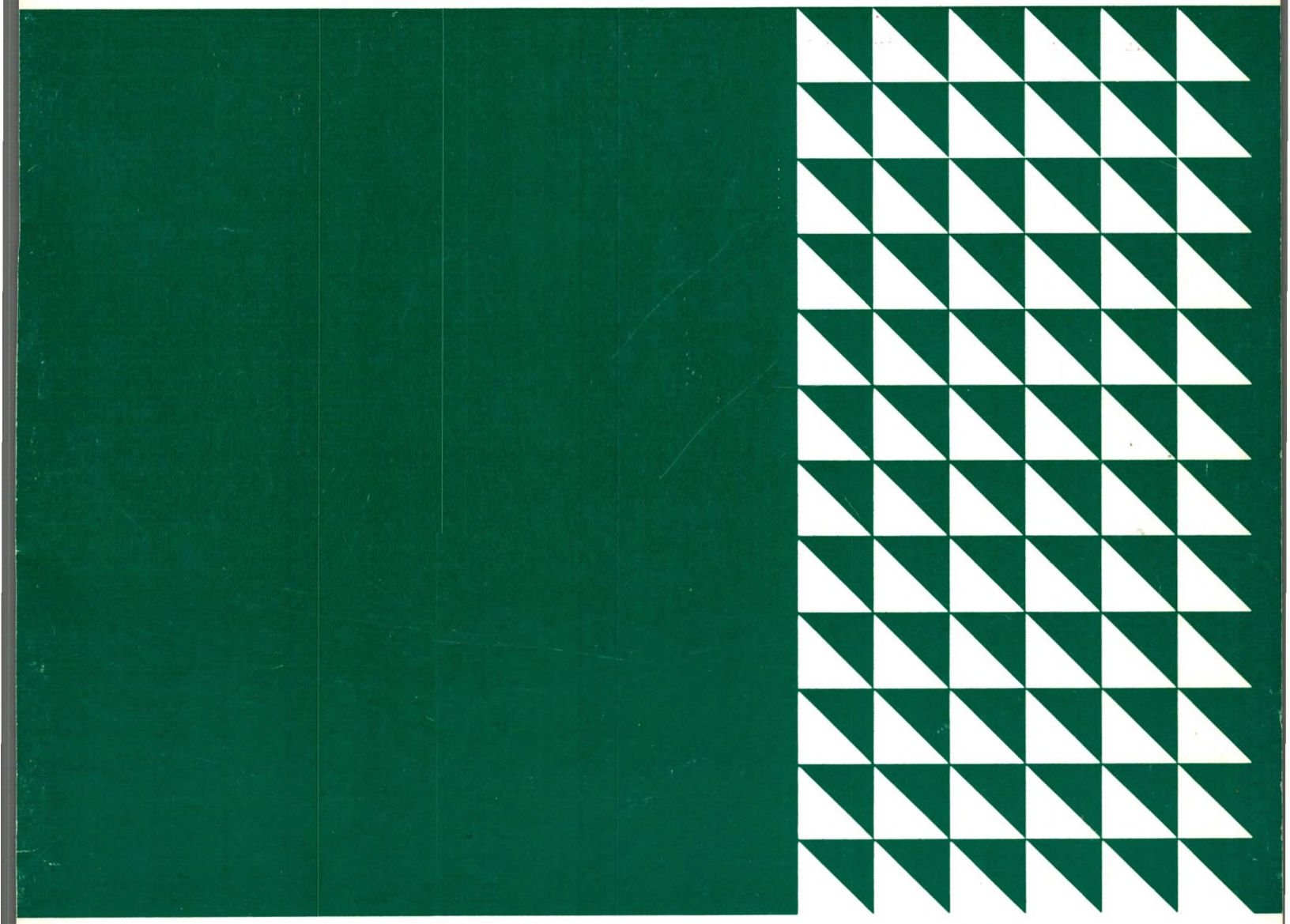
IBM

Assembler Language Coding System Review Text



Programmed Instruction

IBM

Assembler Language Coding System Review Text

Programmed Instruction

ACKNOWLEDGEMENT

We wish to express our appreciation to the Field Engineering Division for providing most of the frames and illustrations used in this course.

In addition, we want to thank the Detroit and Los Angeles DP Education Centers for the frames and problem statements they provided.

Sixth Edition (December 1970)
Reprinted April 1972

This publication, SR29-0231-5, is a minor revision of SR29-0231-4 and incorporates minor changes. The previous edition is not obsoleted.

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.

PREFACE

This book comprises nearly all the contents of books 1 and 2 in the Field Engineering Education Student Self Study Course (Form Nos. R23-2933 and R23-2950).

The sequence of topics has been altered, and some of the more detailed explanations of System/360 functions have been omitted, in order to better suit customer programmer's needs.

The student is assumed to have satisfactorily completed courses in Computing Systems Fundamentals and System/360 Introduction.

How to use this book. This book is intended to serve only as a "teaching test":

In the prerequisite courses you learned many things about computer systems in general, and System/360 in particular. This volume is intended to test your knowledge of, and emphasize, those general topics which are most relevant to coding, assembling, and debugging programs.

Each section begins with a list of learning objectives and a Self-Evaluation test. As you begin a section, you should read the objectives and take the test. Answers are printed on the succeeding pages. For each question and answer there is a reference to a page, in that section of the text which presents the necessary information in self-study form.

You should take each test as follows:

a. Answer every question, before looking at any of the answers. Use scratch paper.

b. If you cannot answer a given question, stop taking the test. Turn to the indicated page, in that section, and begin reading the self-study material. When you have read enough to answer the question, continue with the test.

c. When you have completed the test, check your answers. If one of your answers is wrong, and you don't know why, read the indicated self-study material. When you have read enough to see why the printed answer is correct, continue checking the rest of the answers.

If you have previously learned all that is necessary to answer every question, you would skip from one Self-Evaluation quiz to the next, and you would never read any of the remedial self-study material. This would be ideal: You would take the least time to reach the point where you could begin to learn coding.

You will probably find, however, that you need to read about some topics. For the most efficient use of this volume, try to minimize such reading.

The only other instructions are general, but they are very important: The self-study material is organized into a series of statements called "frames". Each frame gives you some information and then asks you to do something with it (think of an answer, perform a calculation, describe a relationship, etc.). After you respond to the information in the indicated manner, you can check your response against the correct one. It is printed on the left, directly below the frame.

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 you print, 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.

Find your IBM System/360 Reference Data Card (Form X20-1703), turn to the first page of Section I, and begin.

CONTENTS

SECTION I - NOTATION

Note: Be sure that you have read the entire Introduction to this volume, before continuing. You may save time by doing so.

Anyone who codes programs for the System/360 will, at some time, work with numbers expressed in binary or hexadecimal notation.

NUMBERING SYSTEMS

Learning Objectives

When you complete the following test, including any necessary additional reading, you will have demonstrated that you can:

- Convert decimal numbers to binary or hexadecimal notation.
- Convert binary or hexadecimal numbers to decimal notation.
- Add or subtract binary or hexadecimal numbers, by converting to decimal, performing the arithmetic operation, then converting back to the original form.

SELF-EVALUATION QUESTIONS — Reference Pages in Text

Note: Use your Reference Data Card.

1. Given the decimal number 17226: — 3
   a. The high-order position contains the digit _____.
   b. The low-order position contains the digit_____.

2. Express the decimal values 0-15 as a four position binary number and as one hexadecimal digit. — 4 and 5

Decimal	Binary	Hexadecimal
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

3. Express each of the following sums, using the notational system indicated: — 5 - 8

	Binary		Hexadecimal
a.	11011011	b.	FADE
	+ 10000110		+ DEAF

ANSWERS FOR NUMBERING SYSTEMS QUIZ

Reference Pages in Text

1. a. 1
   b. 6

3

2.

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

4 and 5

3. a. 1 0 1 1 0 0 0 0 1
   b. 1D98D

5 - 8

GO TO SECTION II - ORGANIZATION, on page 9.

NUMERIC REPRESENTATION

Numbering systems were developed by man so that he could count. Later the simple act of counting was expanded to the four basic mechanics of arithmetic: addition, subtraction, multiplication, and division. A number is basically a string of symbols. Such a number in the decimal system, with which you are quite familiar, is 360. Each symbol in a number has a definite place value. At this point, let's review some general rules and see how they apply to the numbering systems used by System/360.

A number is a sum of terms. Each term is a product of a digit symbol and its place value. The place value of the digit symbol is some power of the base. The power of the base starts with zero and increases by 1 from right to left.

• • •

1. A number is a sum of _____. Each term is a product of a _____ and its _____ _____. The decimal numbering system has a _____ of ten.

• • •

terms; digit; place value; base

Looking at the decimal number 360 as a sum of terms you can see that:

$360 = 3 \times 10^2 + 6 \times 10^1 + 0 \times 10^0$

Digit; Base; Exponent or Power of Base

TERM

This could also have been expressed in this manner:

$360 = 3 \times 100 + 6 \times 10 + 0 \times 1$

Notice that $10^0 = 1$. Any value to the power of 0 equals 1.

2. $2^0 =$ _____ $16^0 =$ _____

• • •

1; 1

Another rule that you can see from the previous example is that the place value of each digit increases going from right to left. The rightmost digit of a number is called its low-order or least significant position. The leftmost digit is called its high-order or most significant position. Example:

360

High Order or Most Significant — Low Order or Least Significant

3. Given the decimal number 479, express it as a sum of terms and indicate the low-order position.

479 = _____

• • •

$4 \times 10^2 + 7 \times 10^1 + 9 \times 10^0$ ← Low Order

or

$4 \times 100 + 7 \times 10 + 9 \times 1$

Of course, you would not ordinarily express decimal numbers as sums of terms because you are too familiar with the decimal numbering system. However, numbering systems with a base other than 10 can also be expressed as a sum of terms. So as you will see, there are definite similarities between numbering systems regardless of the base.

The System/360 is capable of performing arithmetic instructions involving three different numbering systems. As part of its standard instruction set, the System/360 can do basic arithmetic with binary numbers. With the addition of the decimal feature, it can do arithmetic with binary coded decimal numbers. With the floating point feature, it can do floating point arithmetic operations with hexadecimal numbers.

4. List three numbering systems used by the System/360.

a._____ b._____ c._____

• • •

a. Binary b. Decimal c. Hexadecimal

You have been working with the decimal system most of your life. It uses the value 10 (ten) for its base. This means that each place in a decimal number represents ten raised to a power.

100,000	10,000	1,000	100	10	1

It uses ten digit symbols (0-9). Each time the highest digit value (9) is exceeded by 1 in any place of the number, the result is zero and there is a carry of 1 to the next higher place value. Example:

$$\begin{array}{rl} 09 = & 0 \times 10 + 9 \times 1 \\ +\ 01 = & +\ 0 \times 10 + 1 \times 1 \\ & 0 \times 10 + 0 \times 1 \\ & +\ 1 \times 10 \leftarrow \text{Carry} \\ 10 = & 1 \times 10 + 0 \times 1 \end{array}$$

The principle illustrated here is true for the other numbering systems as well. Let's see if you know the principle.

5. When the highest digit value is exceeded by 1, (in your own words) _____

_____

• • •

The result is zero and there is a carry of 1 to the next higher place value.

A numbering system other than decimal which you may be familiar with is the binary numbering system. It uses the base 2 and has only two digit symbols (0 and 1).

6. Express the binary number 1000 as a sum of its terms. _____

• • •

$1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$

or

$1 \times 8 + 0 \times 4 + 0 \times 2 + 0 \times 1$

Notice that a binary number increases by the powers of 2. That is, each added place to a binary number doubles it. Binary 1000 is double binary 100. In decimal, adding a place multiplies the number by ten. Decimal 1000 is ten times decimal 100.

7. Fill in the place values of a six-position decimal number.

• • •

100,000	10,000	1,000	100	10	1

8. Fill in the place values of a six-position binary number.

• • •

32	16	8	4	2	1

9. Express the decimal value of 25 as binary number.

_____

• • •

1 1 0 0 1

10. Which of the following is not a binary number?

a. 1011 b. 0000 c. 1200

• • •

c. 1200

1200 is not a valid binary number because 2 is not a valid symbol. The binary numbering system has only two valid symbols; 0 and 1.

11. Add 1 to binary number 1001. _____

• • •

1010

12. Express the decimal values 0-15 as four-position binary numbers.

DECIMAL	BINARY
0	_____
1	_____
2	_____
3	_____
4	_____
5	_____
6	_____
7	_____
8	_____
9	_____
10	_____
11	_____
12	_____
13	_____
14	_____
15	_____

• • •

DECIMAL	BINARY
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Because the binary numbering system uses only two symbols (0 and 1), it is ideally suited for use in computers. Each bit position in a computer can be used to represent a Binary Digit. To represent a decimal digit, four bit positions are needed. For instance, a decimal 9 would be represented like this: 1001.

A third numbering system in use in the System/360 is the hexadecimal numbering system. The hexadecimal system uses the decimal value of 16 as its base.

16^3	16^2	16^1	16^0
4096	256	16	1

The binary system can only count as high as 1 before a carry occurs.

(1) 0 + 1 = 1
(2) 1 + 1 = 0 with a carry

The decimal system can count as high as 9 before a carry occurs.

(1) 8 + 1 = 9
(2) 9 + 1 = 0 with a carry

In the hexadecimal numbering system you can count as high as 15 before a carry occurs.

(1) 14 + 1 = 15
(2) 15 + 1 = 0 with a carry

To express the value 10 to 15, the symbols A to F are used. This will probably be the hardest thing for you to get used to; seeing alphabetic characters in a number.

13. Express the following hexadecimal number as a sum of terms:

796 = _____

• • •

$796 = 7 \times 16^2 + 9 \times 16^1 + 6 \times 16^0$

or

$7 \times 256 + 9 \times 16 + 6 \times 1$

The decimal value 112 can be expressed as the hexadecimal number 70.

Hexadecimal 70 is equal to: $7 \times 16^1 + 0 \times 16^0$

14. Add 1 to a hexadecimal 9. 9 + 1 = _____

• • •

A

15. Count from a decimal 10 to a decimal 19 in hexadecimal.

Decimal	Hexadecimal
10	____
11	____
12	____
13	____
14	____
15	____
16	____
17	____
18	____
19	____

• • •

Decimal	Hexadecimal	
10	A	These are the hexadecimal symbols for the decimal values 10 to 15.
11	B	
12	C	
13	D	
14	E	
15	F	
16	10	
17	11	
18	12	
19	13	

Since you think most readily in decimal terms, you will find it very helpful to be able to convert from one numbering system to another. You have already expressed several small decimal values as both binary and hexadecimal numbers. It becomes more difficult as the values get larger. Fortunately, there are a few simple rules to remember for converting any number.

CONVERTING FROM DECIMAL TO HEXADECIMAL

For this section, you will be using your IBM System/360 Reference Data Card (Form X20-1703).

• • •

1. Find the "Hexadecimal and Decimal Conversion" table and notice the six columns of figures. The bottom of the table shows that the columns are numbered 1-6 from _____ to _____.

• • •

right; left

2. Each of the columns stands for one position (place) in a hexadecimal number. The table can therefore be used to convert a decimal number into a hexadecimal number with as many as _____ places.

• • •

six

Given a decimal number, we will use the table to fill in the required hexadecimal positions from left to right. Here is an example:

Convert 50,000 to a hexadecimal.

3. Our first operation with the table is to find the largest decimal number that is equal to, or less than, the number we wish to convert. There is no 50,000 in the table. The closest lowest number is in column 4, and the number is _____.

• • •

49,152

4. This first operation tells us two things:
   a. Our hexadecimal number will have four positions (we started in column 4).
   b. The high order (leftmost) position will be occupied by the hex symbol that we find opposite 49,152. This symbol is _____.

• • •

C

5. Out of our original 50,000, we have taken care of 49,152 with one hexadecimal symbol. There is still a difference of 848 (50,000 - 49,152) to be accounted for. We go to the next column to the right (column 3) and repeat the operation with 848. There is no decimal number equal to it, and the closest lower number is _____. The hex symbol that stands for that number is _____.

• • •

768; 3

So far, we have filled two positions of our four-position hexadecimal number: | C | 3 | | |. We still have a difference of 80 (848 - 768 = 80) to account for.

6. Moving to column 2, we find that _____. (Use your own words).

• • •

a decimal value of 80 can be shown by the hex symbol 5.

We have accounted for our entire 50,000 with the hexadecimal symbols | C | 3 | 5 | |. We have a zero differance, after performing our comparison in column 2, but we still must put something in the low-order position to show that we have a four-position number. As expected, when we move into column 1 with a decimal amount of 0, we find that we show it with the hexadecimal symbol 0.

7. Answer: Decimal 50,000 is expressed in hex as _____.

• • •

C350

If the result of our first table operation is a difference that is less than the smallest significant decimal number in the next column to the right, we give the next position a hexadecimal value of zero. Here's an example:

Convert decimal 2,317 to hex.

Starting in column 3, we find a decimal value of 2,304. This gives us a hexadecimal symbol of 9 and a decimal difference of 13 (2,317 - 2,304= 13). Moving to column 2, we find that the lowest significant decimal number is 16. The only decimal number less than 13 is 0. We follow our rule and use the hexadecimal symbol 0 for this position.

8. We have found the first two hexadecimal symbols to be _____.

• • •

90

9. We still must fill the low-order position, and we still have a difference of 13. Going to column 1, we find that we can take care of it with the hexadecimal symbol _____.

• • •

D

Now practice without help. When you get two correct in succession, go on to the topic "Converting from Decimal to Binary".

10. Convert decimal 72,501 to hex.

    • • •

    11B35

11. Convert decimal 36,879 to hex.

    • • •

    900F

12. Convert decimal 2,986 to hex.

    • • •

    BAA

13. Convert decimal 61,455 to hex.

    • • •

    F00F

CONVERTING FROM DECIMAL TO BINARY

Even with the use of the "Powers of 2" table on your reference card, conversion of large decimal numbers is a fairly lengthy procedure. You can shorten it considerably by converting your decimal number to hex, first, then converting the hex number to binary.

You have seen how to convert a decimal number to hex; the rule for the second part is easy:

a. Each hexadecimal symbol is converted into a four-position binary number.

You have already listed the binary equivalent of every decimal number from 0 to 15. The list is the same, for every hex number from 0 to F. (Incidentally, the comparison is printed above the conversion table on your reference card.)

Read the following example, using your card to go through the steps.

Convert 28,465 to binary.

a. decimal 28,465 becomes hex 6F31.
b. hex 6F31 becomes binary 0110 1111 0011 0001.

• • •

1. Now you practice, using the earlier example 61455 (which became, in hex, F00F).

   • • •

   1111 0000 0000 1111

2. Now convert 42,800 to binary.

   • • •

   hex: A730; binary: 1010 0111 0011 0000

Besides using the hexadecimal numbering system for floating point calculations, the System/360 also uses the hex system in most printed material to express long binary numbers. An example of this is expressing the 24-bit binary addresses of main storage as six hexadecimal digits. The six hex digits can be easily converted to binary if it is necessary to find the actual machine language address.

"Hex" Address ⟶ 0 0 0 4 E C

Binary Address ⟶ 0000 0000 0000 0100 1110 1100

If it is desired to find the decimal byte location, the hex address can be converted to decimal.

CONVERTING FROM HEXADECIMAL TO DECIMAL

Your Reference Data card makes this conversion easy. You simply list and add the decimal numbers that correspond to each of the symbols in a given hexadecimal number.

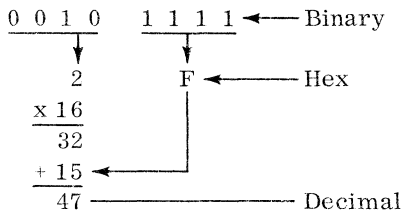
As an example, let's convert the hexadecimal number 538. Since it is a three-position number, we start, in our table, with column 3.

Column	Hex Symbol	Decimal Equivalent
3	5	1280
2	3	48
1	8	+ 8
		1336

The decimal number 1336 is equivalent to the hexadecimal number 538.

CONVERTING FROM BINARY TO DECIMAL

The direct conversion from binary to decimal can be rather lengthy. It is much better to convert from binary to hexadecimal and then to decimal.

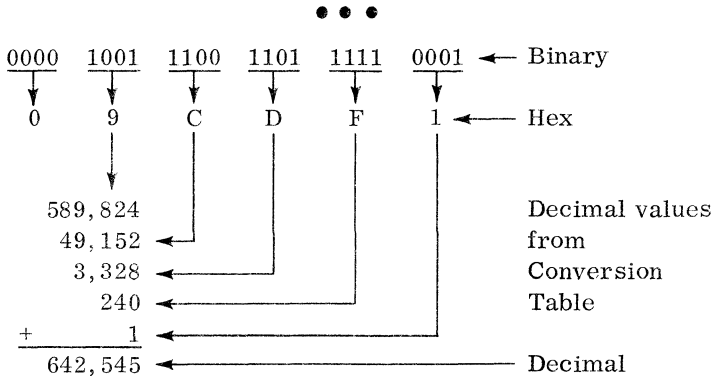


• • •

1. Given the following 24 bit binary address, what is the decimal byte location?

0 0 0 0 1 0 0 1 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1

• • •



So far you know how to count in decimal, binary, or hexadecimal. You can also convert from one numbering system to another. Therefore, you can "add" or "subtract" binary or hexadecimal numbers by converting to decimal, finding the answer, and converting back.

SECTION II - ORGANIZATION

Generally, the closer you come to using machine language, the more consideration you must give to the actual activities of the computer system that you are coding instructions for. While Assembler Language has mnemonic operation codes and allows symbolic addressing, it is a "one-for-one" language - very close to the language of the machine. Therefore, when you begin coding programs for a particular model and configuration, you will need to consider the availability of features, specifics of data management (including file specification and I/O coding), etc.

This section introduces you to some system organization considerations that are fundamental to the use of any model of the System/360.

MAIN STORAGE

Learning Objectives

When you complete the following test, you will have demonstrated that you can:

- Define a byte.
- Identify the relationship between storage organization, in bytes, and the basic addressing scheme.
- Describe halfwords, fullwords, and doublewords of data.
- State the rules for locating halfwords, fullwords, and doublewords in storage, and describe what will happen if these rules aren't followed.

SELF-EVALUATION QUESTIONS	Reference Pages in Text
1. The byte consists of _____ data bits and one _____ bit.	11
2. If a byte has an _____ number of its bits set, a machine check will occur.	12
3. Each main storage address refers to a unique _____ location.	13
4. Data field bit positions are numbered starting with 0, from _____ to _____.	13
5. Data fields are addressed by their _____ -order byte location.	14
6. A _____ is two bytes long.	13
7. A _____ is four bytes long.	13
8. A _____ is eight bytes long.	14
9. What must a storage address be divisible by: a. For halfword boundary alignment? b. For word boundary alignment? c. For doubleword boundary alignment?	14
10. What two program check exceptions could occur when addressing main storage? _____ _____	15
11. The address of fixed length data fields must be divisible by the number of bytes in the field or a _____ exception will occur.	15

ANSWERS

		Reference Pages in Text
1.	eight, parity	11
2.	even	12
3.	byte	13
4.	left, right	13
5.	high	14
6.	halfword	13
7.	word	13
8.	doubleword	14
9.	a. A halfword can start at any address that is divisible by 2. b. A word can start at any address that is divisible by 4. c. A doubleword can start at any address that is divisible by 8.	14
10.	Specification, Addressing	15
11.	Specification	15

GO TO THE NEXT PART OF THIS SECTION "CENTRAL PROCESSING UNIT" on page 17.

The System/360 is a general purpose computer system. By this we mean it is designated to be used for commercial, scientific, and communications applications. In the past, these applications were handled by separate computer families.

Growth ↑	Scientific	Commercial
	7090	7080
	709	705 III
	704	705 II
	701	702

One scientific computer family and its comparable commercial equivalent.

The scientific computers were usually fixed word length machines and used a pure binary form of coding. On the other hand, the commercial computers were usually variable word length (character oriented) machines and used a binary coded representation of decimal information. The System/360 uses binary as well as character data representation and has both fixed and variable length fields.

To fit the cost and volume needs of computer users, the IBM System/360 is available in several models. For instance, to suit the demands of users who need a minimum number of answers per month, a model 30 is available at a minimum cost. A model 75, however, will give better than 50 times as many answers per month. Both models (30 and 75) are, however, program compatible. That is, a program written for a model 30 can run on a model 75 and vice versa. The answers will be the same; the numbers of answers per month will be different.

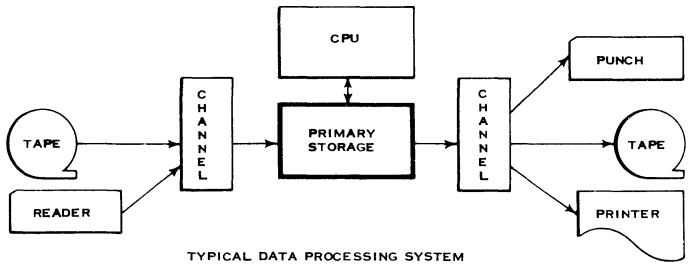
• • •

1. A machine language program written for one model of the System/360 can run on another model. _____(True/False)

• • •

True

MAIN STORAGE



TYPICAL DATA PROCESSING SYSTEM

In the preceding figure, you can see the components that make up a data processing system. You should be familiar with these components either from past experience or because of a computing systems fundamentals course.

Let's learn about these components as they apply to the System/360!

The primary storage is that section of a DP system that contains the program to be executed as well as the data to be processed. All data entering the system goes into the primary storage before it can be processed. After processing, the data must be placed back into primary storage before it can be sent to an output device.

Primary storage is sometimes referred to as main storage. Most computers use ferrite cores as their primary storage device. The System/360 also uses ferrite cores for its main storage.

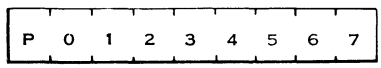
• • •

1. The type of storage used for primary storage in the System/360 is _____ storage.

• • •

core

The smallest addressable unit of main storage in the System/360 is called the byte. The byte consists of eight data bits and one parity bit.



Bit Positions in the Byte

The leftmost bit of a byte is the parity bit. System/360 uses odd parity. That is, an odd number of bits in every byte will be set on (in the 1 state). The remaining bits will be in the 0 state. If an even number of bits are set on, a machine check (error) will be indicated.

2. The smallest addressable unit of main storage is called a_____. It consists of_____ data bits and one_____ bit. The leftmost bit is the_____ bit. A machine error will be indicated if a byte has an_____ number of bits set on.

• • •

byte; eight; parity; parity; even

3. In the System/360, a hexadecimal addressing system is used. A hexadecimal address of 1000_{16} is equivalent to a decimal address of 4096. Therefore, in discussing the capacity of the System/360 it is convenient to use multiples of 1K, where 1K equals $1,024_{10}$ bytes. Thus an 8K machine has a memory capacity of $8,192_{10}$ bytes (2000 hex). A 64K machine has a capacity of_____ bytes.

• • •

65,536

As would be expected, the faster models of System/360 would need more storage bytes than the slower models. Also each model of the system would have as an option several sizes of main or core storage. As can be seen from the following figure, the model 30 comes in four sizes from 8K bytes to 64K bytes.
The model 75, on the other hand, can have either 256K, 512K, or 1,024K bytes in main storage.

MAIN STORAGE						
CAPACITY (BYTES)	SYSTEM MODEL					
	25	30	40	50	65	75
8,192 = C		C				
16,384 = D	D	D	D			
24,576 = DC	DC	DC				
32,768 = E	E	E	E			
49,152 = ED	ED					
65,536 = F		F	F	F		
131,072 = G			G	G	G	
262,144 = H			H	H	H	H
524,288 = I				I	I	I
786,432 = IH					IH	IH
1,048,576 = J					J	J
BYTES/ACCESS	2	1	2	4	8	
MAIN STORAGE SPEED IN USECS	0.9	1.5	2.5	2.0	0.75	

PROCESSING UNIT

- The time required to take a storage cycle varies between models of the System/360.
- The number of bytes accessed during each storage cycle varies with each model of the System/360.

A storage cycle is the period of time during which information is read out of main storage. Either the information that is read out is regenerated or new information is placed back into main storage.

4 The smallest addressable unit of main storage is called a_____.

• • •

byte

You should now realize that main storage addresses start with 0000 for the first byte and increase by 1 for each byte in the particular main storage unit. Valid storage addresses for a model 30 would start with 0000 and continue up to 65,535. On a model 50, valid main storage addresses start with 0000 and continue up to 524,287. To allow for program compatibility as well as for future growth, the System/360 uses a 24-bit binary address to address main storage. A 24-bit binary number allows us to go as high as 16,777,215 for an address. You can see the future growth that is possible here! A binary rather than a binary coded decimal address is used because it is more efficient with large addresses.

5. Picture the 24-bit binary address that would be used to address byte location 0007._____

• • •

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1

You should be familiar enough at this point with the binary numbering system to have done the preceding question without much difficulty. Of course, you might have a slight case of writer's cramps from writing out a 24-bit address. Normally, machine addresses are expressed hexadecimally. Hexadecimal is another numbering system you are familiar with. Binary uses a base of two (2^1) while hexadecimal uses a base of sixteen (2^4). There is a direct 4-to-1 ratio between binary and hexadecimal. Each four binary bits can be expressed as one hexadecimal digit. Address 0007 could be expressed as six hexadecimal digits:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1→0 0 0 0 0 7

6. How would the highest 24-bit binary address be expressed hexadecimally?

• • •

FFFFFF

7. Each main storage address refers to an individual _____. Every byte in main storage can be located by a _____ - _____ binary address.

• • •

byte; 24-bit

8. Every byte has _____ data bits and one _____ bit. The leftmost bit is the _____ bit and is used to give every byte an _____ number of bits set (in the 1 state).

• • •

eight; parity; parity; odd

You are probably a little perplexed about this byte by now. You know that a byte consists of eight data bits and a parity bit! You know that each byte is individually addressable by a 24-bit binary address! You know that main storage size can vary from 8K bytes on a model 30 to over 1000K bytes on a model 75 !
You know that the model 30 accesses one byte per storage cycle while a model 75 accesses eight bytes per storage cycle! However, you are probably asking yourself:

Is the byte a character?
Is it a binary number?
Just what is it?

The answer to these questions is simple. The eight data bits of a byte can be coded to represent characters, binary numbers, or anything you want them to be. The instructions of the System/360 are many and varied. Some of the instructions treat bytes as characters. Some instructions treat bytes as part of a binary number. So the answer to the question, "What does a byte represent?" is that it depends on the particular instruction being executed at the time. This question will be answered more to your satisfaction after you study the data formats and some of the instructions.

As was previously stated, the System/360 is a general purpose data processing system. As such it is designed to operate with fixed length as well as variable length data. The byte as you have already learned is a very versatile unit. It is individually addressable. By further specifying the number of desired bytes, we can have a variable length field in main storage starting and ending at any byte address.

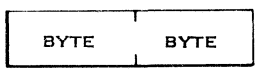
9. The System/360 can operate with variable length data. Variable length data can start at _____byte address.

• • •

any

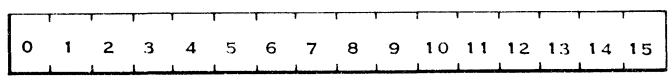
To be of truly general purpose, the System/360 must also be capable of operating with fixed length data. Whereas variable length data has a variable number of bytes, fixed length data always has a fixed number of bytes. So let's go on and define these fixed length fields.

A halfword is two bytes in length.



HALFWORD

The data bit positions of a halfword are numbered 0-15 from left to right.



HALFWORD

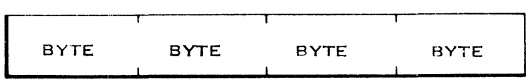
Notice that the parity bits are not shown. They will not be shown from here on, since they do not represent data. However, remember that every byte does contain a parity bit for checking purposes.

10. Two bytes are contained in a _____. Its data bit positions are numbered ___ to ___, left to right. Each halfword has _____ parity bits associated with it.

• • •

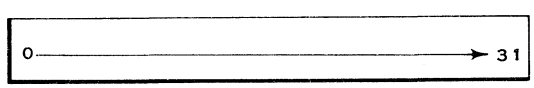
halfword; 0 to 15; two

A word is 4 bytes long.



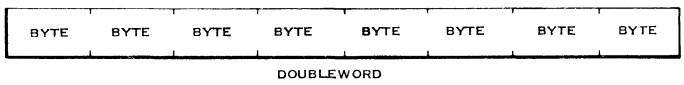
WORD

The data bit positions of a word are numbered 0-31 from left to right.

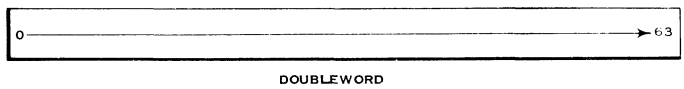


WORD

A doubleword is 8 bytes long.



The data bit positions of a doubleword are numbered 0-63 from left to right.



Remember now that each byte of a halfword, word, or doubleword carries its own parity bit.

11. A byte contains _____ data bits and one _____ bit.

• • •

eight; parity

12. A _____ is two bytes in length.
A _____ is four bytes in length.
A _____ is eight bytes in length.

• • •

halfword; word; doubleword

13. The data bit positions of fixed length data are numbered from _____ (right to left/left to right) starting with bit position 0. Each _____ of fixed length data contains a parity bit.

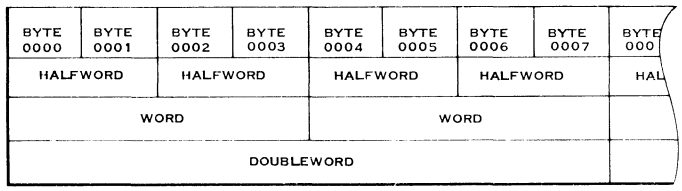
• • •

left to right; byte

The Op code of the instruction determines whether to consider data as variable or fixed. The Op code of the instruction will also determine, in the case of fixed length data, whether it is a halfword, word, or doubleword.

Before leaving the definitions of fixed length data, you must learn the restrictions placed on the use of fixed length data.

The rule is that fixed length data must begin on the correct boundaries in main storage.



Fixed length data is addressed by the high-order byte (leftmost byte) of the field.

The illustration above shows where halfwords, words, and doublewords can start, thus defining their respective boundaries.

For a halfword, the address of the boundary must be divisible by two.

For a word, the address of the boundary must be divisible by four.

For a doubleword, the address of the boundary must be divisible by eight.

Notice that the address of each type of fixed length field must be divisible by the number of bytes in that type of field.

Another way of stating this rule is to say that the 24-bit binary address:

- Of a halfword must have one low-order zero bit.
- Of a word must have two low-order zero bits.
- Of a doubleword must have three low-order zero bits.

14. A fixed length data field is addressed by its _____ (low/high) order byte.

• • •

high

15. The binary address of a word must contain _____ low-order zero bits.

The binary address of a doubleword must contain _____ low-order zero bits.

• • •

two; three

The boundary restriction placed on the use of fixed length fields is a restriction placed on the user. If you violate these rules, it is not a machine check. Instead it is, and rightfully so, considered a program check.

16. Starting a halfword data field at an odd address (such as 000001) will result in a _____ check.

    Incorrect parity in a halfword data field will result in a _____ check.

• • •

program; machine

As there are other restrictions placed on the programmer, you should be able to identify program checks by type. The type of program check caused by a violation of fixed length boundaries is known as a specification exception.

Another exception to valid programming is addressing a byte location that is not available on your particular model of System/360. The largest size main storage available on the model 30 is 65,536 bytes. Any address higher than this would result in a program check. This type of check is known as an addressing exception.

17. What two types of program check exceptions could occur when addressing main storage? _____

    _____

• • •

Specification; Addressing

18. If the binary address of a word does not contain two low-order zero bits, the program check that occurs is identified as a _____ exception.

• • •

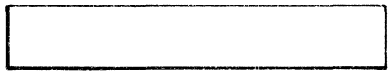
specification

CENTRAL PROCESSING UNIT

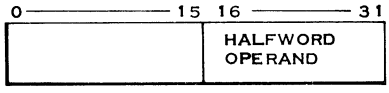
Learning Objectives

When you complete the following test, you will have demonstrated that you can:

- Identify the major system components and their functions.
- Relate component function to logical and arithmetic operations with fixed length and variable length data.

SELF-EVALUATION QUESTIONS	Reference Pages in Text
1. Instructions are decoded by the _____ _____ of CPU.	19
2. For its variable field length operations, the System/360 uses the _____ to _____ concept.	20
3. Variable length fields can start at _____ byte location in main storage.	20
4. The length of variable length fields is specified by (In your own words) _____ .	20
5. When both fixed length operands are in general registers, a _____ to _____ concept may be used.	22
6. For fixed length operations, the System/360 uses a _____ to _____ concept.	21
7. For use as accumulators, the programmer can address _____ _____ _____ .	21
8. Number the bit positions of the general register below. Also show where a halfword operand would be placed. 	21
9. With general register address 8 specified, what two general registers would be used in a fixed length divide operation? _____	22
10. List three main uses of the available general registers. 1. _____ 2. _____ 3. _____	23
11. For floating point operations, the System/360 has _____ floating point registers.	23

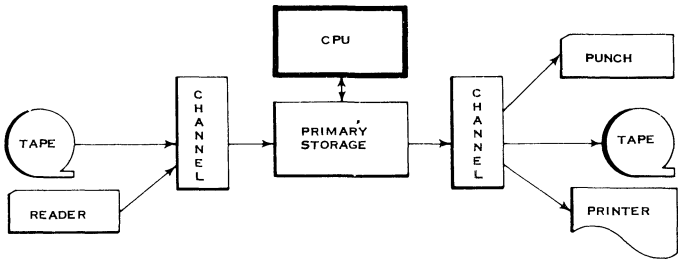
ANSWERS

		Reference Pages in Text
1.	control section	19
2.	storage-to-storage	20
3.	any	20
4.	length code in the instruction	20
5.	register-to-register (or accumulator-to-accumulator)	22
6.	storage-to-register (or accumulator)	21
7.	sixteen general registers	21
8.		21
9.	General Registers 8 and 9	22
10.	1. Accumulators 2. Index Registers 3. Base Registers	23
11.	four	23

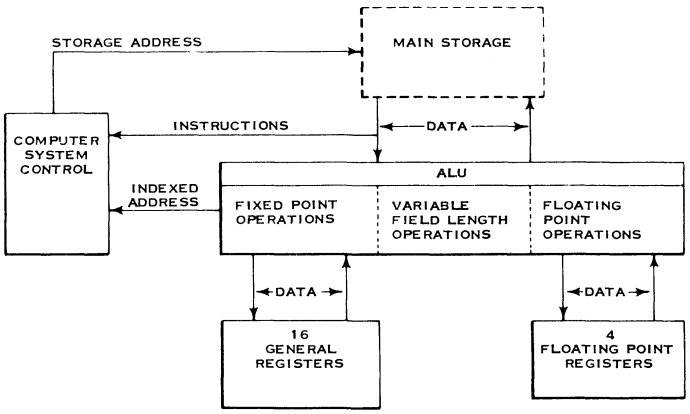
GO ON TO THE NEXT PART OF THIS SECTION "CHANNELS" on page 25.

CENTRAL PROCESSING UNIT

Now that you know the primary storage capabilities of the System/360, let's explore those of the Central Processing Unit (CPU).



TYPICAL DATA PROCESSING UNIT



CENTRAL PROCESSING UNIT LOGIC FLOW

In the preceding illustration, you can see the logical structure of the CPU for the System/360 and its relationship to the main storage.

As you know, there are two main sections in CPU. They are: 1) the control section, and 2) the arithmetic and logical section (called ALU).

From the preceding illustration, you should be able to see some of the functions of the control section. They are:

- All references to main storage, whether for instructions or for data, are made by the control section.
- The control section addresses main storage and causes the instruction to be fetched and sent to the control section. The instruction is then decoded by the control section.

• • •

1. The instruction is brought out of main storage to the _____ section. The control section decodes the _____.

   All addresses are supplied to the main storage by the_____ section.

• • •

control; instruction; control

In general, the arithmetic and logical section of a computer contains the circuits necessary for adding and comparing data fields as well as the other circuits necessary for operating on data fields.

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

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

2. In your own words, what is the function of the arithmetic and logical unit of a computer?_____

   _____

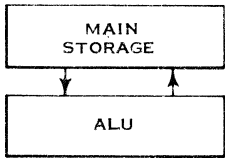
   _____

• • •

The function of the ALU is to perform (execute) on the data fields, the operation defined in the instruction (such as add or subtract).

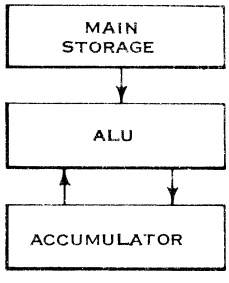
VARIABLE FIELD LENGTH OPERATIONS

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.



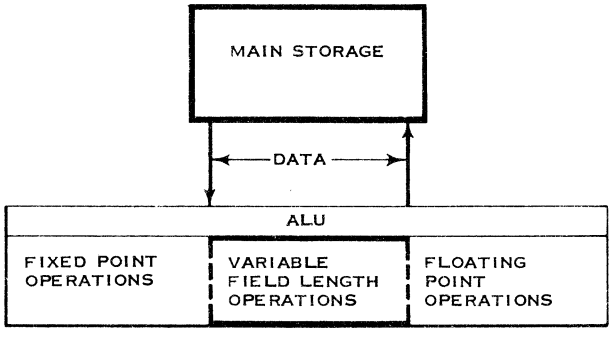
STORAGE-TO-STORAGE CONCEPT

Other computers such as those of the 702 - 705 family used a storage-to-accumulator concept. The accumulator was a small storage device. The storage medium could be core storage, vacuum tube 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.



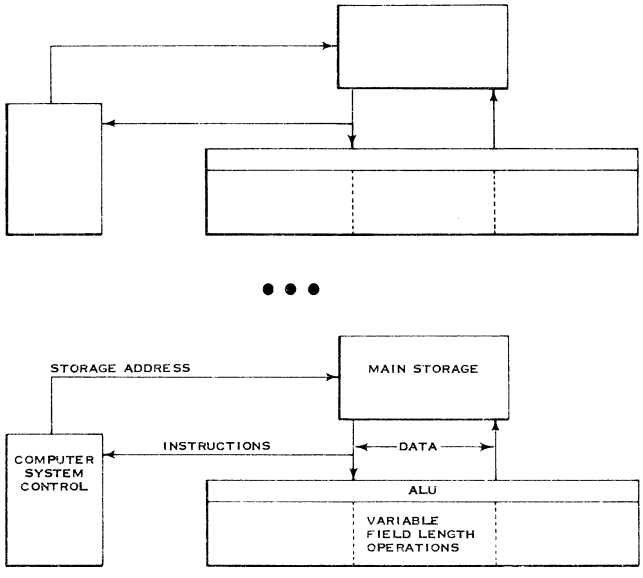
STORAGE-TO-ACCUMULATOR CONCEPT

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



1. Sections of the System/360 necessary for a variable length operation, including I time, are shown in this frame. Label the blocks as to:
   - Control Section
   - Main Storage
   - ALU
   - Variable Field Length Operations

   On the lines connecting the blocks, indicate whether they are:
   - Addresses
   - Instructions
   - Data



Fields of data (fixed or variable length) are often referred to as operands.

2. Instructions usually contain an Op code, the address of a first _____ and the address of a second _____.

• • •

operand; operand

As you have previously learned, variable length fields can start at any byte location in main storage. They are not restricted by storage boundaries as are fixed length operands. However, there must be some way of indicating to the system the length of the fields. System/360 specifies the length of these fields by a length code in the instruction.

3. 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 instruction

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 length can be either 16 or 256 bytes. The value of the code is one less than the total number of bytes.

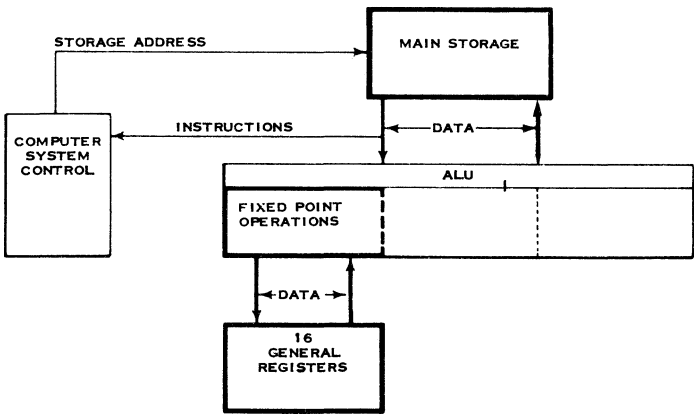
Length code of 0000 = 1 Byte
Length code of 1111 = 16 Bytes
Length code of 11111111 = 256 Bytes

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

• • •

Eight (8)

FIXED LENGTH OPERATIONS



• • •

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

• • •

16; general registers

2. When working with fixed length operations, the System/360 uses a s_____ -to-r_____ concept.

• • •

storage; register

3. 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

It is important to note that there are certain restrictions on the use of general registers. For example, the Disk Operating System uses registers 0, 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.

4. To use general register 2 as an accumulator, what address is given? _____

• • •

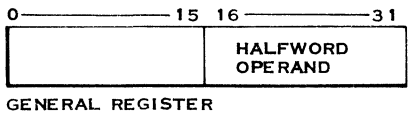
2

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

• • •

four

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.

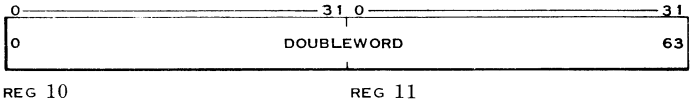
None of the general registers 0-15 can contain a doubleword. For those operations that use a doubleword operand, such as fixed length divide, a pair of adjacent registers is used. In these cases, an even-odd pair of registers (such as 2-3 or 6-7) is used, and the even register is addressed.

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

• • •

10 and 11

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

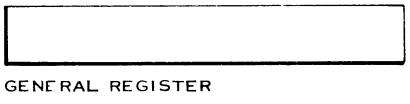


7. Fixed length operands in main storage must be on specific boundaries or a program c_____ will occur indicating a s_____ e_____.

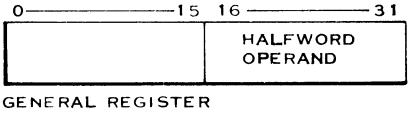
• • •

check; specification exception

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

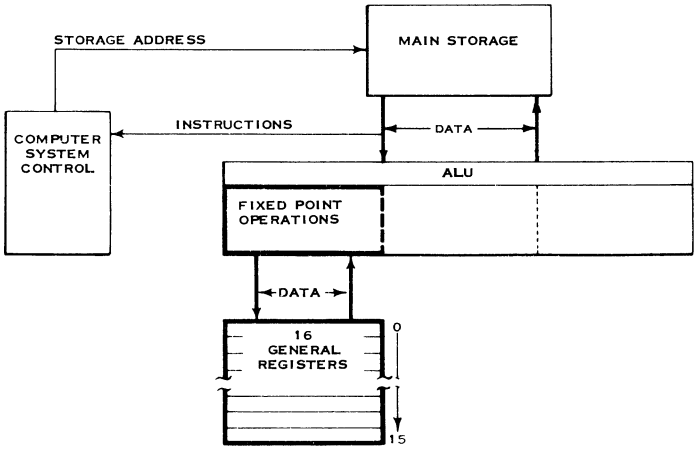


• • •



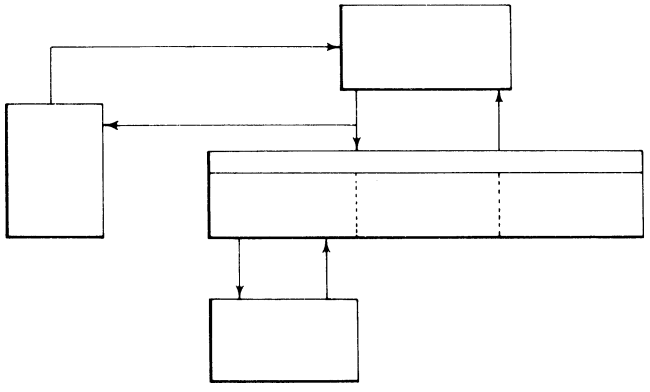
With sixteen general registers, sometimes both fixed length binary operands will be in the general registers. In these cases, another data flow concept is used. The System/360 can do a register-to-register (accumulator-to-accumulator) operation.

Incidentally, "fixed length (binary) operations" include both "fixed point", and "floating point" operations. You will be introduced to the latter shortly.

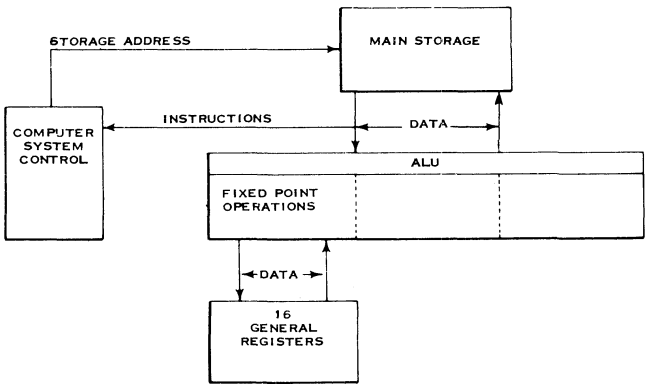


9. Sections of the System/360 necessary for fixed length operations, are shown in this frame. Label the blocks as to:
   - Control Section
   - ALU
   - Main Storage
   - Fixed Point Operations
   - General Registers

   On the lines connecting the blocks indicate whether they are:
   - Addresses
   - Instructions
   - Data



• • •



The general registers are also used for purposes other than accumulating. Two other main uses are as Index Registers and Base Registers. Indexing is a form of indirect addressing. An increment contained in an index register is added to the data address in the instruction to form an effective main storage address. Neither the index register nor the instruction in storage is changed by indexing. The use of the general registers as index and base registers will be explained later in this course. Base registers are similar to index registers.

10. List three main uses of the general registers.
    1._____ 2._____ 3._____

• • •

1. Accumulators
2. Index Registers
3. Base Registers

FLOATING POINT OPERATION

The floating point feature is not an objective of this course. Some information however, is necessary to acquaint you with the term "floating point". Floating point is the term given to arithmetic operations involving a fraction and an exponent. For instance:

217,000 can be expressed as $.217 \times 10^6$

296,000 can be expressed as $.296 \times 10^6$

Fixed point arithmetic would add the numbers as follows:

$$\begin{array}{r} 217,000 \\ +\ 296,000 \\ \hline 513,000 \end{array}$$

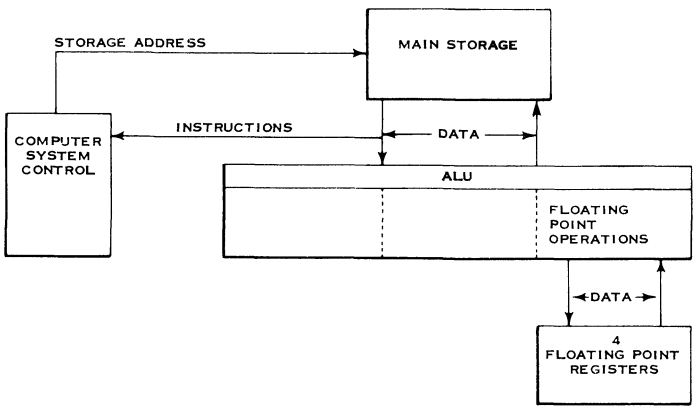
Floating point arithmetic would do it like this:

$$\begin{array}{r} .217 \times 10^6 \\ +\ .296 \times 10^6 \\ \hline .513 \times 10^6 \end{array}$$

Add fraction Retain exponent

The example shown is an example of decimal floating point. The System/360 uses hexadecimal floating point. For instance:

$$\begin{array}{r} .7F \times 16^6 \\ +\ .1F \times 16^6 \\ \hline .9E \times 16^6 \end{array}$$



• • •

1. Floating point arithmetic is most useful for expressing very large numbers and operating on them with much precision. To do floating point arithmetic, the System/360 has_____floating point registers.

• • •

four

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

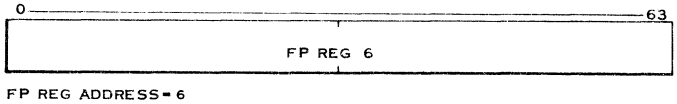
2. There are _____floating point registers numbered ____, ____, ____, ____. The floating point registers are not the same as (in your own words)

_____.

• • •

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

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



3. Floating point registers are_____bits long and can contain a_____.

• • •

64; doubleword

CHANNELS

Learning Objectives

When you complete the following test, you will have demonstrated that you can:

- Describe, in detail, the logical relationship of I/O devices, channels and main storage.
- Identify the modes of operation of each type of channel.

SELF-EVALUATION QUESTIONS — Reference Pages in Text

1. All data flow between I/O devices and the main storage passes through the _____. — 27

2. The channel receives ____ ____ of data at a time from an I/O device. — 27

3. The requesting of a storage cycle by the channel is known as _____. — 27

4. The simultaneous operation of an I/O device on the channel and the processing of instructions in the CPU is known as _____. — 27

5. The I/O device communicates with its channel via a _____ _____ cable. — 27

6. The I/O device logically ties into the standard interface through a ____ ____. — 27

7. The mode of operation in which only one I/O device is functional on the channel, for transmission of the entire record, is known as _____ _____. — 29

8. Each channel has its own _____ _____ cable. — 29

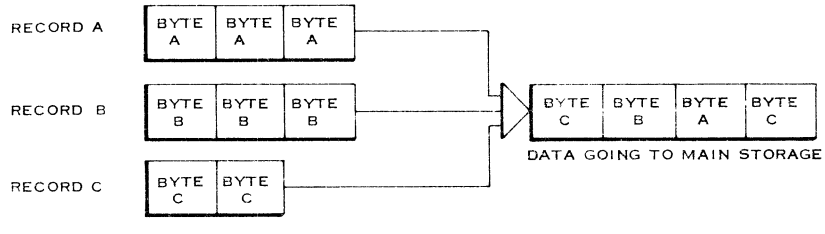
9. _____ channels are designed to operate at high data rates and can operate only in _____ mode. — 29

10. A _____ channel is designed to operate with a number of I/O devices simultaneously. — 30

11. The control information necessary for each I/O device in operation on a multiplexor channel is contained in _____ _____. — 31

12. Multiplexor channels can operate in two modes: _____ _____ and _____ _____. — 31 and 32

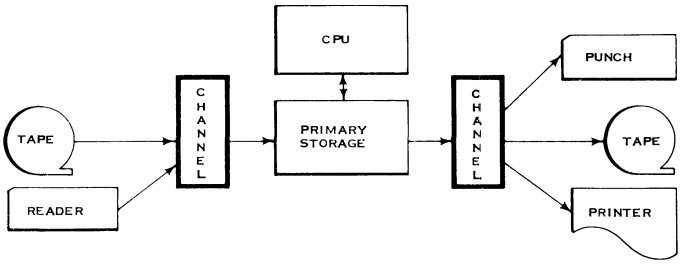
13. The following illustration shows a _____ mode operation. — 30



ANSWERS	Reference Pages in Text
1. channels	27
2. one byte	27
3. interference	27
4. overlap	27
5. standard interface	27
6. control unit	27
7. burst mode	29
8. standard interface	29
9. Selector, burst	29
10. multiplexor	30
11. MPX storage (multiplexor storage, "bump" storage)	31
12. burst mode, multiplex mode	31 and 32
13. multiplex	30

GO TO SECTION III - PROGRAM CONTROL AND EXECUTION, on page 33.

CHANNELS



From your knowledge of basic computer systems principles, you should realize the importance of input-output channels in any computer system. Their main function is to act as an intermediary between the I/O devices and the main storage unit.

• • •

1. Before input data can be processed in the ALU, it must first reside in _____ _____ .

• • •

main storage

2. Before processed data can be sent to an output device, it must be placed in _____ _____ .

• • •

main storage

3. All data flow between I/O devices and the main storage passes through the _____ .

• • •

channels

One of the main functions of a channel is to handle I/O requests for a main storage cycle. The channel receives data from the System/360 I/O devices one byte at a time. When enough data has been received to justify the use of main storage, the channel will request a storage cycle. The amount of data required will vary from one to eight bytes depending on the particular model of System/360. After the data has been placed in main storage the channel will wait for additional information from the input device. For an output device the procedure reverses. The channel requests a main storage cycle and brings out data. It passes this data to the output device one byte at a time. The requesting of a main storage cycle by the channel for I/O data is commonly referred to as "interference".

4. The channel receives _____ _____ of data at a time from an I/O device. Requesting of a main storage cycle by the channel is known as "_____".

• • •

one byte; "interference"

Since the channel is taking care of main storage cycles for the I/O device, the central processing unit now is logically free to continue processing instructions. We say that processing is "overlapped" with the I/O operation.

5. This simultaneous operation of an I/O device and the processing of instructions is known as _____ .

• • •

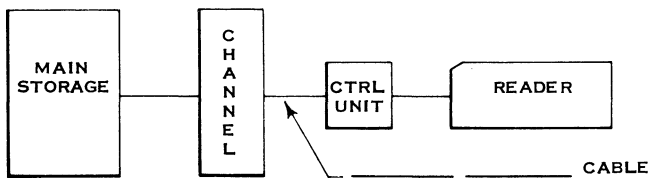
overlap

On some models of the System/360, overlapping the channel with CPU operations is not allowed at certain times. Once the CPU has started a channel operation, it has to wait for the channel operation to finish before it can continue processing instructions.

All data and control information are communicated between the System/360 channel and its I/O devices via a Standard Interface cable. More on this later!

Each I/O device logically ties into the System/360 channel's Standard Interface through a control unit.

6.



Fill in the blanks in the illustration.

• • •

standard interface

7. The I/O device logically ties into the channel's Standard Interface through a _____ _____ .

• • •

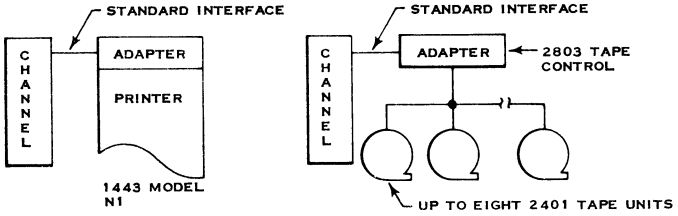
control unit

Another name for a control unit is adapter. For some I/O devices, the control unit or adapter is built into the device. For other devices, the control unit is external to the device.

8. The control unit, or _____, may be housed in the_____ or may be external to it.

• • •

adapter; device



Some adapters can control only one I/O device while others can control a number of similar I/O devices. The 1443 Printer Model N1 is an example of an I/O device with a self-contained adapter which controls only one printer. The 2803 tape control is an example of a stand-alone adapter which can control up to eight 2401 magnetic tape units.

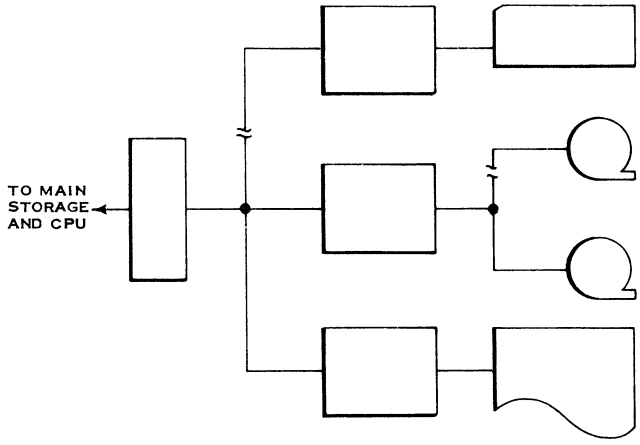
Each channel of the System/360 has the ability to select up to 256 I/O devices. There are physical limitations, of course. One of these is in the standard interface between the channel and the I/O device. There can only be up to eight control units tied into the standard interface cable.

9. The communication lines between the channel and its I/O devices are known as the _____ _____. The maximum number of control units that can tie into the standard interface cable is _____.

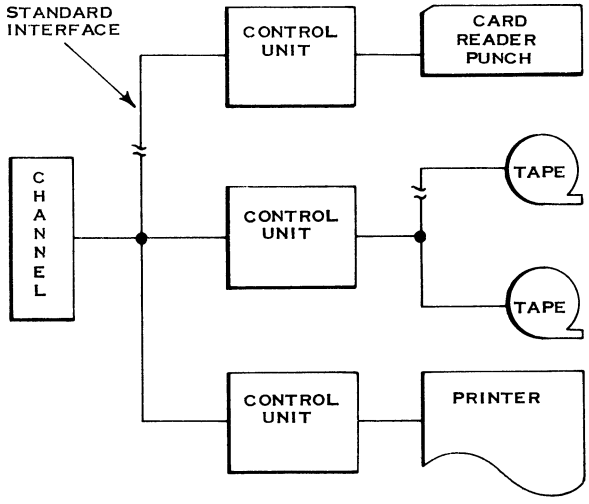
• • •

standard interface; eight

10. Label each block of the channel organization shown below as to channel, control unit, or type of device. Indicate which line is the standard interface.



• • •



Channels are logical concepts in I/O operations. In the System/360, they may be stand-alone units as in the model 75 or may be packaged along with main storage in the CPU housing as in the model 30. In the lower models of the System/360, many of the processing units' circuits are used by the channels for their functions. There are two types of channels used by System/360: 1) selector channels, and 2) multiplexor channels. Let's discuss the selector channels first.

SELECTOR CHANNELS

Selector channels are available on all models of the System/360. The maximum number per model varies from two for a model 30 to six for a model 75. The selector channel is so named because only one I/O device can be selected on the channel at any one time. Once selected, a complete record is transferred over the standard interface one byte at a time.

• • •

1. On a selector channel, only one I/O device can be _____ at a time. Once selected, a _____ _____ is transferred over the_____ _____one byte at a time.

• • •

selected; complete record; standard interface

Once the record has been transferred, the channel is free to select another I/O device. When a channel is transferring an entire record between main storage and an I/O device, it is said to be operating in "Burst Mode". Since a selector channel always transfers an entire record, it can only operate in burst mode.

2. The operation of a channel with only one I/O device for the entire record is known as_____ _____. Burst mode is the only mode in which a _____ channel can operate.

• • •

burst mode; selector

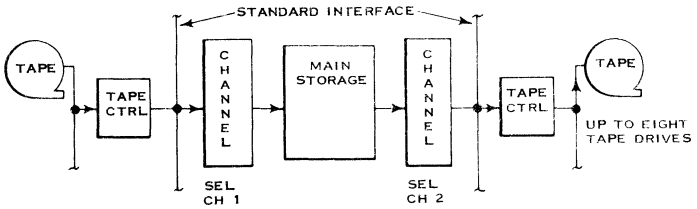
In summary then, on a selector channel, one I/O device transfers an entire record over the channel's standard interface. During this time no other I/O device can be using the channel. The other I/O devices could, however, be in the process of a feed cycle involving no data transfers over the channel. This is most apparent in those I/O devices which have buffers in their control units.

3. On a selector channel, one I/O device transfers an _____ _____ over the channel. This mode of channel operation is known as _____ _____ . During this period of time no other I/O device can be using the _____ .

• • •

entire record; burst mode; channel

Although only one I/O device can be operating on a selector channel at any one time, multiple selector channels can be operating simultaneously. The following illustration shows an input record being read in from tape over selector channel 1 at the same time as an output record is being transferred over selector channel 2. All channels have their own individual standard interface cable.



4. Each channel has its own _____ _____ _____ . All channels can be in operation (your own words) _____ .

• • •

standard interface cable; simultaneously

Selector channels are designed to operate with high data rates. I/O devices such as magnetic tape, disk units, drums, and buffered card devices are the devices most likely to operate on a selector channel. For operating with communication terminals in a real time application and with low data rate devices like an unbuffered card punch unit, a multiplexor channel is used.

5. I/O devices that operate at high data rates usually use_____ channels which can operate only in _____ mode.

• • •

selector; burst

6. For operation with low-speed or real-time I/O devices, a _____ channel is available on System/360.

• • •

multiplexor

MULTIPLEXOR CHANNELS

A Selector channel is designed to operate with only one I/O device at a time on an entire record basis. A Multiplexor channel is designed to operate with a number of I/O devices simultaneously on a byte basis. That is, several I/O devices can be transferring records over the multiplexor channel, time-sharing it on a byte basis.

• • •

1. A number of I/O devices can be operated simultaneously with a _____ channel. The I/O devices time-share the multiplexor channel on a _____ basis.

• • •

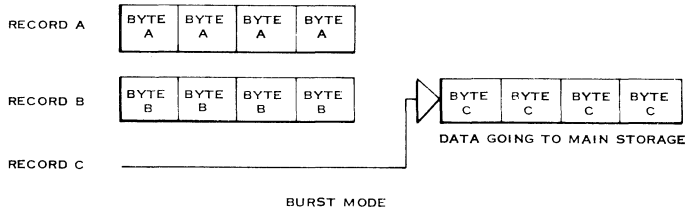
multiplexor; byte

2. This time-sharing mode of operation is known as "Multiplex" mode. Selector channels can only operate in _____ mode. Multiplexor channels can operate in _____ mode.

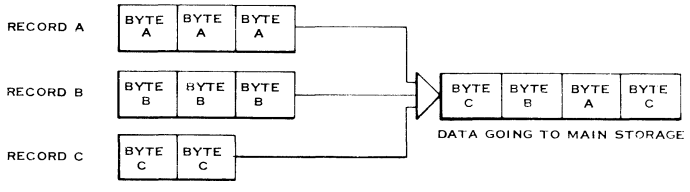
• • •

burst; multiplex

A comparison of burst versus multiplex mode can be seen below.



BURST MODE



MULTIPLEX MODE

To handle data flow from an I/O device, the channel needs to know certain information such as:

- In which direction does data flow (input versus output)?
- Where in main storage should data be placed or taken out of?
- How many bytes should be sent to an output device or accepted from an input device?

Information of this type is contained in the I/O command addressed to a particular I/O device. For a selector channel, which operates with only one I/O device at a time, the information may be placed in the channel registers and left there to control the operation.

3. Information necessary to control a selector channel operation is contained in the channel _____. This information was contained in the I/O _____ addressed to a particular I/O device.

• • •

registers; command

On a multiplexor channel, it is possible to have up to 256 I/O devices operating simultaneously. The actual maximum number varies with the particular model and main storage of the System/360. In any case, it is not feasible to have all this information sitting in the multiplexor channel's registers. A set of registers would be necessary for each I/O device. Instead, the multiplexor channel keeps this information in a compact storage area. As a byte of data comes in from a particular I/O device, the multiplexor channel brings the necessary information out of the compact storage area and places it in its registers. After the byte of data from the I/O device has been serviced, the information in the registers is automatically put back into the compact storage area.

4. A number of I/O devices can operate on a _____ channel simultaneously. The control information necessary for each I/O device is kept in a compact _____ _____.

• • •

multiplexor; storage area

5. As a byte of data comes in from an I/O device, the control information is brought out and placed in the channel _____.

• • •

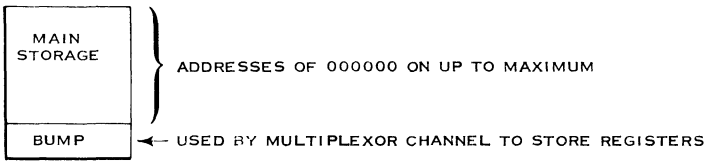
registers

6. After the byte of data has been serviced, the control information is placed back into the compact _____ _____.

• • •

storage area

The compact storage area used by the multiplexor channel is sometimes called "bump" storage because it is physically part of the main storage core array unit. The correct name of this storage area is Multiplexor Storage (abbreviated MPX Storage).



7. As can be seen above, the "bump" does not use any of the main storage addresses. It is a physical part of the core array used by the main storage unit. However, logically it is separate from it and has separate addressing lines. On the model 30, part of the "bump" is also used to contain the sixteen general registers. The remaining portion is used by the multiplexor channel and is called _____ storage.

• • •

multiplexor or MPX

8. The control information necessary for each I/O device on a multiplexor channel is contained in _____ _____. The control information in MPX storage comes from the original I/O c_____ addressed to a particular I/O device.

• • •

MPX storage; command

9. Each I/O device has an area in _____ _____ to contain its control information from the original I/O _____.

• • •

MPX storage; command

10. MPX storage does not use any main storage (your own words) _____ _____.

• • •

addresses or available area or equivalent

11. On the model 30, the "bump" contains more than the MPX storage. It also contains the 16 _____ _____.

• • •

general registers

Each I/O device has an area of MPX storage for its own individual use when operating on a multiplexor channel. In effect then, a multiplexor channel is comprised of a number of subchannels. Each subchannel has its own area of MPX storage. All subchannels (I/O devices) can be transferring records simultaneously. However, the multiplexor channel registers can be used with only one subchannel at a time. When the subchannel has finished servicing a byte of data for its I/O device, its control information is placed back into its area of MPX storage. The multiplexor channel registers are now free to be used by another (or possibly the same) subchannel.

12. The multiplexor channel can be said to be a number of _____. The multiplexor channel registers can contain at any one time the control information for only one _____.

• • •

subchannels; subchannel

13. When not being used to service data bytes, the subchannel information is contained in _____ _____.

• • •

MPX storage

14. Operating several I/O devices simultaneously on a multiplexor channel and servicing their data bytes as needed is known as _____ mode.

• • •

multiplex

15. _____ channels can operate with only one I/O device at a time. This mode of operation is called _____ _____ .

• • •

Selector; burst mode

Multiplexor channels can also operate in burst mode if necessary. Burst mode can be forced on the multiplexor channel by the I/O device. This is done if high-speed devices are placed on the multiplexor channel.

16. Multiplexor channels can operate in two modes: _____ _____ and _____ _____ .

• • •

multiplex mode; burst mode

17. The normal mode of operation for a multiplexor channel is _____ _____ . Burst mode can be forced on a multiplexor channel by the _____ _____ .

• • •

multiplex mode; I/O device

SECTION III - PROGRAM CONTROL AND EXECUTION

One concept basic to the design of System/360 is the use of a Supervisor program to control execution of problem programs. An efficient device, aiding the Supervisor in this task, is the use of a Program Status Word.

Among other things, the Program Status Word carries information on the condition of the CPU and the identification of the instruction being executed.

The activity of the CPU depends on the format of the instruction it is executing.

The format of an instruction, that we choose to perform a particular operation on some data, depends on the format of the data.

Thus data formats, instruction formats, and the contents and purpose of the Program Status Word are all linked together in program control and execution.

This section reviews details, about these topics, that are relevant to Assembler Language Coding.

INSTRUCTION FORMATS

Learning Objectives

When you complete the following test, you will have demonstrated that you can:

- Describe the restrictions on the placement of instructions in storage.
- State the length of each type of instruction.
- Describe the format of each type of instruction.
- State that the Op code specifies
  a. Instruction Length
  b. Format
  c. Specific Instruction
- Describe how the instruction specifies the address of a general register.
- Describe how an effective main storage address is generated using a displacement field and a base address located in a general register.
- Describe how programs can be relocated by updating the base register contents.
- Describe how an effective address may be further indexed with an index factor in a general register.
- Given a mnemonic Op code, determine the location of 1st and 2nd operands.
- State where, with the exception of store-type operations, the result of the operation is located.

SELF-EVALUATION QUESTIONS

Reference Pages in Text

NOTE: Use your Reference Data Card.

1. Instructions are a multiple of _____ in length. 38

2. Instruction addresses must be divisible by _____ or a _____ exception will occur. 38

3. The first byte of every instruction is the _____. 38

4. State in your own words the types of information specified by an Op code. _____ 39

5. For the following mnemonic Op codes, indicate the instruction type and its length in halfwords. 39

	Mnemonic	Type	Length
a.	AR	_____	_____
b.	O	_____	_____
c.	TM	_____	_____
d.	DP	_____	_____

6. All effective storage addresses are generated by adding the instruction's 12-bit _____ to a 24-bit _____ _____ in one of the general registers. 40

7. Some effective storage addresses are generated by also including an _____ factor in one of the general registers. 43

8. Address generation _____ (does/does not) change the contents of the general registers or the instruction in storage. 41

9. A program can be relocated in storage by changing the contents of the _____ _____. 41

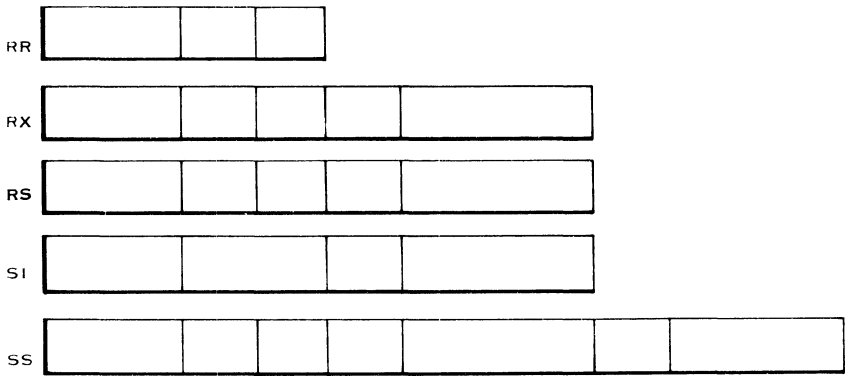
10. The displacement has a range of 0 to _____ bytes. 40

11. Only general registers _____ through _____ can be used as base or index registers. 43

12. What happens if register 0 is specified as a base or index register? _____ 42

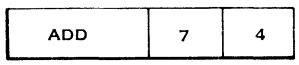
Reference
Pages in Text

13. Label the fields of the following formats: 45



14. For most operations, the results replace the _____(1st/2nd) operand. 45

15. Given the following RR type instruction: 45



The result of the addition will replace the contents of register_____.

16. In the SI format, the 2nd operand is located in the _____ and is one _____ long. 47

17. Only the _____ _____format uses an index register for address generation. 48

18. Only the_____ _____format involves variable length data. 48

19. What is the relationship between the number in the length code of the SS format and the number of bytes in the referenced data field? _____ 49

_____

ANSWERS TO REVIEW QUESTIONS

		Reference Pages in Text
1.	halfwords	38
2.	two, specification	38
3.	Op code	38
4.	a. the operation (such as add, subtract, load, etc.) which is to be performed with the data, b. the data format, whether the data is variable or fixed in length and, if fixed, whether the operands are halfwords or fullwords, c. The location of data, d. the length of the instruction.	39

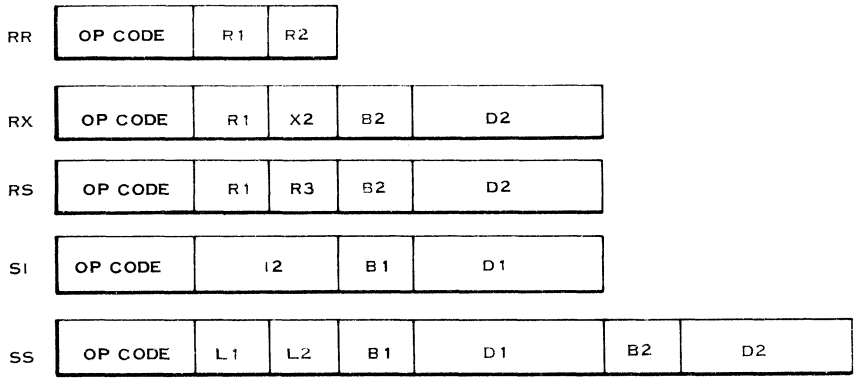
5.

	Mnemonic	Type	Length
a.	AR	RR	one halfword
b.	O	RX	two halfwords
c.	TM	SI	two halfwords
d.	DP	SS	three halfwords

39

6.	displacement, base address	40
7.	index	43
8.	does not	41
9.	base registers	41
10.	4095	40
11.	1, 15	43
12.	The contents of register 0 are ignored and a value of zero is used for the base or index factor.	42

Reference
Pages in Text

13. 

45

14. 1st 45

15. 7 45

16. instruction, byte 47

17. RX 48

18. SS 48

19. The number of bytes in the data field is one greater than the number in the length code. 49

GO TO THE NEXT PART OF THIS SECTION "DATA FORMATS" on page 50.

INSTRUCTION LENGTH

Let's learn about the instruction formats of the System/360. As you know, instructions specify the operation to be done and the location of data. Data may be located either in main storage or in general registers or a combination of the two. Main storage is addressed with a 24-bit binary address while the general registers are addressed with a 4-bit binary address. As a result, instructions will be of different lengths depending on the location of data. System/360 instructions may be one, two, or three halfwords in length.

• • •

1. System/360 instructions are one, two, or three _____ in length depending on the location of data.

• • •

halfwords

2. When both operands or data are in general registers, only eight binary bits are needed for addresses. As a result, the System/360 can use the shortest instruction which is _____ _____ in length.

• • •

one halfword

3. One halfword cannot contain an 8 bit op code, a 4-bit register address and a storage address. So when one of the operands is in main storage (storage to register concept), the next larger size instruction is used. This would be _____ halfwords in length.

• • •

two

4. When both operands are in storage (storage to storage concept), the maximum size instruction must be used. This would be _____ _____ in length.

• • •

three halfwords

5. For the following processing concepts, fill in the necessary System/360 instruction length.

Concept	Instruction Length
Storage to Storage	_____
Register to Register	_____
Storage to Register	_____

• • •

Instruction Length

three halfwords; one halfword; two halfwords

Since instructions are a multiple of halfwords in length, they are considered as fixed length information as far as storage boundaries are concerned.

6. Instructions are a multiple of _____ in length. As a result, instruction addresses must be divisible by two or a _____ exception will occur.

• • •

halfwords; specification

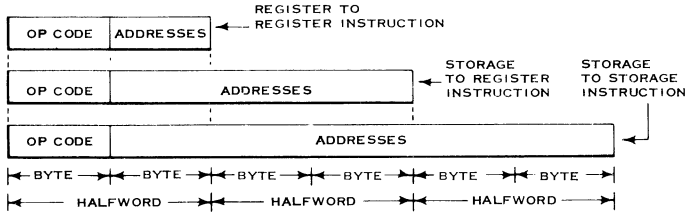
7. If the address of an instruction has a low-order 1 bit, a _____ exception will occur.

• • •

specification

OP CODE

In discussing data formats earlier, you had been told that the Op code portion of the instruction would specify whether the data was binary or decimal in nature. We are now at the point where we should find out all about the Op code.



• • •

1. Shown above are the three instruction lengths which are used depending on the location of data. One thing to notice is that the 1st or high-order byte of each instruction contains the _____ _____.

• • •

Op code

2. All instructions have an Op code which is contained in the high-order _____.

• • •

byte

Op codes in the System/360 give specific information:

- The Op code specifies the operation (such as add, subtract, or branch).
- The Op code specifies data format - whether the data is variable or fixed in length, and, if fixed, whether the data is binary or decimal.
- The Op code specifies the general location of the data - whether the operands are in main storage or general registers. The Op code, however, does not give the address of data.
- The Op code specifies the length of the instruction of which it is a part.

To repeat, the Op code specifies the operation, the data format, the general location of the data, and the instruction length. Reread this statement, until you can repeat its information from memory.

Each type of instruction that has been mentioned so far has been given a "type abbreviation" that shows where the operands are and therefore specifies the length.

a. Type RR indicates that each of the operands is in a register.
b. Type RS indicates that one of the operands is in a register and the other is in storage.
c. Type SS indicates that each of the operands is in main storage.

Every instruction listed on your reference card has a mnemonic (easier to remember than the hex equivalent of an eight-bit Op code) and a type abbreviation.

3. Find the following instructions and state their type and length:

Name	Mnemonic	Type	Length
Compare Logical	CLC	_____	_____
Divide	DR	_____	_____
Load Multiple	LM	_____	_____

• • •

Name	Mnemonic	Type	Length
Compare Logical	CLC	SS	three halfwords
Divide	DR	RR	one halfword
Load Multiple	LM	RS	two halfwords

OPERAND ADDRESSING

At this point, you should have a good idea that an Op code of an instruction specifies its length, the type of data, and what to do with the data. Before taking a further look at instruction format, let's examine how to address main storage and the general registers. We'll take a look at general register addressing first because of its simplicity. But first answer the following review questions.

• • •

1. For use as accumulators when working with fixed length binary operands, the programmer has available _____ general registers. They are numbered _____ through _____ .

• • •

sixteen; 0; 15

2. Each general register is one _____ in length.

• • •

word

3. Besides being used as accumulators, the general registers can also be used as base registers and i_____ registers.

• • •

index

4. The programmer specifies the general registers to be used by their decimal number (0-15). In machine language, however, the general registers are addressed by the 4-bit binary equivalent of their decimal numbers. The address of general register 7 is _____.

• • •

0111

5. When both operands are in general registers, the instruction is _____ _____ in length.

• • •

one halfword

6. The first byte of an instruction is the _____ _____.

• • •

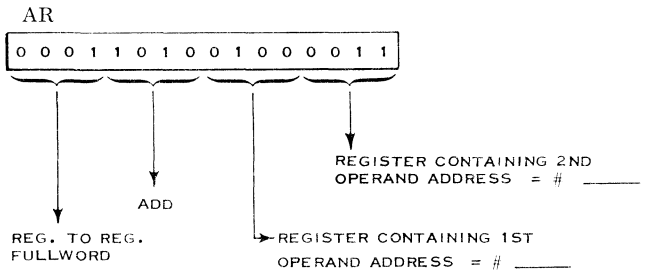
Op code

7. When an instruction is one halfword in length, the 2nd byte will contain the a _____ of the two general registers.

• • •

addresses

8. Given the following instruction, state the numbers of the general registers involved.



• • •

1st operand register 4 ; 2nd operand register 3 ; (AR 4,3)

Main storage addressing is a little more difficult. To use a 24-bit address in the instruction for each operand would consume storage space that could be used for other purposes. One solution would be to use 24-bit addresses on the larger models such as model 75 and to use shorter addresses on the smaller models. This would mean that programs used on the various System/360 models would no longer be compatible because of the different length addresses. So we must look for another solution that will reduce the length of the instructions and still maintain complete compatibility.

There are other features desirable in main storage addressing besides a simple reduction in the length of instruction.

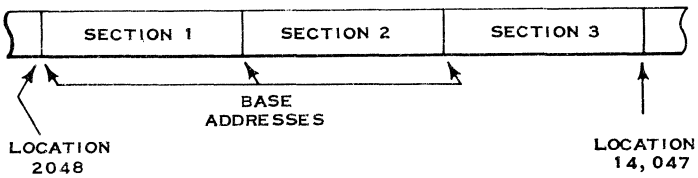
It is also desirable that any time the program is loaded into the computer, the program can be put in a different area of main storage. We would like to do this without having to change the addresses in each instruction. This is known as program relocation, which is a valuable tool in IBM's latest programming systems.

Besides the features of program relocation and shorter instructions, it is also desirable to be able to index instructions.

To see how main storage is addressed in the System/360, we must make some assumptions.

The first assumption is that System/360 programs will be written in sections. Each section will be 4096 bytes in length. Of course, programs that are less than 4096 bytes can be written as one section. The beginning of each section is called the Base Address for that section.

Consider the case of a program that required 12,000 bytes. By sectioning it into 4096 byte groups, we would have three sections of our program with a base address for each. The program can start anywhere. In the example shown below, the program starts at byte location 2048.



SECTIONALIZED PROGRAM IN MAIN STORAGE

As can be seen in the above example, our 12,000 byte program starts at location 2048 and runs through location 14,047. We have divided the program into three sections. The first two sections are 4096 bytes each, while the remainder of the program (the last 3808 bytes) is in section 3.

9. The 1st location of each section of a program is called its _____ _____ .

• • •

base address

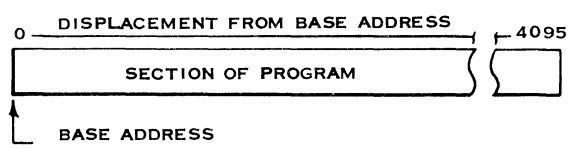
10. In the preceding example, the base address of the 1st section is location _____ . The 1st section is 4096 bytes long. As a result, the base address of the 2nd section is location _____ . Location 10,240 is the _____ address of the 3rd section.

• • •

2048; 6144; base

Now that the program has been sectionalized and base addresses are known, how can this help in addressing main storage?

Since each section is a maximum of 4096 bytes long, any byte in a section can be located by adding to the Base Address a number in the range of 0 to 4095. This number is called its Displacement. That is, each byte is displaced from the base address by 0 to 4095 places.



11. Assuming a base address of 2048, the displacement for location 3170 is _____.

• • •

1122

12. Assuming a base address of 6144, the displacement for location 9170 is_____.

• • •

3026

13. Assuming a base address of 6144, the displacement for location 6144 is_____.

• • •

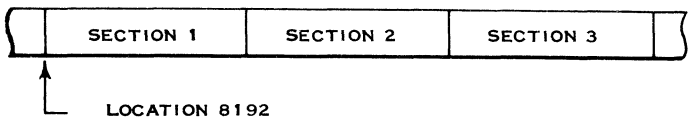
zero

14. Any byte in a program can be located by adding its _____ to its _____ _____.

• • •

displacement; base address

Supposing that the program that we've been using as an example was moved so that it started at location 8192.



15. The base address for Section 1 is now 8192. The base addresses for Section 2 and 3 are now _____ and _____.

• • •

12,288; 16,384

16. The displacement for each byte in the program _____ (has/has not) changed.

• • •

has not; The last byte of section 1 is still displaced from its base address by 4095.

17. The preceding frames demonstrate the ease with which a System/360 program can be relocated. To relocate a System/360 program, the _____ _____of all sections are changed, while the _____remain the same.

• • •

base addresses; displacements

18. As you know, main storage addresses are 24 bits long. This allows for compatibility throughout the range of System/360 as well as for addressing up to 16 million bytes. Since a program can start anywhere in main storage, this means that the base addresses for the program must be____ bits long.

• • •

24

19. The displacement range for any particular base address is 0 to 4095. To express this range will require_____ binary bits. (You can answer this by using the table in your Reference Data card. Convert 4095 to hexadecimal and then to binary.)

• • •

12 (4095 = FFF = 1 1 1 1 1 1 1 1 1 1 1 1)

20. Any byte in main storage can be located by adding a ____ bit displacement to a _____ bit base address.

• • •

12; 24

The use of a base address and a displacement certainly makes it easy to relocate a program each time it is loaded into the computer. However, we also wanted a shorter instruction. To put both the base address and displacement in the instruction would make the instruction longer. It would also mean that each instruction would have to be changed (base address) every time the program is relocated. The manner in which the System/360 handles this is to carry the base address in one of the General Registers. When a general register contains a 24-bit base address, it is referred to as a Base Register. The address of the base register and the 12-bit displacement are carried in the instruction.

21. To obtain a main storage address, the 12-bit _____carried in the instruction is added to the 24-bit _____ _____ in the base register.

• • •

displacement; base address

Let's take a look at a typical instruction used to add one operand in main storage to another operand in main storage.

22. When both of the operands are in main storage, the instruction is ____ (1/2/3) halfword(s) in length.

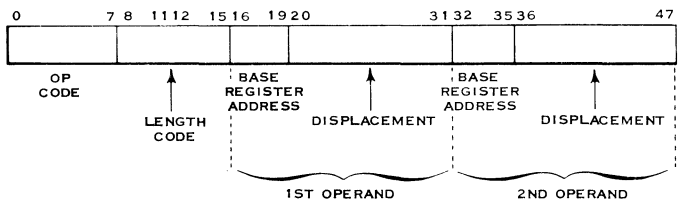
• • •

3

To add a main storage operand to another main storage operand, several items are necessary. They are:

- 8 bit Op Code
- 8 bit Length Code
- 4 bit 1st Operand's Base Register Address
- 12 bit 1st Operand's Displacement
- 4 bit 2nd Operand's Base Register Address
- 12 bit 2nd Operand's Displacement

The instruction format for this type operation would look like this:



Bits 8 - 15 of this instruction are used for specifying the length of the data field. We will ignore it for the present and cover it later.

23. The location of either main storage operand would be determined by adding its _____ in the instruction to the contents of the _____ _____ specified by the instruction.

• • •

displacement; base register

24. If bits 16 - 19 of the instruction contained 1011, the base address of the 1st operand is in general register _____.

• • •

11

25. If general register 11 contains the value of 2048, and the 1st operand displacement field in the instruction has the value 118, the effective storage address of the 1st operand would be _____.

• • •

2166

26. The base address of 2048 in the previous problem is a _____ bit binary address and appears in bits 8 - 31 of general register 11.

• • •

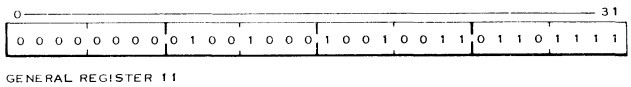
24

27. Only the low-order _____ bits of the base register are used in generating an effective storage address.

• • •

24

28. Given a displacement of 1 0 0 1 1 0 1 1 0 0 1 0 and base register 11 whose contents are shown below, the effective storage address would be _____. (Give the answer in decimal.)



• • •

4,758,817

If you missed the above, check your arithmetic. Remember that you were attempting to add the 12 binary bit displacement to the low-order 24 binary bits of the base register.

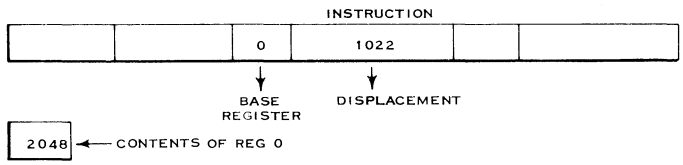
The address generated by adding the displacement and base address is used for addressing main storage. The original instruction and the contents of the base register remain unchanged.

29. If the displacement value is 1022 and the base register contains 2048, the effective storage address would be _____. After generating the address, the base register will contain _____ and the instruction will have a displacement field of _____.

• • •

3070; 2048; 1022

Only general registers 1 - 15 can be used as base registers. If general register 0 is specified as the base register, the base address is assumed to be zero, regardless of the contents of register 0.



30. Given the above address portion in the instruction and the contents of register 0, the effective storage address would be _____.

• • •

1022; Because register 0 was specified as the base register, a base address of 0 is used. The contents of register 0 are ignored.

In order to insure that you understand main storage addressing, let's look at a simple, typical application.

31. A certain small job requires 2,500 bytes of storage. These 2,500 bytes will contain the instructions, data read-in area, constants, work area and data output area. In other words, everything that pertains to this job is contained in the _____ bytes.

• • •

2,500

The program used to do this job consists of 500 instructions.

Program { Instruction #1
Instruction #2
↓
Instruction #500

32. It is decided to use main storage bytes 5000 through 7500 for the job. One of the program's first few i_____ will load the number 5000 into a _____ register.

• • •

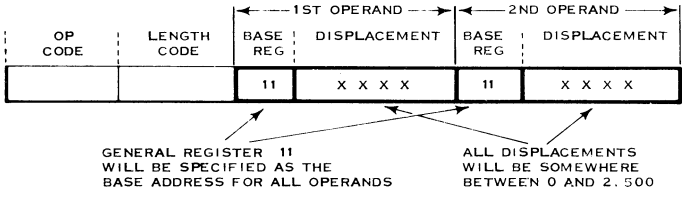
instructions; general

33. 5000 is the b_____ a_____ and we will select general register 11 as the base register.

• • •

base address

All instructions in this program will be alike in the following areas:



34. For the job in the preceding example, how many different base addresses were required?_____

• • •

one

35. The base address was loaded into a general register at the _____ (beginning/end) of the program.

• • •

beginning

36. If the application required 8,000 bytes of storage, _____ base addresses would have been needed.

• • •

two

37. The second base address would probably be used for the s_____ half of the program.

• • •

second

Let's summarize what you have learned so far about main storage addressing:

- Storage addresses are generated by adding a displacement value to a base address.
- The instruction contains the displacement value as well as the address of the general register containing the base address.
- The general register that contains the base address is called the base register.
- Only registers 1 - 15 can be used as base registers.
- If register 0 is specified as the base register, its contents are ignored. Instead, a base address of 0 is used.
- The generation of storage addresses does not change the instruction or the base register contents.

All storage addresses are generated by using base and displacement. In some instructions, however, a 3rd base factor is used. The 3rd factor is called the Index value. It is also contained in a general register.

The purpose of the index factor (indexed program) is to reduce the number of instructions in a program. This will be illustrated in just a moment.

38. A 3rd factor which is sometimes used in addressing main storage is called the_____ value. The index value is held in one of the_____ _____.

• • •

index; general registers

39. Just like the base value, the index value can only be in general registers _____ through _____.

• • •

1; 15

40. Just like the base value, if register 0 is specified, its contents are ignored and the index value is assumed to be _____.

• • •

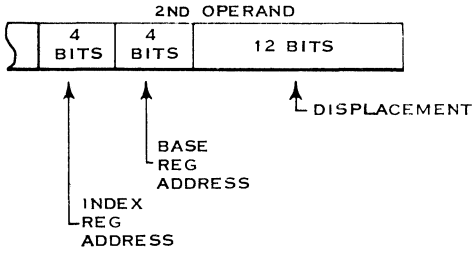
zero

41. Only registers ____ through ____ can be used as index registers and their contents remain unchanged by the address generation.

• • •

1; 15

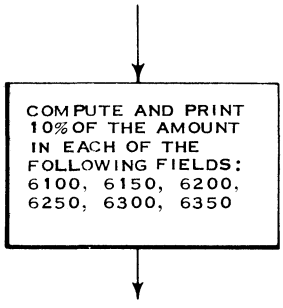
In those instructions that include an indexing factor, the address portion looks like this:



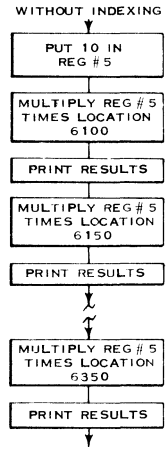
The effective storage address would be generated by adding:

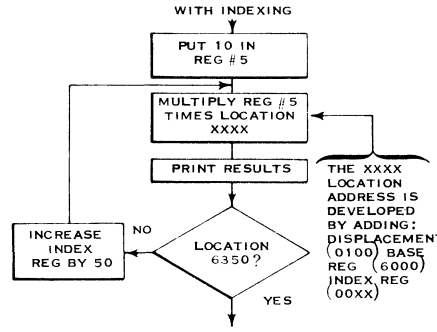
Displacement + Contents of Base Register + Contents of Index Register

The following illustration shows part of a theoretical program flowchart.



Let's see how the preceding program function would be accomplished with and without indexing.





42. An index register is a register whose contents are systematically altered. Since the contents of an index register are part of the effective address of an operand, this systematic alteration allows one instruction to operate on successive data fields. The index factor is used where a number of similar instructions can be replaced by one i_____ that has its operand address modified.

• • •

instruction

43. Given an address portion of

6	7	1012

2ND OPERAND

Reg 6 contents	2048
Reg 7 contents	6024

and the register contents shown, indicate the following:

(a) the effective storage address _____
(b) address portion of instruction after address generation _____
(c) base register contents after address generation _____
(d) index register contents after address generation _____

• • •

a. 9084; b. | 6 | 7 | 1012 | ; c. 6024; d. 2048

FORMAT TYPES

At this point, let's take a look at different types of instruction formats of the System/360. As you read this section, locate each of the format types at the bottom of page 2 on your Reference Data card. As you know, the instructions are of three lengths: one, two, or three halfwords depending on the location of the operands.

A one halfword instruction is used when both operands are in general registers. What is required is:

- An 8-bit Op code.
- A 4-bit register address for 1st operand.
- A 4-bit register address for 2nd operand.

Instructions that involve register-to-register operations are considered to be of the RR format.

RR FORMAT

• • •

1. An RR type instruction involves a _____ -to- _____ operation and is _____ _____ in length.

• • •

register-to-register; one halfword

2. The first byte of every instruction is the___ ____. Bits 0 and 1 of the Op code indicate the length of the instruction and the location of the operands.

• • •

Op code

3. The addresses of the two general registers are given in the 2nd byte of the __ __ format.

• • •

RR

4. The 2nd byte of the RR format is divided into two fields: R1 and R2. The R1 field gives the register address of the first operand while the ____ field is the address of the 2nd operand.

• • •

R2

The numbers in the address fields of the RR formats (and all other formats) indicate whether the operand is the 1st or 2nd (and is in some cases, the 3rd) operand.

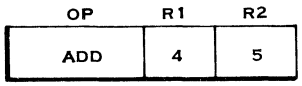
RR FORMAT

5. For most operations, the results replace the _____ (1st/2nd) operand.

• • •

1st

6. For the given instruction, the contents of registers 4 and 5 are added together and the sum goes into register_____.

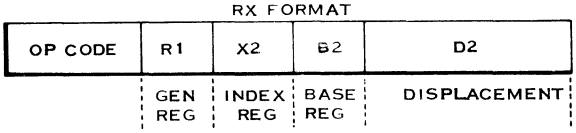


• • •

4

Instructions which are <u>two halfwords</u> in length may have three different formats. If bits 0 and 1 of the Op code are either 01 or 10, the instruction is two halfwords in length. Furthermore, it bits 0 and 1 of the Op code are 01, it indicates a specific format known as the **<u>RX format</u>**.

RX FORMAT



7. The__ __ <u>format</u> is used for storage-to-register operations. The register address is specified by the _____ field.

• • •

RX; R1

8. In the RX format, the effective address is generated by adding the contents of the base register and the _____ register and displacement.

• • •

index

9. When the effective storage address includes an indexing factor, the instruction is said to be the ___ ___ format.

• • •

RX

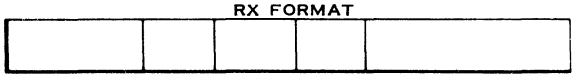
10. In the RX format, the index register is specified by the____ field while the base register is specified by the_____field.

• • •

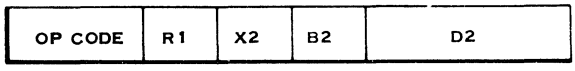
X2; B2

11. Fill in the correct names for the fields of the RX format.

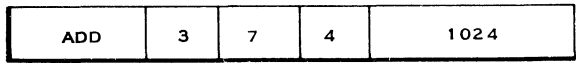
RX FORMAT



• • •



12.



For the above RX type instruction, the storage address is generated by adding the contents of registers____ and____ and the displacement value of _____.

• • •

7, 4; 1024

13. In the preceding instruction, the storage operand is added to the contents of register_____ and the sum is placed in register_____.

• • •

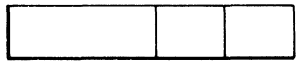
3; 3

14. Register-to-register operations use the ____ _____ format.

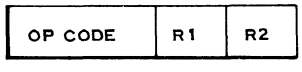
• • •

RR

15. Label the fields in the RR format.



• • •

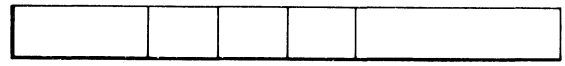


16. Storage-to-register operations, where the storage address includes an indexing factor, use the ___ ___ format.

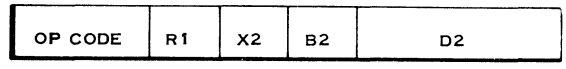
• • •

RX

17. Label the fields in the RX format.

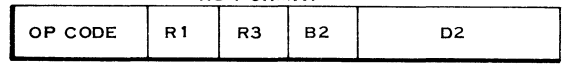


• • •



Storage-to-register instructions in which the storage address does not include an indexing factor are called the R S format. The 4 bits normally used for the X2 field are used for a 3rd operand.

RS FORMAT



18. In the__ __ format, the effective address of the 2nd operand is obtained by adding the contents of the _____ _____ to the _____.

• • •

RS; base register; displacement

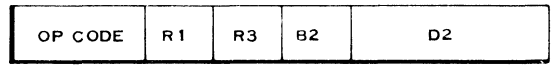
19. In the RS format, the 1st operand is specified by the _____ field while the third operand is specified by the _____ field.

• • •

R1; R3

20. Label the fields of the RS format.

• • •



The RS format is identified by a 10 in bits 0 and 1 of the Op code. The R3 field in the RS format specifies the general register used for the 3rd operand. In some RS instructions, the R3 field is ignored. An example of an instruction which uses the R3 field is an instruction called Load Multiple. In the load multiple instruction, the data in main storage is loaded (or placed) into the general registers. Loading begins with the register specified by the R1 field and continues consecutively until the register specified by the R3 field has been loaded.

For example:

LOAD MULTIPLE	4	7	0	0 1 0 0

21. In the preceding example, the effective storage address is_____.

• • •

0100; This is because register 0 is specified as the base register and its contents are ignored.

22. In the preceding example, registers____ through ____will be loaded with the data in main storage.

• • •

4; 7

23. Since each register can hold one fullword, registers 4 - 7 will be loaded with the data in storage location 0100 through_____.

• • •

0115; Each storage address represents a byte of data.

24. In the RS format, the effective storage address _____ (does/does not) include an indexing factor.

• • •

does not

25. Label the fields of the RS format

• • •

OP CODE	R1	R3	B2	D2

There is another instruction format that is two halfwords in length. It is called the SI Format. This format is used when one operand is in main storage and the other operand (called the immediate operand) is carried in the instruction itself.

SI FORMAT	BYTE		
OP CODE	I2	B1	D1

26. In the __ __ Format, one operand is in main storage while the immediate operand is in the_____.

• • •

SI; instruction

27. In the SI format, the storage operand is the_____ (1st/2nd) operand. Its effective address_____ (does/does not) include an indexing factor.

• • •

1st; does not

28. In the SI format, the immediate operand is fixed in length and is one_____ long.

• • •

byte

29. Since the results of instruction execution usually replace the 1st operand, an SI format instruction would change the operand in _____ (storage/the instruction).

• • •

storage

30. Label the fields of the SI format.

• • •

OP CODE	I2	B1	D1

An example of an SI format is an instruction called "Move Immediate". This instruction will move the immediate operand (I2) in the instruction to the storage location.

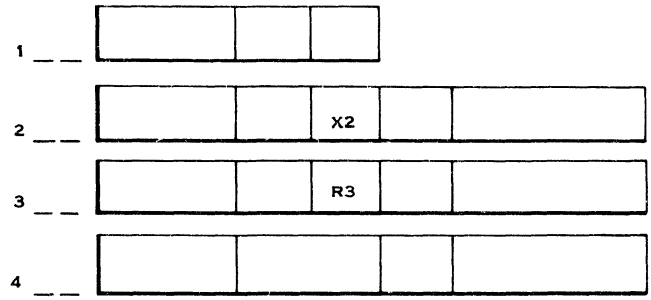
OP CODE	I2	B1	D1
MOVE IMMEDIATE	XX	0	1 0 0 0

31. In the previous instruction, the contents of the_____ field will be placed in storage location _____.

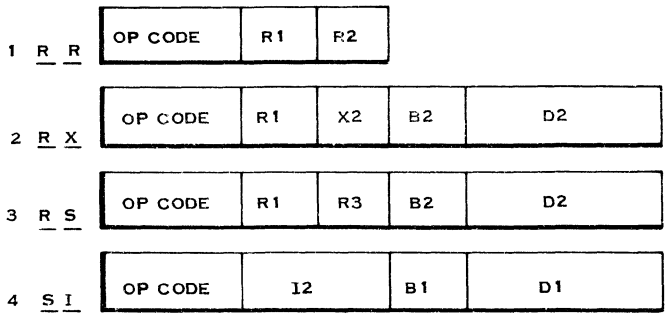
• • •

I2; 1000

32. So far you have learned four instruction formats. Fill in their names and their blocks.



• • •



33. Which format does not involve a storage operand? _____

• • •

RR

34. Which format does not involve a general register operand? _____

• • •

SI

35. Which format includes an indexing factor when addressing main storage? _____

• • •

RX

36. Did any of these formats include an operand length field? _____

• • •

no

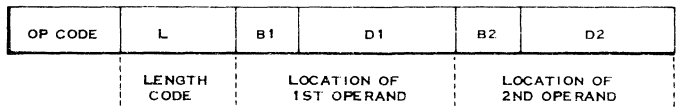
In the four previous formats, the operands were of fixed length. Now let's take a look at the instruction format for variable length operations.

37. As you should recall, variable length operation involves the_____ -to- _____ concept.

• • •

storage-to-storage

Variable length operation uses a storage-to-storage concept. The instruction format is called the SS Format and looks like this:



38. The _ _ format, because it must address two storage operands, is_____ (one/two/three) halfwords in length.

• • •

SS; three

39. In the SS format, an indexing factor_____ (is/is not) included in the generation of storage addresses.

• • •

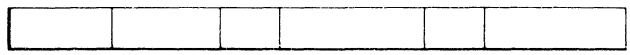
is not

40. The length code for the variable length storage operands is in the 2nd byte of the _ _ format.

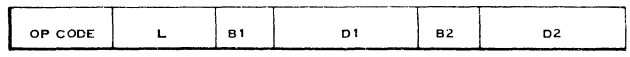
• • •

SS

41. Label the fields of the SS format.



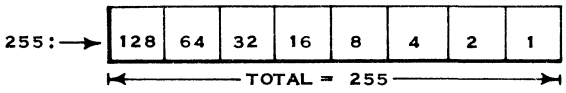
• • •



The 2nd byte of the SS format is the length code which consists of 8 binary bits.

42. The maximum value that can be expressed with 8 binary bits is _____.

• • •



Since all operands are at least 1 byte long, the length code is used to tell how many additional bytes are needed. For instance, a length code of 15 would tell us that the operand is 16 bytes long.

43. A length code of 33 would indicate an operand length of _____ bytes.

• • •

34

44. If an operand is 1 byte long, the length code would be _____.

• • •

zero

45. The maximum operand length that can be expressed by an 8-bit length code is _____ bytes.

• • •

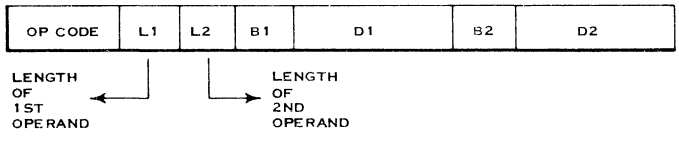
256

46. An operand that is a word in length would have a length code of _____.

• • •

3

So far, we have been treating the length code as one 8-bit binary number. However, we are dealing with two operands. Do they both have to be of the same length? The answer is no. It depends on the particular operation. If we are concerned with moving a data field from one area of storage to another, we only need one length code. If, however, we are adding one storage field to another, then we need to know the length of both operands. For arithmetic-type SS operations, the length code is split in two:

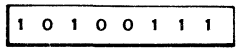


47. With the length code split into two 4-bit fields, the maximum length of arithmetic variable length operands is _____ bytes.

• • •

16; The length of variable length fields is one more than the length code.

48. Given the following binary length code:

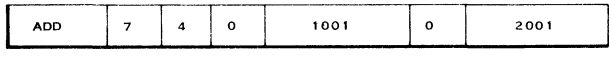


The 1st operand is _____ bytes long.
The 2nd operand is _____ bytes long.

• • •

11; 8

49. Given the following SS-type add instruction



This instruction would cause bytes 2001 through _____ to be added to 1001 through _____.

• • •

2005; 1008

50. At this point, you have learned the five instruction formats. List them.
1.__ __ 2.__ __ 3.__ __ 4.__ __ 5.__ __

• • •

1. RR; 2. RX; 3. RS; 4. SI; 5. SS

51. For the given instruction formats, specify
a. instruction length in halfwords.
b. location of 1st operand, such as storage or register.

	a.	b.
RR	____	_____
RX	____	_____
RS	____	_____
SI	____	_____
SS	____	_____

• • •

	a.	b.
RR	1	register
RX	2	register
RS	2	register
SI	2	storage
SS	3	storage

DATA FORMATS

Learning Objectives

When you complete the following test, you will have demonstrated that you can:

- Identify the character or Op code represented by any given bit combination in the EBCDIC code.
- Describe packed and zoned decimal data fields and their relationship to input, calculations, and output.
- Describe halfword and fullword binary operands.
- State which system operation concepts are used in decimal and binary arithmetic.
- Describe the conditions under which decimal and fixed point overflow will occur.

SELF-EVALUATION QUESTIONS

Reference Pages in Text

Note: Use your Reference Data Card.

EBCDIC

1. The Extended BCD code uses _____ bits to express a character. 53

2. The numeric portion of a character uses bits _____ through _____ of a byte. 53

3. The zone portion of a character uses bits _____ through _____ of a byte. 53

4. What is the Extended BCD code for the character "2"? 54

5. A numerical field read in from a punched card would be stored in the _____ format. 55

6. If we wish to punch a card with a field that is the result of an arithmetic operation using the decimal instruction feature, we must first _____ the field. 57

7. The packed format has _____ digits in each byte except the low-order byte which has the sign in bits _____ . 56

8. Show a three-digit field in the zoned format. [ ] 56

9. Show a three-digit field in the packed format. [ ] 56

10. To use the instructions of the decimal feature, decimal fields must be in the _____ format. 56

11. The length of a decimal field is specified by (in your own words) _____ . 56

12. Decimal fields are processed using the _____ -to- _____ concept. 57

13. Results of decimal arithmetic operations replace the _____ operand. 57

Reference
Pages in Text

14. Name two items that can cause a decimal overflow. 58

a. _____
b. _____

BINARY

15. A halfword is _____ bytes long while a word is _____ bytes long. 59

16. A halfword in main storage is addressable by its _____ (high/low) order byte location. 59

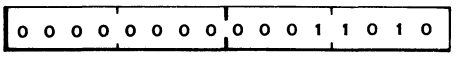
17. The main storage address of a halfword must be divisible by _____ or a _____ exception will occur. 59

18. The results of binary arithmetic operations replace the _____ (1st/2nd) operand. 60

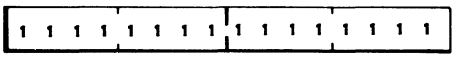
19. Positive binary numbers are represented in their _____ form with a _____ (1/0) in the high-order bit position. 60

20. Negative binary numbers are represented in their _____ form with a _____ bit in the high-order bit position. 60

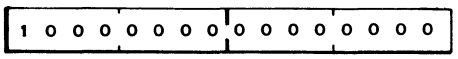
21. What is the decimal value of this halfword binary operand? 60



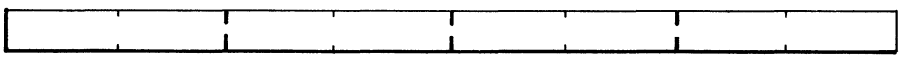
22. What is the decimal value of this halfword binary operand? 61



23. A halfword operand can exist, as such, only in main storage. When it is loaded into a general register, _____.
(complete in your own words). 61



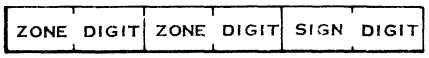
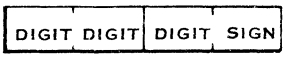
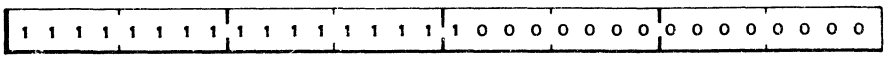
24. Show the above halfword after it has been placed in a general register. 61



25. Halfword operands are processed using the _____-to-_____ concept. 60

26. Whenever the largest negative or positive number is exceeded in a binary operation, a _____ _____ _____ will occur. 63

ANSWERS

		Reference Pages in Text
1.	eight	53
2.	4 - 7	53
3.	0 - 3	53
4.	1 1 1 1 0 0 1 0	54
5.	zoned or unpacked	55
6.	unpack	57
7.	two, 4-7	56
8.		56
9.		56
10.	packed	56
11.	a length code in the instruction	56
12.	storage-to-storage	57
13.	1st	57
14.	a. High-order carry during true addition. b. 2nd operand has more significant digits than the 1st operand has room for.	58
15.	two, four	59
16.	high	59
17.	two, specification	59
18.	1st	60
19.	true, 0	60
20.	complement, 1	60
21.	+26	60
22.	- 1	61
23.	it is placed in bit positions 16-31 of the general register and it is expanded into a fullword by filling all positions to the left with the contents of the bit position 16 (halfword sign).	61
24.		61
25.	Storage-to-Register (or accumulator)	60
26.	fixed point overflow	63

GO TO THE NEXT PART OF THIS SECTION
"INSTRUCTION SEQUENCING AND BRANCHING" on page 64.

EBCDIC

It is assumed that you have had experience with the IBM card code or, as it is sometimes called, the Hollerith Card Code. From your previous computer experience or from a basic computer systems principles course, you are also familiar with the Standard BCD (Binary Coded Decimal) code.

• • •

1. The seven bits of the standard BCD code are _____ .

• • •

C, B, A, 8, 4, 2, 1

2. The ____ bit is the parity bit.

• • •

C

3. The zone bits are the _____ and _____ bits.

• • •

A; B

The basic unit of information in the System/360 is the byte. Just as each card column can be contained as a character in the Standard BCD code, it can also be contained as a character in the System/360 byte.

4. The byte in the System/360 can be used to contain a _____ .

• • •

character

The character code used in the System/360 is known as the Extended BCD Interchange code. Neglecting parity for now, the extended BCD code uses 8 bits to express a character, whereas the standard BCD code uses only 6 bits.

5. The character code used in the System/360 is known as the _____ BCD Interchange Code. This code uses ____ bits to express a character.

• • •

extended; 8

The use of 8 bits may seem inefficient. However, the extended code has some definite advantages not contained in the standard BCD code:

- 256 different bit configurations are possible.
- Both upper and lower case alphabetic information can be coded.
- All possible 256 bit combinations can be punched into an IBM card. This allows pure binary information to be coded on an IBM card, with each column representing 8 bits of binary information.

6. There are ____ possible bit combinations in the extended BCD code. All 256 possible bit combinations can be punched into an IBM _____ .

• • •

256; card

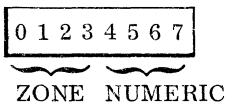
Basically, most character codes are divided into zone and numeric parts. The extended BCD interchange code is no exception. Let's take a look at the System/360 byte and see how it is divided.

7. The bit positions of a byte are numbered ____ through _____ from left to right.

• • •

0; 7

The EBCDIC (Extended BCD Interchange Code) divides the eight bits of a byte as shown below:



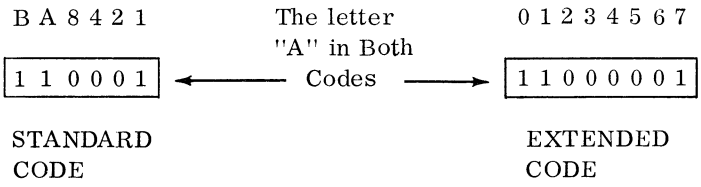
Bit positions 0 - 3 are used to express the zone portion of a character while bits 4 - 7 are used to express its numeric portion.

8. The numeric portion of a character uses bits _____ through ____ of a byte. The zone portion of a character uses bits ____ through ____ of a byte.

• • •

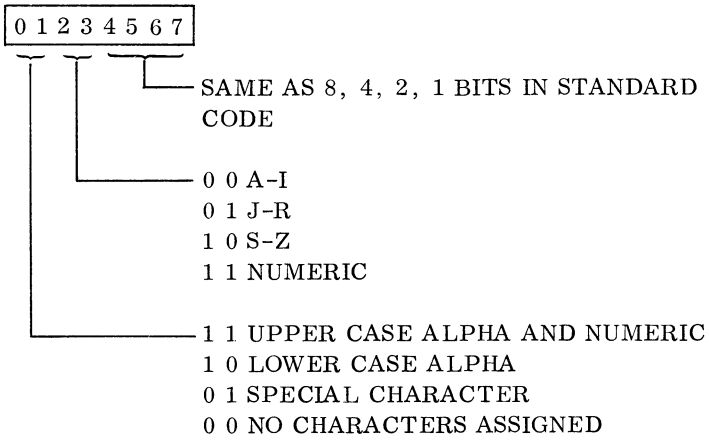
4; 7; 0; 3

Let's see how alphameric characters are expressed in the Extended BCD code as compared with the Standard BCD code.



Notice that bits 4 - 7 of the extended BCD code are used just like bits 8, 4, 2 and 1 of the standard code.

THE EXTENDED BCD BYTE



Pages 7 - 10 of your Reference Data card contain tables showing every graphic (letter, digit, or special character) instruction Op code, etc., represented by the EBCDIC code. Use these tables while practicing translating characters in the next few frames.

9. Bits 4-7 of the extended BCD code are used just as bits ____ , ____ , ____ , and ____ of the standard code.

• • •

8, 4, 2, 1

Bits 2 and 3 of the extended code are used like the B and A bits of the standard code but in reverse order. In the standard code, the presence of B and A bits indicates the letters A-I and the absence of them indicates the numbers 0-9. In the extended code the absence of bits 2 and 3 indicates the letters A-I while the presence of them indicates the numbers 0-9.

10. Bits 2 and 3 of the extended code are the reverse counterpart of the _____ bits of the standard BCD code.

• • •

B and A

Bit-positions 0 and 1 of the extended code are used to group the characters. 1 bits in both positions indicate numeric characters as well as upper case letters. A 1-bit in position zero only indicates lower case letters, while a 1-bit in position 1 only, indicates special characters.

Examples:

Character	Bit Combination
A	11 00 0001
a	10 00 0001
1	11 11 0001
/	01 10 0001

11. What is the extended BCD code for the character "B"? _____ (Use your Reference Card.)

• • •

11000010

DECIMAL DATA FORMATS

Decimal data consists of numeric fields which are coded to represent decimal numbers. For instance, the decimal number "17" can be represented in two columns of an IBM card by a 1-hole punch and a 7-hole punch. In the extended BCD code, this same number can be represented in two bytes like this:

1 1 1 1 0 0 0 1	1 1 1 1 0 1 1 1
1	7

• • •

1. Decimal data consists of _____ fields which are coded to represent (in your own words) _____ _____.

• • •

numeric; decimal numbers

In the preceding illustration, the number 17 did not have a sign and was considered plus.

2. How is the sign of a numeric field conventionally indicated in an IBM card? _____.

• • •

By a zone punch over the low-order or units position of the field.

3. A minus field is indicated by an _____ hole punch while a plus field used a _____ hole punch. The absence of zone punches in a card can also be used to indicate a _____ field.

• • •

11; 12; plus

4. A 12 and 1 punch in the low-order of a field would indicate a _____ sign and a low-order digit of _____.

• • •

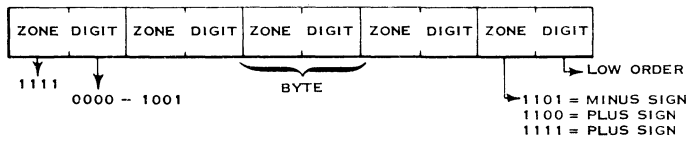
plus; 1

5. When not dealing with decimal data fields a 12 and 1 punch can also be used to represent the character _____, and in the extended BCD code would have this bit configuration: _____. (Use your Reference Card.)

• • •

A; 1100 0001

Decimal numeric fields in the extended BCD code are said to be in the zoned or unpacked format. The zoned or unpacked decimal format looks like this:



Note: Although you will need to be able to determine the sign of a decimal number in storage, later on in this course, there is no need for you to memorize the meanings of zone bits shown above.

Make a note of them on your Reference Data card. One convenient place is to the right of the binary representations of the Interruption Code, at the top of page 6. If you put them there, your card would look like this:

CODE FOR PROGRAM INTERRUPTION

Interruption Code			Program Interruption Cause
DEC	HEX	BINARY (Sign)	
1	01	0000 0001	Operation
2	02	0000 0010	Privileged operation
3	03	0000 0011	Execute
4	04	0000 0100	Protection
5	05	0000 0101	Addressing
6	06	0000 0110	Specification
7	07	0000 0111	Data
8	08	0000 1000	Fixed-point overflow
9	09	0000 1001	Fixed-point divide
10	0A	0000 1010	Decimal overflow
11	0B	0000 1011	Decimal divide
12	0C	0000 1100 +	Exponent overflow
13	0D	0000 1101 -	Exponent underflow
14	0E	0000 1110	Significance
15	0F	0000 1111 +	Floating-point divide

6. Decimal data in the extended BCD code are said to be in the _____ or zoned format.

• • •

unpacked

7. If the input to the System/360 is in card form, the sign of the unpacked format is indicated by a _____ punch over the low-order digit.

• • •

zone

It is a waste of storage space and processing speed to use the unpacked or zoned format for decimal arithmetic operation. The decimal feature of the System/360 uses a more efficient format for decimal arithmetic. It is called the packed or unzoned format. Since only four binary bits are needed to express a decimal digit, why not pack two digits into each byte of a decimal field? This is the packed format as used by the System/360.

8. The decimal feature of the System/360 uses the _____ format. The packed format has _____ decimal digits in a byte

• • •

packed; two

What about the sign of a packed field? It is contained in the low-order bits of the low-order byte. A comparison of the unpacked and packed low-order byte is shown below.

9. The packed format has two digits in each byte except for the low-order byte which has the _____ in bits 0 to 3 and the _____ in bits 4 to 7.

• • •

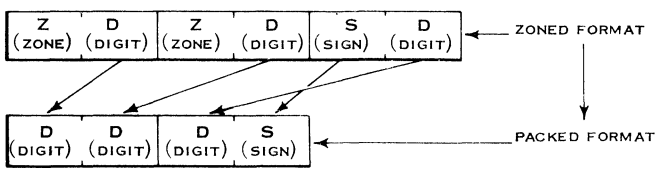
low-order digit; sign

10. In the unpacked format, the sign is in bits _____ of the low-order byte.

• • •

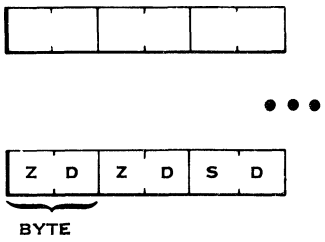
0-3

The next question is: If the System/360 will only process decimal data when it is in the packed format, how do you pack it? The System/360 has an instruction called "pack" which will take a decimal field in the zoned format and change it to the packed format as follows:



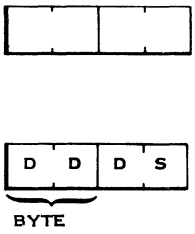
You are not expected to know this instruction at this time. You should be aware, however, that zoned decimal fields can be changed to the packed format by a machine instruction.

11. Show a three-digit field in the zoned format.



• • •

12. Show a three-digit field in the packed format.



• • •

13. A zoned decimal field can be changed to the packed format (in your own words) _____ _____.

• • •

by a machine instruction called "pack"

14. To be able to use the instructions of the decimal feature, decimal fields must be in the _____ format.

• • •

packed

Decimal fields are variable in length and, as such, are processed using the storage-to-storage concept as previously discussed.

15. Variable length fields can start at _____ byte location in main storage.

• • •

any

16. The length of a variable length field is specified by (in your own words) _____ _____ .

• • •

length code in the instruction

17. Decimal fields are _____ in length, and are processed using the _____ -to- _____ concept.

• • •

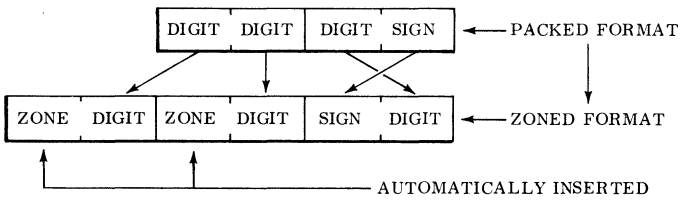
variable; storage-to-storage

18. To process fields using the decimal feature instructions, the fields must be in the _____ format.

• • •

packed

After decimal fields have been processed, they may be left in the packed format unless the output medium is code sensitive. A code sensitive device is one which will accept only certain bit configurations for a given byte. If the output device is code sensitive, the packed fields must be changed to the zoned format. An example of a code sensitive output device is a card punch unit. Just as there was an instruction called "pack", there is an instruction called "unpack." The "unpack" instruction will change a packed format field to the zoned format as shown below.



19. After packed fields have been processed, they may be changed to the zoned format by (in your own words) _____ .

• • •

an instruction called "unpack"

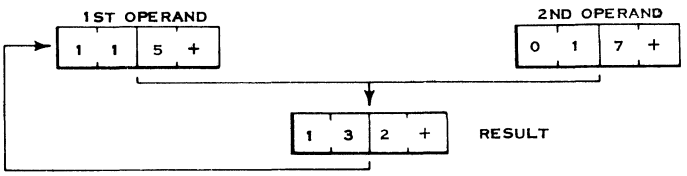
Let's take a few examples of decimal arithmetic. Decimal fields, of course, will be in the packed format. These arithmetic operations will involve the storage-to-storage concept. This means that both operands are located in main storage and that the result will go back into main storage. You will recall that in the System/360 the operands are referred to as the 1st and 2nd operands. In most System/360 operations involving two operands, the result of the operation replaces operand 1.

20. In the System/360, the results of decimal arithmetic operations replace the _____ operand.

• • •

1st

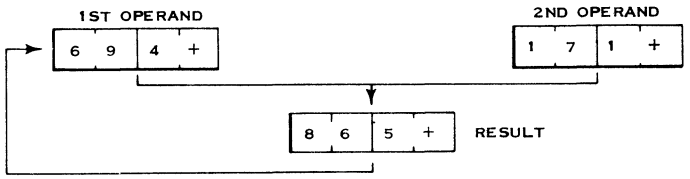
Suppose you wanted to add + 17 to + 115.



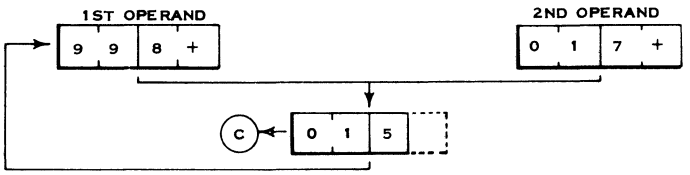
Notice that in the above example:

- The result will replace the 1st operand.
- The 2nd operand has a high-order zero digit. Packed decimal fields are variable by byte length, not by digit length.

In similar fashion as the preceding example, show the addition of + 171 to + 694.



Here is the addition of + 17 to + 998.



Notice that, in the preceding problem, there was a carry out of the high-order position. This carry is lost.

21. A carry out of the high-order during true addition is called a decimal overflow and occurs because the result is too _____ (large/small) to be contained in the 1st operand location.

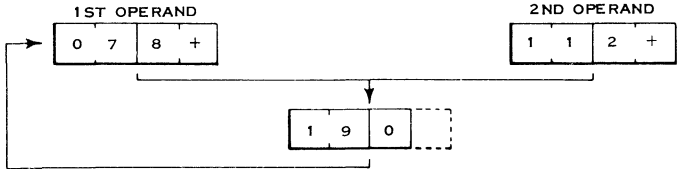
• • •

large

A decimal overflow will occur any time all the significant digits of true addition cannot be contained in the length of the 1st operand.

A decimal overflow will also occur if the 2nd operand contains more significant digits than the 1st operand has room for.

Example: Add + 112 to + 78

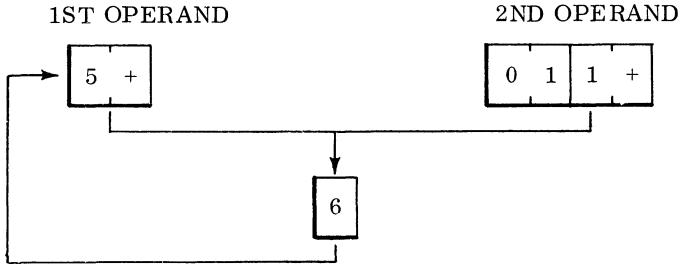


22. In the above example a decimal overflow _____ occur because the 1st operand has room for all significant digits.

• • •

will not

Example: Add + 11 to + 5

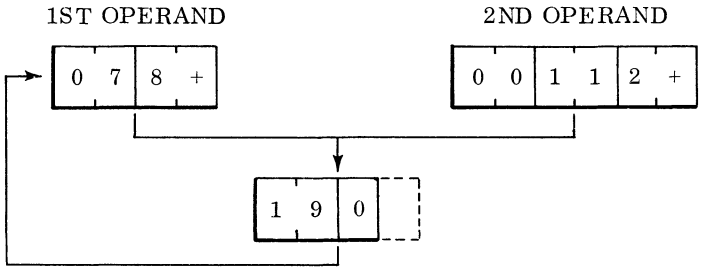


23. In the above example a _____ _____ would occur because all significant digits of the result cannot be contained in the 1st operand.

• • •

decimal overflow

Example: Add + 00112 to + 078



24. In the above example a decimal overflow _____ occur because the 1st operand location can contain all significant digits of the result.

• • •

will not

The next question is: How can a programmer make sure a decimal overflow will not occur during decimal arithmetic operations? A good method would be to make the 1st operand long enough (by having high-order zeros) to accommodate 1) any possible high-order carry as well as 2) all significant digits from the 2nd operand.

25. You wish to add + 11 to + 5. Show the operands necessary to avoid a decimal overflow.

1ST OPERAND

5 +

2ND OPERAND

1 1 +

• • •

1ST OPERAND

0 0 5 +

2ND OPERAND

0 1 1 +

Remember, a partial byte cannot be used as an operand.

BINARY DATA FORMATS

You have just learned the decimal data formats. Decimal data is variable in length and is processed with the storage-to-storage concepts. Binary data is fixed in length and is processed with both the storage-to-register and register-to-register concepts. Let's see what you remember about fixed length operations on the System/360.

• • •

1. Each main storage address refers to a unique _____ location.

• • •

byte

2. Data fields are addressed by their _____ (high/low) order byte location.

• • •

high

3. A halfword is _____ bytes long.

• • •

two

4. A word is _____ bytes long.

• • •

four

5. A doubleword is _____ bytes long.

• • •

eight

6. The storage address of fixed length data fields must be divisible by the number of _____ in the field or a s_____ exception will occur.

• • •

bytes; specification

7. Fixed length operations use the storage-to-register concept. For use in fixed length operations, the programmer has available (in your own words)

_____

• • •

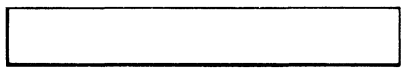
16 general registers

8. How many bytes may be contained in a general register? _____

• • •

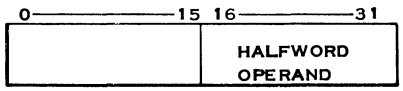
four (4)

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



GENERAL REGISTER

• • •

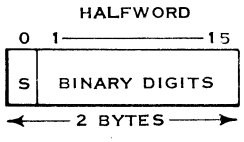


GENERAL REGISTER

Whereas the length of decimal data was specified by a length code in the instruction, the length of binary data is implied by the Op code of the instruction. Binary operands may be either a halfword or a word in length, depending on the INSTRUCTION. Let's discuss halfword operands first.

HALFWORD BINARY OPERANDS

A halfword binary operand is two bytes in length and can be used to express numbers which do not exceed a value of $2^{15} - 1$ (32,767).



As can be seen above, the high order bit position of a halfword is used to represent the sign.

• • •

1. A halfword binary operand is _____ bytes in length. The sign of a halfword operand is represented by (in your own words) _____ .

• • •

two; the high-order bit

Halfword operands use only the storage-to-accumulator concept. The 1st operand is located in the low-order (bits 16-31) of a general register and the 2nd operand is located in main storage. As with decimal arithmetic operations, the results of binary arithmetic operations replace the 1st operand.

2. The address of the 2nd operand in halfword operation must be divisible by _____ or a _____ exception will occur.

• • •

two; specification

3. The 1st operand is located in (in your own words) _____ .

• • •

bits 16-31 of a general register

4. The result of a binary arithmetic operation replaces the _____ operand.

• • •

1st

The System/360 does its binary calculations in a rather unique way. As you have already seen, decimal numbers were represented in their true form (absolute value) with a + or - sign. The System/360 does not represent binary numbers in this manner.

Positive binary numbers are represented in their true form while negative numbers are represented in their complement form. The sign or high-order bit is 0 for positive numbers and is 1 for negative or complement numbers.

5. In the System/360, decimal numbers are represented in their _____ form with a _____ .

• • •

true; + or - sign

6. Positive binary numbers are represented in their _____ form with a _____ in the high-order bit position.

• • •

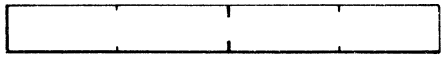
true; 0

7. Negative binary numbers are represented in their _____ form with a _____ in the high-order bit position.

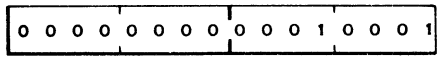
• • •

complement; 1

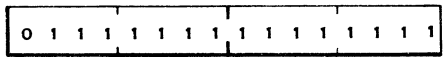
8. Represent the decimal value + 17 as a halfword binary operand. (The small vertical lines within the operand box have no significance; they simply break up the operand into groups of 4 bits so that the number is easier to read)



• • •

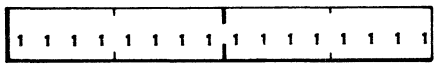


The sign bit position in the preceding answer is 0. This indicates that the binary number is positive and is in its true form. Here is the largest positive binary number that can be in a halfword operand:



Any positive binary number larger than the preceding answer would need a 1 in the high-order bit position. The high-order bit is reserved as a sign bit. A sign bit of 1 would indicate that the number is negative and is in its "twos" complement form. It is very important to remember that negative numbers are always represented in their "twos" complement form.

Here is the value of -1 in a halfword binary format:



The value of zero cannot be complemented. Since negative numbers are represented in their complement form in the System/360, there can be no minus zero. This is desirable in arithmetic operations.

9. What is the value of this binary halfword?
0000 0000 0000 0001. _____

• • •

+1

10. What is this (in your own words)?
0111 1111 1111 1111. _____

• • •

The largest positive binary number that can be represented in a halfword.

Here is the value of zero in a halfword binary format:
0000 0000 0000 0000.

In the System/360 binary numbers are contained in fixed length operands. At this point we are discussing halfword formats. A halfword consists of 16 bits (2 bytes). Of these sixteen bits, bits 1 to 15 represent the number, while bit 0 represents the sign. However, bit 0 does not actually represent a plus or minus sign. Instead it indicates whether bits 1 to 15 contain a true number or a complement number. The total range of the sixteen bits of a halfword operand would look like this:

Complement or Negative Numbers

1111111111111111 → (Minus one)
↓ ↓
1000000000000000 → (Largest negative value)

True or Positive Numbers

0111111111111111 → (Largest positive value)
↑ ↑
0000000000000001 → (Plus one)
0000000000000000 → (Zero)

11. If bit position 0 contains a 0 bit, it indicates that the halfword is the _____ form of a _____ number.

• • •

true; positive

12. If bit position 0 contains a 1 bit, it indicates that the halfword is the _____ form of a _____ number.

• • •

complement; negative

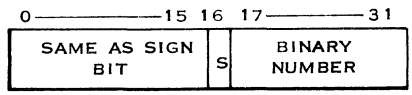
13. A halfword binary number is placed into bit positions _____ of a general register.

• • •

16 to 31

When a halfword is placed or loaded into a general register, the halfword is expanded to a fullword by propagating the sign bit to the left. In other words, bits 0 to 16 will be the same. Bit position 16, of course, will contain the sign of the halfword operand.

HALFWORD OPERAND IN A GENERAL REGISTER



14. When a halfword is loaded into a general register, it is expanded to a _____.

• • •

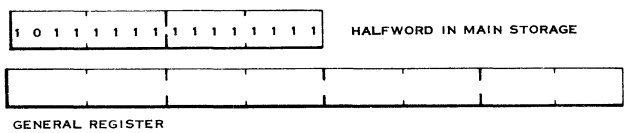
word or fullword

15. A halfword is expanded to a fullword by (in your own words) _____.

• • •

propagating the sign bit to the left.

16. Show the contents of a general register after it has been loaded with the following halfword operand from main storage:



• • •

17. Show the contents of the preceding general register as 8 hex digits. _____

• • •

FFFFBFFF

From the preceding discussion you should now realize that halfword operands exist only in main storage. When a halfword is loaded into a general register, it is expanded to a fullword.

FULLWORD BINARY OPERANDS

For use with the binary arithmetic instructions, binary operands in the System/360 may be either a halfword or a word in length. Halfwords are processed using the storage-to-register concept. Binary operands in the general registers are always a word in length. Halfwords are expanded to fullwords whenever they are placed into a general register.

1. Halfword binary operands are processed using the _____ to _____ concept. Binary operands in a general register are a _____ in length.

• • •

storage; register; word

Binary operands which are a word in length may reside either in main storage or in a general register. Since there are sixteen general registers, word operands may be processed using either the storage-to-register concept or the register-to-register concept.

2. Word operands may reside in either ____ ____ or in the _____ _____ . When in main storage they must have an address divisible by _____ or a specification exception will occur.

• • •

main storage; general registers; four

3. A specification exception is a type of p_____ check.

• • •

program

4. Besides the storage-to-register concept, word operands may be processed using the _____ -to- _____ concept.

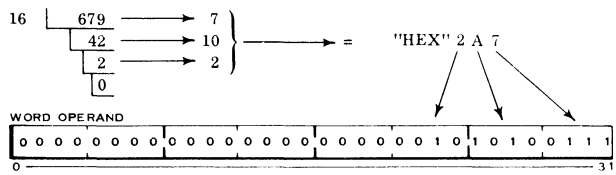
• • •

register-to-register

As with halfword operands, bit position 0 of a word operand is the sign bit and indicates whether the word is a positive number in true form or a negative number in complement form.

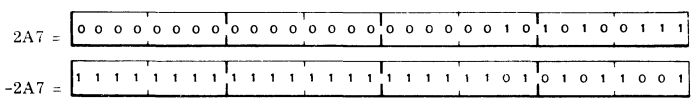
Let's show the value (+ 679) as a word operand. We'll convert the decimal value to hexadecimal first and then to binary.

Example:



Here is the value (-679) as a word operand. Remember this is a negative number and will appear in complement form.

"DEC" 679 = "HEX" 2A7



From the two preceding problems, you can see that binary operands basically are unsigned numbers in either true or complement form.

BINARY ARITHMETIC OPERATIONS

The System/360 uses halfword and word binary operands. The arithmetic principles involved, however, are the same regardless of length.

• • •

1. Binary operands in the System/360 are a _____ or _____ in length.

• • •

halfword; word

2. The length of a binary operand is implied by the I_____ .

• • •

Instruction; More specifically the length is implied by the Op code portion of the instruction.

3. Positive numbers are in _____ form while negative numbers are in _____ form.

• • •

true; complement

4. Bit position _____ of a binary operand indicates whether the operand is in true or _____ form. If bit position 0 is a 1, the operand is in _____ form and represents a _____ number.

• • •

zero; complement; complement; negative

There is one final principle of System/360 binary operations to be learned. This is the Fixed Point Overflow. Earlier you learned about the decimal overflow and what caused it. Read the following review frames on decimal overflow.

5. The presence of a high-order carry during true addition of decimal fields indicates a _____ _____ .

• • •

decimal overflow

6. Any time all the significant digits of a decimal operation cannot be contained in the resulting field a _____ _____ will occur.

• • •

decimal overflow

A decimal overflow indicates that the decimal result is not correct. The result has exceeded the maximum value that could be contained in the result field (1st operand).

7. The maximum decimal value that can be contained by a three-byte 1st operand is _____ .

• • •

3 BYTE PACKED OPERAND

A fixed point overflow indicates that the result of a binary operation is not correct. The result has exceeded the maximum value that could be contained in the result field (1st operand). Of course binary operands are fixed in length and may be either a halfword or word in length.

Here is the largest positive binary number that can be represented in a halfword: 0111 1111 1111 1111

Here is the largest negative binary number that can be contained in a halfword: 1000 0000 0000 0000

Here is the largest positive binary number that can be contained in a word: 01111111111111111111111111111111

8. If the value of + 1 were added to the largest positive binary number, a _____ _____ overflow would occur.

• • •

fixed point

9. When the result of a binary operation exceeds the maximum value that can be contained in the result field (1st operand), a _____ _____ _____ will occur.

• • •

fixed point overflow

10. If the value of -1 were added to the largest negative binary number, a _____ _____ _____ would occur.

• • •

fixed point overflow

INSTRUCTION SEQUENCING AND BRANCHING

Learning Objectives

When you complete the following test, you will have demonstrated that you can:

- Describe the format of the program status word.
- Describe how the PSW affects the sequential nature of instruction fetching.
- Describe how branching affects the address portion of the PSW.
- Describe how the CPU status is indicated in the PSW.
- Describe the operation of the "branch on condition" instruction.

SELF-EVALUATION QUESTIONS — Reference Pages in Text

Note: Use your Reference Data card.

1. PSW is short for _____ _____ _____. — 68

2. The PSW is a _____ in length. — 68

3. The address of the next instruction to be fetched is contained in _____. — 68

4. The instruction address portion of the PSW contains the address of the _____(current/next sequential) instruction. — 68

5. The PSW which is being used to fetch instructions is sometimes referred to as the "_____" PSW. — 69

6. This PSW is located in _____(main storage/ a general register/some type of internal register or storage area). — 69

7. Branching is accomplished by replacing the _____ _____ in the PSW with the "branch to"_____. — 70

8. The condition code _____(is/is not) contained in the PSW. — 70

9. The condition code has _____ possible settings. — 70

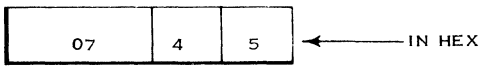
10. The condition code in the PSW is changed by _____(all/some) instructions that are executed. — 71

11. Indicate the condition code setting after the following results: — 71

Algebraic Add	Compare	Condition Code
zero	Equal	_____
< zero	Low	_____
> zero	High	_____
overflow		_____

12. Following an algebraic add instruction, the following "branch on condition" instruction would test for what result?_____ — 74

BRANCH ON CONDITION



Reference
Pages in Text

13. Following a compare instruction, the following instruction would test for what result? _____ 76

    BCR 12,8

14. The following instruction would _____(always/never) result in a branch. 76

    BCR 15,7

15. If the following instruction results in a branch, the instruction address in the PSW would be replaced by bits 8 through 31 of _____. 76

    BCR 8,5

16. If the following instruction resulted in a branch, the instruction address in the PSW would be replaced by _____. 76

    BC 8,0(4,7)

ANSWERS

	Reference Pages in Text
1. Program Status Word	68
2. Doubleword	68
3. PSW	68
4. next sequential	68
5. "current"	69
6. some type of internal register or storage area	69
7. instruction address, address (location)	70
8. is	70
9. 4	70
10. some	71
11. Condition Code 00 (0) 01 (1) 10 (2) 11 (3)	71
12. < zero	73
13. equal or low	75
14. always	76
15. general register 5	76
16. The effective address generated by adding the contents of register 4 and register 7 and a displacement factor of 0.	76

GO ON TO THE NEXT PART OF THIS SECTION "INTERRUPTS" on page 78.

INSTRUCTION SEQUENCING AND BRANCHING

In the System/360, there is a doubleword which is used to indicate the status of the program as well as to control the program. This doubleword is called the Program Status Word or PSW for short. The PSW, which is being used with the program, is not kept in either main storage or the general registers. It is part of the internal machine circuitry and as such is not easily changed. You will learn more about the PSW after you answer a few questions concerning stored program concepts.

• • •

1. Coded information which causes a computer to perform a specific task (such as add or subtract) is called an _____ .

• • •

instruction

2. A series of instructions used to solve a problem on a computer is called a _____ .

• • •

program

3. A program is sometimes referred to as a stored program because of the fact that it is kept in _____ _____ when it is executed.

• • •

main storage

4. The instructions of the stored program are read out of main storage one at a time. The instruction is then decoded in the _____ section of the CPU (Central Processing Unit).

• • •

control

5. After being decoded in the control section of the CPU, the instruction is then executed in the _____ section of the CPU.

• • •

ALU (Arithmetic and Logic Unit)

6. For every instruction, there are two periods of time. The time during which the instruction is read out or "fetched" from main storage is known as __ _____ .

• • •

I Time (or Instruction Time)

7. The operation specified by the instruction is performed during __ _____ .

• • •

E Time or (Execution Time)

8. Instructions have two periods of time associated with them: ___ time and ___ time.

• • •

I; E

In the System/360, there is no clear division between I time and E time. That is, before the instruction has been completely read out and analyzed by the control section, some part of the execution may have already been started. But for all practical purposes, we can think of I time as being separate from E time.

9. Data is the name generally given to information read out of main storage during ___ time.

• • •

E

10. Instructions are information read out of main storage during ___ time.

• • •

I

11. An instruction may be treated as data and changed if it is read out during ___ time.

• • •

E

12. Information in main storage is treated as either _____ or _____ depending on when the information is read out of main storage.

• • •

instructions; data

13. The instructions of a stored program are generally read out and executed in a _____ (random/sequential) manner.

• • •

sequential

14. The sequential manner of instruction fetching and execution can be changed by instructions known as _____ instructions.

• • •

branch

15. When the next instruction is read out of a non-sequential location in main storage, we say that _____ occurs.

• • •

branching

16. Instructions are generally thought of as having two parts. One part of the instruction is used to tell the computer what to do (such as to add or branch). This portion of the instruction that tells the computer what to do is known as the ____ _____ .

• • •

Op code

17. The other portion of the instruction generally tells the computer where the data is located. For this reason, it is sometimes called the _____ portion.

• • •

address or operand

PSW - INSTRUCTION ADDRESS

The address portion of an instruction may sometimes contain other information besides data addresses. In a branch instruction, it would give the address of the next instruction to be executed. In some instructions, the data to be operated on may be contained in the address portion. Later on when you study the instructions of the System/360, you will learn what is contained in the address portion of each instruction. Let's continue now with the study of the System/360 and its Program Status Word (PSW).

As mentioned earlier, the PSW indicates the status of the program being executed by the System/360. The program status word would include status information such as:

- The location of the next instruction.
- Whether an arithmetic operation has resulted in a positive or negative answer. Possibly, the operation ended with a zero balance or an overflow.

Information such as indicated above, as well as other information, is contained in the program status word.

• • •

1. The PSW is a doubleword and contains ____ bytes of information.

• • •

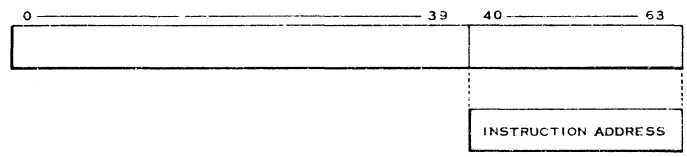
8

2. Like all doublewords, the bits of the PSW are numbered ____ through ____ from left to right.

• • •

0; 63

For right now, let's examine only one portion of the PSW and identify its contents.



3. Bit positions 40 - 63 of the PSW are called the _____ _____ portion. Note the format at the bottom of page 5 of your Reference Data card.

• • •

instruction address

4. The location of the next instruction to be fetched from main storage is indicated by bits 40 - 63 of the PSW. These bits are the _____ _____ portion of the PSW.

• • •

instruction address

5. As you learned earlier, main storage is accessed by binary address. As such, bits 40 - 63 of the _ _ _ would contain the 24-bit binary address of the next sequential _____ .

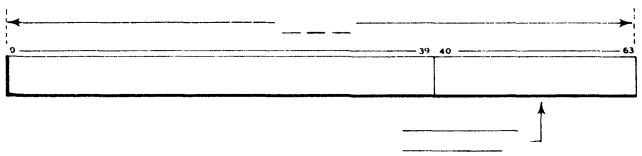
• • •

PSW; instruction

6. The instruction address of the next sequential instruction is contained in a doubleword called the _ _ _ .

• • •

PSW

7. 

Fill in the blanks.

• • •

PSW; instruction address

The PSW is a doubleword which reflects the status and controls the program "currently" being executed. For this reason, it is often referred to as the "current" PSW.

8. The address of the next sequential instruction is contained in the instruction address portion of the "_____" PSW.

• • •

"current"

9. The status of the program being executed is contained in the "_____" __ __ __ .

• • •

"current" PSW

Before examining more of the "current" PSW, you may be wondering where this doubleword is kept. For one thing, the "current" PSW does not use any of the 16 general registers or addressable locations in main storage. It is therefore kept in some internal area of the System/360 that is not addressable by the program. It may be a conventional doubleword register or it may be kept in the same "local store" used by the general registers in some models of System/360. In other words, it all depends on which particular model of System/360 we are discussing. The "current" PSW may actually be kept in a number of smaller registers. For all practical purposes, the "current" PSW is considered as one doubleword of information.

10. The "current" PSW _____ (is/is not) kept in main storage or any of the general registers.

• • •

is not

11. The address of the next sequential instruction is kept in the "_____" __ __ __ .

• • •

"current" PSW

The instruction address portion of the "current" PSW is automatically updated for each instruction that is fetched and executed. That is, if an RR type instruction is fetched from location 1000, the instruction address portion of the "current" PSW must be updated.

12. Since an RR type instruction is one halfword (2 bytes) in length, the location of the next sequential instruction would be _____ .

• • •

1002; Each storage address refers to a single byte.

13. After the RR type instruction at location 1000 has been executed, the instruction address portion of the PSW which now contains _____ will be used to fetch the next _____ .

• • •

1002; instruction

14. If the instruction at location 1002 is the RX type, the instruction address portion of the "_____" __ __ __ will then be changed to _____ .

• • •

"current" PSW; 1006

15. Since instruction length is always a multiple of halfwords, the instruction address portion of the "current" PSW is always updated by some multiple of _____ (1/2/3).

• • •

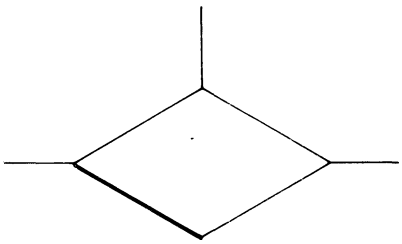
2

16. The instruction address in the "current" PSW is increased by 2, 4, or 6, depending on the _____ of the current instruction.

• • •

length

You should be familiar with the use of flowcharts in writing a program. Decision blocks in a program are represented by this symbol:



The use of this symbol in a program represents a decision as to what to do next. Should the program continue with its present sequence of instructions or should it "branch out" to another sequence of instruction? Sometimes a decision block represents leaving a sequence of instructions. In this case, the program is trying to decide which of two or more new sequences to "branch to."

As you know, the instruction address portion of the "current" PSW is used to fetch the next sequential instruction. What then happens to the instruction address portion of the "current" PSW when a "branch" is taken?

Whenever a branch is executed, the contents of the instruction address portion of the "current" PSW are replaced by the address of the instruction being branched to.

For example:

17. If an RX instruction at location 1000 is fetched, the instruction address portion of the "current" PSW would normally be changed to _____ .

• • •

1004

18. If, however, the instruction at 1000 says to branch to location 2000, the instruction address portion of the "_____ " PSW is changed to _____ .

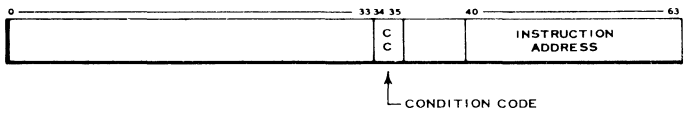
• • •

"current"; 2000

In the preceding example, the instruction address of the "current" PSW might actually be first updated to 1004 and then changed to 2000. This will depend on the particular branch-type instruction and the model of the System/360. However, at the time the system decides that it will branch, the address of the "branch to" location is placed in the instruction address portion of the "current" PSW.

PSW - CONDITION CODE

At this point, you should clearly understand the function of the instruction address portion of the "current" PSW. It is used to fetch instructions from main storage and to indicate the current location in the program.



• • •

1. As illustrated above, there is another field in the PSW and it is called the _____ _____ . It is located in bits ___ and ___ of the "current" PSW.

• • •

condition code; 34; 35

2. Bits 34 and 35 of the "current" PSW are used to reflect the status of the CPU. These bits are known as the _____ _____ .

• • •

condition code

The question that now arises is: "How does the condition code reflect the status of the central processing unit?" First of all, since the condition code has two binary bits, it can have four possible bit combinations:

1) 00
2) 01
3) 10
4) 11

3. The condition code is set to one of its ____ possible combinations after an instruction has been executed.

• • •

four

4. List the four possible settings for the condition code:

1.____ 2.____ 3.____ 4.____

• • •

00; 01; 10; 11

5. After the instruction is executed, one of four possible settings is placed in the _____ _____ portion of the "current PSW."

• • •

condition code

I don't want to mislead you. Not all instructions affect the condition code. Later, when you learn the instructions, one of the items you should be interested in is each instruction's effect on the condition code. At this point let's take a good look at the condition code and see how it is used.

One of the uses of the condition code is to indicate the result of arithmetic operations such as add or subtract. There are 4 possible results of an algebraic add or subtract, whether it is performed by binary or decimal arithmetic. The result could be a 1) positive number, 2) negative number, 3) zero balance or, 4) an overflow. The condition code reflects these results with these settings:

Arithmetic Result	Condition Code
zero balance	00
< zero (or negative)	01
> zero (or positive)	10
overflow	11

6. A zero (00) condition code after algebraic addition indicates a _____ result.

• • •

zero

7. Let's assume that at the end of an add operation the condition code is set to 01. This indicates that the algebraic addition resulted in a _____ _____ (zero balance/negative number).

• • •

negative number

8. If the condition code is set to 10, the algebraic addition resulted in a _____ _____ (zero balance/positive number).

• • •

positive number

9. Besides a zero, negative or positive result, an overflow is also possible with algebraic addition. This is indicated with a condition code setting of _____ .

• • •

11

10. If algebraic addition results in a negative number, the condition code is set to _____ .

• • •

01

11. Indicate the condition code setting for the following results.

Algebraic Result	Condition Code
zero balance	a. _____
< zero (or negative)	b. _____
> zero (or positive)	c. _____
overflow	d. _____

• • •

a. 00; b. 01; c. 10; d. 11

The condition code is set at the end of algebraic add or subtract operations (either decimal or binary). The condition code in the PSW will retain this setting until the end of the next instruction that can change the condition code. Remember now that not all instructions affect the condition code.

12. Indicate the condition code setting for the following algebraic results.

Result	Condition Code
overflow	a. _____
zero	b. _____
positive	c. _____
negative	d. _____

• • •

a. 11; b. 00; c. 10; d. 01

Notice that you have learned how the condition code indicates the results of algebraic addition. By algebraic addition, we mean the addition or subtraction of signed numbers. Another use of the condition code is to indicate the result of a compare operation. A compare operation consists of comparing the 1st operand to the 2nd operand. The condition code is set to indicate the result. Neither operand is changed. The condition code is set and indicates whether the 1st operand is equal to, less than, or greater than the 2nd operand as follows:

Comparison	Condition Code
equal	00
1st operand low	01
1st operand high	10

Notice that a condition code setting of 11 is not possible after a compare operation.

13. Besides indicating the results of algebraic addition, the condition code is also used to indicate the result of a _____ operation.

• • •

compare

14. Which one of the following is always true:
    a. A condition code of 00 indicates a zero result.
    b. A condition code of 00 indicates an equal comparison.
    c. A condition code of 00 depends on the instruction just executed.
    d. A condition code of 00 depends on the last instruction that could possibly change the condition code.

• • •

d; Not all instructions affect the condition code.

15. If the instruction just executed were a compare operation, a condition code setting of 00 would indicate that the 1st and 2nd operands were _____ .

• • •

equal

16. After a compare operation, the condition code indicates whether the _____ (1st/2nd) operand is equal to, lower than, or higher than the _____ (1st/2nd) operand.

• • •

1st; 2nd

17. After a compare operation, a condition code of 01 would indicate that the _____ (1st/2nd) operand was low compared to _____ (1st/2nd) operand.

• • •

1st; 2nd

18. After a compare operation, a condition code of 10 would indicate that the 1st operand was _____ (low/high) compared to the 2nd operand.

• • •

high

19. Indicate the condition code for the following comparisons.

Comparison	Condition Code
a. 1st and 2nd operands are equal	_____
b. 1st operand is low	_____
c. 1st operand is high	_____

• • •

a. 00; b. 01; c. 10

20. Indicate the meaning of the following condition codes for algebraic and compare operations.

Condition Code	Algebraic Result	Comparison Result
a. 00	_____	_____
b. 01	_____	_____
c. 10	_____	_____
d. 11	_____	_____

• • •

a.	zero	equal
b.	negative	low
c.	positive	high
d.	overflow	not possible

You should now have a good idea of how the condition code of the PSW indicates the status of the central processing unit. The condition code is used to indicate more than just the result of an algebraic or comparison operation. You will learn these other possible indications as you learn the individual instructions of System/360.

Incidentally, when you look up an instruction in the SRL manual IBM System/360 Principles of Operation (Form #A22-6821), you will find references to the condition code setting expressed as follows:

Condition Code in PSW (binary)	"Condition Code" as shown in the manual
00	0
01	1
10	2
11	3

Since you can already state the condition given the binary code, you should be able to do the same with the decimal equivalent. Try it:

21. Suppose that a compare results in a condition 2. What is the condition?

• • •

2(decimal)=high. The first operand is greater than the second.

The next question which you may have is: "Now that I know how the condition code indicates CPU status, how can the condition code be used to control the program?"

One type of instruction in the System/360 is called "branch on condition". This instruction causes the system to examine the condition code and branch if its setting matches that of a code in the "branch on condition" instruction.

22. The condition code can be tested by means of an _____.

• • •

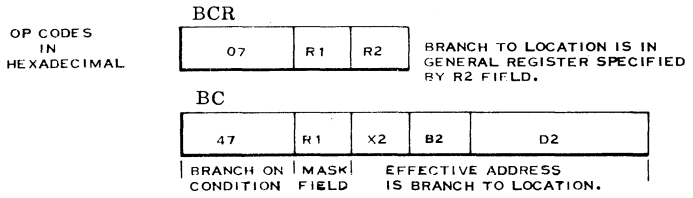
instruction

23. An instruction that tests the condition code is called "_____ __ _____." This "branch on condition" instruction will cause a branch if the condition code matches a coded field in the_____.

• • •

"branch on condition"; instruction

A "branch on condition" instruction can be either of the RR or the RX format. In either case, the R1 field is coded so that the condition code can be tested.



24. In the "branch on condition" instruction, the condition code in the PSW is tested against the R____ field in the instruction.

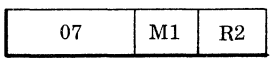
• • •

R1

We showed an "R1" field, in the instructions above, because we knew you were used to seeing it, in an RR or RX format. Actually, the "R1" field in a branch on condition instruction is called a "mask" field and is designated as "M1".

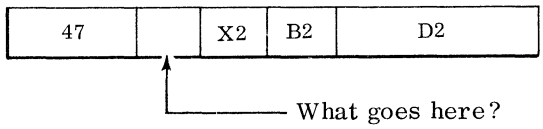
25. The instruction in the RR format should have been shown as:

BCR



The instruction in the RX format should have been shown as:

BC



• • •

M1

26. Look at the RR and RX formats on your Reference Data Card. In each case, the R1/M1 means that M1 is used to represent the first operand when it is a mask field. So, to make our earlier statement strictly correct, "The condition code is tested by being matched against the M1 or _____ field in the instruction".

• • •

mask

As you know, the condition code can mean many things. For instance, it could indicate a low or equal compare, a negative arithmetic result, an overflow and so forth. However, it can have only one value (0, 1, 2, 3) at any one time. This setting can represent only one thing, depending on the last instruction that affected the condition code.

27. Select one of the following answers:

At any one time, a condition code value of 0 can represent:

a. Both an equal compare and an arithmetic result of zero.
b. Either an equal compare or an arithmetic result of zero.
c. Never an equal compare or an arithmetic result of zero.

• • •

b

28. The instruction that tests the condition code is known as " _____ __ _____ ."

• • •

"branch on condition"

29. The "branch on condition" instruction can be in either RR or the RX format. In the RR format, the "branch to" address is in the register specified by the _____ (1st/2nd) operand of the instruction which is the R_____ field.

• • •

2nd; R2

30. In the RX format, the "branch to" address is in the _____ (1st/2nd) operand of the instruction and consists of b___ _____ , i_____ _____ and d_____ .

• • •

2nd; base address; index address; displacement

31. In either the RR or RX format, the M1 field, also called the _____ field, is tested against the _____ _____ .

• • •

mask; condition code

32. Bit positions 8 - 11 of a "branch on condition" instruction are called the _____ field.

• • •

mask or M1

33. The four bits of the mask field are tested against the _____ possible settings of the condition code.

• • •

four

Depending on what the programmer specifies as the contents of the four-bit M1 field, the result of the condition test will be to:

a. Never branch
b. Branch if one of the four conditions is present
c. Branch if one or another of the conditions is present
d. Always branch

Let's take b. first.

The mask field is tested against a single condition code according to the following table:

Mask Field	Condition Code in PSW	"Condition Code" as shown in the manual
1000	00	0
0100	01	1
0010	10	2
0001	11	3

As you can see from the table, any one of the possible condition code settings can be tested by setting one appropriate bit of the instruction's mask field.

34. If bits 8 - 11 of a "branch on condition" instruction contain 1000, a branch will occur only if the condition code has a value of _____ .

• • •

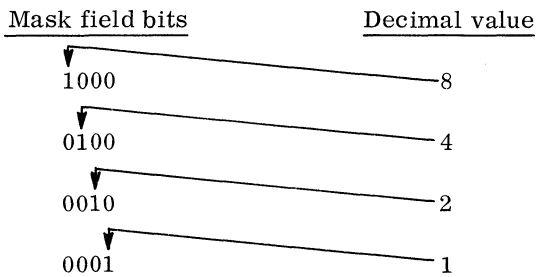
0

35. If the condition code setting were 01 and the mask field were 0010, a branch (would/would not) _____ occur.

• • •

would not

We have mentioned that the Principles of Operation manual refers to condition code settings as 0, 1, 2, 3. The Reference Data Card refers to the four bits of the mask field by their decimal equivalents:



Now look at the chart labeled "Condition Code Setting" at the top of page 5 of your Reference Data Card to find these values. The conditions checked by each mask field bit (for every applicable instruction) are listed below the corresponding position value.

Let's check the chart against what you have already learned:

If our branch on condition instruction has a mask field (M1) with a value of 4 (0100):

a. The result of an algebraic addition, whether binary (Add H/F) or decimal (Add Decimal) will be tested for < zero (negative).
b. The result of a compare, whether binary (Compare H/F) or decimal (Compare Decimal) will be tested for 1st operand (A) low.

By the way, the H/F means that the test and branch conditions are the same, for a given mask field, regardless of whether the binary operation involved halfwords or fullwords.

Now see if you can use the chart.

36. What condition would be tested, after a compare, if the value of the mask field is 2?

	Name of Instruction	Condition
Binary set	_____	_____
Decimal set	_____	_____

• • •

	Name	Condition
Bi	Compare H/F	A high
Dec	Compare Decimal	A high

37. Now suppose that there has been a fixed point subtraction operation with fullwords, and M1 of the branch on condition instruction has a value of 8. For what condition will the result be tested?

• • •

a result of zero

The reason that we have stressed the position values of the mask bits is that, when you are coding the instruction, you will use a decimal number to establish M1. For example, suppose that you write the RR type branch on condition instruction:

BCR	8,	7
↑	↑	↑
Mnemonic Op code	Decimal Value of M1	R2

This is just like the last example. The mask field bit with a value of 8 will be set on (i. e. , the mask field will be 1000), and if the condition set by the last effective instruction is matched, the contents of general register 7 will be placed in the instruction address portion of the PSW.

38. If the last instruction that set a condition code was a compare, what condition would cause a branch?

• • •

equal

39. Suppose that you coded BCR 1,5 to test the result of binary addition:
    a. How would M1 look in binary?
    b. What condition will be tested?
    c. What will happen to the PSW if M1 matches the condition code setting?

• • •

a. 0001; b. overflow; c. The address specified by general register 5 will be placed in the instruction address of the current PSW.

40. The branch on condition instruction in the RX format has a main storage location for its second operand. Looking on page 1 of your Reference Data Card, find the operands that are used with the mnemonic BC.

• • •

M1, D2(X2, B2)

Assuming that you wished to use this form of the instruction to test for "A high" as the result of a compare, you might write: BC 2,100(8,4)

This would:

a. Establish M1 as 0010
b. Test for A high (see your card)
c. Given a condition code setting of 10 (value of 2) cause the contents of (base) register 4, plus the contents of (index) register 8, plus a displacement of 100 to become the new instruction address in the current PSW.

41. If you wrote the instruction:

BC 4,0(3,2)

a. What would the M1 field look like?
b. Given a decimal add operation, what result would be tested?
c. What would be the condition code setting if that result occurred?
d. What would happen to the current PSW after the condition code and M1 matched?

• • •

a. 0100; b. <zero; c. code setting 01;
d. The contents of (base) register 2, plus (index) register 3, plus a displacement of zero would become the new instruction address in the current PSW.

The preceding questions are the sort that you will be asking yourself, as you code.

Using page 5 of your Reference Data Card, you should now be able to determine which condition is being considered, regardless of whether the PSW condition code setting, the IBM System/360 Principles of Operation SRL designation, or the mask field value is referred to.

We have seen how you would write a branch on condition instruction to set any one of the M1 bits on. This will test for any one of the four possible condition code settings.

If we set two of the M1 bits on, a branch will occur on either of the respective conditions.

For example, suppose that we want to branch to an address specified by the contents of register 6, if the result of a subtraction is zero or less (negative).

The mask field bit that tests for zero has a position value of 8.

The mask field bit that tests for less than zero (<zero) has a position value of 4.

To turn both of these bits on, we must specify an M1 with a value of 12 (8 + 4).

We would write:
BCR 12,6

42. What would we write, if we wanted to branch to an address specified by base register 2, index register 3, and a displacement of 1000, if the result of an addition was >zero or showed an overflow?

• • •

BC 3,1000(3,2)

We would branch only on overflow, to an address specified by register 6, if we write the instruction: BCR 1,6.

43. How would we write an instruction that would cause a branch to the address specified by register 6 for any condition except an overflow?

• • •

BCR 14,6 (By specifying an M1 of 14, we turn on every mask bit except the one that tests for overflow)

There will usually be a point in a program when we want to branch to the address of a specific instruction, regardless of the condition code. This is called an "unconditional branch".

For example, we may have successfully completed a series of operations on data from one card, and we wish to go back to the first instruction in the series so that we can begin operations with data from another card. We would unconditionally branch to the address of that first instruction.

44. We use the branch on condition instruction to do this, by setting all four mask field bits on. What value would we give M1?

• • •

15 (8+4+2+1)

This use of branch on condition relies on the fact that one condition code or another will always be set, during the execution of any program.

By contrast, we may want to make a branch instruction temporarily inoperative. We would do this by setting all M1 bits to zero. None of the four conditions would be tested, so there would be no branch.

The last point to be made about the branch on condition instruction (and any other, for that matter) is:

a. You will specify it with a mnemonic Op code, a decimal value for M1, and decimal value or a symbolic label for R2.
b. You will be reading it, on a machine listing of your program, with the Op code and any decimal values expressed in hex.

We will leave symbolic operands for later; the following shows the sort of translation that occurs when you specify registers and displacements in decimal notation:

a. If you write the RR type instruction BCR 12, 10 it will be printed out as

07	C	A ← in hex.
Op code	M1	R2

b. Given that the instruction tests the results of a compare operation; if you were checking your program listing you would say, "If A is equal to or lower than B, the contents of register 10 will become the instruction address in the current PSW."

45. Here is a branch on condition instruction from a machine listing:

    07 E B

    How was it originally coded?

• • •

BCR 14,11

Because of the difference between the coding of the operand for an RX instruction and its format in a hex printout, the situation is a bit more complex.

Suppose that you code BC 4,100(7,8). If the test is to be made after a subtraction operation, this instruction means; "Given a condition code setting for a negative result, branch to a location whose address is the sum of the contents of (base) register 8, (index) register 7, and a displacement of 100.

In hex, the instruction would print out as

47	4	7	8	064
↑	↑	↑	↑	↑
Op code	M1	X2	B2	D2

Check the RX format on your card.

INTERRUPTS

Learning Objectives

When you complete the following test, you will have demonstrated that you can:

- Relate the handling of interrupts to a supervisor or control program.
- Describe how an interrupt affects the current PSW.
- Define: Current, Old, New PSW's.
- State the five (5) classes of interrupts.
- Describe how the old PSW shows the cause of the interrupt.

SELF-EVALUATION QUIZ

Reference Pages in Text

Note: Use your Reference Data card.

1. List the five classes of interrupts — 85
   a. _____
   b. _____
   c. _____
   d. _____
   e. _____

2. Define: — 85
   a. "Current" PSW _____
   _____
   b. "Old" PSW _____
   _____
   c. "New" PSW _____
   _____

3. List the information contained in the PSW which is of principal interest to the programmer: — 90
   a. _____
   b. _____
   c. _____

4. What are the first actions taken by the computer when an interrupt occurs? — 87
   a. _____
   b. _____
   c. _____

5. Describe the additional normal actions taken by the system following a program interrupt. — 88
   a. _____
   b. _____
   c. _____

6. Describe the additional normal actions taken by the system following a machine check interruption. — 88
   a. _____
   b. _____
   c. _____

Reference Pages in Text

7. Describe the additional normal actions taken by the system following an I/O interrupt. 89
   a. _____
   b. _____

8. After handling an I/O interrupt, how does the machine return to the interrupted program? _____ 89
   _____

9. An interrupt action replaces: (Choose the most correct answer) 90
   a. The "current" PSW with any "old" PSW.
   b. The "current" PSW with the contents of a general register.
   c. A "new" PSW with a doubleword from main storage.
   d. The "current" PSW with a specified doubleword from main storage.

ANSWERS

Reference Pages in Text

1. a. External — 85
   b. Supervisor Call
   c. Program
   d. Machine
   e. I/O

2. a. "Current" PSW is the doubleword being used by CPU to control the execution of a sequence of instructions. There is only one "current" PSW. — 85
   b. "Old" PSW is the doubleword placed in main storage as a result of an interrupt. Prior to the interrupt it was the "current" PSW. There are five locations reserved in main storage, one for each class of interrupt.
   c. "New" PSW is the doubleword fetched from main storage as a result of an interrupt. It then becomes the "current" PSW. Bits 40 - 63 of this doubleword would switch the machine to a new sequence of instructions.

3. a. Condition Code — 90
   b. Interrupt Code
   c. Instruction Address

4. a. Set Interruption Code — 87
   b. Store Old PSW
   c. Fetch New PSW

5. a. Diagnostic message is posted. — 88
   b. Contents of core storage are printed out (core dump), unless suppressed.
   c. Program is cancelled.

6. a. Error recovery routines are entered, if applicable. — 88
   b. If unrecoverable error, diagnostic message is posted.
   c. Machine is placed in the WAIT state.

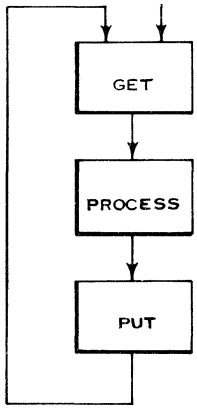
7. Two general causes for I/O interrupts exist, 1. the end of an I/O operation, 2. an I/O error. — 89
   1. a. Channel end routine is entered and pending I/O requests are accepted.
      b. Problem program then continues.
   2. a. Error recovery routine is entered.
      b. If unrecoverable, message is posted and system is placed in WAIT state.

8. By issuing a "load PSW" instruction addressing the "old" PSW for an I/O interrupt. — 89

9. d; There are five specific locations reserved in main storage for storing the "new" PSW's, one for each class of interrupt. These are used to replace the "current" PSW upon an interrupt and permit entry into the interrupt handling routine. — 90

CONTROL INTERRUPTS

A program has previously been defined as a sequence of instructions designed to solve a problem. A problem typical of those solved by a stored program is a payroll application. A payroll problem would consist of (1) getting an employee's record, (2) calculating gross and net pay, and (3) putting the results out in the form of a pay check. The payroll program would get the next employee's record and repeat the process. This sequence of instructions would continue until all employee's records had been processed. Admittedly, this is a gross simplification of a payroll problem. However, most programs can be broken down into the three operations of (1) get record, (2) process record, and (3) put record in output file. These problem solving programs are sometimes referred to as Problem Programs.

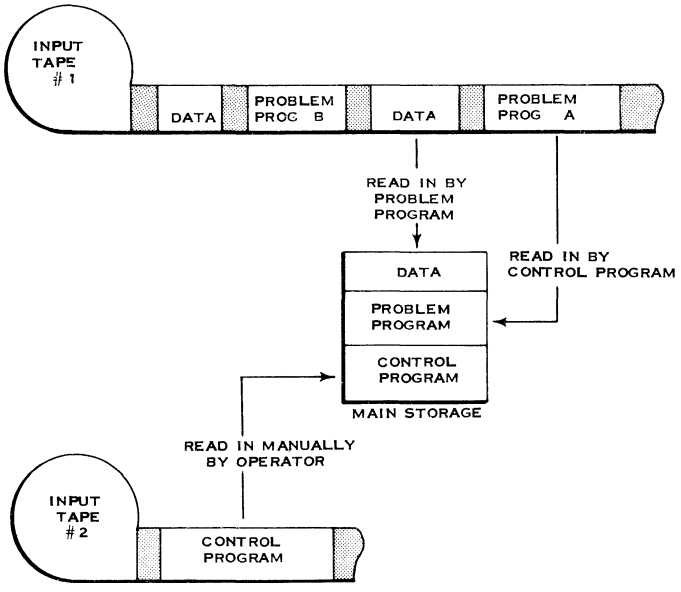


TYPICAL PROBLEM PROGRAM

During the past years, data processing machines have been developed with faster and faster internal processing speeds. As a result, the execution times for these problem programs have been continually reduced with no corresponding reduction in the time it took for an operator to load in the next problem program and manually set up its input data. In some data processing installations, the average "set up" time was about equal to the average "execution" time. In other words, the data processing system was idle about half the time while the operator was "setting up" for the next problem program. Clearly this was an inefficient way to control an installation. In an attempt to reduce this idle time and keep the system running, installations began to use stored programs to control the execution of problem programs. These programs in turn were called Control Programs. Other names used were "monitors" or "supervisors". These Control Programs were at first written only for the requirements of a particular installation. Later, as the similarities between control programs became obvious, IBM began to supply generalized control programs which could then be tailored to the requirements of each installation.

The simplest type of control program would be used to supervise the loading of problem programs. It would operate like this:

- An input tape would be prepared containing the problem programs and their associated data.
- The operator would load the control program into main storage.
- The control program would load in the 1st problem program and then pass control (via a branch) to the problem program.
- The problem program would read in its data and perform its assigned task.
- When the problem program is finished, it would not issue a halt instruction. Instead it would pass control (by branching) back to the control program.
- The control program would then load in the next problem program and pass control to it.
- This operation would continue until all problem programs had been executed.

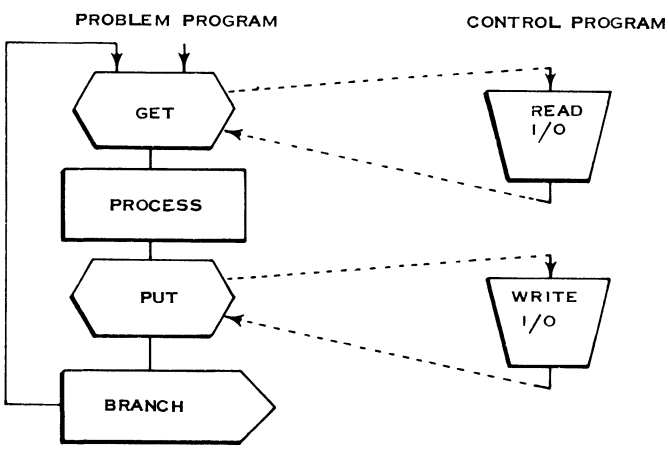


Notice several things about the use of a control program in the preceding example:

- The system never halted between jobs.
- The control program remained in main storage as the problem programs were executed.
- The control program served only as a linkage between jobs. Its only function was to bring in a new problem program as each job was finished.

What you have read is just one example of the use of a control program. Its functions were limited. Thus, the control program was small enough that it could be left in main storage.

Other functions can be included as part of a control program. One such function is the initiation of input-output operations. The problem program is mainly interested in processing data. The actual read and write operation necessary to transfer data between the input-output devices and main storage can be handled by the control program. Each I/O operation that is to be handled by the control program may consist of many instructions. Beside telling the I/O device to start, the instructions check for error conditions, I/O device status, etc.

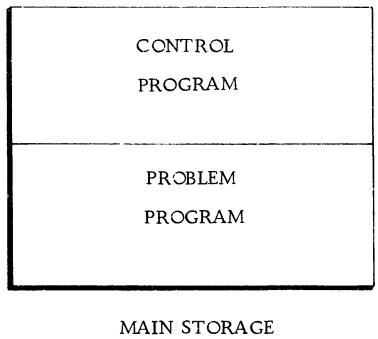


In this function of a control program, control will pass back and forth between the problem and control programs during the execution of the problem program.

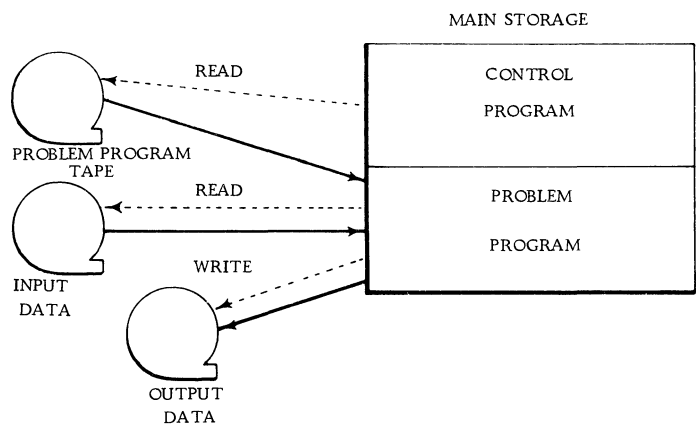
The preceding example differs from the original example of using the control program just to load in new problem programs. In that example, the only time the control program was in control was between jobs. In our new example, the control program will do the following:

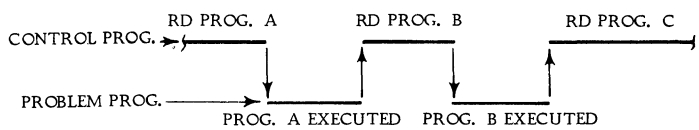
- Read in new problem programs when necessary (same as preceding example).
- Also, it will start the necessary I/O units for handling I/O data during the execution of the problem program.

So in the control program concept, there are always two programs in main storage: the control program and a problem program.

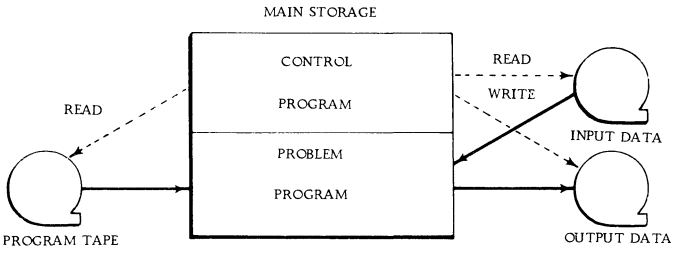


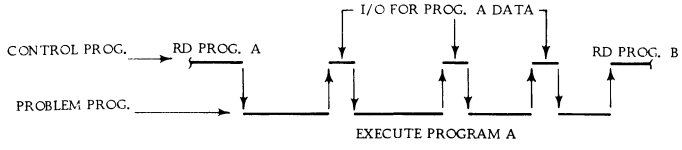
In the simplest utilization of the control program, it was used only to bring in the next problem program. The problem programs handled their own input-output operations.





In its expanded function the control program would not only read in the problem programs but would also handle the input-output data operation during the execution of the problem program. The problem program would transfer control to the control program whenever an input-output operation was necessary.





• • •

1. The preceding sequence chart shows that the control program will not only read in the _____ programs, it will also be used during the execution of the problem program to handle the ____ operation for data.

• • •

problem; I/O

2. The control program can be given other functions as well. In fact, some control programs have reached a very high degree of sophistication. Of course, the more functions that a control program has, the more main storage space it requires. This problem is somewhat solved by placing those sections of the control program that have infrequent usage on a high speed fast access I/O device such as a disk storage unit. Only those sections that are necessary to supervise the running of problem programs are kept in main storage. The portion of the control program that remains in main storage is known as the _____ program.

• • •

supervisor

In review then, control programs have come into general acceptance because of the need to reduce machine idle time and manual intervention and to increase the over-all efficiency of a data processing installation.

3. As the size of control programs increased to meet the demands of more and more efficiency, it became necessary to keep most of the control programs on a high speed I/O device. Preferably, it would be a direct access device such as a drum or a disk storage unit. The portion of the control program that was kept in m_____ s_____ was called the _____. The supervisor would call in the other sections of the control program when necessary.

• • •

main storage; supervisor

Let's continue on now and learn how a System/360 is controlled. As you go through these pages, you will see more clearly how the design of System/360 is such as to facilitate the use of a control program. In fact, the System/360 needs some type of a control program in order to run. These control programs may be written by IBM or by the user. The smaller models of System/360 with limited main storage space may use a control program with a minimum number of functions. Nevertheless, it will be a control program! In the System/360, the part of a control program that resides in main storage is called the supervisor. The remainder of the control program will be assumed to be on a high speed I/O unit.

In the previous section of this text, you learned how the instruction address portion of the "current" Program Status Word (PSW) was used to sequentially fetch instructions. You saw that the sequence of instruction being executed could be changed by "branching". Branch instructions in the System/360 cause the instruction address portion of the PSW to be replaced by the address of the "branch to" location.

INTERRUPT ACTION

In this section, you will be learning how the System/360 can change the sequence of instruction execution without the use of a branch instruction. This method is called an Interrupt.

The System/360 was designed to be used with a control program. One of the reasons why a control program is used at all is to reduce machine idle time. Realizing this, the designers or architects of System/360 did not design a halt instruction. A problem program on a System/360 cannot issue a halt instruction when it is finished because there is no halt code. When finished, the problem program must pass control back to the supervisor. The supervisor is that portion of the control program that resides in main storage.

• • •

1. Which of the following is most correct?

   When a problem program is done on the System/360, the problem program:
   a. Reads in the next problem program.
   b. Issues a halt instruction.
   c. Effects some type of branch to the supervisor.
   d. Reads in the control program.

• • •

c; Effects some type of branch to the supervisor. (The function of the supervisor is then to bring in the next problem program.)

There is another way the "architects" of System/360 have attempted to reduce idle time besides not having a halt instruction. Normally, in past computers, a machine or program check would cause an error stop but not in System/360! A machine check (such as an even number of bits in a byte) or a program check (such as locating a halfword operand on an odd byte address) in the System/360 causes an automatic branch to the supervisor instead of stopping the machine.

2. Which of the following is most correct:

   A machine check on the System/360:
   a. Is taken care of by the problem program.
   b. Is impossible.
   c. Causes an automatic branch to the supervisor which then issues a halt instruction.
   d. Causes an automatic branch to the supervisor.

• • •

d; Causes an automatic branch to the supervisor.

So far we have discussed the use of a control program to bring in new problem programs when the old ones are finished. Since there is no halt instruction in System/360, a problem program when finished must be able to somehow "branch" into the supervisor (that portion of the control program which resides in main storage). We also saw that when a machine or program check occurs, an automatic "branch" to the supervisor usually occurs.

These automatic branches into the supervisor are called Interrupts. That is, the current sequence of instructions is interrupted and an automatic branch is taken to a new sequence of instructions.

3. Usually when a machine check occurs, an automatic branch is taken into the supervisor. This automatic branch is called an _____ .

• • •

interrupt

4. Interrupts can be caused by m_____ checks and p_____ checks.

• • •

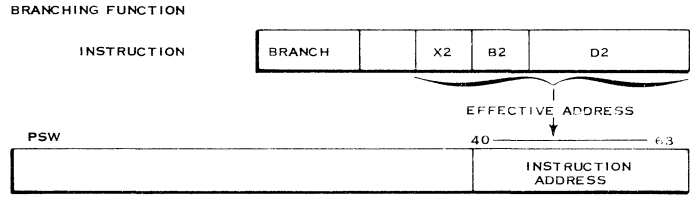
machine; program

5. When a problem program is finished, it signals the supervisor via an i_____.

• • •

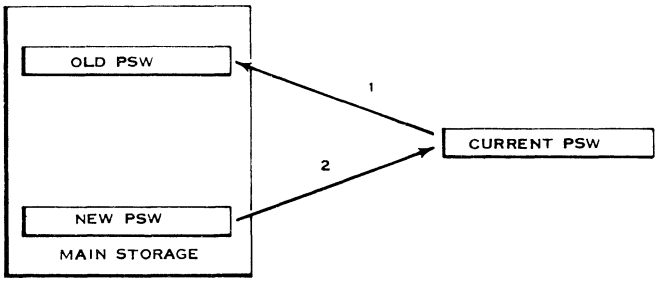
interrupt

An interrupt is quite similar to a branch. However, it does much more than a simple branch instruction. A branch instruction only replaced the instruction address portion of the "current" PSW.



An interrupt replaces the entire "current" PSW. It does this by (1) placing the "old" PSW in main storage and then (2) fetching a "new" PSW from main storage.

INTERRUPT FUNCTION



6. A branch instruction replaces only the _____ _____ portion of the "_____" PSW. The "_____" PSW is the one which is being used to control the program.

• • •

instruction address; "current"; "current"

7. An interrupt replaces the entire "current" PSW. It does this by storing it as the "____" PSW and bringing out a "new" PSW.

• • •

"old"

8. The "____" PSW is now controlling the program and is therefore the new "current" PSW.

• • •

"new"

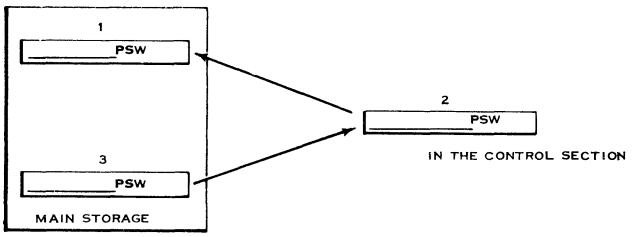
9. The "current" PSW that was controlling the program prior to the interrupt has been stored in main storage. It is, therefore, referred to as the "____" PSW.

• • •

"old"

Actually, "old" and "new" PSWs reside only in main storage. There is only one "current" or controlling PSW and it does not reside in main storage but in the control section of CPU. When an interrupt occurs, the "current" PSW is automatically placed in main storage where it is called the "old" PSW, and a "new" PSW is automatically brought out of main storage and becomes the "current" PSW.

10.



Fill in the blanks above.

• • •

1. "old"; 2. "current"; 3. "new"

11. When an interrupt occurs, the "current" PSW is placed in main storage in the location reserved for the "____" PSW.

• • •

"old"

12. The location of the last instruction executed prior to an interrupt can be determined by examining the "____" PSW.

• • •

"old"

13. The new sequence of instructions will be under control of the PSW brought out from the main storage location reserved for a "____" PSW.

• • •

"new"

14. Assuming that the instruction address portion of a "new" PSW contains 1096, the 1st instruction after an interrupt would be at location _____.

• • •

1096

By now you should have the idea that these "new" and "old" PSWs are in fixed doubleword locations in main storage. Just what are these locations? The answer will depend on just what class of interrupt it is. There are five distinct classes of interrupts:

- External — Can be caused by pressing an interrupt key on the operator's console.
- Supervisor — Caused by an instruction known as "supervisor call".
- Program — Caused by a program check.
- Machine — Caused by a machine check.
- I/O — Can be caused by the end of an I/O operation.

Each of the five classes of interrupts has its own distinct locations for "new" and "old" PSWs as follows:

Interrupt	"Old" PSW	"New" PSW
External	0024	0088
Supervisor	0032	0096
Program	0040	0104
Machine	0048	0112
I/O	0056	0120

As you can see from the above chart, a machine check will cause the "current" PSW to be placed in location 0048 and a "new" PSW will be brought out from location 0112. Notice that these locations are all divisible by eight since they contain doublewords.

Also note that you do not have to memorize these locations. They and others are listed under "Permanent Storage Assignment" at the bottom of page 4 of your Reference Data Card.

Locate them on your card.

15. A program check causes an i_____. This program check interrupt will cause the "current" PSW to be placed in location _____ and a "new" PSW to be brought out from location _____ .

• • •

interrupt; 0040; 0104

16. The handling of program check interrupts, like all interrupts, is taken care of by the _____ program.

• • •

supervisor

17. The portion of the control program that resides in main storage and handles all interrupts is called the _____ program.

• • •

supervisor

18. When a program check occurs, the PSW is stored in the main storage location reserved for program interrupts and becomes the " _____ " (old/new) PSW. A " _____ "(old/new) PSW is then brought out from its reserved location in main storage.

• • •

"old"; "new"

Although an interrupt may be initiated by an instruction (such as the instruction "supervisor call" initiating a supervisor interrupt), the actual storing and loading of the PSW is done automatically by the internal circuitry of the System/360.

19. The storing of the "old" PSW and the loading of the "new" PSW is:
    a. Taken care of by machine instructions in the supervisor program.
    b. Accomplished automatically by the "hardware" (internal circuitry) of System/360.
    c. Taken care of by machine instructions in the problem program.

    Choose one of the above.

• • •

b; Accomplished automatically by the "hardware" (internal circuitry) of System/360.

20. There are _____ classes of interrupts. Each class has its own fixed doubleword locations in main storage for a "_____ " and "_____" PSW.

• • •

five; old and new (in either order)

21. An entry into the correct routine in the supervisor program will be caused by the instruction address portion of the " _____ " (old/new) PSW.

• • •

"new"

22. The particular routine that will be used in the supervisor program is determined by the class of the _____ .

• • •

interrupt

23. The location of the first instruction to be executed after the interrupt is contained in the " _____ " — — — .

• • •

"new"; PSW

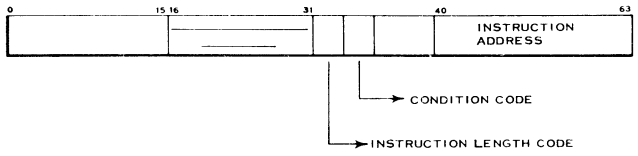
24. The location of the last instruction executed prior to the interrupt can be determined from the " ____ " — — — .

• • •

"old"; PSW

There are five classes of interrupts. Each of these interrupt handling routines would handle the interrupts in a different way. Not all of them would be interested in the last instruction executed. In the case of program, machine or supervisor interrupts, it is an instruction in the problem program that caused the interrupt. In the case of external and I/O interrupts, the problem program did not cause the interrupts. As a result, the supervisor is not concerned about what instruction was last executed in the problem program. It would only want to be able to return to the next instruction.

Another field in the PSW that may be of value to the supervisor is the Interruption Code. It appears in bits 16 - 31 of the PSW.

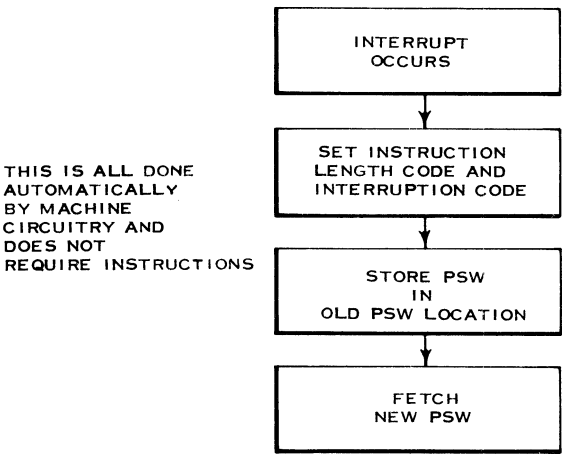
25. 

Fill in the blanks above.

• • •

Interruption Code

When an interrupt occurs, the "current" PSW is stored in one of five locations reserved for the "old" PSW. It is at this time that the interruption code of the PSW is set.



26. The _____ code is contained in the "old" PSW after an interrupt has occurred. The supervisor program can, by examining the "old" PSW, determine the _____ _____ .

• • •

interruption; interruption code

27. The interruption code of the "current" PSW is not set until an _____ occurs.

• • •

interrupt

The interruption code in the "old" PSW gives the supervisor the specific reason for the interrupt. The five classes of interrupts tell the supervisor only the general reason for the interrupt. For instance, the fact that the "new" PSW was brought out of location 0104 will tell the supervisor that the interrupt was caused by a program check. The supervisor still needs to know what type of program check occurred. This is the function of the interruption code in the PSW. By examining the interruption code in the "old" PSW, the program check routine in the supervisor program can tell specifically whether it was a specification, addressing or some other type of exception. In the case of I/O interrupts, the interruption code will tell the supervisor what channel and I/O unit are causing the I/O interrupt.

Refer briefly to the chart called "Code for Program Interruption" at the top of page 6 of your Reference Data Card.

Note that only the right hand 8 bits are shown. These are the only ones which will show any settings in the current system.

28. To determine the specific reason for a program interrupt, the supervisor program would have to examine the " _____ " (old/new) PSW.

• • •

"old"

29. When a program interrupt is caused by a fixed point overflow, the eight low-order bits in the interruption code of the "old" PSW will contain _____ . (Refer to your Reference Data card.)

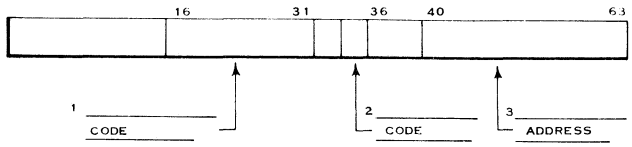
• • •

00001000

For brevity's sake, the interruption code would be represented as 2 hexadecimal digits:

Binary		Hexadecimal
00001000	⟶	08

30. Using your Reference Data Card, identify the names of the indicated portions of the PSW.



• • •

1. Interruption; 2. Condition; 3. Instruction

Since there are five "old" PSW storage locations in main storage, how does the supervisor know which one to use? The answer is, of course, that the class of interrupt which occurs determines which "new" PSW is fetched. Before fetching the "new" PSW, however, the interrupt will store the current PSW in the "old" PSW storage location corresponding to the class of interrupt that occurred. For example, an External Interrupt will store the "old" PSW in storage location 0024 and fetch the "new" PSW from storage location 0088; a Supervisor Interrupt will store the "old" PSW at 0032 and fetch the "new" PSW from 0096; etc. The "new" PSW will cause an entry into the proper routine in the supervisor program.

Interrupt	"Old" PSW	"New" PSW
External	0024	0088
Supervisor	0032	0096
Program	0040	0104
Machine	0048	0112
I/O	0056	0120

31. In the case of an interrupt caused by a machine check, the PSW that was controlling the program prior to the interrupt is stored automatically in location _____ . Then the doubleword at location _____ is brought out and becomes the controlling ("current") PSW.

• • •

0048; 0112

32. This PSW at 0112 will direct the system to that area of the supervisor program that handles _____ checks. The machine check handling routine of the supervisor is written so that the doubleword at location _____ will be processed as the "old" PSW.

• • •

Machine; 0048

Machine check interrupts are caused by various types of machine errors and hardware malfunctions. Ordinarily the system will execute an error recovery routine, if the nature of the machine check indicates a possibility of error recovery. If the error is recovered the problem program continues. If not, or if the error was not of a recoverable kind, a message to the operator is posted and the system goes into the WAIT state.

33. In the case of an interrupt caused by a program check, the PSW that was controlling the program prior to the interrupt is stored automatically in location _____ . Then the doubleword at location _____ is brought out and becomes the controlling PSW.

• • •

0040; 0104

34. This PSW at 0104 will direct the system to that area of the supervisor that handles _____ checks. The program check handling routine of the supervisor is written so that the doubleword at location _____ will be processed as the "old" PSW.

• • •

program; 0040

A program check interrupt is caused by such errors as overflow, improper addressing, and invalid instructions. In such cases, the system will ordinarily print out a diagnostic message and then terminate the program being executed. The contents of storage are printed. (This can be suppressed.) This is the interrupt that is normally of primary concern to the programmer.

35. In the case of a supervisor interrupt the "current" PSW (prior to the interrupt) is stored in location _____ , where it is referred to as "____" __ __ __ . Then the doubleword at location 0096, referred to as the "_____" __ __ __ , is brought out and becomes the controlling or "current" PSW.

• • •

0032; "old" PSW; "new" PSW

A supervisor interrupt is different from the other interrupts in that it is under program control in a "supervisor call". Its normal function is to switch the system from the problem state to the supervisory state, in order that certain instructions (which are executable only in the supervisory state) may be carried out. An example of this is an input-output operation. The programmer is usually not aware of this interrupt.

36. If the interrupt key on the operator's console is depressed, an external interrupt will occur. In this case, the "current" PSW will be automatically stored at location _____ where it is known as the "____" PSW.

• • •

0024; "old"

37. On an external interrupt, the doubleword at location 0088, known as the "____" PSW, is brought out and becomes the new "current" PSW.

• • •

"new"

An external interrupt is initiated by the operator's interrupt key, the interval timer (when a predefined value has been reached), or by an external signal preparing the system for computer-to-computer communications. Operator interrupts are handled by "Operator Communications Routines" for man-machine communications, the interval timer turns control over to a user routine, and an external signal is used to control computer-to-computer operations.

38. An interrupt may also be caused by the end of an I/O operation. An I/O interrupt causes the PSW to be stored at location _____ where it is called the "____" PSW. Then the "____" PSW at location 0120 is brought out and becomes the "current" PSW.

• • •

0056; "old"; "new"

39. This PSW will direct the system to that section of the supervisor program that handles I/O _____.

• • •

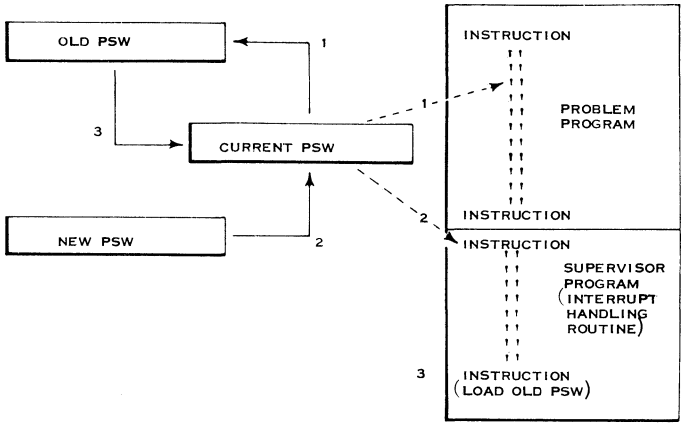
interrupts

An I/O interrupt signals one of two possible conditions:

- An I/O operation has just been completed.
- An I/O device requires attention.

When the interrupt signals the end of an I/O operation, the supervisor will execute a "channel end" routine. This routine will determine whether or not there are any waiting requests to use the channel; if there are, the routine supplies the channel with the next request, thus beginning another I/O operation. Control then passes back to the problem program.

If the interrupt signals that an I/O device needs attention the supervisor executes a routine to determine the nature of the problem. It may be a parity error, a wrong length record, a read or write failure due to bad tape, an end of tape condition, or any of several others. If it is an error that might be corrected an error recovery routine is entered. If the error is corrected control returns to the problem program. If unrecovered, a message is posted for the operator and the system is placed in the WAIT state.

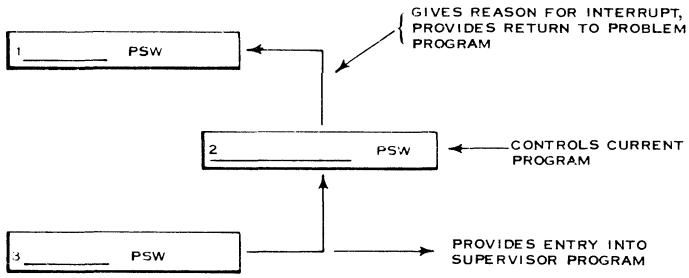


As can be seen from the preceding figure, interrupt action is as follows:

- At the time of the interrupt, the "current" PSW which is controlling the problem program is stored in the "old" PSW location. The "old" PSW gives the reason for the interrupt. It also contains in its instruction address portion the point at which we left the problem program. This is done automatically by machine circuits.
- A "new" PSW is then brought out of storage and becomes the "current" PSW. This "new" PSW points to the first instruction of the interrupt handling routine which is part of the supervisor program.
- After the interrupt has been taken care of, the last instruction of the interrupt handling routine will be "load PSW". This will cause the "old" PSW to once again become the "current" PSW and we are back in the problem program.

We can summarize the interrupt concept by saying that the interrupt or "branch" is completed automatically by the internal circuitry or "hardware" of System/360. The "current" PSW is placed in a fixed location in main storage and becomes the "old" PSW. The "old" PSW basically gives the specific reason for the interrupt and also provides a return to the interrupted program. A "new" PSW is fetched from a fixed location in main storage and becomes the "current" PSW. The "new" PSW provides an entry into the correct routine in the supervisor program.

40.



Identify PSWs 1, 2, and 3, above.

• • •

1. "old"; 2. "current"; 3. "new"

We have identified three items of information contained in the PSW: the Instruction Address, the Condition Code, and the Interruption Code. Many additional items are included in the PSW, most of which are out of the control of the programmer and are rarely used in the average problem programs. As such, they will not be covered here. If required, information on these items is covered in the System/360 Principles of Operation SRL manual.

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]

SR29-0231-5

Assembler Language Coding System Review — Printed in USA SR29-0231-5