IBM Field Engineering Education
Student Self-Study Course

SYSTEM/360

Introductory Programming
Book 2 – Program Control and Execution

Preface

This is Book 2 of the System/360 Introductory Programming Student Self-Study Course.

Course Contents

Book 1:	Introduction	R23-2933
● Book 2:	Program Control and Execution	R23-2950
Book 3:	Fixed Point Binary Operations	R23-2957
Book 4:	Branching, Logical and Decimal Operations	R23-2958
Book 5:	Input/Output Operations	R23-2959

Prerequisites

- Systems experience (1400 series with tapes, 7000 series with tapes) or a basic computer concepts course.
- Book 1 of this Introductory Programming course.

Instructions to the student and advisor

- This course is to be used by the student in accordance with the procedure in the Instructions to the Student section in Book 1 of this course.
- The course is to be administered in accordance with the procedure in the System/360 Introductory Programming Administrator Guide, Form #R23-2972.

This edition, R23-2950-1 is a minor revision of the preceding edition, but it does not obsolete R23-2950-0. Numerous changes of a minor nature have been made throughout the manual.

Issued to: _____

Branch Office: _____ No: _____

Department: _____

Address: _____

If this manual is mislaid, please return it to the above address.

Copies of this and other IBM publications can be obtained through IBM Branch Offices. Address comments concerning the content of this publication to: IBM, FE Education Planning, Dept. 911, Poughkeepsie, N. Y., 12602

How to use this book

There are four sections to this text. At the beginning of each section, there is a list of Learning Objectives. These are items that you will be expected to learn as a result of studying that particular section. At the end of each section (or subsection) is a list of Review Questions so that you can evaluate your progress. You will go through this book in a serial fashion. That is, you will not be expected to skip or branch around. The answer to each frame is in the next frame. You may find it helpful to use a standard IBM card to cover the answers as you read the frames.

Periodically, as you go through this book, you will be directed to study areas of the System/360 Principles of Operation manual. This will help you to become familiar with the manual so that it may be used as reference material at a later date.

THE CONTENTS OF THIS BOOK

This book deals mainly with instruction formats and the Program Status Word.

System/360 Program Control and Execution

- Section I: Instruction Formats
  Section II: Instruction Sequencing and Branching
  Section III: Interrupts
  Section IV: Storage Protection

SECTION I LEARNING OBJECTIVES

At the end of this section, you should be able to:

1. State that instructions:
   a. can be one, two, or three halfwords in length.
   b. must reside in main storage on halfword boundaries.
2. State that the first byte of every instruction is the Op code.
3. Convert hexadecimal Op codes to actual machine language.
4. Describe how the Op code states:
   a. Instruction Length
   b. Format
   c. Specific Instruction
5. Describe how the instruction specifies the address of a general register.
6. Describe how an effective main storage address is generated using a displacement field and a base address located in a general register.
7. Describe how programs can be relocated by updating the base register contents.
8. Describe how an effective address may be further indexed with an index factor in a general register.
9. Given an Op code, determine the location of 1st and 2nd operands.
10. State that, with the exception of store-type operations, the result is located in the first operand.
11. Describe the five instruction formats.

Instruction Formats

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.

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

halfwords

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

When both operands are in main storage, a total of 48 bits are needed for the addresses. Accordingly, the System/360 uses its longest instruction which is _____ halfwords in length.

three

When only one of the operands is in main storage, a shorter instruction can be used. However, one halfword cannot contain a 24-bit binary address; therefore, the System/360 also uses instructions which are _____ _____ in length.

two halfwords

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.

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

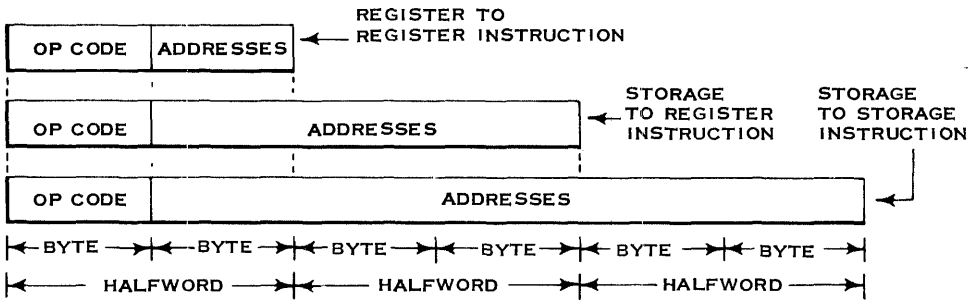
halfwords specification

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

OP CODE

specification

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.



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

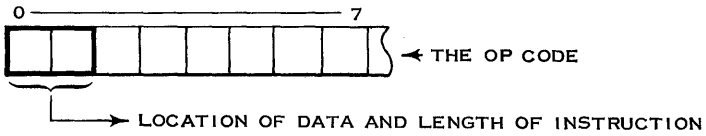
All instructions have an Op code which is contained in the high-order _____.

byte

Op codes in the System/360 give specific information:

1. The Op code specifies the operation such as add, subtract, or branch.
2. The Op code specifies whether the data is variable or fixed in length.
3. The Op code specifies whether the data is in binary or decimal format.
4. The Op code specifies whether the operands are in main storage or general registers. The Op code, however, does not give the address of data. It only says the data is in main storage or in general registers.
5. The Op code specifies the length of the instruction of which it is a part.

As you can see, there is much information in the eight bits which make up the Op code. So let's break down the Op code byte and see how it gives us this information.



Bits 0 and 1 of the Op code specify whether data is in main storage or in the general registers. Since the instruction length depends on the location of the data, the instruction length would also be specified by bits ___ and ___ of the Op code.

0, 1

There are four possible combinations of bits 0 and 1 of the Op code: 00, 01, 10, 11.

If both bits are zero (00), both operands are in general registers and the instruction is _____ _____ in length.

---

one halfword

If both bits are one (11), both operands are in main storage and the instruction is _____ _____ in length.

---

three halfwords

If bits 0 and 1 of the Op code are either 01 or 10, only one of the operands may be in main storage and the instruction is _____ _____ in length.

---

two halfwords

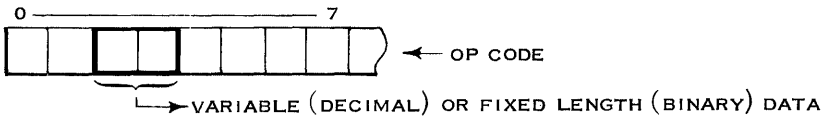
For the following Op codes, fill in the instruction length:

Op Codes	Instruction Length
a. 0 1 0 0 0 1 1 1	_____
b. 0 0 0 1 1 0 1 0	_____
c. 1 1 1 1 1 0 0 0	_____
d. 1 0 0 1 1 1 0 0	_____

---

two halfwords
one halfword
three halfwords
two halfwords

You have just learned how bits 0 and 1 of the Op code specify the location of operands and the length of instruction. Now, let's look at bits 2 and 3. We won't be concerned with the actual coding of the bits, but rather, their purpose.



As can be seen above, bits 2 and 3 of the Op code are used to specify the type of data. In the case of fixed length data, bits 2 and 3 of the Op code would further specify whether the operands were halfwords or fullwords.

In an instruction that is used with fixed length (binary) operations, bits 2 and 3 of the Op code would be _____ (the same/different) for halfword and fullword operands.

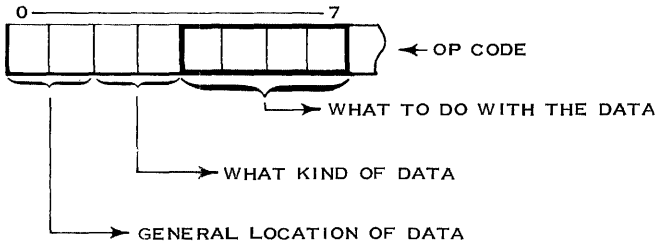
---

different

Instruction length is specified by bits ___ and ___ of the Op code, while the type of data is specified by bits ___ and ___.

0, 1
2, 3

The last thing the Op code must indicate is the specific task to be performed such as add or subtract. This is done by bits 4 - 7 of the Op code as shown below. The actual meaning of the various bit combinations (add, subtract, etc.) will be covered later.



The fact that both operands are in general registers would be indicated by bits ___ and ___ of an instruction's Op code.

0, 1

The fact that an operand is either a halfword or fullword is indicated by bits ___ and ___ of an instruction's Op code.

2, 3

The fact that the two operands are to be added together is indicated by bits ___ through ___ of the Op code.

4, 7

Quite often the Op code byte is referred to with two hexadecimal digits rather than eight binary bits. Given the hexadecimal Op codes below, write the Op code as eight binary bits and indicate the length of the instruction.

Hex Op Code	Actual Op Code	Length in Halfwords
5A	_____	_____
4A	_____	_____
FA	_____	_____
18	_____	_____
58	_____	_____
94	_____	_____

Hex	Actual Op Code	Length in Halfwords
5A	0 1 0 1 1 0 1 0	Two Halfwords
4A	0 1 0 0 1 0 1 0	Two Halfwords
FA	1 1 1 1 1 0 1 0	Three Halfwords
18	0 0 0 1 1 0 0 0	One Halfword
58	0 1 0 1 1 0 0 0	Two Halfwords
94	1 0 0 1 0 1 0 0	Two Halfwords

OPERAND ADDRESSING

At this point, you should have a good idea of how the 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.

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

Each general register is one _____ in length.

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

---

sixteen
0, 15
word
index

As you learned when you were first introduced to the general registers, they are addressed by a 4-bit binary address. The address of general register 7 is _____.

0111

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

---

one halfword

The first byte of an instruction is the ____ _____.

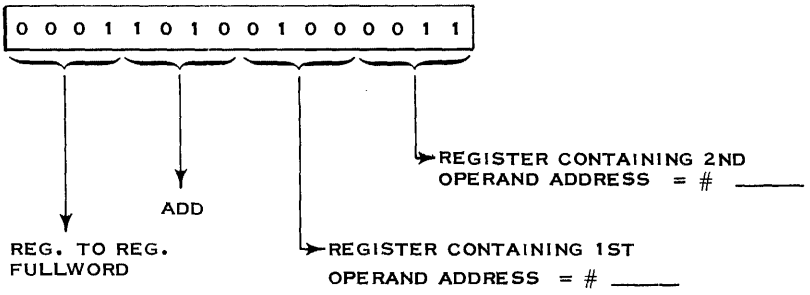
Op code

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

---

addresses

Given the following instruction, state the number of the general registers involved.



1st operand register 4
2nd operand register 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. In the smaller models of System/360 (such as the model 30 with approximately 8K storage), the amount of main storage space is definitely limited. One solution would be to use 24-bit addresses on the larger models such as model 70 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 instructions.

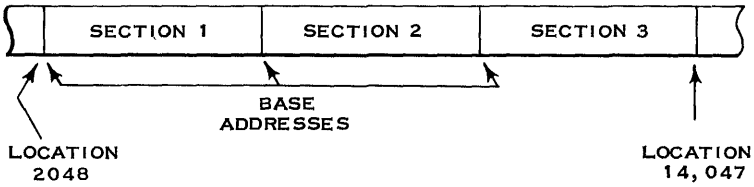
It is also desirable that any time the program is loaded into the computer, the program can be put in a different area of main storage. We would like to do this without having to change the addresses in each instruction. This is known as program relocation, which is a valuable tool in IBM's latest programming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

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

---

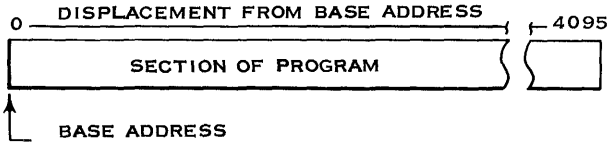
base address

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.



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

1122

Assuming a base address of 6144, the displacement for location 9170 is _____.

3026

Assuming a base address of 6144, the displacement for location 6144 is _____.

---

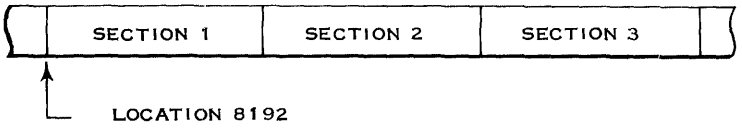
zero

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.



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

12,288; 16,384

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.

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

---

base addresses
displacements

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

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 the Appendix of the IBM System/360 Principles of Operation manual. Convert 4095 to hexadecimal and then to binary.)

4095 (Decimal) = FFF (Hex) = 1 1 1 1 1 1 1 1 1 1 1 1 (Binary)

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

The use of a base address and a displacement certainly make 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.

12
24

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.

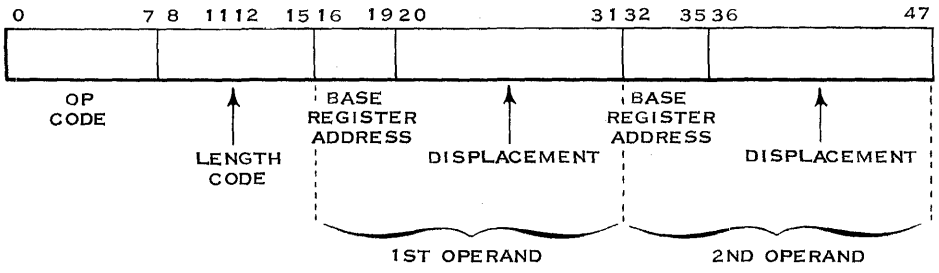
When both of the operands are in main storage, the instruction is ____ (1/2/3) halfword(s) in length.

3

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

1.	8 bit	Op Code
2.	8 bit	Length Code
3.	4 bit	1st Operands Base Register Address
4.	12 bit	1st Operands Displacement
5.	4 bit	2nd Operands Base Register Address
6.	12 bit	2nd Operands 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.

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

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

11

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

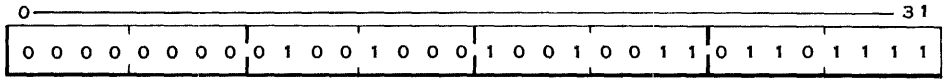
The base address of 2048 in the previous problem is a ____ bit binary address and appears in bits 8 - 31 of general register 11.

24

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

24

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 binary address would be _____.



GENERAL REGISTER 11

---

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

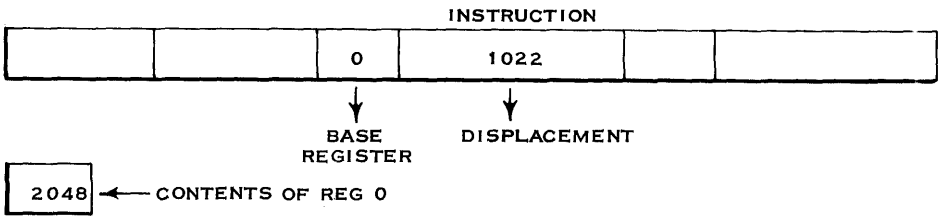
If you missed the above, check your arithmetic. Remember that you were attempting to add the 12 binary bit displacement to the low-order 24 binary bits of the base register.

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

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.



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 is ignored.

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

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

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.

---

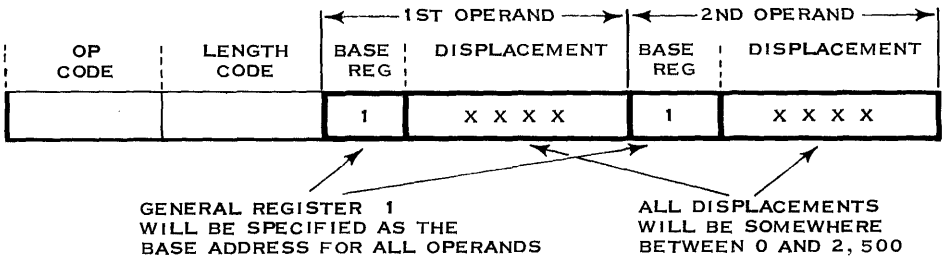
instructions
general

5000 is the b_____ a_____ and is put in general register #1.

---

base address

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



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

---

one

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

---

beginning

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

---

two

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:

1. Storage addresses are generated by adding a displacement value to a base address.

2. The instruction contains the displacement value as well as the address of the general register containing the base address.

3. The general register that contains the base address is called the base register.

4. Only registers 1 - 15 can be used as base registers.

5. If register 0 is specified as the base register, its contents are ignored. Instead, a base address of 0 is used.

6. 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.

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

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

1, 15

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

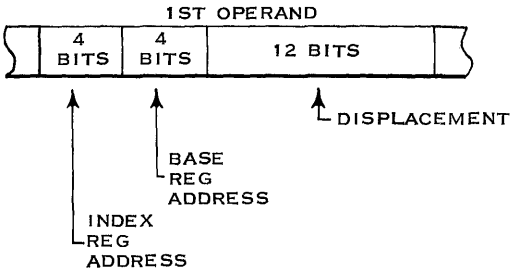
---

zero

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:

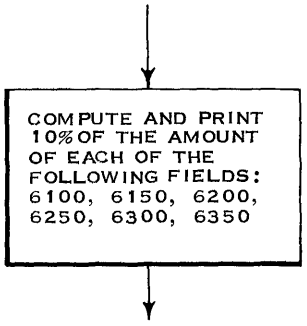


The effective storage address would be generated by adding:

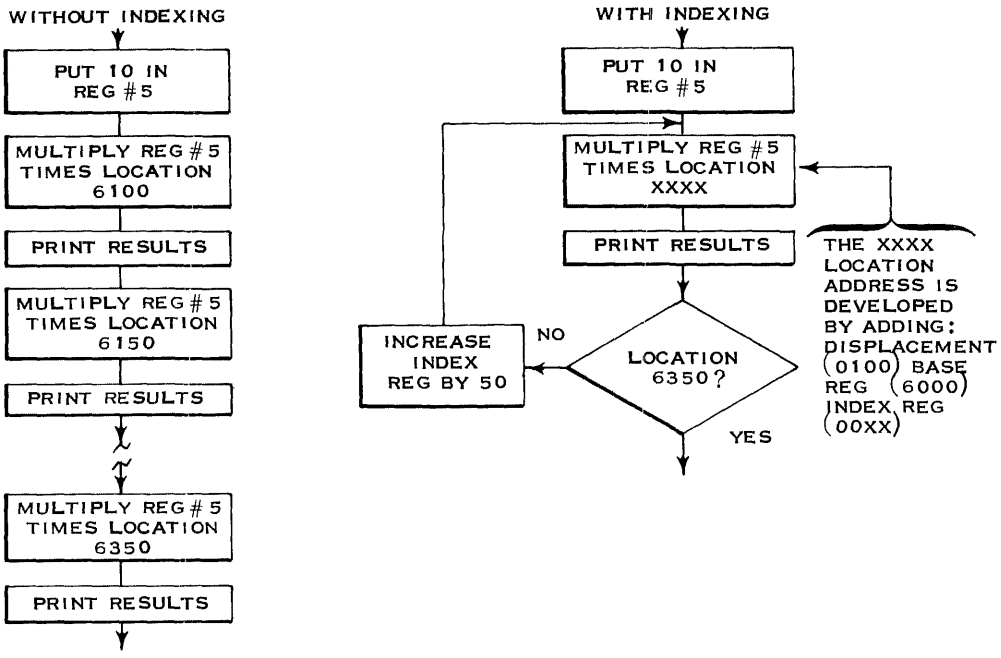
Displacement + Contents of Base Register + Contents of Index Register

---

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.



The index factor is used where a number of similar instructions can be replaced by one i_____ that has its operand address modified.

---

instruction

Given an address portion of

6	7	1012

1ST 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

At this point, let's take a look at different types of instruction formats of the System/360. 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:

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

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

RR FORMAT

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

---

register-to-register
one halfword

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. For the R R format, bits 0 and 1 will be ____.

Op code
00

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

R R

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

RR FORMAT

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

1st

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

OP	R1	R2
ADD	4	5

4

Instructions, which are two halfwords in length, may have three different formats. As you recall, if bits 0 and 1 of the Op code are either 01 or 10, the instruction is two halfwords in length. Furthermore, if bits 0 and 1 of the Op code are 01, it indicates a specific format known as the R X format.

RX FORMAT

OP CODE	R1	X2	B2	D2
	GEN REG	INDEX REG	BASE REG	DISPLACEMENT

The ___ ___ format is used for storage-to-register operations. The register address is specified by the ____ field.

R X
R1

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

---

index

When the effective storage address includes an indexing factor, the instruction is said to be the ___ ___ format and has ____ in bits 0 and 1 of the Op code.

RX
01

In the RX format, the index register is specified by the ____ field while the base register is specified by the ____ field.

X2
B2

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

RX FORMAT

OP CODE	R1	X2	B2	D2

ADD	3	7	4	1024

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

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

3
3

Register-to-Register operations use the ___ ___ format.

R R

Label the fields in the RR format.

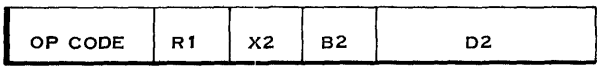
The RR format is identified by a ____ in bits 0 and 1 of the Op code.

00

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

R X

Label the fields in the RX format.

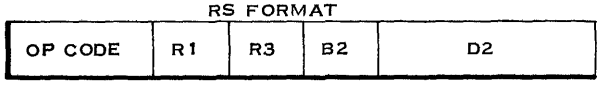


The RX format is identified by a ____ in bits 0 and 1 of the Op code.

01

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



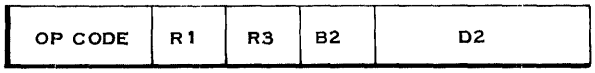
In the ___ ___ format, the effective address of the 2nd operand is obtained by adding the contents of the _____ _____ to the _____.

R S
base register
displacement

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

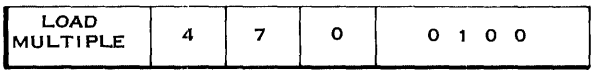
R1
R3

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 specified 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:



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.

In the preceding example, registers ___ through ___ will be loaded with the data in main storage.

4, 7

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

0115; Each storage address represents a byte of data.

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

---

does not

A 10 in bits 0 and 1 of the Op code identifies the ___ ___ format.

R S

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

In the ____ Format, one operand is in main storage while the immediate operand is in the _____.

S I
instruction

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

In the SI format, the immediate operand is fixed in length and is one _____ long.

byte

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

storage

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

In the above instruction, the contents of the ____ field will be placed in storage location _____.

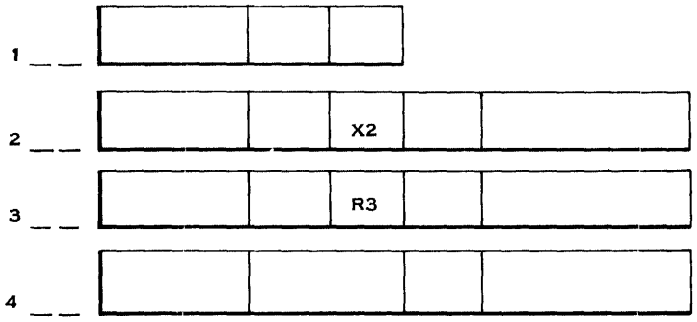
I2
1000

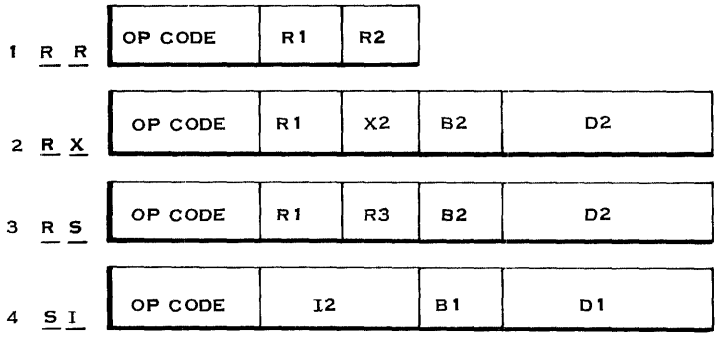
The SI format is identified by a 10 in bits 0 and 1 of the Op code, just like the ___ ___ format.

R S

Since bits 0 and 1 of the Op code are the same for both the RS and SI formats, the remaining bits of the Op code 4 - 7 would have to tell the computer whether it is the RS or SI format.

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





Which format does not involve a storage operand? _____

Which format does not involve a general register operand? _____

Which format includes an indexing factor when addressing main storage?

_____

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

RR
SI
RX
no

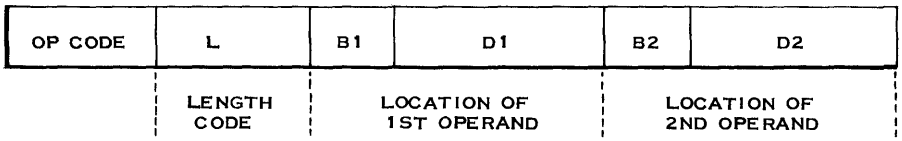
In the four previous formats, the operands were of fixed length. Now let's take a look at the instruction format for variable length operations.

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:



The ___ ___ format, because it must address two storage operands, is _____ (one/two/three) halfwords in length.

S S
three

Because both operands are in storage and the instruction is three halfwords in length, the SS format is identified when bits 0 and 1 of the Op code contain ____.

11

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

---

is not

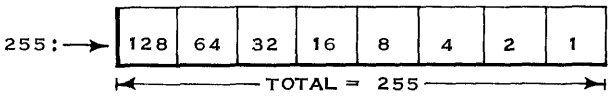
The length code for the variable length storage operands is in the 2nd byte of the ___ ___ format.

S S

Label the fields of the SS format.

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

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.

A length code of 33 would indicate an operand length of ____ bytes.

34

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

---

zero

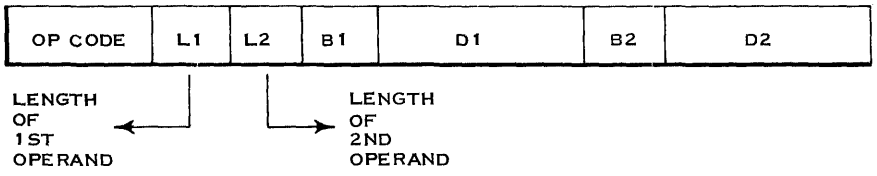
The maximum operand length that can be expressed by an 8-bit length code is _____ bytes.

256

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:



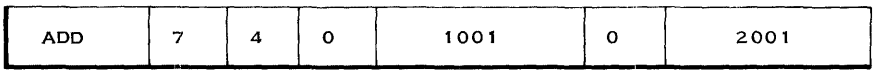
With the length code split into two 4-bit fields, the maximum length of arithmetic variable length operands is ____ bytes.

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

Given the following binary length code: 1 0 1 0 0 1 1 1
The 1st operand is ____ bytes long.
The 2nd operand is ____ bytes long.

11
8

Given the following SS-type add instruction



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

2005
1008

At this point, you have learned the five instruction formats. List them.

1. ___ ___
2. ___ ___
3. ___ ___
4. ___ ___
5. ___ ___

---

1. R R
2. R X
3. R S
4. S I
5. S S

For the given instruction formats, specify

a. instruction length in halfwords.
b. bits 0 and 1 of the Op code.
c. location of 1st operand, such as storage or register.

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

RR	1,00	register
RX	2,01	register
RS	2,10	register
SI	2,10	storage
SS	3,11	storage

Go to the IBM System/360 Principles of Operation manual and briefly study the following areas of the System Structure section:

Program Execution
Instruction Format
Address Generation

REVIEW QUESTIONS ON INSTRUCTION FORMATS

- Try to answer the questions without referring to the material. However, if you do require aid, refer to this book and/or the System/360 Principles of Operation manual and consider reviewing the area where aid is required.

1. Instructions are a multiple of _____ in length.

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

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

4. For the following Op codes expressed hexadecimally, indicate the binary bit structure of the Op code and its length in halfwords.

	Hex	Binary	Length
a.	1A	_____	_____
b.	56	_____	_____
c.	9C	_____	_____
d.	FD	_____	_____

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

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

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

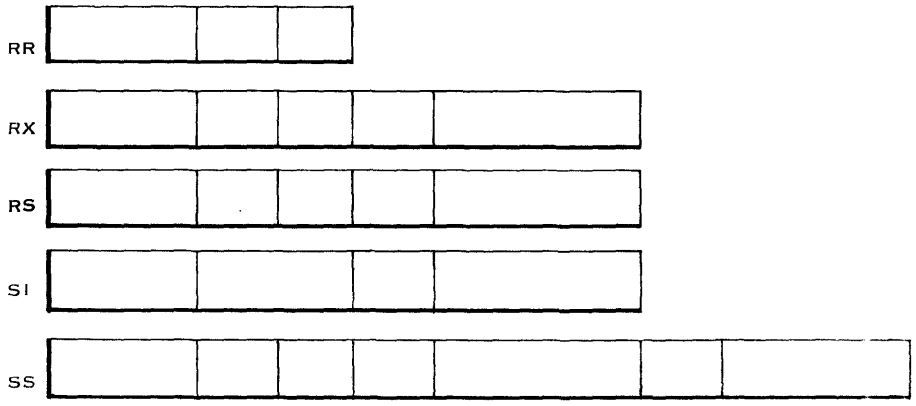
8. A program can be relocated in storage by changing the contents of the _____ _____.

9. The displacement has a range of 0 to _____ bytes.

10. Only general registers ___ through ____ can be used as base or index registers.

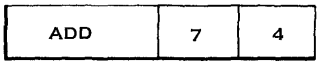
11. What happens if register 0 is specified as a base or index register?
_____
_____.

12. Label the fields of the following formats:



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

14. Given the following RR type instruction:



The result of the addition will replace the contents of register ___.

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

16. Only the ___ ___ format uses an index register for address generation.

17. Only the ___ ___ format involves variable length data.

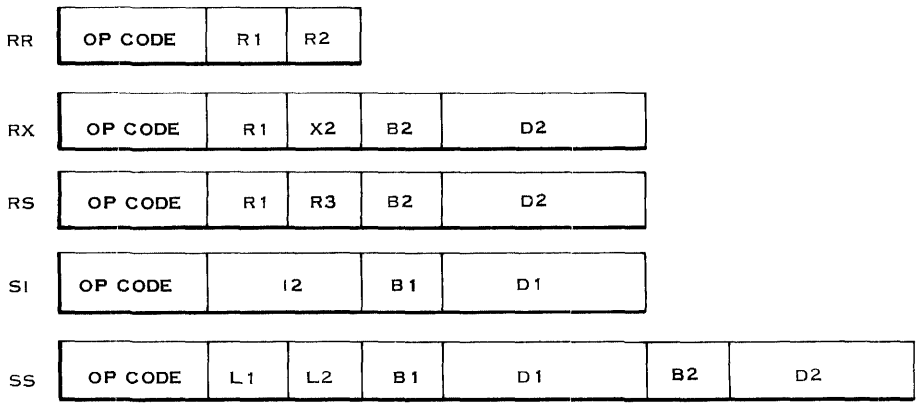
18. What is the relationship between the number in the length code of the SS format and the number of bytes in the data field. _____ _____ _____.

ANSWERS TO REVIEW QUESTIONS

1. halfwords
2. two, specification
3. Op code
4.

	Hex	Binary	Length
a.	1A	0 0 0 1 1 0 1 0	1
b.	56	0 1 0 1 0 1 1 0	2
c.	9C	1 0 0 1 1 1 0 0	2
d.	FD	1 1 1 1 1 1 0 1	3

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



13. 1st

14. 7

15. instruction, byte

16. RX

17. SS

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

System/360 Program Control and Execution

Section I: Instruction Formats
● Section II: Instruction Sequencing and Branching
Section III: Interrupts
Section IV: Storage Protection

SECTION II LEARNING OBJECTIVES

At the end of this section, you should be able to:

Referring to the program status word:

1. Describe how the PSW affects the sequential nature of instruction fetching.

2. Describe how branching affects the address portion of the PSW.

3. Show how the CPU status is indicated by the condition code in the PSW.

4. Describe the "branch on condition" instruction.

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.

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

---

instruction

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

---

program

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

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

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)

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

The operation specified by the instruction is performed during ___ _____.

E Time (or Execution Time)

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.

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

E	Instructions are information read out of main storage during ___ time.
I	An instruction may be treated as data and changed if it is read out during ___ time.
E	Information in main storage is treated as either _____ or _____ depending on when the information is read out of main storage.
instructions data	The instructions of a stored program are generally read out and executed in a _____ (random/sequential) manner.
sequential	The sequential manner of instruction fetching and execution can be changed by instructions known as _____ instructions.
branch	When the next instruction is read out of a non-sequential location in main storage, we say that _____ occurs.
branching	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 ____ _____. The other portion of the instruction generally tells the computer where the data is located. For this reason, it is sometimes called the _____ portion.
Op code address or operand	Much of the remainder of this book is devoted to the Program Status Word. You may find that the total or over-all function of the PSW is difficult to keep in mind. This is because the PSW has such a large number of detailed functions. As you study the PSW, do the following: 1. Remember that many of the detailed functions of the PSW are not meant to be completely memorized and are, therefore, listed as reference in the Principles of Operation manual. 2. Do not hesitate to go back and review or refer to the Principles of Operation manual to help you keep an over-all view of the PSW.

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:

1. The location of the next instruction.
2. 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.

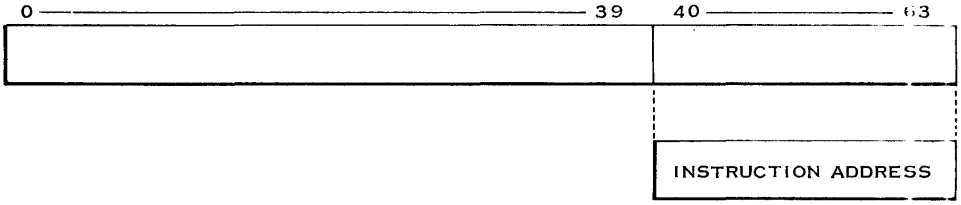
The PSW is a doubleword and contains ___ bytes of information.

8

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.



Bit positions 40 - 63 of the PSW are called the _____ _____ portion.

---

instruction address

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.

---

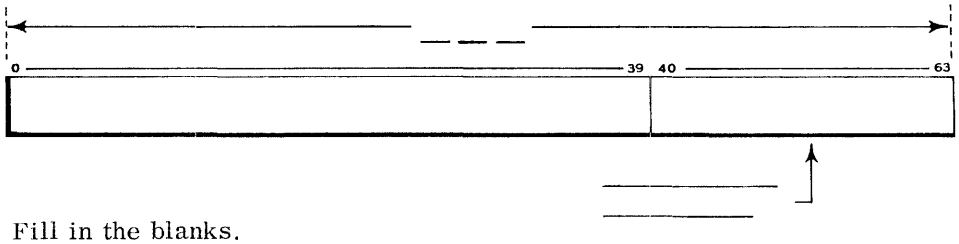
instruction address

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

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

PSW



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.

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

"current"

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.

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

---

is not

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.

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.

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

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

"current" PSW 1006

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

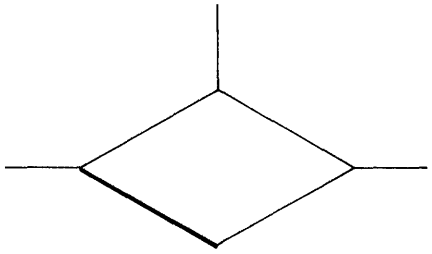
The instruction address in the "current" PSW is increased by 2, 4, or 6 depending on bits ___ and ___ of the current instruction's ____ _____.

0, 1 Op code

If bits 0 and 1 of the current instruction's Op code contain 11, the instruction address in the "current" PSW will be increased by ___.

6

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 is replaced by the address of the instruction being branched to.

For example:

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

1004

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

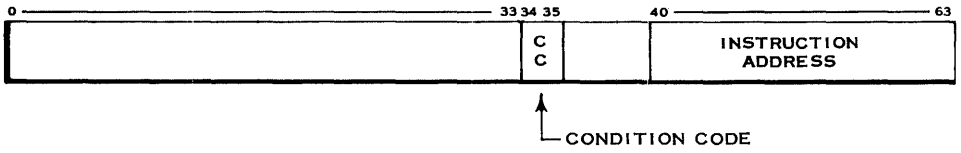
"current"
2000

In the preceding example, bits 40 - 63 (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 bits ____ through ____ (instruction address portion) of the "current" PSW.

PSW - CONDITION CODE

40, 63

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.



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

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

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

---

four

List the four possible settings for the condition code.

1. _____ 2. _____ 3. _____ 4. _____

---

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

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. 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

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

---

zero

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

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

---

positive number

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

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

01

Indicate the condition code setting for the following results.

Algebraic Result	Condition Code
zero	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.

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
low	01
high	10

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

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

---

compare

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.

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

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

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

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

---

high

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

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	You should now have a good idea of how the condition code in bits 34 and
b. negative	low	35 of the PSW indicates the status of the central processing unit. The
c. positive	high	condition code is used to indicate more than just the result of an algebraic
d. overflow	not possible	or comparison operation. You will learn these other possible indications as you learn the individual instructions of System/360.

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 of the instructions of the System/360 is an instruction 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.

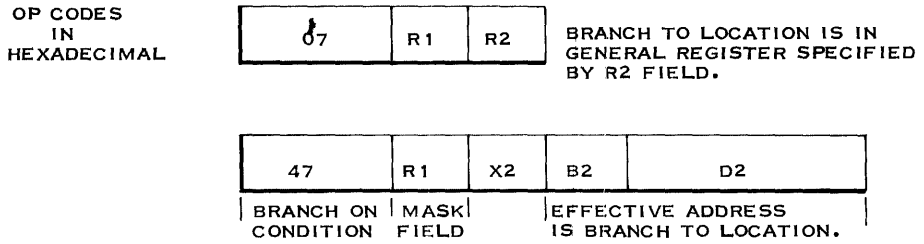
The condition code in the ___ ___ ___ can be tested by means of an _____.

PSW instruction

The 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

The "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.



In the "branch on condition" instruction, the condition code in the PSW is tested against the R____ field in the instruction.

R1

The R1 field in the "branch on condition" instruction is sometimes referred to as the "mask" field. The condition code is tested by being matched against the R1 or _____ field.

---

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 setting (00, 01, 10, 11) at any one time. This setting can represent only one thing depending on the last instruction that affected the condition code.

Circle one of the following answers:

At any one time, a condition code setting of 00 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

The instruction that tests the condition code is known as "_____ ____ _____."

"branch on condition"

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

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

In either the RR or RX format, the R1 field, also called the _____ field, is tested against the PSW's _____ _____.

---

mask
condition code

Bit positions 8 - 11 of a "branch on condition" instruction are called the _____ field.

---

mask or R1

The four bits of the mask field are tested against the _____ possible settings of the PSW's condition code.

---

four

The mask field is tested against the condition code according to the following chart:

Mask Field	Condition Code
1000	00
0100	01
0010	10
0001	11

As you can see from the above, any of the possible PSW condition code settings can be tested by setting the appropriate bit of the instruction's mask field.

If bits 8 - 11 of a "branch on condition" instruction contain 1000, a branch will occur only if the PSW condition code has a setting of ____

00

If the condition code were 01 and the mask field were 0010, a branch _____ occur.

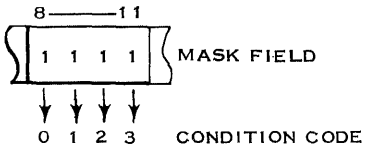
---

would not

Sometimes the four possible settings of the PSW condition code are referred to by hexadecimal digits.

Condition Code	Referred To As
00	0
01	1
10	2
11	3

The bits of the "branch on condition" mask field correspond to the condition code settings in a left-to-right fashion.



To test for a specific condition code setting, the corresponding bit of the mask field must contain a ____ (0/1).

1

If the mask field contained 0000, _____ (all/none) of the possible condition code settings will be tested.

---

none

A branch would never occur if the instruction's mask field contains _____. If the mask field contains 1111, _____ of the possible condition code settings will be tested.

0000
all or any

Since the condition code will always contain one of its four possible settings, a "branch on condition" instruction mask field of 1111 will always result in a _____.

---

branch

SUMMARY:

With its 4-bit mask field, the "branch on condition" instruction has many uses:

a. It can be used as a no-Op instruction, by having a mask field of 0000.
b. It can test for a specific result (such as an equal compare) by setting one of the bits of the mask field (1000).
c. It can test for a multiple result (such as an equal or low compare) by setting two or more bits of the mask field (1100).
d. It can be used as an unconditional branch by having a mask field of 1111.

Go to the IBM System/360 Principles of Operation manual and briefly study the following areas of the System Structure section.

Sequential Instruction Execution
Branching
Program Status Word

REVIEW QUESTIONS ON INSTRUCTION SEQUENCING AND BRANCHING

- Try to answer the questions without referring to the material. However, if you do require aid, refer to this book and/or the System/360 Principles of Operation manual and consider reviewing the area where aid is required.

1. PSW is short for _____ _____ _____.

2. The PSW is ____ bits long.

3. The address of the next instruction to be fetched is contained in bits ____ through ____ of the PSW.

4. After an instruction has been fetched (read out), the instruction address portion of the PSW is usually incremented by ___, ___, or ___.

5. The amount to increase the instruction address in the PSW is determined by bits ___ and ___ of the instruction's ____ _____.

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

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

8. Branching is accomplished by replacing the _____ _____ in the PSW with the "branch to" _____.

9. The condition code is in bits ____ and ____ of the PSW.

10. The condition code has ___ possible settings.

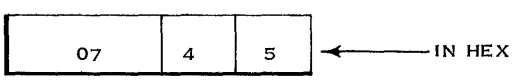
11. The condition code in the PSW is changed by _____ (all/some) instructions that are executed.

12. Indicate the condition code setting after the following results:

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

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

BRANCH ON CONDITION



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

BRANCH ON CONDITION

07	C	8	← IN HEX

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

BRANCH ON CONDITION

07	F	7	← IN HEX

16. If the following instruction resulted in a branch, the instruction address in the PSW would be replaced by bits ___ through ____ of _____.

BRANCH ON CONDITION

07	8	5	← IN HEX

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

BRANCH ON CONDITION

47	8	4	7	0 0 0	← IN HEX

ANSWERS TO REVIEW QUESTIONS

1. Program Status Word

2. 64

3. 40, 63

4. 2, 4, 6

5. 0, 1, Op code

6. "current"

7. some type of internal register or storage area

8. instruction address, address (location)

9. 34, 35

10. 4

11. some

12. Condition Code

    00
    01
    10
    11

13. < zero

14. equal or low

15. always

16. 8, 31, general register 5

17. The effective address generated by adding the contents of register 4 and register 7 and a displacement factor of 0.

System/360 Program Control and Execution

Section I: Instruction Formats
Section II: Instruction Sequencing and Branching
● Section III: Interrupts
Section IV: Storage Protection

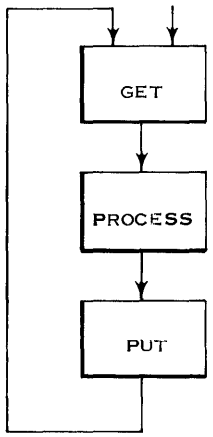
SECTION III LEARNING OBJECTIVES

At the end of this section, you should be able to:

1. Relate the handling of interrupts to a supervisor or control program.
2. Describe how an interrupt affects the current PSW.
3. Define: Current, Old, New PSW's.
4. State the five (5) classes of interrupts.
5. Describe how the old PSW shows the cause of the interrupt.
6. Describe how the current PSW allows or masks interrupts.
7. State which interrupts cannot be masked.
8. Relate the problem state bit and the wait state bit to interrupts.

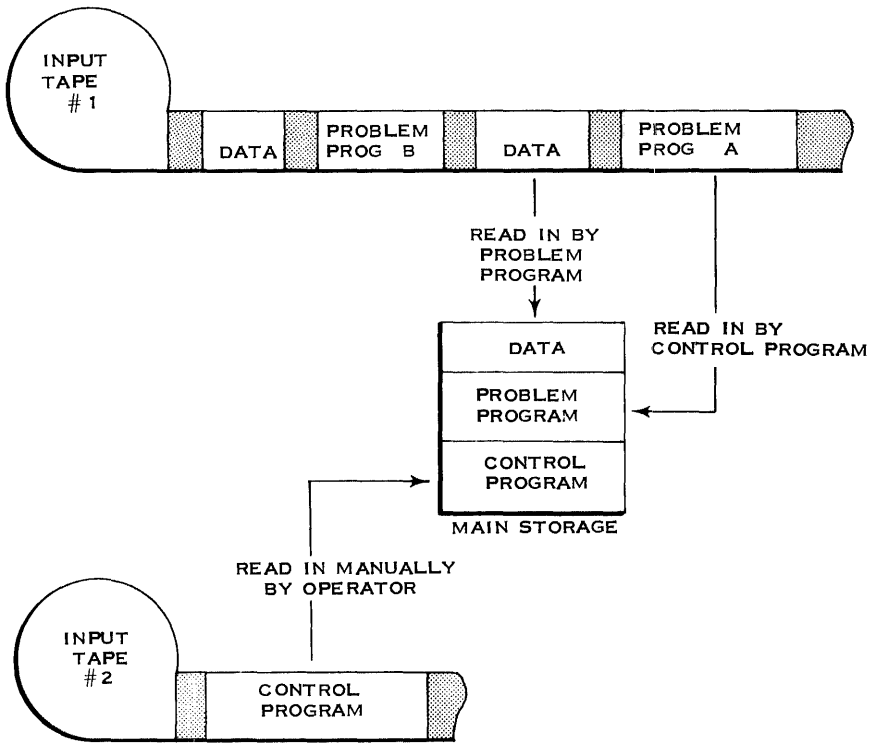
Interrupts

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 problem 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

Another example of a Problem Program is an assembly program. Here the problem is different, but the three operations are basically the same. The problem consists of (1) getting a symbolic (source language) statement, (2) processing it by translating the statement into machine language and (3) putting the results in the output file (object program).


INPUT
TAPE
#1
DATA
PROBLEM
PROG B
DATA
PROBLEM
PROG A
READ IN BY
PROBLEM
PROGRAM
READ IN BY
CONTROL PROGRAM
DATA
PROBLEM
PROGRAM
CONTROL
PROGRAM
MAIN STORAGE
READ IN MANUALLY
BY OPERATOR
INPUT
TAPE
#2
CONTROL
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:

1. An input tape would be prepared containing the problem programs and their associated data.

2. The operator would load the control program into main storage.

3. The control program would load in the 1st problem program and then pass control (via a branch) to the problem program.

4. The problem program would read in its data and perform its assigned task.

5. 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.

6. The control program would then load in the next problem program and pass control to it.

7. This operation would continue until all problem programs had been executed.

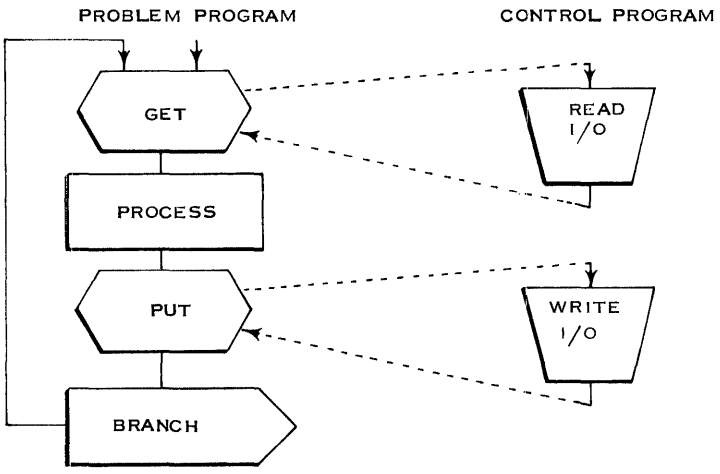
Notice several things about the use of a control program in the preceding example:

1. The system never halted between jobs.

2. The control program remained in main storage as the problem programs were executed.

3. 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. As such the entire control program 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 operations 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. Besides telling the I/O device to start, the instructions check for error conditions, I/O device status, etc.

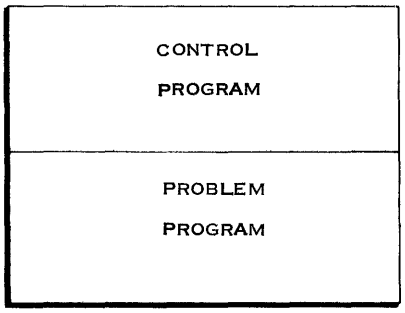


In this function of a control program, control will pass back and forth between the problem and control programs during the execution of the problem program.

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:

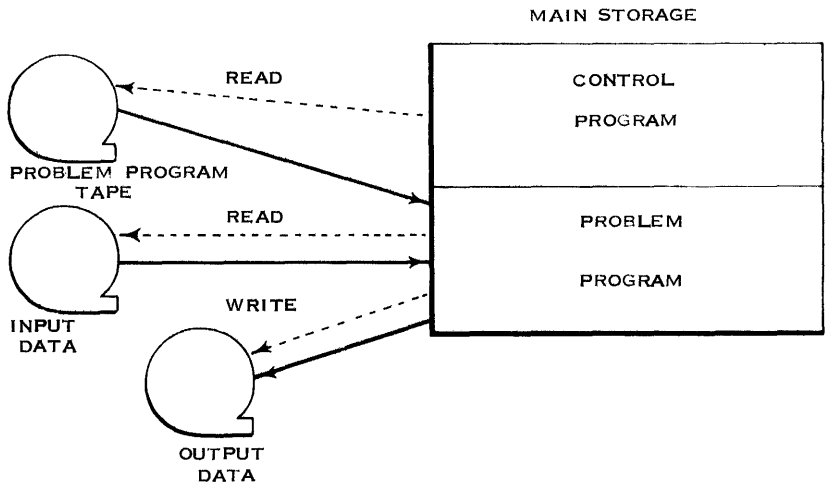
> Read in new problem programs when necessary (same as preceding example).
>
> Also, it will start the necessary I/O units for handling I/O data during the execution of the problem program.

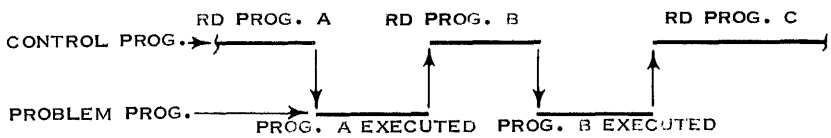
So in the control program concept, there are always two programs in main storage: the control program and a problem program.



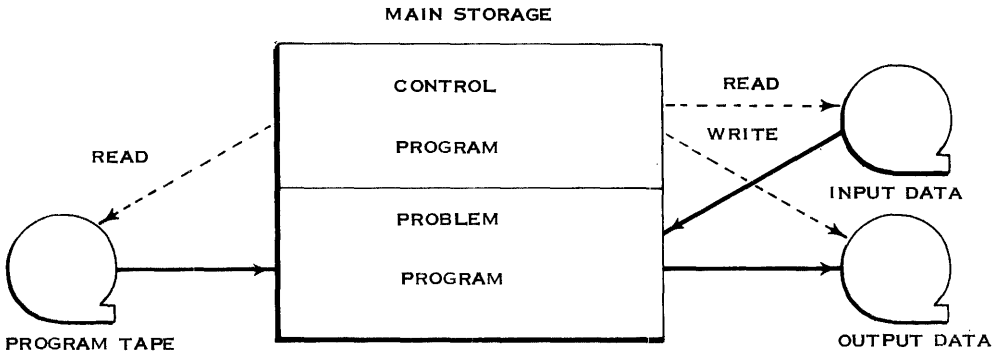
MAIN STORAGE

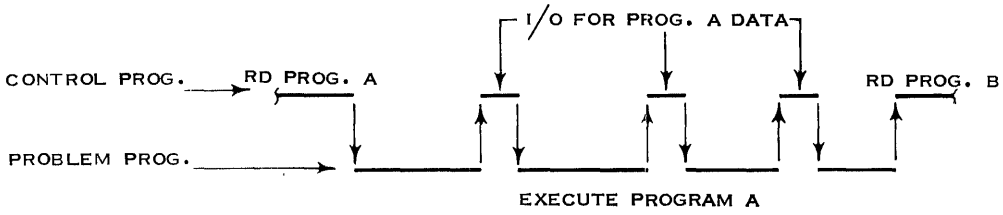
In the simplest utilization of the control program, it was used only to bring in the next problem program. The problem programs handled their own input-output operations.





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.





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

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.

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! For the purposes of our discussion, 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 eliminate 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.

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 cause an automatic branch to the supervisor instead of stopping the machine.

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.

As you have just learned, a machine check on System/360 causes an automatic branch to the supervisor. This is the usual manner of System/360 operation. However, as an aid to the Customer Engineer, there are two additional items concerning a machine check:

1. There is a switch on the system control panel which can allow a stop when a machine check occurs.

2. There is a bit (position 13) in the PSW which when set to 0 will cause machine checks to be ignored. That is, there will be no automatic branch to the supervisor.

Which of the following is/are true?

a. A machine check can cause an automatic branch to the supervisor.
b. A machine check can cause an error stop.
c. A machine check can be ignored.
d. All of the above.
e. None of the above.

---

d; It is true that a machine check can:

1. Cause an automatic branch to the supervisor. This is the usual method of operation.
2. Cause an error stop. This is under control of a switch on the system control panel.
3. Be ignored. If bit position 13 of the PSW is set to zero, machine checks are ignored and no automatic branches to the supervisor will occur. Note: The machine check is remembered. When bit position 13 of the PSW is set back to 1, the automatic branch will occur.

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.

Usually when a machine check occurs, an automatic branch is taken into the supervisor. This automatic branch is called an _____.

---

interrupt

Interrupts can be caused by m_____ checks and p_____ checks.

---

machine
program

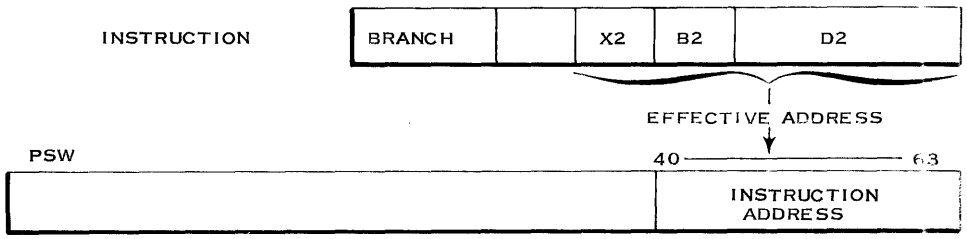
When a problem program is finished, it signals the supervisor via an i_____.

---

interrupt

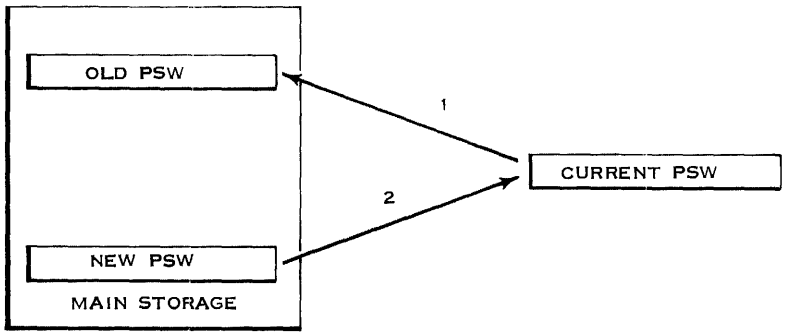
An interrupt is quite similar to a branch. However, it does much more than a simple branch instruction. A branch instruction only replaced the instruction address portion of the "current" PSW.

BRANCHING FUNCTION



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



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"

An interrupt replaces the entire "current" PSW. It does this by storing it as the "_____" PSW and bringing out a "new" PSW.

"old"

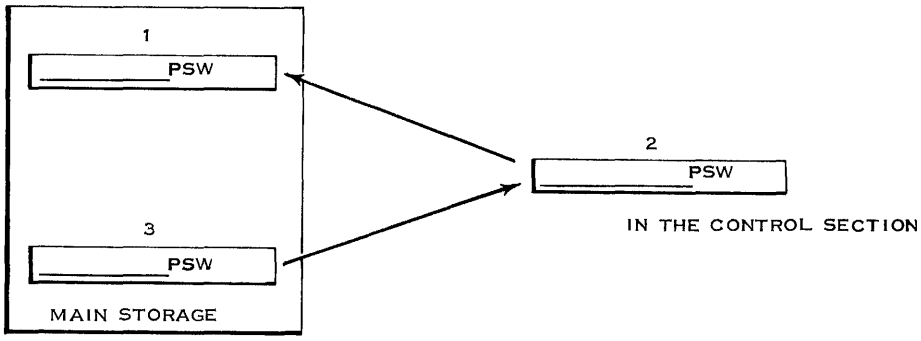
The "_____" PSW is now controlling the program and is therefore the new "current" PSW.

"new"

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.



Fill in the blanks above.

---

1. "old"
2. "current"
3. "new"

When an interrupt occurs, the "current" PSW is placed in main storage in the location reserved for the "_____" PSW.

"old"

The location of the last instruction executed prior to an interrupt can be determined by examining the "_____" PSW.

"old"

The new sequence of instructions will be under control of the PSW brought out from the main storage location reserved for a "_____" PSW.

"new"

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:

1.	External	Can be caused by pressing an interrupt key on the operator's console.
2.	Supervisor	Caused by an instruction known as "supervisor call."
3.	Program	Caused by a program check.
4.	Machine	Caused by a machine check.
5.	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.

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

The handling of program check interrupts, like all interrupts, is taken care of by the _____ program.

---

supervisor

The portion of the control program that resides in main storage and handles all interrupts is called the _____ program.

---

supervisor

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.

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.

Circle one of the above.

---

b; Accomplished automatically by the "hardware" (internal circuitry) of System/360.

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)

An entry into the correct routine in the supervisor program will be caused by the instruction address portion of the "_____" (old/new) PSW.

"new"

The particular routine that will be used in the supervisor program is determined by the class of the _____.

---

interrupt

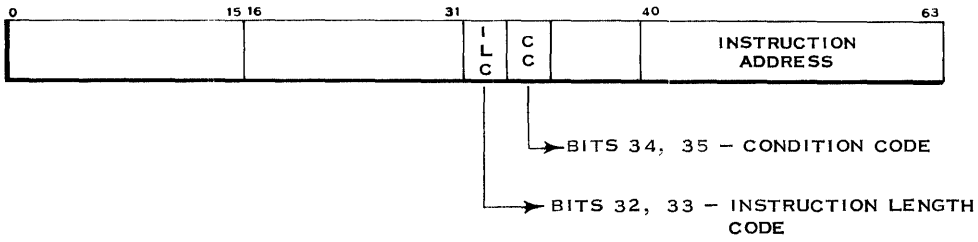
The location of the first instruction to be executed after the interrupt is contained in the "_____" _ _ _ _.

"new"
PSW

The location of the last instruction executed prior to the interrupt can be determined from the "_____" _ _ _ _.

"old" PSW

As you will recall from our earlier discussion of the PSW, the instruction address portion of the "current" PSW is used to read out an instruction. Once the instruction has been read out, the instruction address portion of the PSW is updated so as to point to the next instruction. Interrupts can only occur after an instruction is finished. Therefore, the instruction address portion of the "old" PSW will not contain the address of the last instruction executed. Instead it will contain the address of the next instruction that would have been executed if the interrupt hadn't occurred. In order to get the location of the last instruction executed, the instruction address portion of the "old" PSW must be decremented by the supervisor program. The supervisor must then know the length of the last instruction executed. This is taken care of by an instruction length code (ILC) in the PSW. The instruction length code is contained in bits 32 and 33 of the PSW.



Bits 32 and 33 (ILC) of the PSW will be set to 1, 2, or 3 depending on the length of the instruction.

PSW Bits 32 and 33	Instruction Length
01	1 Halfword
10	2 Halfwords
11	3 Halfwords

If the instruction address portion of the "old" PSW contains 4000 and its instruction length code contains 3, the last instruction executed prior to the interrupt is located at _____.

3994

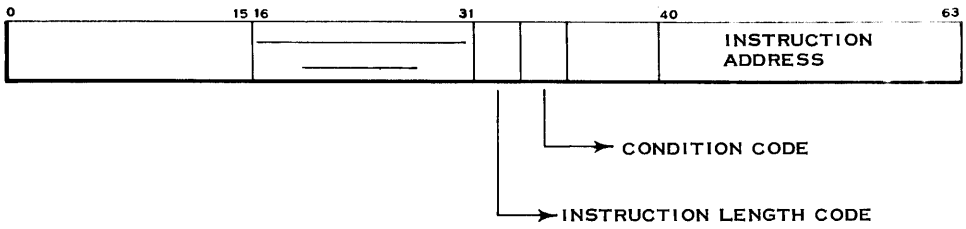
If the last instruction executed prior to the interrupt was of the RX format, the instruction length code of the "old" PSW will contain ___.

The length in halfwords of the last instruction executed prior to the interrupt is contained in the ILC (_____ _____ _____) of the "_____" (old/new) PSW.

10
instruction length code
"old"

You have just seen how the instruction length code in the "old" PSW would indicate the length of the last executed instruction in the interrupted program. 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.

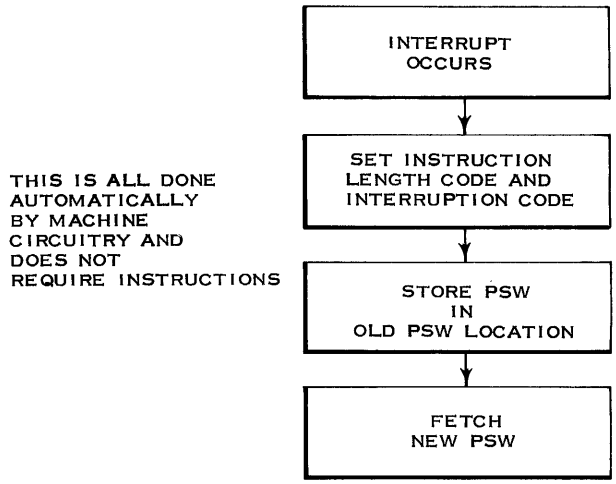
Another field in the PSW that may be of value to the supervisor is the Interruption Code. It appears in bits 16 - 31 of the PSW.



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.



Bits 16 - 31 of the "old" PSW will contain the _____ code. The supervisor program, by examining bits 16 - 31 of the "old" PSW, can determine the _____ _____.

---

interruption
interruption code

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 bits 16 - 31 of 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.

Go to the IBM System/360 Principles of Operation manual and briefly study the Interruption Action chart in the Appendix or the Interruptions section. Use this chart as reference when reading the following frames.

To determine the specific reason for a program interrupt, the supervisor program would have to examine bits 16 - 31 of the " _____ " (old/new) PSW.

"old"

Bits 16 - 31 of the PSW are called the _____ _____.

---

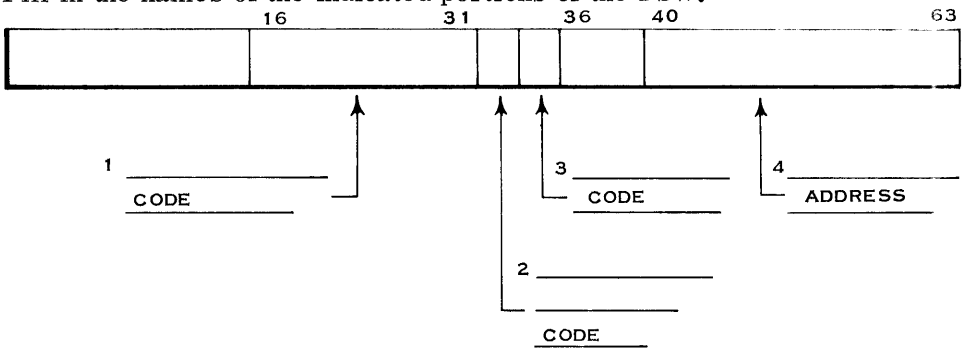
interruption code

When a program interrupt is caused by a fixed point overflow, the interruption code of the "old" PSW will contain _____. (Refer to the Interruption Action chart in the Principles of Operation manual.)

0000000000001000; For brevity's sake, the interruption code would be represented as 4 hexadecimal digits:

Binary		Hexadecimal
0000000000001000	⟶	0008

Fill in the names of the indicated portions of the PSW.



---

1. Interruption Code
2. Instruction Length Code
3. Condition Code
4. Instruction Address

Since there are five "old" PSWs 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 the type of "new" PSW that is fetched. The "new" PSW will cause an entry into the proper routine in the supervisor program. The routine in turn will use the "old" PSW that corresponds to the particular class of interrupt. For instance, the program check routine in the supervisor will use the "old" PSW at location 0040 while the supervisor call routine will use the "old" PSW at location 0032.

Interrupt	"Old" PSW	"New" PSW
External	0024	0088
Supervisor	0032	0096
Program	0040	0104
Machine	0048	0112
I/O	0056	0120

Notice that the "old" and "new" PSW locations are shown on the Interruption Action chart in the Principles of Operation manual.

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

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

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

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

In the case of an interrupt caused by the instruction "supervisor call," 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

This "new" PSW will direct the system to that portion of the supervisor program which handles "supervisor calls."

One way a problem program could notify the supervisor program that the program is finished is to issue a "_____ _____" instruction.

"supervisor call"

The last instruction of a problem program would probably be a "_____ _____" instruction.

"supervisor call"

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.

On an external interrupt, the doubleword at location 0088, known as the " _____ " PSW, is brought out and becomes the new "current" PSW.

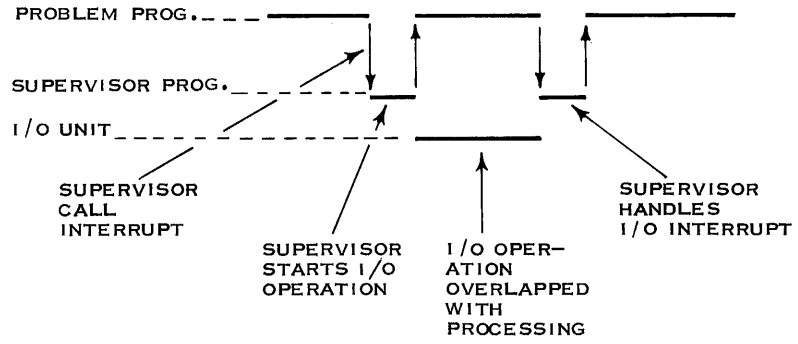
0024
"old"
"new"

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.

This PSW will direct the system to that section of the supervisor program that handles I/O _____.

0056
"old"
"new"
interrupts

You will learn more about I/O interrupts when you study I/O programming. For now, it is sufficient to realize that I/O interrupts generally occur at the end of an I/O operation. Most I/O operations are overlapped with processing. The I/O interrupt is an efficient way of signaling the supervisor that the I/O operation is finished.



At the end of the I/O operation, the I/O interrupt would cause the PSW used in the processing of the p_____ program to be stored in location 0056.

The doubleword at location 0120 would be brought out of main storage and used as the PSW to control the processing of the I/O interrupt routine in the _____ program.

---

problem
supervisor

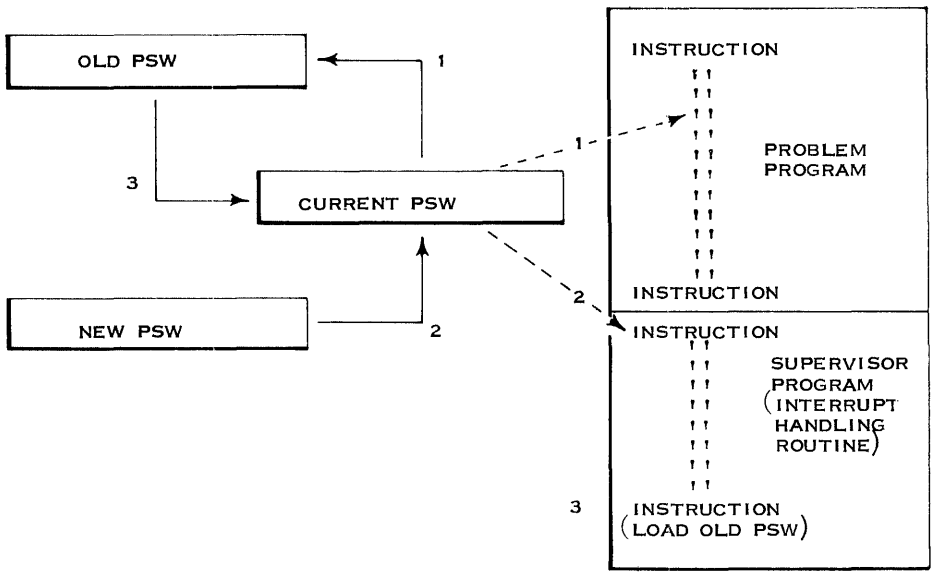
After the end of the I/O interrupt routine in the supervisor, it is desirable to return back to processing the problem program. Which of the following sounds as if it were the better method?

a. The supervisor should issue a branch instruction back to the problem program.
b. The supervisor should load the "old" PSW in location 0056 back as the controlling PSW.

---

b

To simply branch back to the problem program would not be desirable. A branch instruction only affects the instruction address portion of the PSW. Other parts of the PSW are also important in controlling the processing of a program. For one thing, the condition code setting in the controlling PSW for the I/O interrupt routine would not necessarily be the same as it was before the I/O interrupt occurred. It would be best to be able to give control back to the problem program with the same PSW that the problem program was using when the I/O interrupt occurred. This can be done in the System/360 with an instruction known as "Load PSW." This instruction can be used by the supervisor to load the "old" PSW back into the system's control section. This would necessarily be the last instruction in the supervisor's interrupt handling routine. This return to the problem program by replacing the PSW is done by means of an instruction ("load PSW") and is not automatic as an interrupt was.



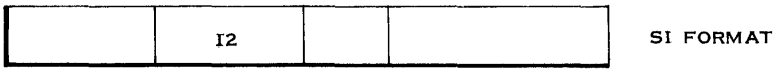
As can be seen from the preceding figure, interrupt action is as follows:

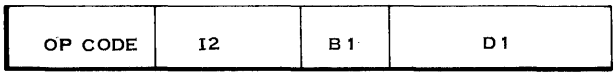
(1) At the time of the interrupt, the "current" PSW which is controlling the problem program is stored in the "old" PSW location. The "old" PSW gives the reason for the interrupt. It also contains in its instruction address portion the point at which we left the problem program. This is done automatically by machine circuits.

(2) 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.

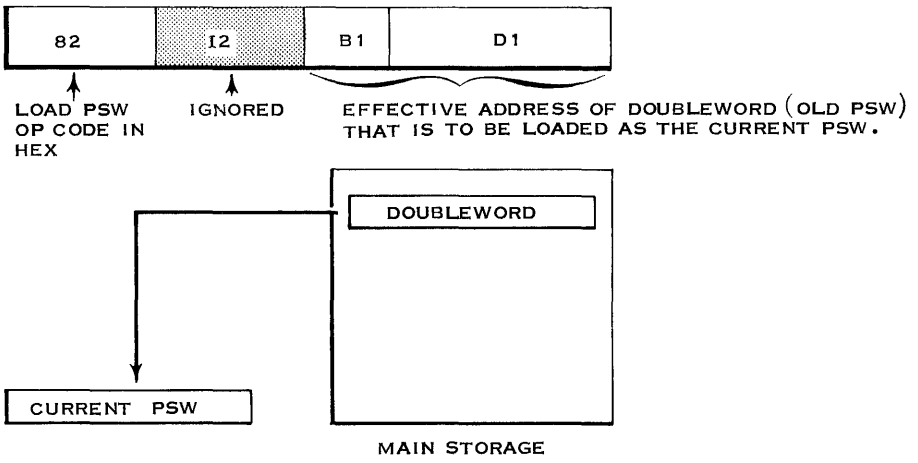
(3) After the interrupt has been taken care of, the last instruction of the interrupt handling routine will be "load PSW." This will cause the "old" PSW to once again become the "current" PSW and we are back in the problem program.

The "load PSW" is of the SI format. Label the fields of the SI format.





In the "load PSW" instruction the I2 field is ignored.



The "load PSW" instruction can be used by a supervisor program any time it wants to change the "current" PSW. One of its main uses will be to return to the problem program by making the "old" PSW become the "current" PSW after an I/O, "supervisor call," or external interrupt has been serviced. It could also be used to load the PSW for new problem program after it had been read into the machine by the supervisor program.

To return to a problem program after an I/O interrupt has been serviced, the effective address generated by the B1 and D1 fields of a "load PSW" instruction should be _____. (Refer to the Interrupt Action chart.)

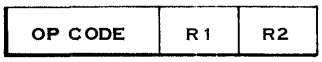
0056: "old" PSW for I/O interrupt

The "supervisor call" interrupt as was previously explained is used by the problem program to pass control to the supervisor program. There are a number of reasons why the problem program might want to call the supervisor program. Two of the major reasons are:

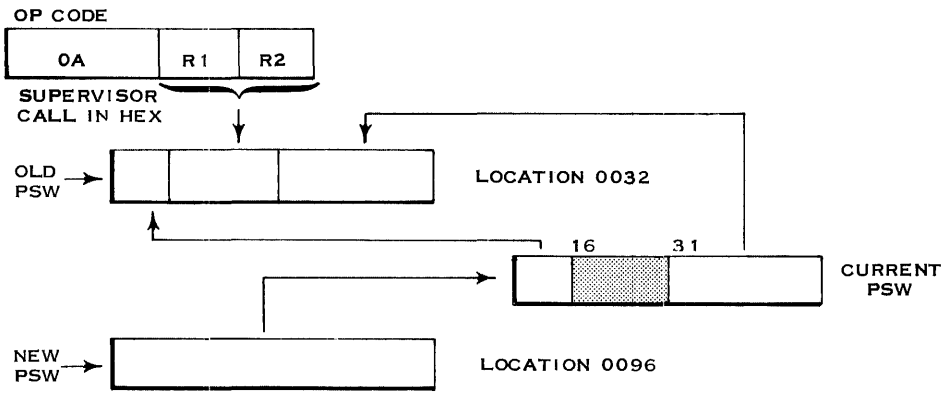
1. To tell the supervisor program that it (the problem program) is done. The supervisor could then read in a new problem program and load its PSW.

2. To request the supervisor program to start an I/O operation for the problem program.

The "supervisor call" instruction is of the RR format. Label the fields of the RR format.



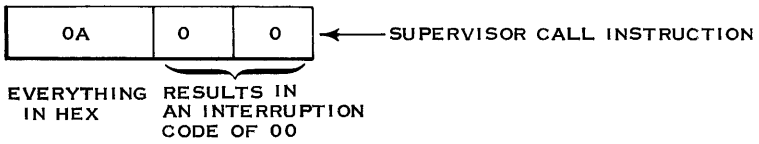
The "supervisor call" instruction causes a "supervisor call" interrupt. The eight bits of the R1 and R2 fields are placed in the interruption code of the "old" PSW.



The "old" PSW will be the previous "_____" PSW with the exception of its i_____ code which will come from the "supervisor call" instruction.

"current" interruption

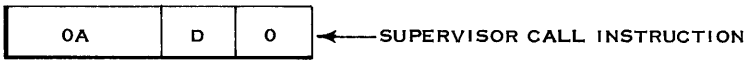
Since the bits of R1 and R2 field are stored as the interruption code, they can be used as pre-arranged signals to tell the supervisor program the reason for the interrupt. These "pre-arranged signals" would depend on who (IBM or the user) wrote the supervisor program. For instance:



This interruption code of 00 might be used to signal a supervisor program that the problem program is finished.

Given the following "supervisor call" instruction (in hex), what binary bit structure would be placed in the interruption code of the "old" PSW?

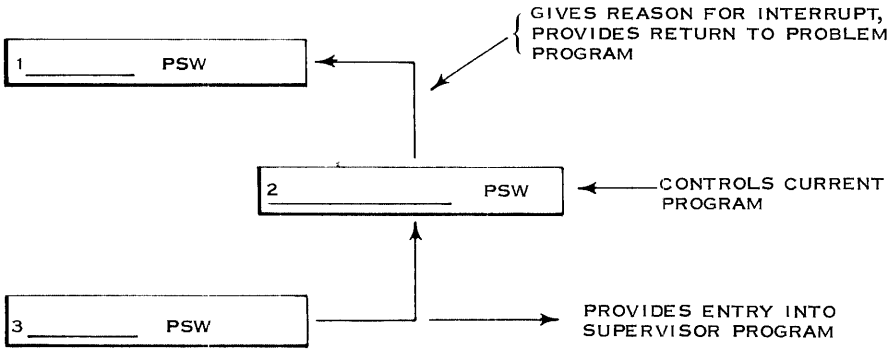
_____



0000000011010000-bits 16-31 of the "old" PSW in location 0032

As you know by now, an interrupt is a type of "branch." It can occur at any time in a program although never in the middle of an instruction. Although you have been told that interrupts occur only at the end of an instruction and never in the middle of one, this might be a little misleading. It is definitely and absolutely true that the current instruction will be completed before an I/O, external, or "supervisor call" interrupt is taken. In the case of program and machine interrupts (which indicate programming and hardware errors), the interrupt still occurs at the end of the instruction. However, in these two cases, the end may be forced by suppressing the instruction's execution where a programming error is detected during instruction fetch time or by terminating its execution when a programming or machine error is detected during E time. Refer to the Execution Column of the Interruption Action chart.

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.

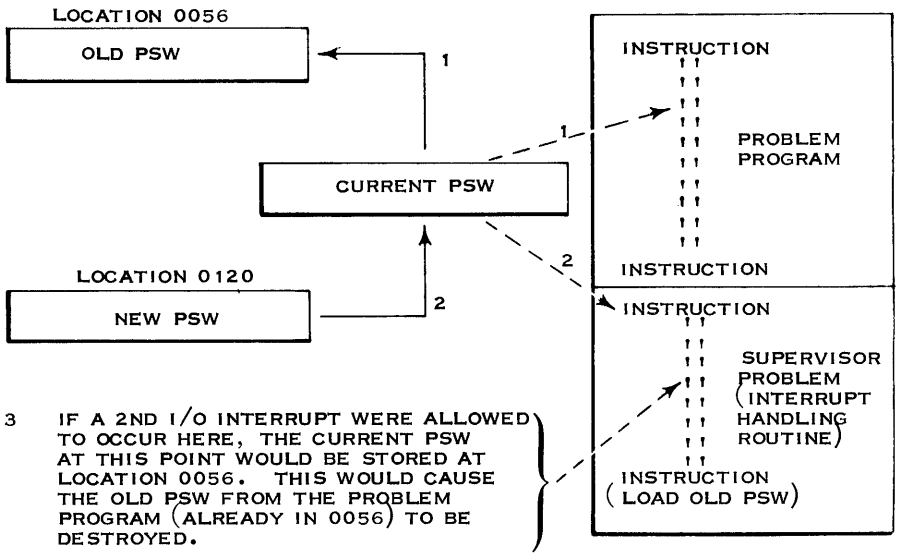


Fill in the blanks above.

---

1. "old"
2. "current"
3. "new"

Sometimes it is not desirable to allow an interrupt. This is most apparent when we consider an I/O interrupt because in the System/360 it is possible to have simultaneous I/O operations on two or more channels. This will be explained in the following illustration.



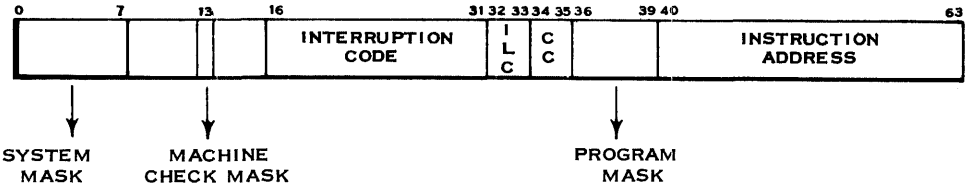
As can be seen from the preceding figure, interrupt action is as follows:

(1) When one I/O operation is completed, an I/O interrupt will usually occur. The "current" PSW will be stored to give the supervisor program the reason (which I/O unit) for the interrupt. This "old" PSW also gives the supervisor program a way in which to return to the interrupted problem program.

(2) A "new" PSW is then brought out of storage and becomes the "current" PSW. This "new" PSW points to the first instruction of the I/O interrupt handling routine.

(3) A 2nd I/O interrupt caused by the completion of operation of another I/O channel will result in a loss of the "old" PSW.

How can the supervisor program prevent this 2nd and undesirable I/O interrupt until it has processed the first one? It does this by proper usage of "Mask" bits in the PSW.



As can be seen above:

1. Bits 0 - 7 are known as the _____ _____ bits.
2. Bit 13 is the _____ _____ _____ bit.
3. Bits 36 - 39 are known as the _____ _____ bits.

SYSTEM MASK

---

1. system mask
2. machine check mask
3. program mask

When these mask bits are set to zero, the corresponding interrupts are masked or prevented. Let's first consider the system mask bits. These 8 bits can be used selectively or collectively to mask all I/O and external interrupts as follows:

PSW Bits 0 - 7	System Mask
0	Multiplexor Channel
1	Selector Channel 1
2	Selector Channel 2
3	Selector Channel 3
4	Selector Channel 4
5	Selector Channel 5
6	Selector Channel 6
7	External

To prevent (mask) all I/O and external interrupts, bits 0 - 7 of the "current" PSW must contain _____ (zeroes/ones).

---

zeroes

A system mask of all zeroes would mask: (Circle one or more.)

a. All external interrupts
b. All program interrupts
c. All supervisor call interrupts
d. All I/O interrupts
e. All machine interrupts

---

a and d

Notice that there is only one I/O interrupt. However, each of the six selector channels and the multiplexor channel can be selectively prevented from causing the I/O interrupt.

Use the "mask bits" column of the Interruption Action chart as reference for the following frames.

A system mask of 00111110 would mask (prevent): (Circle one of the following.)

a. All I/O and external interrupts
b. Some I/O and all external interrupts
c. Some I/O and some external interrupts
d. All I/O and some external interrupts

---

b

A system mask of 10000001 would prevent I/O interrupts by:

a. Multiplexor channel only
b. All selector channels
c. Multiplexor and selector channels

---

b

Bits 0 - 7 of the PSW are known as the _____ _____.

The system mask can be used to prevent these two classes of interrupts:
1. _____ 2. _____.

---

system mask
1. I/O
2. external

To prevent all I/O and external interrupts, the _____ mask must contain all _____ (zeroes/one).

---

system
zeroes

Although there is only one I/O interrupt, each of the _____ can be selectively masked so that they will not cause the interrupt.

---

channels

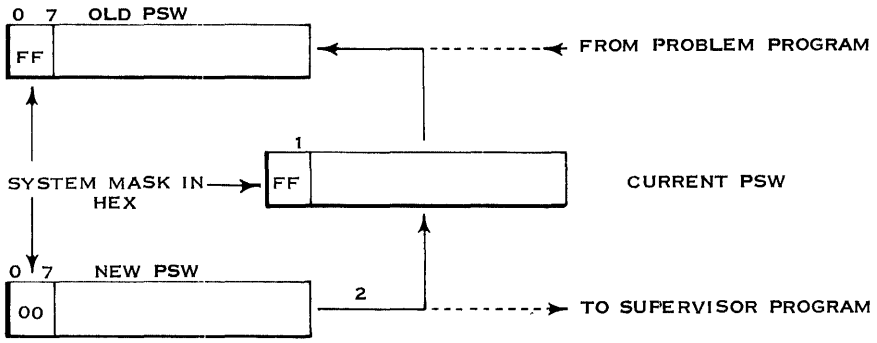
The system mask that determines whether or not to mask I/O and external interrupts is in the "_____" (old/current/new) PSW.

"current"

In case of an I/O interrupt, the channel causing the interrupt will be contained in the interruption code of the "_____" (old/current/new) PSW.

"old"

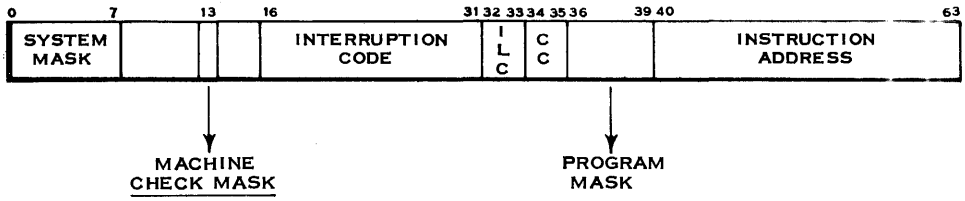
To prevent a second I/O interrupt before a first one has been completely processed, the system mask of the "new" PSW should contain zeroes.



One more point that should be made concerning the system mask. When it contains zeroes, I/O and external interrupts are prevented. However, any I/O and external interrupts will remain pending. As soon as the system mask is set to 1's, another interrupt will be taken.

The last instruction in the I/O interrupt routine of the supervisor program would be "load PSW." The "old" PSW in main storage would be brought out and placed back in action as the "current" PSW. Once this is done I/O interrupts can once more occur. This is because the system mask of the problem program's PSW would probably contain all 1's (FF). Of course, a system mask of all 1's would allow not only I/O interrupts but also external interrupts.

MACHINE CHECK MASK



There are five classes of interrupts. I/O and external interrupts can be masked by means of the System Mask in bits 0 - 7 of the PSW. A third class of interrupt can be masked by means of bit 13 of the PSW. If this bit contains a zero, machine checks will be ignored and no machine interrupt will occur. Of course, this is not the usual state of the machine check mask bit. It is usually set to 1, so that machine checks will cause an interrupt. Remember also that a switch on the CE section of the system control panel can be used to cause an error stop rather than have an interrupt. The usual mode of operation is to have this switch off and PSW bit 13 set to 1. This means that when a machine check (such as even parity) occurs, an error stop does not occur. Instead a machine interrupt occurs.

When PSW bit 13 is set to 1, a machine check _____ (will/will not) cause an interrupt. The bit is usually set so that an interrupt _____ (will/will not) occur whenever a machine check is encountered.

---

will
will

When PSW bit 13 is set to zero, a machine check will cause: (Circle one of the following.)

a. An error stop
b. An interrupt
c. None of the above

---

c; The machine check mask bit only determines whether a machine check causes an interrupt or is ignored. If PSW bit 13 is zero, machine checks are ignored and the System/360 continues merrily on its way. Obviously, this mask bit is rarely set to zero.

Which of the following will cause an error stop when a machine check happens?

a. PSW bit 13 set to 1
b. PSW bit 13 set to zero
c. Switch on CE control panel is on
d. Switch on CE control panel is off

In summary then, there are three possible courses of action when a machine check occurs:

1. It will cause machine interrupt; the PSW is stored in location 0048 and a "new" PSW is fetched from location 0112.

2. It will cause an error halt.

3. It will be ignored if PSW bit 13 is zero.

There is one other item of information concerning machine checks. It is called "log out." Unless the machine check is being ignored, information concerning the status of internal circuitry is automatically placed in storage starting at machine location 0128. This "log out" occurs prior to the machine interrupt or error stop.

The automatic storing of status information at location 0128 when a machine check occurs is known as "_____ _____."

"log out"

"Log out" consists of storing information concerning the status of the machine when the error occurred. This "log out" begins at location _____.

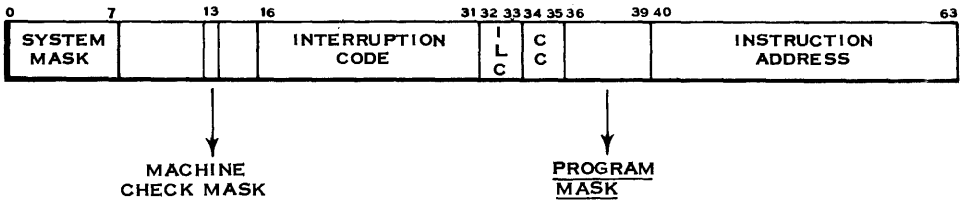
0128

Just how much information is contained in a "log out" and what it means will depend on the particular model of System/360. Until you learn one of the models of System/360, "log out" doesn't mean much. However, "log out" always occurs prior to a machine interrupt and places information in storage starting at location 0128. The size of this "log out" area may vary from a couple of bytes to as much as almost 200 bytes, depending on the model of System/360. This information reflects the status of the machine's internal circuitry. As such it is meaningful only to someone who has a knowledge of the machine's internal circuitry.

PROGRAM MASK

Program checks (such as a specification exception) also can cause an interrupt. While machine checks cause machine interrupts, program checks will cause a program interrupt. On a program interrupt, the PSW is stored in location 0040 and a "new" PSW is fetched from location 0104. It might be interesting at this time to note for what it's worth that the location of any "new" PSW is 64 higher than that of its corresponding "old" PSW.

Class of Interrupt	"Old" PSW		"New" PSW
External	0024	+64=	0088
Supervisor Call	0032	+64=	0096
Program	0040	+64=	0104
Machine	0048	+64=	0112
I/O	0056	+64=	0120



We have already seen that I/O, External, and Machine interrupts can be masked. Program interrupts can also be masked by use of bits _____ through ____ of the PSW.

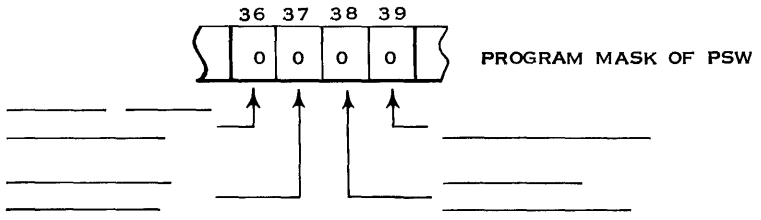
36, 39

There are 15 possible exceptions which can cause a program check. Go to the Interruption Action chart and briefly review the 15 exceptions.

Four of these exceptions may on occasion not be considered as program checks. These four exceptions are:

1. Fixed Point Overflow
2. Decimal Overflow
3. Exponent Underflow } ⟶ Concerned with
4. Significance } ⟶ Floating Point

When one of the general registers is being used as a counter in a program, it may be desirable to test the counter for an overflow. In such cases, an overflow should not be treated as a program check. As a result the program mask in the PSW is available to the programmer to mask program check interrupts caused by the four exceptions mentioned earlier. Use the Mask Bits Column of the Interruption Action chart and fill in the blanks in the illustration.



---

36 - Fixed Point Overflow
37 - Decimal Overflow
38 - Exponent Underflow
39 - Significance

All other programming exceptions (such as specification) are always treated as programming errors and will always cause a program interrupt.

What four programming exceptions can be masked by bits 36 - 39 of the PSW?

1. _____
2. _____
3. _____
4. _____

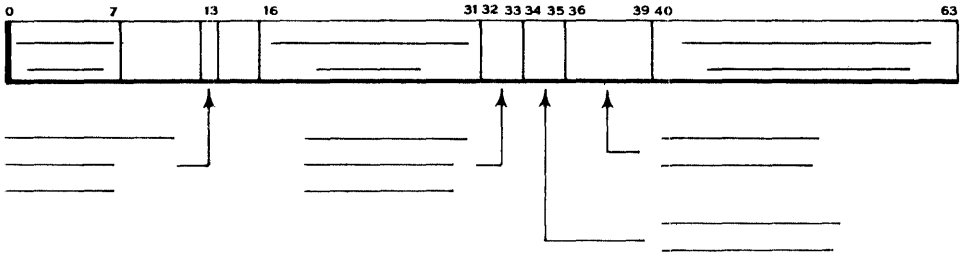
---

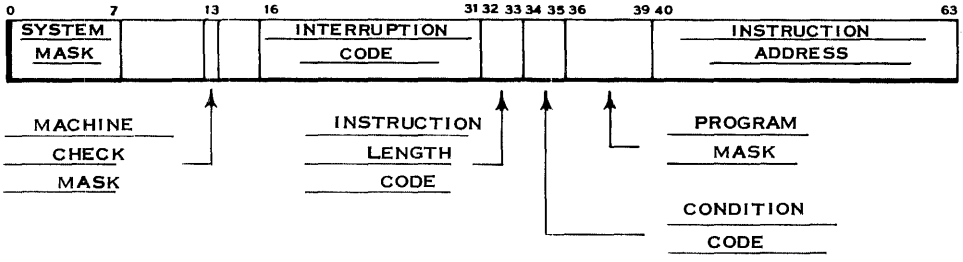
1. Fixed Point Overflow
2. Decimal Overflow
3. Exponent Underflow
4. Significance

You learned these when you studied Data Formats.

These are part of Floating Point which is not an objective of this course.

Label the indicated fields of the PSW.

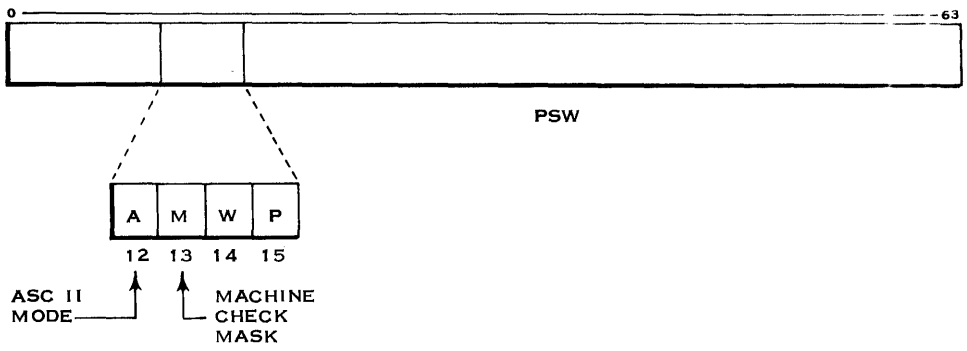




It is also important to know which classes of interrupts cannot be masked. They are the "supervisor call" interrupt and program interrupts caused by all but the four programming exceptions indicated in bits 36 - 39 of the PSW.

At this point you have covered most of the PSW and can relate it to the control of the program and the System/360 interruption system. Let's finish up this PSW!

ASC II MODE



Of bits 12 - 15, you are already familiar with bit 13. It is the m_____ c_____ mask bit. Bit 12 is the _____ mode bit. ASCII is a computer code adopted by the American Standards Association. It differs from binary coded decimal mainly in the way it represents its zone bits. If bit 12 of the PSW contains a 1, the ASCII code will be internally generated rather than the extended BCD code.

Example:

The number 1 in EBCDIC looks like this:

11110001

Zone ← → Numerics

The number 1 in ASCII looks like this:

01010001

Zone ← → Numerics

---

machine check
ASC II

When processing data with the instructions of the decimal feature, the following are the standard signs generated:

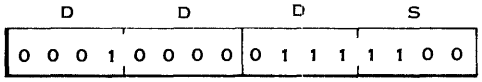
Plus	=	1100	EBCDIC
Minus	=	1101	

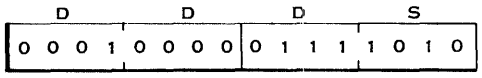
If bit 12 of the PSW contains a 1, the signs that will be generated when using the decimal feature are:

Plus	=	1010	ASCII
Minus	=	1011	

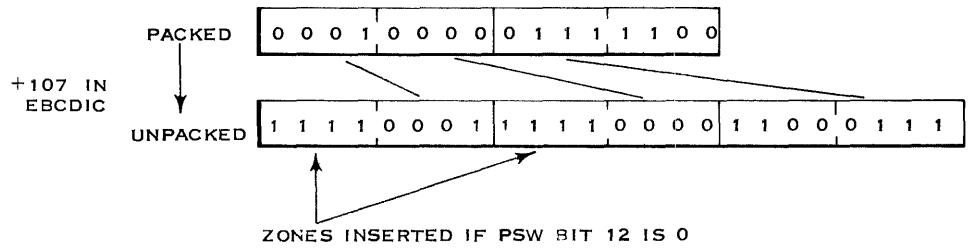
The remainder of the packed fields used by the decimal feature are the same.

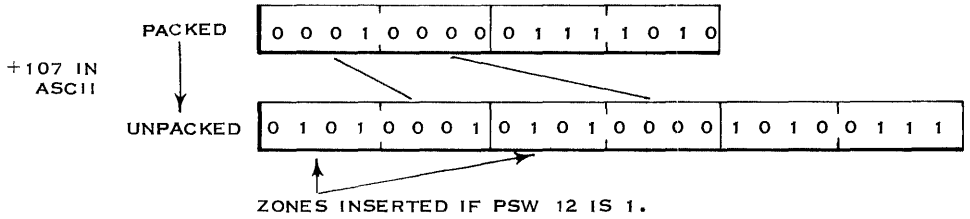
For instance, a +107 would look like this:





When a packed field is converted back to the unpacked format by the "unpack" instruction, the zone bits that are inserted will depend on the ASCII mode bit in the PSW.
For instance:



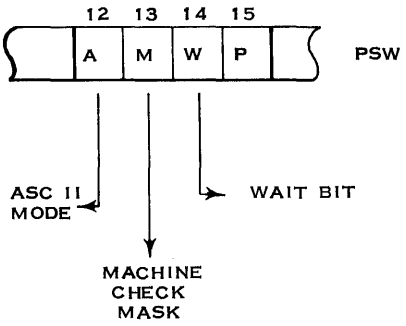


Most System/360 I/O devices which are code sensitive such as punches or printers use the Extended BCD Interchange Code (EBCDIC) rather than the extended ASCII code. As a result it would be expected that in most cases bit 12 of the PSW would be _____ (zero/one).

WAIT BIT

---

zero



Bit position 14 of the PSW is called the wait bit. If this bit is zero, instructions are fetched and executed in the normal manner. Once an instruction is executed, the next instruction is fetched under control of the instruction address portion of the PSW. If the wait bit contains a 1, instructions are no longer fetched and executed. Instead the System/360 will wait until an interrupt occurs and changes the PSW. Of course, the "new" PSW would contain a zero in bit position 14.

If bit 14 of the PSW contains a _____ (zero/one) the CPU will wait until an i_____ occurs and changes the ___ ___ ___.

---

one
interrupt
PSW

Only the I/O and external interrupts can change the status of the CPU from a wait state to a running state. Machine, program and "supervisor call" interrupts can occur only when the CPU is in a running state and processing instructions.

If bit 14 of the PSW contains a 1, the CPU will wait until either an _____ or an _____ interrupt occurs.

I/O
external

We have previously discussed the use of a supervisor program to control a computer installation. Also we discussed the use of a supervisor program to handle the input and output requirements of a problem program. We have also discussed several features of the System/360 which necessitate the use of some type of a supervisor. These features included:

1. The lack of a halt instruction.

2. No stopping of the machine when a program or machine check was encountered.

Another feature of the System/360 which necessitates the use of a control program or supervisor is this:

There are certain instructions in the System/360 which are legal only when the supervisor program fetches them! If the problem program should attempt to execute one of these instructions a program check would occur. These instructions are called privileged operations.

If the problem program should attempt to execute a privileged instruction, a _____ interrupt would occur. The program interrupt is handled by the _____ program.

program
supervisor

The fact that the problem program attempted to execute a _____ instruction is indicated to the supervisor program by the interruption code in the "_____" (old/new) PSW.

privileged
"old"

Those instructions that may be executed by the supervisor program but not by the problem program are called _____ operations.

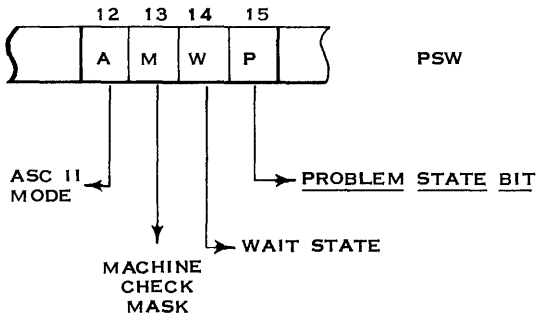
privileged

Two questions now arise:

1. How does the machine know whether the supervisor or the problem program is being executed?

2. What instructions are considered as privileged instructions?

To answer the first question, let's take a look at bit 15 of the PSW.



The machine assumes that the problem program is being executed if there is a 1 in bit 15 of the PSW. This bit is called the _____ _____ bit.

If PSW bit 15 contains a 1, the fetching of a privileged instruction would cause a _____ interrupt.

---

problem state program

The machine assumes that the supervisor program is being executed if PSW bit 15 contains a _____ (zero/one).

---

zero

We would expect that bit 15 of the five "new" PSWs in main storage would contain a _____ (zero/one).

---

zero

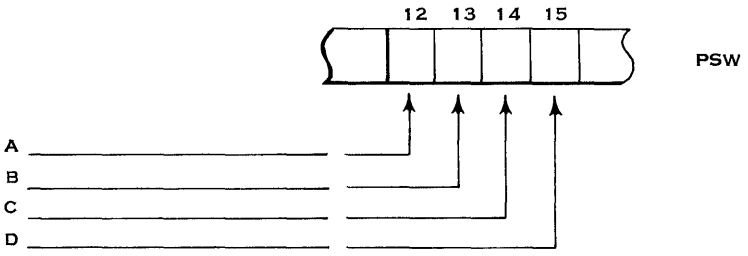
We would expect that bit 15 of any of the "old" PSWs in main storage would contain a _____ (zero/one).

---

one

The "old" PSWs would contain a 1 because they indicate where we left the problem program while a "new" PSW indicates where we are entering the supervisor program.

Label the indicated bits of the PSW.



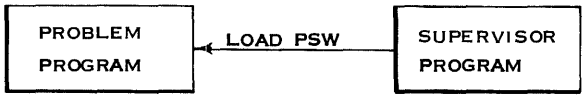
---

a. ASCII Mode
b. Machine Check Mask
c. Wait State
d. Problem State

What instructions are considered privileged? We do not intend at this time to list all privileged instructions. However, you should be aware of some instructions that should be considered privileged. First of all, it would be expected that the supervisor program should be able to change any part of the PSW any time it wanted to. However, there are only certain parts of the PSW that should be changed by the problem program. Let's take a look at the fields of the PSW.

Bits	Field	Changed By
0-7	System Mask	An instruction called "Set System Mask"
8-11	(We'll examine this field later)	
12-15	(AMWP)	
16-31	Interruption Code	An Interrupt
32-33	Instruction Length	An Interrupt
34-35	Condition Code	Many Instructions
36-39	Program Mask	An Instruction called "Set Program Mask"
40-63	Instruction Address	Execution of Program

From the above we can see that some of the PSW fields can be changed by a special instruction. The other fields can be changed only by changing the entire PSW. Basically, there are two ways of changing the entire PSW. One is by way of an interrupt. The other is by way of the instruction "load PSW." It would not be desirable to allow the problem programmer to use the "load PSW" instruction since this instruction changes all parts of the PSW. You would not want the problem programmer to have this much control over the machine. Only the supervisor program should retain this control. As a result, the "load PSW" is a privileged instruction. It can only be used by the supervisor program (indicated by bit 15 of the PSW). The supervisor program could use the "load PSW" to change any part of the PSW. It would also use this instruction to return to the problem program after an interrupt had been serviced.



The problem program would enter the supervisor program by way of an interrupt. This interrupt would normally be a result of the instruction "supervisor call."

Notice that a branch instruction is not used in either example above. This is because a branch instruction cannot change the problem state bit (bit 15) in the PSW.

The supervisor program can change the state of the machine any time it wants to by use of the "load PSW" instruction. The problem program cannot use the "load PSW" instruction because it is a privileged operation. The problem program can only use the "supervisor call" instruction to go from the problem state to the supervisor state (PSW bit 15). Of course, this assumes that the "new" PSW in location 0096 (for "supervisor call" interrupts) has a zero in bit 15.

SET SYSTEM MASK - SET PROGRAM MASK

Besides the "load PSW" instruction, there are two other instructions which can change the PSW. They are "Set System Mask" and "Set Program Mask." The "Set Program Mask" is not a privileged instruction. As such, the problem programmer can use it to change the Program Mask portion of the PSW.

Whereas the "load PSW" changes the entire PSW, the "Set Program Mask" changes the _____ _____ portion of the PSW.

---

program mask; Actually the "Set Program Mask" instruction changes bits 34-39 of the PSW. This means that the condition code is also changed.

The "Set System Mask" instruction is a privileged instruction. This is because the system mask affects I/O interrupts. In the System/360, it was the intention of the designers to have the supervisor handle all I/O operations. For this reason, the "Set System Mask" instruction and the four I/O instructions are privileged operations.

Which of the following instructions are valid when the machine is in the problem state (PSW bit 15 is 1)? (Circle one or more.)

a. Set System Mask
b. Load PSW
c. Start I/O (one of the 4 I/O instructions)
d. Set Program Mask
e. Supervisor Call

---

d and e

Which of the following instructions may be issued only by the supervisor program (PSW bit 15 is 0)?

a. Load PSW
b. Set Program Mask
c. Set System Mask
d. Start I/O
e. All of the above

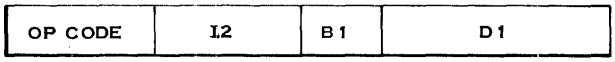
---

a, c, d; The instruction "Set Program Mask" may be issued by both supervisor and problem programs.

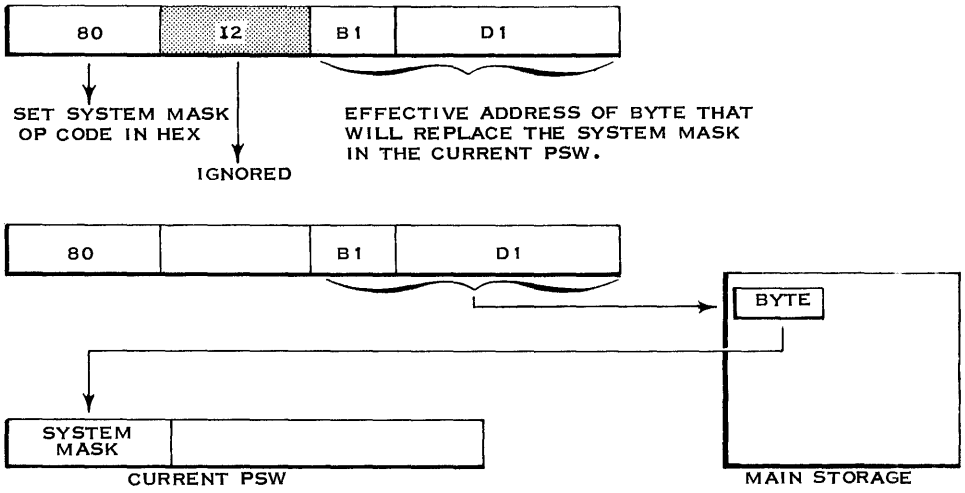
Let's take a look at the "Set System Mask" and "Set Program Mask" instructions.

The "Set System Mask" instruction is of the SI format. Label the fields of the SI format.

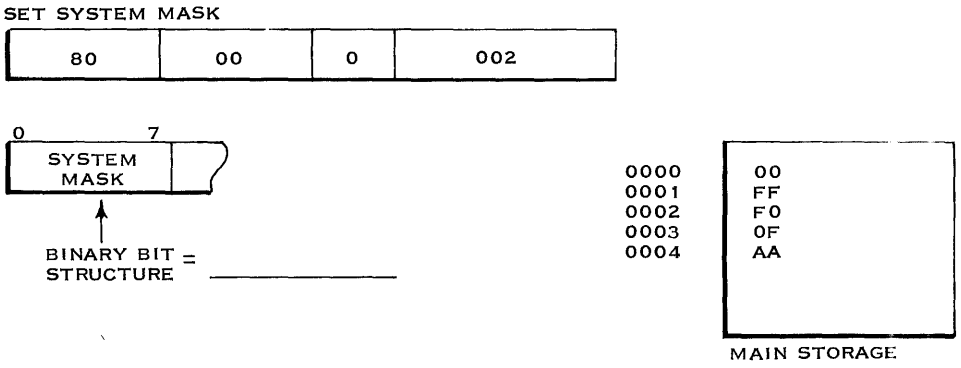
SI FORMAT



This "Set System Mask" instruction is similar to "load PSW" instruction in that the I2 field is ignored.

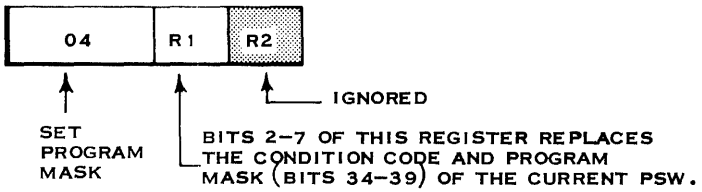


Given the following "Set System Mask" instruction (in hex), what binary bit structure will be placed in bits 0 - 7 of the "current" PSW.

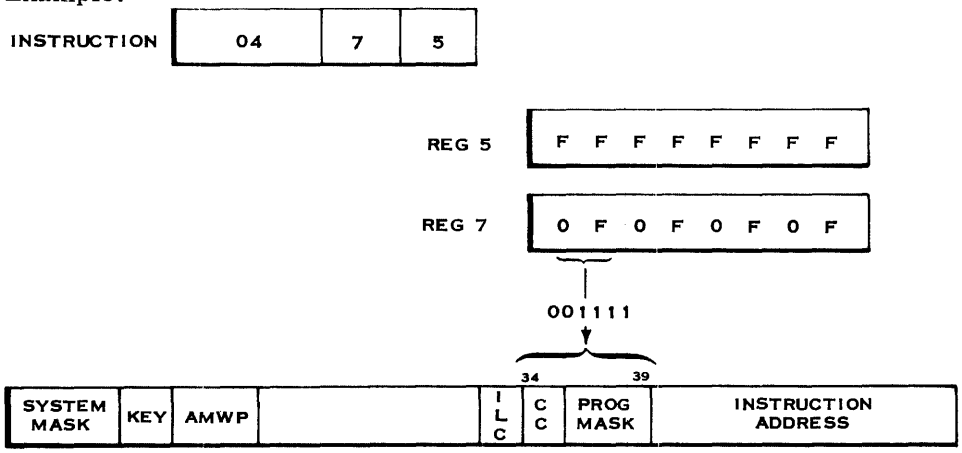


11110000

The "Set Program Mask" instruction is of the RR format.

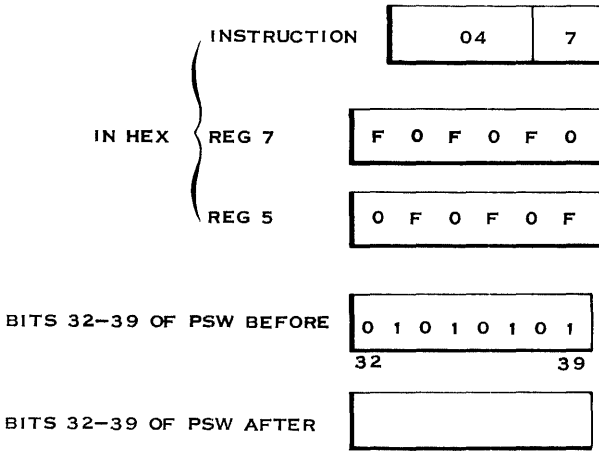


Example:



As can be seen in the above example, reg 5 was ignored. Bits 2 to 7 (001111) of reg 7 were placed in bits 34 to 39 of the PSW. This action replaced the condition code and program mask. With a program mask of all ones, any fixed point and decimal overflows would be treated as errors and a program interrupt would occur.

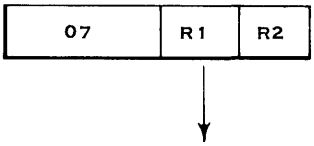
Given the following "Set Program Mask" instruction, indicate the binary bit structure of bits 32 to 39 of the "current" PSW after the instruction is executed.



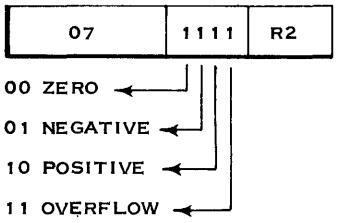
01110000; Bits 32 and 33 are the instruction length code.

With a program mask of all zeroes, a fixed point or decimal overflow will not be treated as a programming error and no program interrupt will occur. However, the overflow will set the condition code to a binary 11 ("hex" 3). Now the problem programmer can use the "branch on condition" instruction to test for an overflow.

Indicate the binary bit structure in the R1 field of the following "branch on condition" instruction that would be necessary to test for an overflow. Refer back to the Condition Code area of the Instruction Sequencing and Branching section of this book for review if necessary.



0001; The R1 field of the "branch on condition" instruction is laid out left to right to test for condition code settings of 0, 1, 2, or 3.



Go to the IBM System/360 Principles of Operation manual and briefly study the following areas.

In the System Structure section, study:

Interruption
Input/Output Interruption
Program Interruption
Supervisor Call Interruption
External Interruption
Machine Check Interruption
Priority of Interruptions
Program States

In the Appendix, study:

Permanent Storage Assignment chart
Condition Code Setting chart
Privileged Operation chart

REVIEW QUESTIONS ON INTERRUPTS

- Try to answer the questions without referring back into this book.

  Do use the System/360 Principles of Operation manual. Consider reviewing any area where aid from this book was required.

1. List the five classes of interrupts

   a. _____
   b. _____
   c. _____
   d. _____
   e. _____

2. Define:

   a. "Current" PSW _____
   _____

   b. "Old" PSW _____
   _____

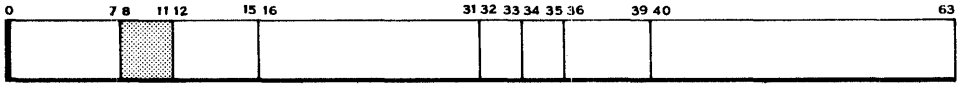
   c. "New" PSW _____
   _____

3. The area of main storage reserved for "old" PSWs is from 0024 to _____.

4. The area of main storage reserved for "new" PSWs is from _____ to 0127.

5. The area of main storage reserved for machine check "log outs" starts at _____.

6. Label the fields of the PSW.



7. Which interrupts cannot be masked? _____
   _____

8. To prevent an interrupt, a mask bit must be _____ (0/1).

9. How can a system/360 be taken out of a wait state? _____
   _____

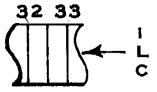
10. What can switch the system from a problem state to a supervisor state? _____

11. What is placed in the interruption code on an I/O interrupt?

_____

12. What is placed in the PSW interruption code on a supervisor call interrupt? _____

13. Show what is placed in the instruction length code when an RX type instruction was the last instruction executed prior to an I/O interrupt.



14. In the problem state _____ instructions cannot be used or a _____ _____ will occur.

15. After handling an I/O interrupt, how does the machine return to the interrupted program? _____

_____

16. What is the difference between a "break-in" and an I/O interrupt?

_____

_____

_____

17. Which of the following instructions may not be given by a problem program? (Circle one or more.)

a. Set system mask
b. Set program mask
c. Load PSW
d. Supervisor call
e. Any I/O instruction

18. The "Set System Mask" instruction causes the system mask to be replaced by _____.

19. The "Set Program Mask" instruction replaces the _____ and _____ with _____.

20. The "load PSW" instruction replaces: (Circle the most correct answer)

    a. The "current" PSW with an "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 doubleword from main storage.

ANSWERS TO REVIEW QUESTIONS

1. a. External
   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.

   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.

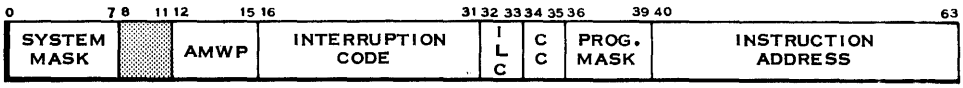
   c. "New" PSW is the doubleword fetched from main storage as a result of an interrupt. It then becomes the "current" PSW. Bits 40 - 63 of this doubleword would switch the machine to a new sequence of instructions.

3. 0063; There are 5 "old" PSWs of 8 bytes each. Each main storage address refers to an individual byte.

4. 0088

5. 0128; It uses the area of main storage just above the area for "new" PSWs.

6.



7. Supervisor call interrupts and those program interrupts not caused by:
   a. fixed point overflow
   b. decimal overflow
   c. exponent underflow
   d. significance

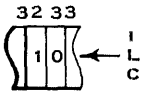
8. 0

9. Only by an I/O or external interrupt.

10. Any interrupt

11. The address of the channel and I/O unit.

12. The 8 bits in the R1 and R2 field of the "supervisor call" instruction.

13. 

14. privileged, program interrupt

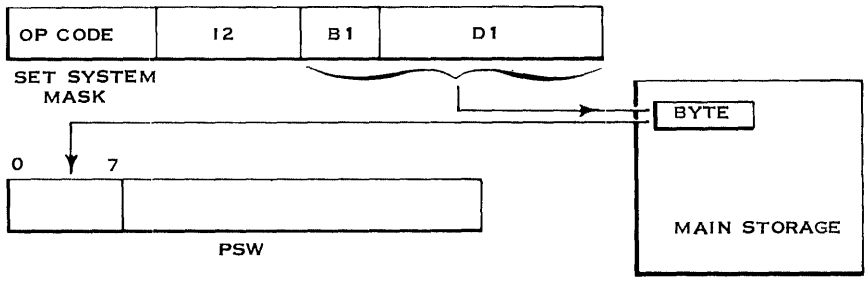
15. By issuing a "load PSW" instruction addressing the doubleword at location 056 (the "old" PSW for an I/O interrupt).

16. A "Break-In" is a request by an I/O channel to use the main storage unit to put data in (read) or take data out (write). It can occur at any time. In some models of System/360, the I/O channels and CPU shared common circuitry and a "Break-In" might involve temporarily delaying the CPU from executing an instruction.

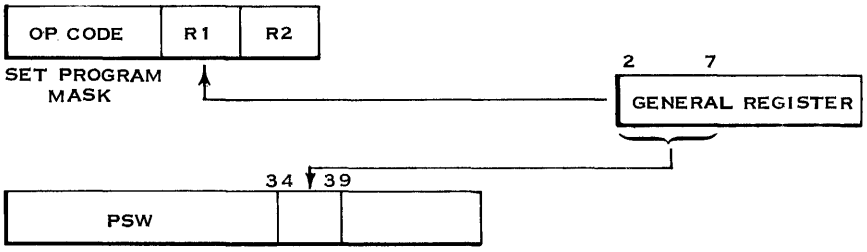
    An I/O interrupt occurs after CPU has executed an instruction and before it executes the next one. The I/O interrupt has nothing to do with requesting data. Basically, the I/O interrupt is a signal to the program that a channel has finished transferring a record to or from an I/O unit. It could also mean that an operation that did not involve a data transfer over the channel (such as rewinding a tape unit) has ended.

17. a,c,e

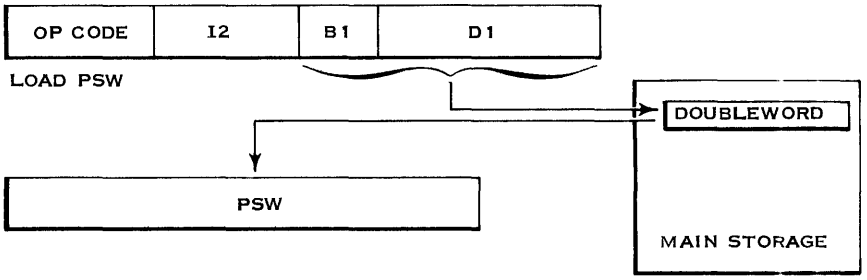
18. A byte from main storage



19. Condition code and program mask with bits 2 - 7 of the register addressed by the R1 field.



20. d; Although the "load PSW" instruction is used to return to an interrupted program by loading the "old" PSW, this "old" PSW has to be addressed from main storage just like any other doubleword.



SECTION IV LEARNING OBJECTIVES

At the end of this section you should be able to:

1. Define a storage key and state how it may be changed.
2. Define a protection key and state how it may be changed.
3. Given a storage and a protection key, determine whether a protection violation would occur.
4. Describe what effect a protection violation would have on the PSW.

Storage Protection

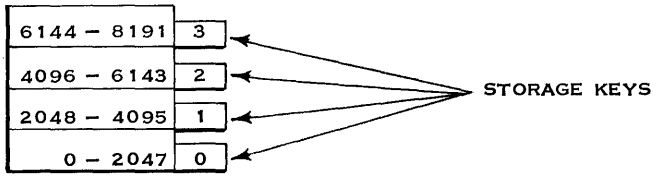
It is the PSW that determines whether the system is operating in the supervisor or in the problem state. Bit 15 of the PSWs used by the supervisor program should contain a 0, while bit 15 of the PSWs used by the problem program should contain a 1. To keep the problem program from asserting too much control on the operation of a System/360 the system was designed so that a program that is operating in the problem state cannot easily change the PSW. It is for this reason that the "load PSW" and "set system mask" instructions are privileged operations.

The problem programmer may not be able to change the "current" PSW easily because of the concept of privileged instructions. However, what is to prevent the problem programmer from modifying the "new" PSWs which are in main storage? After all, any information in main storage can be treated as data and modified. The five "new" PSWs in storage locations 0088-0127 are no different in this respect. In fact, we would want the supervisor program to be able to modify this area of storage. However, we would not want the problem program to be able to modify this same area. It is undesirable to have any part of the supervisor program changeable by the problem program. What is needed here is some means by which the supervisor program can change any area of main storage while the problem program can only change its own assigned area. This concept is known as Storage Protection. The System/360 has available a tamper-proof storage protection feature. It is optional on models 30, 40 and is standard equipment on models 50,70.

The feature which prevents one program (such as the problem program) from modifying another program (such as the supervisor program) is known as _____ _____.

storage protection

To implement the storage protection feature, each main storage block of 2048 bytes has a key associated with it. This key is four bits long and may contain any number from 0 to 15. These numbers are referred to as Storage Keys. For instance, the 8K storage unit below has a key for each block of 2048 bytes of 8K main storage.



Each block of 2048 bytes in the above example has a different _____ _____.

storage key

There is a storage key for each main storage block of _____ bytes.

A 16K (actually 16,384) main storage unit would need ___ storage keys.

2048
8

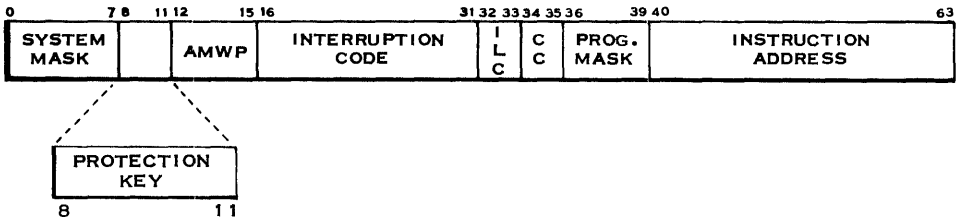
The hardware necessary for the storage keys is part of the _____ _____ feature. The keys are stored in special hardware and do not use any of the 2048 bytes with which they are associated.

---

storage protection

Besides the storage key associated with each block of 2048 bytes, there is a protection key in the "current" PSW.

THE PSW



Bits 8 - 11 of the PSW contain the _____ _____.

---

protection key

Any time the main storage unit takes a store cycle, the storage protection feature is in operation. A store cycle is one in which the information brought out of main storage is not regenerated. Instead new information is placed back into main storage.

The fetching of an instruction during I time _____ (is/is not) an example of store cycle.

---

is not

The operation of the storage protection feature is as follows:

1. On every store cycle, the protection key in the "current" PSW is compared with the storage key associated with that block of main storage.

2. A protection exception will result in a program interrupt if:
   a. The two keys are not identical and the protection key does not contain zero.

If the protection key in the "current" __ ___ ___ contains a six and a store cycle is attempted in an area whose storage key is five, a _____ interrupt will occur.

PSW program

A program interrupt will not occur if the protection key in the PSW contains a _____.

---

zero

If the key in the PSW is zero and the storage key is six, a program interrupt _____ (will/will not) occur.

---

will not; As long as the PSWs protection key is zero, the store cycle is allowed.

If the key in the PSW is six and the storage key is zero, a program interrupt _____ (will/will not) occur.

---

will

If the key in the PSW is six and a store cycle is attempted in an area whose key is five, a _____ _____ will occur.

---

program interrupt

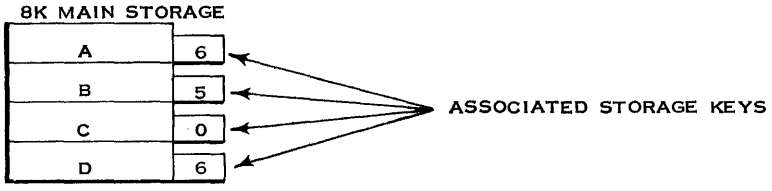
Whenever a program interrupt occurs, the interruption code placed in the "old" PSW indicates the reason for the interrupt. When storage protection is violated, a protection exception will be indicated. Refer to the Interruption Action chart.

When the keys do not agree and a program interrupt occurs, a protection exception will be indicated in the i_____ c_____ of the "old" PSW.

---

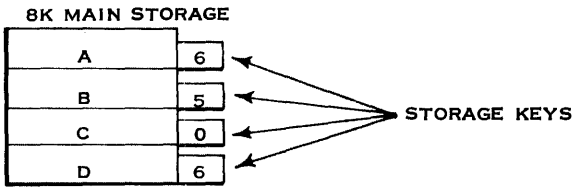
interruption code

Assuming the PSW has a protection key of six, which of the following 2K blocks of main storage can be successfully stored into? _____



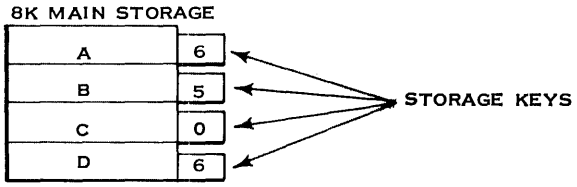
A and D; Blocks A and D have a storage key of six to match the key in the PSW.

Assuming a PSW key of five, which of the following 2K blocks can be successfully stored into? _____



B; Block B has the same key as the PSW.

Assuming a PSW key of zero, which of the following 2K bytes can be successfully stored into? _____



All of them; When the PSW has a protection key of zero, the "current" program can successfully store data anywhere in main storage. A protection key of zero would probably be in a PSW used by a supervisor program.

Let's review what you have learned so far concerning the storage protection feature.

Storage protection is optional on models 30, 40, and is _____ on models 50 and up.

---

standard

There is a storage key for each main storage block of _____ bytes.

2048

There is a protection key in bits 8 - 11 of the ___ ___ ___.

PSW

Every time a store cycle is attempted, the associated storage key and the PSW key are _____.

The store cycle is taken if the two keys are _____ (alike/different) or if the protection key in the PSW is _____.

---

compared
alike
zero

If the keys don't match and the protection key is not zero, the store cycle is not taken. Instead the current instruction is terminated and a _____ _____ occurs.

---

program interrupt

When the program interrupt occurs, a protection exception will be indicated in the interruption code of the "_____" PSW. This program interrupt would be handled by a _____ (supervisor/problem) program.

"old"
supervisor

The interruption code in the "old" PSW would indicate to the supervisor program that a problem program was trying to store data in an area of main storage that _____ (had/had not) been assigned to it.

SET STORAGE KEY

---

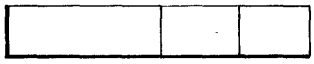
had not

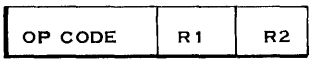
The protection key in bits 8 through 11 of the PSW cannot be altered except as a result of changing the entire PSW. The entire PSW is changeable only by the "load PSW" instruction or by an interrupt. However, the storage keys for each block of 2048 bytes can be changed by an instruction known as "set storage key." This instruction will set the storage key for one block of 2048 bytes.

To set the storage keys for each 2K block of a 16K main storage unit would require 8 executions of the "_____ _____ _____" instruction.

"set storage key"

The "set storage key" instruction is of the RR format. Label the fields of the RR format below.



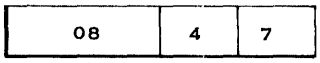


The Op code of the "set storage key" instruction is a hex 08. Show the binary bit structure of this Op code. _____

00001000

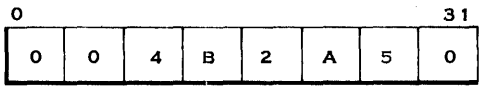
The desired storage key (0 through 15) is in bits 24 through 27 of the general register specified by the R1 field. The remainder of the register is ignored.

Given the following instruction (shown in hex), the desired storage key is in bits 24 through 27 of general register _____.



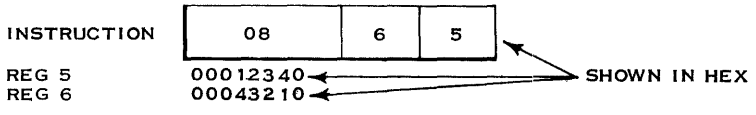
4

Given register 4 (as shown below in hex), the storage key of the 2048 byte block will be set to ____.



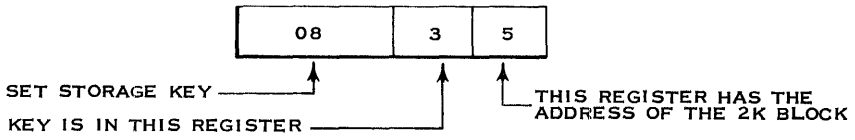
5: Only bits 24 through 27 are used.

Given the following, the storage key of a 2048 byte block will be set to ____.

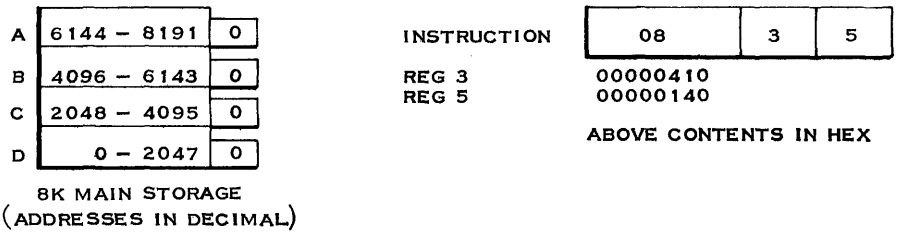


1

The question now arises: "Which 2048 byte block will have its storage key set?" This is determined by the address in the general register specified by the R2 field.



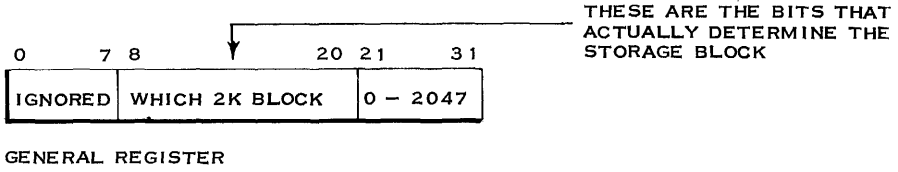
Given the following, the storage key of block ____ (A/B/C/D) will be set to ___.



D

1; In the previous problem, register 5 contains an address that indicates which block of storage was to have its storage key set. Register 5 had a hex 140 in it. You should have been able to convert this to decimal 320 by using the conversion table in the Principles of Operation manual.

Storage addresses are 24 bits in length and general registers are 32 bits in length. As you learned when you studied base addresses and displacements, addresses use the low order of a general register (that is, bits 8 through 31). Since we are concerned with blocks of 2048 bytes, only bits 8 through 20 are necessary to determine which block should have its storage key set. The low-order 11 bits of an address add up to less than 2048.



The address in the register specified by the R2 field does not have to be divisible by 2048 even though only bits 8 through 20 are needed. Any address in the 2K block may be used as long as it has 4 low-order zero bits (that is, divisible by 16).

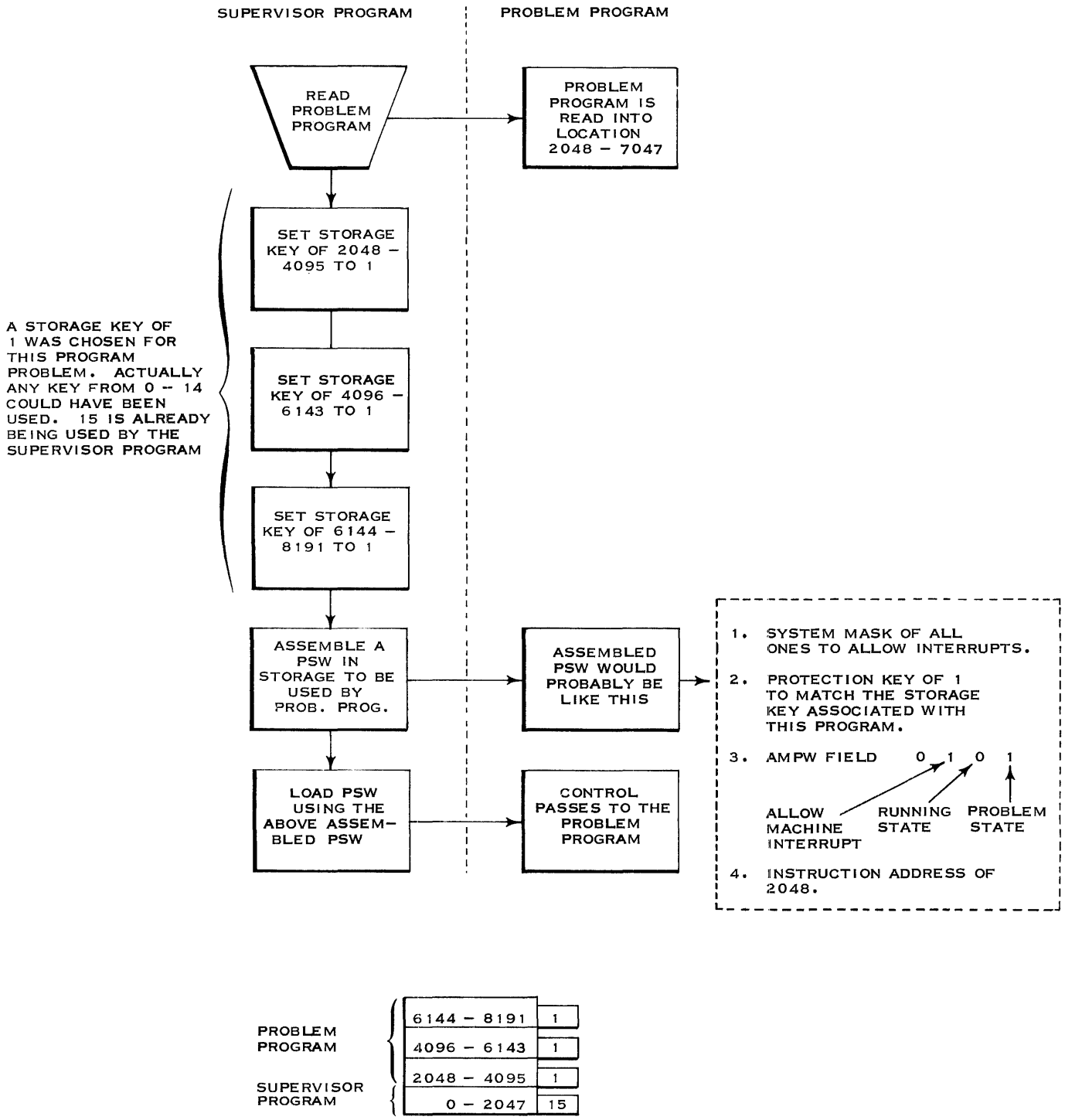
Which of the following addresses may be used to set the storage key for addresses 2048 - 4095?

a. 000A60
b. 000A61 ← Addresses in Hex
c. 001040

---

a. Would be used.
b. Would cause a program interrupt because it does not have 4 low-order zeroes.
c. Would cause the storage key to set for block 4096 - 6143.

Assume: 1. That the problem program takes 5,000 bytes and will begin at location 2048.
2. That the supervisor is in locations 0000-2047 and has a storage key of 15 and a protection key of 0.



Explanation of the preceding illustration:
The "set storage key" instruction is a privileged operation. It may be issued only when bit 15 of the PSW (problem state bit) is zero. Again, it was the intention of the "architects" of System/360 that the storage keys would be assigned by the supervisor program. In a typical supervisor-controlled operation, the supervisor would cause a problem program to be read into main storage. The supervisor would set the storage keys for the area of storage used by the problem program. The supervisor would "assemble" the PSW to be used by the problem program. This "assembled" PSW would have a protection key that matched the storage keys associated with the problem program. Now that its function of loading a problem program into main storage and assigning the keys for storage protection is done, the supervisor would pass control to this problem program. It would do this by using the "load PSW" instruction and specifying the "assembled" PSW.

NOTICE!

1. The protection key in the PSW used by the supervisor program is zero. This allows the supervisor program to store data anywhere in main storage.

2. The main storage area occupied by the supervisor program has a storage key of 15. This means that unless a problem program has a key in its PSW of 0 or 15, it will not be able to store or change information in the area being used by the supervisor program. This would be unlikely since it is the supervisor program that will be assigning storage and protection keys to the problem program.

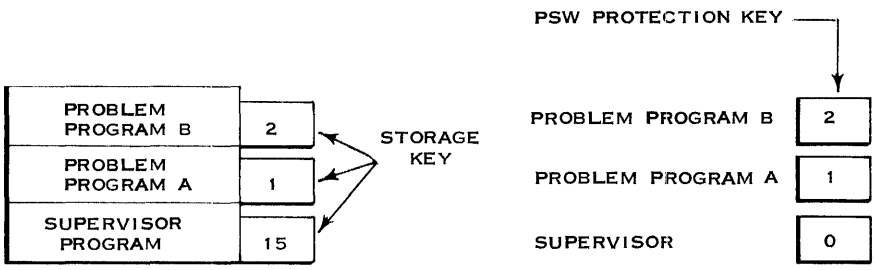
As you saw in the preceding example, each block of 2048 bytes does not need to have a different number set in its storage key. Each program in main storage should, however, have a different storage key assigned to protect one program from another. For instance, in the example the supervisor program took up one block of 2048 bytes which was assigned a storage key of 15. This storage key would most likely be assigned by the supervisor program just after it had been read into the system. The problem program was then read into the machine (as a result of a section of the supervisor program). This program took up 3 blocks of 2048 bytes. Each block was assigned the same storage key (1) by the supervisor program. The PSW for the problem program was given a protection key that matched its storage keys. This would allow the problem program to alter itself if necessary but would prevent it from altering another program.

So far, we have only discussed the concept of two programs in the computer: A supervisor program and problem program. For lack of a better name, we can consider this as the concept of a supervisor-controlled system. We have seen that much of the System/360 "hardware" is designed to take advantage of this concept, such as:

1. No halt instruction.
2. Supervisor vs. Problem State (PSW bit 15).
3. Wait vs. Running State (PSW bit 14).
4. The complex interruption system.
5. No halt on a machine or program check.
6. Storage protection.

MULTI-PROGRAMMING

There is another concept known as multi-programming. In this concept, there are two or more problem programs in the machine. Of course, just as in the supervisor-controlled concept, only one program is being executed at any one time.

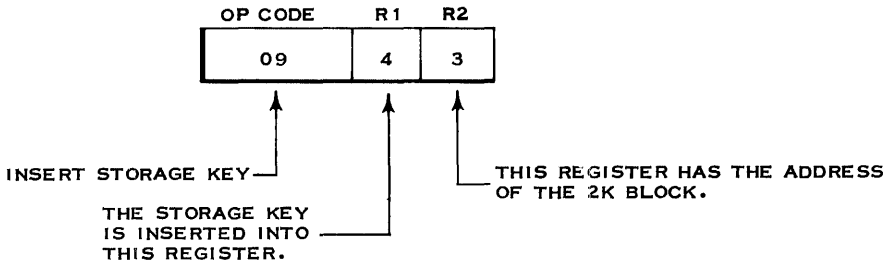


In the above example, each problem program would have a different storage key. The protection keys used by each program would also be different and would match their respective storage keys. Notice that the supervisor's protection key would not match its storage key. Since the supervisor's protection key (in its PSW) is zero, it does not have to match.

A problem program with a storage key of 7 would probably use a PSW with a protection key of ___.

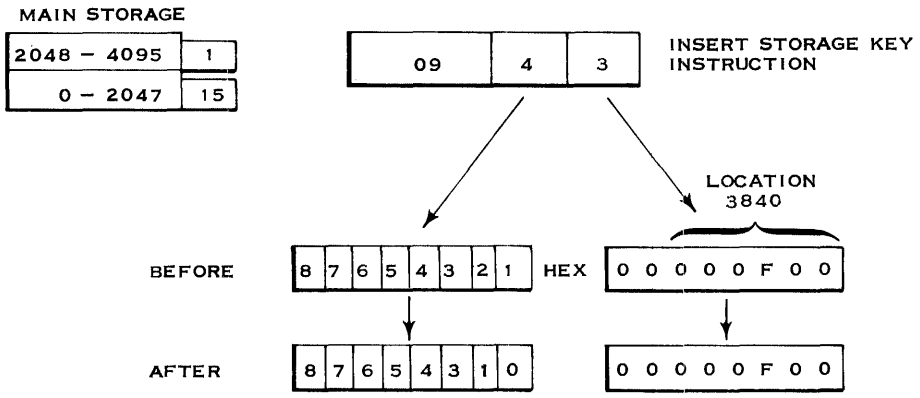
7 A supervisor program, regardless of its storage key, would probably use a PSW with a protection key of ___.

0	An area of storage with a storage key of 6 can only be altered by a program with a protection key of ____ or ____.
6, 0 (in either order)	An area of storage with a storage key of 0 can be altered: a. Only by a program with a protection key of zero. b. By any program (Circle one of the above.)
a	When there is more than one problem program in the computer, this is known as _____.
multi-programming	How many possible programs (including the supervisor) can be in the System/360 and still be protected? _____
16; 0-15 are the possible storage keys.	Besides the "set storage key" instruction, there is another instruction to help a supervisor program assign storage keys. It is called "Insert Storage Key." This instruction does not change any storage keys. Its purpose is to inspect or examine a storage key. The "insert storage key" instruction is also of the RR format.



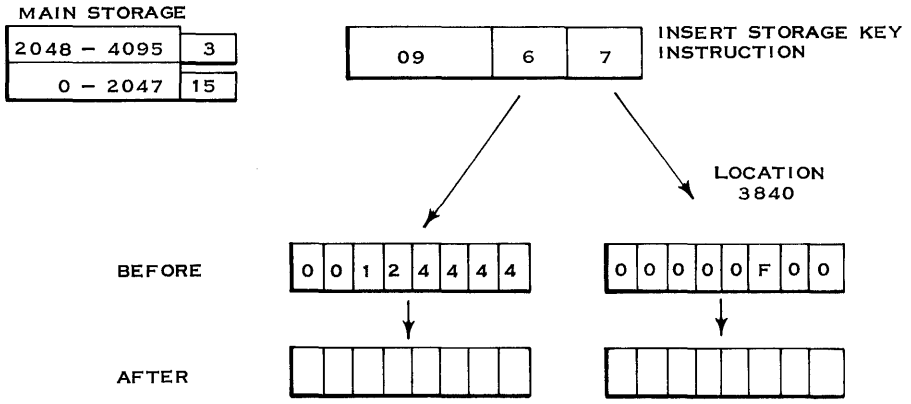
Notice that this instruction works just opposite to the "set storage key" instruction. Here the storage key of the block addressed by the contents of the register specified by the R2 field is inspected. This storage key is then inserted into bits 24 through 27 of the register specified by the R1 field. Bits 28 through 31 of this register are made zero and bits 0 through 23 remain unchanged.

Example:



Notice the storage key (1) of block 2048 - 4095 was inserted into bits 24 through 27 of register 4 while bits 28 through 31 were made zero. The remainder of the register was unchanged. The storage keys themselves were unchanged.

Given the following, indicate the register contents after the instruction is executed.



REG 6

0	0	1	2	4	4	3	0

REG 7

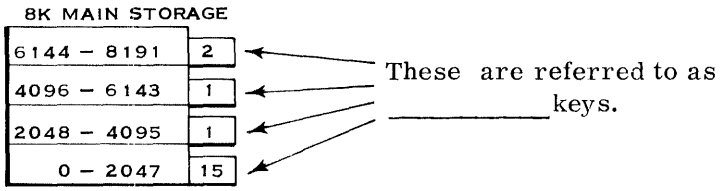
0	0	0	0	0	F	0	0

Go to the IBM System/360 Principles of Operation manual and briefly study the Protection Feature area of the System Structure section.

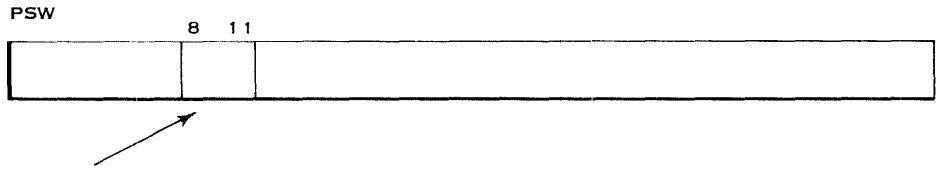
REVIEW QUESTIONS ON STORAGE PROTECTION

- Try to answer the questions without referring to the material. However, if you do require aid, refer to this book and/or the System/360 Principles of Operation manual and consider reviewing the area where aid is required.

1.



2.



This field is referred to as the _____ _____.

3. The keys in question 1 can only be changed by _____.

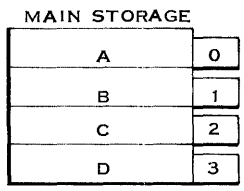
4. Bits 8 through 11 of the PSW can only be changed by (1) _____ _____ or (2) _____.

5. Storage protection applies to:

   a. Store-type main storage cycles
   b. Fetch-type main storage cycles
   c. All main storage cycles

   (Circle one of the above.)

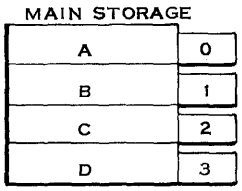
6.



Assuming a PSW key of 2:

   a. From which blocks may information be fetched? _____
   b. In which block(s) may information be altered? _____

7.



Assuming a PSW key of 0:

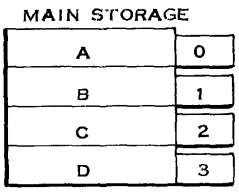
a. From which blocks may information be read out of?

_____

b. Into which blocks may information be written?

_____

8.



If a program (assuming a PSW key of 3) attempts to store data in block C:

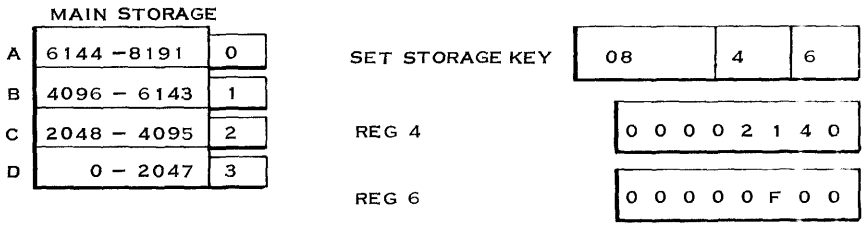
a. The data will be stored and the program will continue.

b. The data will be stored and the program will be interrupted.

c. The data will not be stored and the program will continue.

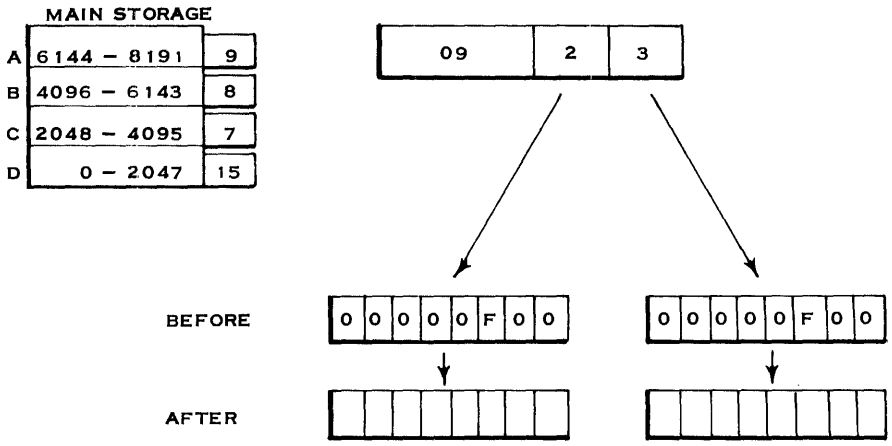
d. The data will not be stored and the program will be interrupted.

9.



The _____ (storage/protection) key of block ____ (A/B/C/D) will be set to ____.

10.



Show the register contents after executing the "insert storage key" instruction.

11. A violation of storage protection will result in a:

    a. Machine interrupt with a protection violation indicated in the interruption code of the "new" PSW.
    b. Program interrupt with a protection violation indicated in the interruption code of the "new" PSW.
    c. Program interrupt with a protection violation indicated in the interruption code of the "old" PSW.
    d. Machine interrupt with a protection violation indicated in the interruption code of the "old" PSW.

ANSWERS TO REVIEW QUESTIONS

1. storage

2. protection key

3. "Set storage key" (a privileged instruction)

4. (1) A privileged instruction called "load PSW."
   (2) An interrupt which changes the entire PSW.

5. a. Store-type main storage cycles

6. a. All blocks; Storage protection does not apply to fetch-type cycles.
   b. Block C

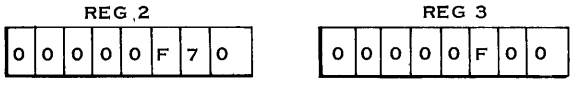
7. a. All blocks
   b. All blocks

   Note: As long as the protection key is zero, the store cycle is allowed and no interrupt occurs.

8. d

9. storage, C, 4

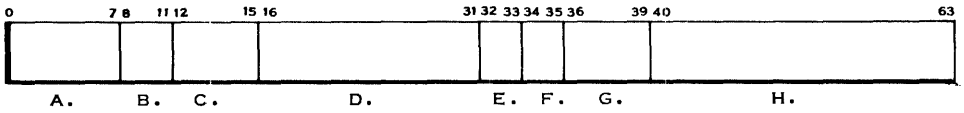
10.



11. c

PSW REVIEW

At this point we have concluded our discussion of the storage protection feature. We have also studied all fields of the program status word. Let's see if you can label the fields of the PSW.



a. _____
b. _____
c. _____
d. _____
e. _____
f. _____
g. _____
h. _____

a. System Mask
b. Protection Key
c. AMWP
d. Interruption Code
e. Instruction Length Code
f. Condition Code
g. Program Mask
h. Instruction Address

Do you need a review? If you think that you may require a review of areas of this book, do the following:

Read the learning objectives at the beginning of each section.

You should review only those areas where you think that you cannot do what the objective indicates.

Now that you have been exposed to the various fields of the program status word and their functions, as well as the instructions that affect the PSW, you are in a position to start programming a System/360. The next self-study text for this course will cover programming with fixed point arithmetic.

Remember that the important facts concerning the material you just covered in this text, are available for reference in the IBM System/360 Principles of Operation manual.

Before proceeding to the next book of this System/360 Introductory Programming Course, do the following:

1. Fill out the Course Evaluation Sheet (located in the back of this book).

2. Ask the person that is administering this course for the Mid-Course Examination.

Alphabetical Index

Book 2 System/360 Program Control and Execution
Student Course Evaluation

You can make this course and all future courses more useful by answering the questions on both sides of this sheet and giving us your comments.

Do you feel that you have an adequate understanding of the learning objectives that are listed at the beginning of the following sections?

Section I:	Instruction Formats	Yes ☐	No ☐
Section II:	Instruction Sequencing and Branching	Yes ☐	No ☐
Section III:	Interrupts	Yes ☐	No ☐
Section IV:	Storage Protection	Yes ☐	No ☐

List any technical errors you found in this book.

Comments

Please complete the information block on the opposite side. Thank you for your cooperation.
For form R23-2950-1

Field Engineering Education - Student Course Evaluation

IBM

Student Name	Man Number	B/O Number	Area Number

Student: Please review this evaluation with the person administering the course; then remove it from the book and send to the FE Education Center via IBM mail.

- Were you given a copy of this text to write in and keep? . . . . . Yes ☐ No ☐
- How many hours per day were scheduled for this course? _____
- Were you interrupted during this time? . . . . . Yes ☐ No ☐
- How many hours were needed to complete this course? _____
- Did you require assistance during this course? . . . . . Yes ☐ No ☐
  (If your answer is yes, explain in the comments section)

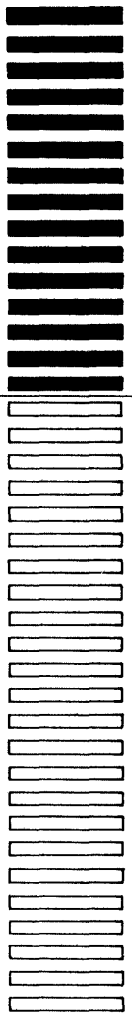
- Indicate your understanding of the total course. Excellent ☐ Good ☐ Fair ☐ Poor ☐

To be completed by course administrator	Date
Reviewed by:	

To be completed by FE Education Planning	Date
Reviewed by:	

FOLD

IBM Corporation
FE Education Planning
Department 911
South Road
Poughkeepsie, N. Y. 12602



FOLD

TEAR HERE

TEAR HERE

R23-2950-1

IBM

International Business Machines Corporation
Field Engineering Division
112 East Post Road, White Plains, N.Y. 10601

System/360 Intro-Book 2 Student Self-Study Course Printed in U.S.A. R23-2950-1