F1 program.

●●●

PSW
registers
program name

22 To resume execution of the BG program, information in the BG .... must first be loaded in their proper locations.

●●●

savearea
Execution of the BG program can then resume.

23 When the dialing of the account number is complete, an I/O interrupt occurs, signaling that the channel program has been completed. Because the F1 program has top priority, the supervisor now goes about returning control to the F1 program for execution. But first the ...., ..., and .... of the BG program must be saved.

●●●

PSW
registers
program name

24 This same data for the F1 program is loaded and execution of the F1 program is resumed.

After handling the inquiry, the F1 program requests more input and finding none, once again goes into a wait state and execution of the BG program may resume.

●●●

25 To get an idea of how much processing can be done during an input operation, let us consider our example.

The input data to the F1 program is the account number. Assume that this number contains seven digits. TheF1 program has requested input data (the account number), then the F1 program goes into the wait state while the channel program is being executed. The customer now dials in the 7 digits.

This takes approximately 11 seconds. Remember, the F1 program is in the wait state during this time and no instructions will be executed. This time is available to the BG program.

Assuming that execution time of an instruction averages 50 microseconds on the S/360 Model 30 being used, that 11 seconds is sufficient time to execute a total of 220,000 instructions in the background (BG) program.

●●●

26 Sequence the following multiprogramming operation. Assume that one foreground and one background program are in storage and that the background program is running.

a. Store registers, PSW and program name of background program; and load registers, PSW and program name of foreground program and re-enter the foreground program.

b. Foreground program requests I/O and returns control to supervisor.

c. Input data is complete for foreground program.

d. Inquiry signal requests service by foreground program.

e. Inquiry is processed by foreground program.

f. Store registers, PSW and program name of background program; and load registers, PSW and program name of the foreground program; and enter foreground program.

g. Resume execution of background program.

h. Store registers, PSW and program name of foreground program.

i. Repeat steps 4, 5 and 6.

j. Load registers, PSW and program name of background program.

●●●

1. d
2. f
3. b
4. h
5. j
6. g
7. c
8. a
9. e
10. i

27 Define multiprogramming in your own words.

● ● ●

Multiprogramming is the concurrent execution of two or more programs residing in main storage.

28 What type of job is particularly suited for execution in a foreground partition?

● ● ●

I/O bound.

29 It can readily be seen that in order to contain a supervisor and two or more problem programs for multiprogramming, main storage will have to be .... in size.

● ● ●

large

30 Also, in order to be able to share a CPU among two or more programs, I/O operations must be handled by ....

● ● ●

channels (channel programs)

31 Of course, a control program, or supervisor, designed for the function of determining which program should be serviced as well as the other supervisor functions previously discussed, must be available.

Finally, in order to prevent accidental destruction of instructions or data in another partition, storage .... is required in multiprogramming.

● ● ●

protection

32 These last four items are the principal characteristics required for a multiprogramming system. List them.

● ● ●

1. Large main storage
2. Channels
3. Supervisor/control program
4. Storage protection

MULTIPROCESSING

33 Multiprocessing is defined as communication between two computers.

Multiprocessing (multiple processors) is sometimes confused with multiprogramming (multiple programs) or with teleprocessing.

Multiprogramming involves concurrent execution of programs which share a .... and ....

● ● ●

supervisor
CPU

34 In multiprocessing, two or more processors are required. Each may run as an independent computer and one or all may directly access the main storage area of another computer. To access another processor's main storage, one CPU must be able to signal the other CPU that such an operation is about to take place. In other words, the CPU's must be able to communicate.

Communication is required because, as you remember, only one operation may take place in main storage at any given time. Thus both CPU's may not access main storage simultaneously.

Multiprogramming requires one CPU and one main storage. What is the minimum number of each required for multiprocessing?

● ● ●

Two CPU's and two main storages

35 Communication between two CPU's requires a special feature called "Direct Control". Through the direct control feature, two or more CPU's can share (read from or write into) main storages. This characteristic is called "shared storage".

Computer communications via TP is not necessarily multiprocessing as one CPU does not control the other, but merely serves as an I/O device. True, it is a very fast and sophisticated I/O device, but neither CPU relinquishes control of its main storage.

Other characteristics are the same as those required for multiprogramming. These are:

●●●

1. Large main storage
2. Channels
3. Storage Protection
4. Control program/supervisor

36 The two additional characteristics are:

1. ....
2. ....

●●●

1. Direct control
2. Shared main storage

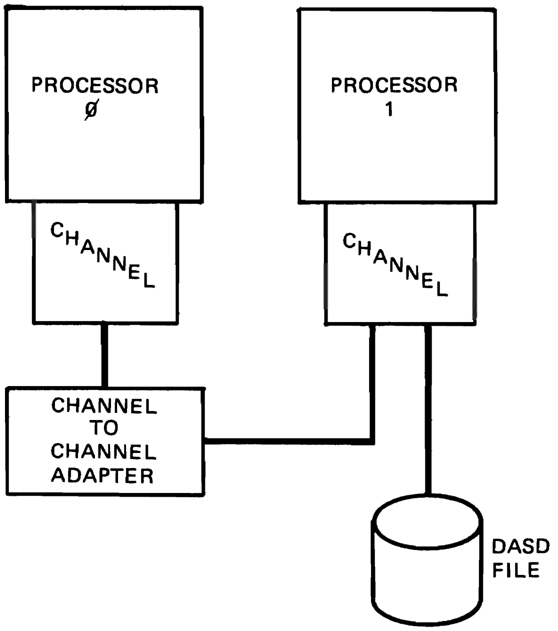
37 System/360 also provides a very useful feature called a channel-to-channel adapter. This permits multiprocessing computers to share files. For example, one computer may be continuously updating an inventory file while another computer is accessing the same file to check stock levels for order entry.

Accessing of the same file by different computers is accomplished by using a ....

●●●

channel-to-channel adapter

SHARED FILES



38 High speed data transfer between computers is simplified through the use of a channel-to-channel adapter. This special feature thus permits one processor to serve as an I/O device for another processor. High speed transfer of data from one main storage to the other is then done through the channels.

Define multiprocessing in your own words.

●●●

Multiprocessing is communication between two computers.

39 Multiprocessing is only available on the largest computers in the S/360 line and only under the full Operating System (OS).

●●●

You have completed this section. At this time, you should fill in your notes and take the self-evaluation quiz.

Self Evaluation Quiz

Multiprogramming - Multiprocessing

QUESTIONS

1. Execution of programs is ________________ (concurrent/simultaneous) under multiprogramming.

2. The CPU can execute __________ instruction(s) at a time.

3. Execution of channel programs ____________ (may/may not) occur while the CPU is executing an instruction.

Match the following job types:

4. I/O bound a. Process time exceeds I/O time

5. Process bound b. Process time exceeds I/O time during part of the run only

6. Balanced c. I/O time exceeds process time

7. Complex d. I/O time equals process time

8. In multiprogramming, foreground area describes:
   a. the front partition of the machine
   b. the front part of core
   c. a high priority storage area
   d. a low priority storage area

Match the following program characteristics with the partitions which best correspond:
   a. foreground
   b. background

9. I/O bound

10. Low priority

11. Process bound

12. TP

13. Tape to Card Utility

14. Accounts Receivable

15. For the storage protect feature to operate, storage is allocated in multiples of ____________ for partitions.

16. What data is required to restore a program to the proper operation point following a multiprogramming program switch?

17. Define multiprocessing.

18. What is the principal factor that differentiates multiprocessing from multiprogramming.

19. The ability to signal another CPU that the first computer is going to use the main storage data or instructions in the second computer is called ________________ .

20. High speed data transfer between computers is simplified through the use of a ___________.

Self-Evaluation Ouiz
Multiprogramming and Multiprocessing

ANSWERS			Frame Reference
1.	concurrent		(1)
2.	one		(1)
3.	may		(1)
4.	c.	I/O time exceeds process time.	(1)
5.	a.	Process time exceeds I/O time.	(1)
6.	d.	I/O time equals process time.	(2)
7.	b.	Process time exceeds I/O time during part of the run only.	(2)
8.	c.	a high priority area.	(7)
9.	a.	foreground.	(14)
10.	b.	background.	(10)
11.	b.	background	(16)
12.	a.	foreground.	(16)
13.	a.	foreground.	(16)
14.	b.	background.	(17)
15.	2K (2048 bytes)		(6)
16.	a. b. c.	PSW contents of general registers. If used, contents of floating-point registers.	(20)
	The program name is useful in case of a program malfunction.		
17.	Multiprocessing is defined as communications between two computers.		(33)
18.	Communication between 2 or more computers is required. Only one computer is required in multiprogramming.		(34)
19.	Direct control.		(35)
20.	Channel-to-channel adapter.		(37)

Transition

TRANSITION

1 For most of the data processing installations going to a System/360, this new computer will make possible an expansion of the facilities provided by the present computer. Defining, planning, and performing the functions necessary to prepare for and make use of their new System/360 is called transition.

The major tasks of transition answer the questions posed by our old friends, WHAT, HOW, WHO, WHEN and CONVERSION. Let us take a brief look at how they are implemented.

1. WHAT

   Define the transition objectives.

   The objectives specify in detail exactly what is required for the smooth conversion from the old data processing system to the new.

2. HOW

   Plan well to reach the objectives defined.

   The amount and care of the planning involved is a large factor in reaching the objectives of the installation. Planning will include the scope of the applications to be served by the new system, the selection of standards and languages, the priority of the various functions and the method of implementation of the programs (immediate reprogramming, conversion, new applications or emulation).

3. WHO

   Staff and train properly.

   Staff and train management, programming, and operations personnel who will be involved in the transition to, and operation of the new system. This will normally consist of internal (standards and procedures) and external (IBM Education Centers and IBM self-study material)training.

4. CONVERSION

   Converting data and programs from old to new format.

   Data and programs for the old system may or may not be acceptable to the new system. If they are not, they must be converted immediately through reformatting or reprogramming. Data and programs for 1400 and 7000 series systems may often be used unchanged. Conversion to the new formats may be postponed until more time is available.

5. WHEN

   Transition schedule complete through changeover.

   The changeover is the target date for going to the new system operation. However, realistic scheduling is required throughout the transition period to assure efficient use of manpower and facilities.

● ● ●

2 Transition is the time and all of the activities required for converting from one data processing system to another.

Match the following tasks which make up transition.

1. WHAT
2. HOW
3. WHO
4. WHEN
5. CONVERSION

a. Convert data and procedures
b. Planning to reach objectives
c. Changeover
d. Staffing and training
e. Defining transition objectives

● ● ●

1. e
2. b
3. d
4. c
5. a

3 IBM provides a large variety of transition aids for customers. Combinations of these aids should be selected as required during transition. Some of these may be used before the programming of the new system begins, some are of value before testing begins, and others are useful both during and after transition. Some aids are in the form of hardware, others are not, and some are a combination of both hardware and programming.

The Document Aids System (DAS) programs may be used early in transition to assist in existing program maintenance and documentation. The DAS programs are executed on a 1401 or 1460 computer or on a 1410, 7010 or S/360 in compatibility mode. There are four DAS programs.

a. Update.
This program permits changes and additions to existing programs.

b. Analysis of source program.
This program produces a cross-reference listing for existing programs. (A cross-reference listing is a table listing the symbolic labels used in a program, the statement number where each was defined and all the statement numbers which use each particular symbolic label.)

c. Flowchart.
This program generates a program flowchart from the source program.

d. Storage Map.
This program produces a storage map of the object program, indicating the locations and sizes of the program phases as they would be stored in main storage ready to execute.

Match the following DAS programs with the proper functions:

1. Update
2. Analysis
3. Flowchart
4. Storage Map

a. Generate flowchart from source program.
b. Produce storage map from object program.
c. Change or add to existing program.
d. Produce cross-reference listing.

● ● ●

1. c
2. d
3. a
4. b

4 Several Language Conversion Programs (LCP) are available. The LCP's convert RPG, COBOL and FORTRAN II programs written for the 1400 and 7000 series equipment, into S/360 acceptable format.

LCP's can be used to reduce the programming task and to reduce debugging and testing time.

What do the initials LCP stand for?

● ● ●

Language Conversion Programs

5 Which of the following languages can be converted to S/360 format by LCP's?

a. COBOL
b. SPS
c. ASSEMBLER
d. FORTRAN II
e. FORTRAN IV
f. FARGO
g. RPG
h. AUTOCODER

● ● ●

a, d, g

6 Another aid to reducing the reprogramming load during transition is “simulation”. Simulation is done by programming the new computer to act like the old one. The simulator program analyzes each instruction from an existing program, then translates it into an instruction or instructions which can be performed by the new computer. The performance of the simulating computer is greatly compromised.

A System/360 computer may be used to simulate the following computers by using IBM-supplied simulator programs:

1. 1410/7010
2. 1620
3. 7080
4. 7090/7094

●●●

7 Sometimes portions of the simulation functions are performed by modifications to the hardware. This modification of the hardware to make the new system act like the old one is called “emulation”.

Simulation is making one computer act like another by means of ....

●●●

programming.

8 Using hardware modifications to make one computer act like another is called ....

●●●

emulation.

9 Emulation provides a distinct advantage over simulation in terms of relative performance on the same program.

Existing programs can thus be run on the new computer by using three transition aids which are:

1. ....
2. ....
3. ....

●●●

1. LCP
2. simulation
3. emulation

10 The most common emulation in use for System/360 is the special feature called 1401 compatibility.

In compatibility mode of operation, the System/360 thinks it is and accepts programs written for the .... computer.

●●●

1401

11 To run in 1401 compatibility mode, the compatibility special feature must be installed in the System/360. This is a hardware modification and thus the 1401 is being .... (simulated/emulated).

●●●

emulated.

12 The System/360 is initialized and switched over to compatibility mode by means of a Compatibility Initialization Deck. The CID assigns the I/O devices, loads the instruction set, etc.

When initialization is complete, a console signal will light up, indicating that the computer is now operating in COMP MODE. 1401 programs may now be run on the S/360 as though it were a 1401.

●●●

13 The efficient processing of programs in 1401 Compatibility mode can be further improved through the use of the Compatibility Operating System (COS).

COS is a program by means of which the S/360 will execute 1401 instructions in Compatibility mode but will do I/O operations requested in Native mode (Native mode is the normal S/360 mode of operating).

Native mode I/O makes full use of channel operation and all the advantages that accrue from the use of channels. Thus the throughput of Compatibility run programs can be significantly increased.

COS is a .... (hardware/program) aid.

● ● ●

program

14 COS I/O looks like .... (1401/S/360) I/O to the 1401 program being run.

● ● ●

1401

15 A significant advantage in running 1401 COS is that the 1401 programs may be run in the DOS job stack. That is, they may be intermixed in the job stream with S/360 jobs and DOS will handle the job-to-job transition.

● ● ●

16 The tape drives used in 1400 series and 7000 series machines were 7-track drives. That is, six tracks were written on the tape to store data in BCD form (6 bits) and the seventh track was used for the parity bit. The S/360 tapes are 9-track tapes to store a byte plus a parity bit across the tape.

To permit processing of 7-track tapes as input to the S/360, S/360 tape drives may be modified by means of a hardware special feature. With this feature, the tape files which have been accumulated over the life of the old system do not automatically become obsolete. This results in a saving of time and money. Tape drives modified for 7-track tape cannot process 9-track tape.

● ● ●

17 The 7-track files which can be run on the 7-track drive special feature may have labels and data which were standard formats for the system under which they were created. Thus, a 7090 tape file may contain a 7090-standard tape label and 7090-standard data representation. These differ from S/360-standard labels and data representation.

A means for processing and writing these non-S/360 standard labels and data formats is a group of programs called the Data Conversion Programs.

Once this data has been converted, it may be written out on the S/360 I/O devices in standard S/360 format.

DASD data is handled by first copying it onto tape in the old system then converting these records to standard S/360 DASD formats for output.

Data Conversion programs handle data directly from .... (tape/disk/tape and disk) files.

● ● ●

tape

18 Standard labels other than S/360-standard labels .... (may/may not) be written on output tape files by a S/360 and the Data Conversion programs.

● ● ●

may

19 A very important transition aid is the availability of IBM-supplied application programs. IBM has a large library of pre-written programs which can be tailored to the specific needs of an installation. These programs are available to IBM customers upon request.

Examples of these programs are:

1. Demand Deposit Accounting.
   A banking program providing such functions as account balances, stop-hold operations, reports, history files, etc.

2. Mortgage Loan Program.
   This program provides such functions as billing, accrued interest, payments, etc.

3. Bill of Materials Program.
   This program provides functions such as bill of materials, indented parts list, next assembly listing, summarized explosion, etc.

4. Medical Information System.
   This program provides such functions as patient records, patient accounting, etc., for hospitals.

●●●

20 The various options provided by the IBM transition aids for the S/360 give users a good set of tools for smooth conversion of the system. The LCP's, simulators, and emulators permit deferment of some design and coding tasks. This helps balance the personnel and equipment resources with the job at hand. New, high-volume, and other important applications may be selected for immediate programming by the powerful language and system facilities available for the S/360.

Test Center and Data Center facilities are available for testing programs prior to installation of the S/360 at the user location. Centrally located education centers and self-study programmed instruction materials for user location use are available for all levels of data processing personnel training.

Intelligent planning and careful follow-up can make transition easy. S/360 transition aids make conversion from other systems easier than ever before.

●●●

You have completed this section. At this point, you should fill in your notes and take the self-evaluation quiz.

Self-Evaluation Quiz
Transition

QUESTIONS

List the major tasks of transition brought to mind by the following:

1. WHAT
2. HOW
3. WHO
4. CONVERSION
5. WHEN

6. What is transition?

7. What are the functions of the Document Aids System (DAS)?

8. What languages can be converted by the Language Conversion Programs (LCP)?

9. What distinguishes simulation from emulation?

10. Is the 1401 Compatibility feature a simulation or emulation item?

11. Do Data Conversion Programs convert disk files directly?

Self-Evaluation Quiz
Transition

ANSWERS — Frame Reference

1. Define the transition objectives. (1)

2. Plan to reach those objectives. (1)

3. Staff and train properly. (1)

4. Convert data and programs from the old format to the new format. (1)

5. Schedule the transition completely through changeover. (1)

6. Transition is the time and all the activities required for converting from one data processing system to another. (2)

7. To assist in existing program maintenance and documentation. (3)

8. (4)
   a. COBOL
   b. FORTRAN II
   c. RPG

9. Simulation is programming a computer to act like another, and emulation does this with hardware modifications. (6,7)

10. Emulation (10)

11. No, the disk records must be stored on tape on the old system, then converted to the S/360 format on the S/360. (17)

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
[U.S.A. only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

GR29-0256-3

Introduction to System/360 Text Printed in USA GR29-0256-3