

Computer Structure and Language

Hamid Sarbazi-Azad
Department of Computer Engineering
Sharif University of Technology (SUT)
Tehran, Iran



(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #6: IBM360 Machine

2

IBM System 360/370:

- IBM360 was introduced in 1964 (IBM370 in 1970); a CISC machine that could perform about 35 KIPS (compatible to USSR's Ryad).
- IBM370 has 13 instructions more than IBM360.
- In 1989, IBM360/370 machines accounted for more than half of the estimated \$260B value of large computer systems worldwide.



(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #6: IBM360 Machine

3

Machine Structure:

Memory size:

2²⁴ bytes

Addressable unit:

8 bits (byte)

Word length:

32 bits, Big Endian, Aligned.

ISA types:

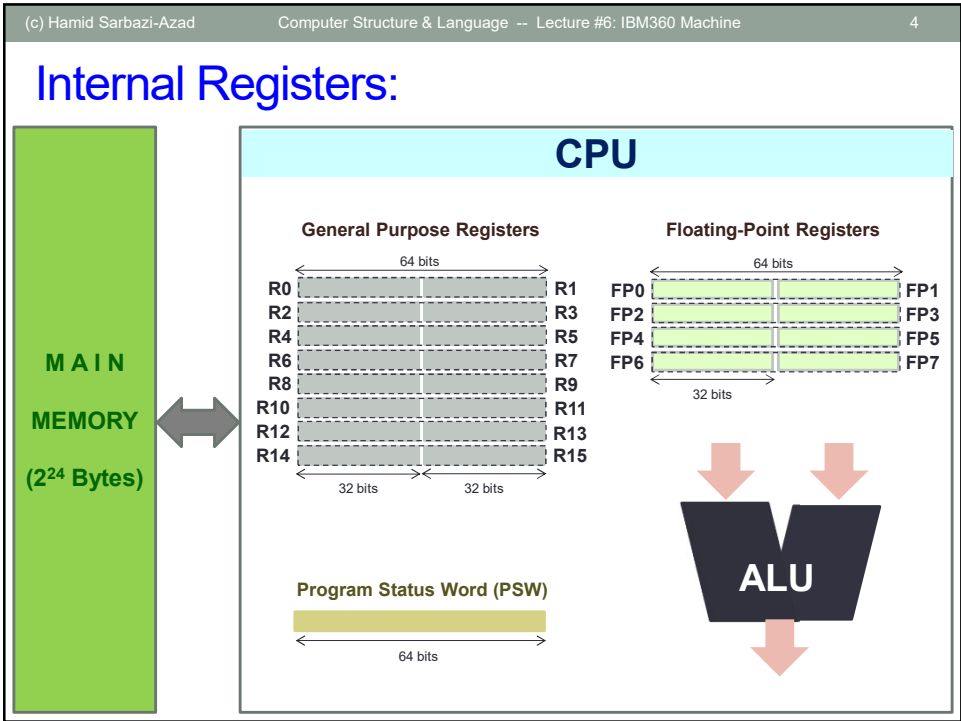
Register-Register,
Register-Memory,
Memory-Memory.

Addressing modes:

Segmented,
Indexed,
Register (direct/indirect),
Implied, and Immediate.

Data types:

Signed integer (2's complement),
Unsigned binary,
Character (byte),
Decimal,
Floating-point.



(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #6: IBM360 Machine

5

Base-Displacement Addressing:

Storage address = (Base register) + Displacement

```
graph TD; BR[Base Register R0-R15 32 bits] --> Adder[24-bit Adder]; D[Displacement 12 bits] --> Adder; Adder --> SA[Storage Address 24 bits]
```

(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #6: IBM360 Machine

6

Data Alignment in Main Memory:

Main Memory (word wide view)

210000h	11	11	00	AF
210004h	21	23	22	66
210008h	23	43	33	01
21000Ch	35	11	8C	EF
210010h	77	0C	18	27
210014h	D6	33	D7	A2
210018h	21	32	23	33
21001Ch	35	8C	77	8D
210020h	6D	07	32	FF
210024h	35	11	8C	EF
210028h	23	43	33	01
21002Ch				
210030h				
210034h				
210038h				
21003Ch				
210040h				
210044h				
210048h				

Main Memory (half-word-wide view)

210000h	11	00
210002h	21	23
210004h	22	66
210006h	23	43
210008h	33	01
21000Ah	35	11
21000Ch	8C	EF
21000Eh	77	0C
210010h	18	27
210012h	D6	33
210014h	D7	A2
210016h		
210018h		
21001Ah		
21001Ch		
21001Eh		
210020h		
210022h		
210024h		

Main Memory (byte-wide view)

210000h	11
210001h	21
210002h	22
210003h	23
210004h	33
210005h	35
210006h	8C
210007h	77
210008h	18
210009h	D6
21000Ah	D7
21000Bh	
21000Ch	
21000Dh	
21000Eh	
21000Fh	
210010h	
210011h	
210012h	

3

(c) Hamid Sarbazi-AzadComputer Structure & Language -- Lecture #6: IBM360 Machine7

Data Alignment in Main Memory (cont.):

Main Memory (double-word wide view)

210000h	11	11	11	00	AF	00	FF	A0
210008h	21	23	22	66	23	43	33	01
210010h	35	11	8C	EF	77	0C	18	78
210018h	D6	33	D7	A2	21	32	23	33
210020h	35	8C	77	8D	6D	07	32	FF
210028h	35	11	8C	EF	23	43	33	01
210030h	21	22	23	33	35	8C	77	81
210038h	8D	6D	71	66	23	43	33	01
210040h								
210048h								
210050h								
210058h								
210060h								
210068h								
210070h								
210078h								
210080h								

(c) Hamid Sarbazi-AzadComputer Structure & Language -- Lecture #6: IBM360 Machine8

Instruction Formats:

RR (Register-Register): 2 Bytes

OPCODE	r1	r2
0 0 x x x x x x	4 bits	4 bits

RX (Register-Indexed): 4 Bytes

OPCODE	r1	X2	B2	D2
0 1 x x x x x x	4 bits	4 bits	4 bits	12 bits

RS (Register-Storage): 4 Bytes

OPCODE	r1	r3	B2	D2
1 0 x x x x x x	4 bits	4 bits	4 bits	12 bits

SI (Storage-Immediate): 4 Bytes

OPCODE	I2	B1	D1
1 0 x x x x x x	8 bits	4 bits	12 bits

SS1 (Storage-Storage 1-Length): 6 Bytes

OPCODE	L-1	B1	D1	B2	D2
1 1 x x x x x x	8 bits	4 bits	12 bits	4 bits	12 bits

SS2 (Storage-Storage 2-Length): 6 Bytes

OPCODE	L1 - 1	L2 - 1	B1	D1	B2	D2
1 1 x x x x x x	4 bits	4 bits	4 bits	12 bits	4 bits	12 bits

(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #6: IBM360 Machine

9

Instruction Formats:

RR (Register-Register)

OPCODE

r1

r2

0 0 x x x x x x

4 bits

4 bits

All mnemonics of this format have character R at the end.

Example 1:

Assembly instruction: SR 12,5

Operation: $R12 \leftarrow (R12) - (R5)$

Machine code: 1BC5

Example 2:

Assembly instruction: NR 6,10

Operation: $R6 \leftarrow (R6) \wedge (R10)$

Machine code: 146A

(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #6: IBM360 Machine

10

Instruction Formats:

RX (Register-Indexed)

OPCODE

r1

X2

B2

D2

0 1 x x x x x x

4 bits

4 bits

4 bits

12 bits

Most machine instructions are coded in this format.

Example 1:

Assembly instruction: S 10,NUM

Operation: $R10 \leftarrow (R10) - (M_{NUM})$

Machine code: 5BA0C01A

NUM address = (R12) + 01Ah

Example 2:

Assembly instruction: L 15,N10

Operation: $R15 \leftarrow (M_{N10});$

Machine code: 58F0B11B

N10 address = (R11) + 11Bh

(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #6: IBM360 Machine

11

Instruction Formats:

RS (Register-Storage)

OPCODE

r1

r3

B2

D2

1 0 x x x x x x

4 bits

4 bits

4 bits

12 bits

This format includes some 3-operand instructions.

Example 1:

Assembly instruction: SLL 5,3

Operation: $R5 \leftarrow (R5) \ll 3$

Machine code: 89500003

Example 2:

Assembly instruction: LM 2,5,A

Operation: $R2 \leftarrow (M_A); R3 \leftarrow (M_{A+4}); R4 \leftarrow (M_{A+8}); R5 \leftarrow (M_{A+12});$

Machine code: 9825C01B

A address = (R12) + 01Bh

(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #6: IBM360 Machine

12

Instruction Formats:

SI (Storage-Immediate)

OPCODE

I2

B1

D1

1 0 x x x x x x

8 bits

4 bits

12 bits

Only byte immediate can be used in this format. We will see later how programmer can use other immediate data types.

Example 1:

Assembly instruction: MVI T1,X'5F'

Operation: $M_{T1} \leftarrow 5F$

Machine code: 925FC0BA

T1 address = (R12) + 0BAh

Example 2:

Assembly instruction: OI 100(3),C'*

Operation: $M_{(R3)+100} \leftarrow (M_{(R3)+100}) \vee 5C;$

Machine code: 965C3064

EBCDIC(*) = 5Ch

(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #6: IBM360 Machine

13

Instruction Formats:

SS1 (Storage-Storage 1-Length)

OPCODE

L-1

B1

D1

B2

D2

1 1 x x x x x x

8 bits

4 bits

12 bits

4 bits

12 bits

Example 1:

Assembly instruction: MVC DST(100),SRC

Operation: Move 100 bytes from M_{SRC} to M_{DST}.

Machine code: D263C00AC110

DST address = (R12) + 0Ah

SRC address = (R12) + 110h

Example 2:

Assembly instruction: OC AREA(10),10(4)

Operation: Or 10 bytes in M_{AREA} with 10 byte from M_{10+(R4)}

Machine code: D609911B400A

AREA address = (R9)+11Bh

(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #6: IBM360 Machine

14

Instruction Formats:

SS2 (Storage-Storage 2-Length)

OPCODE

L1 - 1

L2 - 1

B1

D1

B2

D2

1 1 x x x x x x

4 bits

4 bits

4 bits

12 bits

4 bits

12 bits

Example:

Assembly instruction: AP NUM1(6),NUM2(5)

Operation: Add decimal number of 5 bytes at M_{NUM2} to a decimal number of 6 bytes in M_{NUM1}.

Machine code: FA54C10ACF10

NUM1 address = (R12) + 10Ah

NUM2 address = (R12) + F10h

(c) Hamid Sarbazi-AzadComputer Structure & Language -- Lecture #6: IBM360 Machine15

Program Structure:

PROG-NAME

START

Initial Address

→ Default Initial Address is 0

.

.

START-LABEL

...

.

.

Some directives/instructions to define Base register and initialize it

.

.

Instructions

.

.

Some instructions to return the control to OS

.

.

Some directives to define variables

.

.

END

START-LABEL

(c) Hamid Sarbazi-AzadComputer Structure & Language -- Lecture #6: IBM360 Machine16

Notations:

• General format of an assembly instruction is:

Label

Mnemonic

Opr1,Opr2,Opr3

Inline Comment can come after last operand

Note that:

- Labels should start at first column.

- Full-line comment starts with * at the first column

- Mnemonic come at column 2+.

- Label, Mnemonic, Operands and then Inline comment are separated by 1+ blanks.

Example:

LOOP AR 5,6 Add the content of R6 into R5

* "LOOP" is a label, "AR" is mnemonic, "5,6" show operands, and "Add the content of R6 to R5"

* is an inline comment

L 6,ADAD1

* No label is used, "L" is mnemonic, "6,ADA1" show operands, no comment is used

8

(c) Hamid Sarbazi-AzadComputer Structure & Language -- Lecture #6: IBM360 Machine17

Notations:

You can use:

- Numbers in the assembly program in (default) decimal, Octal, Binary and Hexadecimal bases using O, B, and X prefixes.

Example: B'000001010111" == O'0127" == X'057' == 87

- **DS** (define storage) directive to define a variable (allocate space for it)

Example: ARRAY DS 20F

- **DC** (define constant) directive to define a variable and initialize it

Example: NUM1 DC F'-5'

- **EQU** (Equate) directive to define a constant.

Example: R5 EQU 5 then use as: AR R5,R5 == AR 5,5

(c) Hamid Sarbazi-AzadComputer Structure & Language -- Lecture #6: IBM360 Machine18

Examples of Variable Definition:

.....

* We define some WORD variables.

* Lets assume Location Counter = 0000FEh, here.

VAR1 DS 3F

VAR2 DC F'1'

VAR3 DC F'1',F'-2',2F'100'

.....

Symbol	Address
VAR1	000100h
VAR2	00010Ch
VAR3	000110h

Main Memory

0000FCh

000100h

000104h

000108h

00010Ch

000110h

000114h

000118h

00011Ch

000120h

--

--

--

--

00 00 00 01

00 00 00 01

FF FF FF FE

00 00 00 64

00 00 00 64

9

(c) Hamid Sarbazi-AzadComputer Structure & Language -- Lecture #6: IBM360 Machine19

Sample Program:

STOCK START 256
BEGIN BALR 11,0
 USING *,11
 L 3,OLDOH
 A 3,RECPT
 S 3,ISSUE
 ST 3,NEWOH
 EOJ
OLDOH DC F'9'
RECPT DC F'4'
ISSUE DC F'6'
NEWOH DS F

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT
000100				1	STOCK	START 256
000100	05B0			2	BEGIN	BALR 11,0
000102				3		USING *,11
000102	5830 B012		00114	4		L 3,OLDOH
000106	5A30 B016		00118	5		A 3,RECPT
00010A	5830 B01A		0011C	6		S 3,ISSUE
00010E	5030 B01E		00120	7		ST 3,NEWOH
				8		EOJ
000114	00000009			9	OLDOH	DC F'9'
000118	00000004			12	RECPT	DC F'4'
00011C	00000006			13	ISSUE	DC F'6'
000120				14	NEWOH	DS F
000100				15	END	BEGIN

The listing of the program

(c) Hamid Sarbazi-AzadComputer Structure & Language -- Lecture #6: IBM360 Machine20

End of Slides