

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 #7: IBM360 Machine

2

RR Instructions:

OPCODE

r1

r2

Load Register

Mnemonic: LR r1,r2

Operation: $r1 \leftarrow (r2);$

OPCODE: 18h

Example 1:

Assembly instruction: LR 3,5

Operation: $R3 \leftarrow (R5);$

Machine code: 1835

Example 2:

Assembly instruction: LR B'101',X'D'

Operation: $R5 \leftarrow (R13);$

Machine code: 185D

- CC ← 00; if the result = 0
- CC ← 01; if the result < 0
- CC ← 10; if the result > 0
- CC ← 11; on overflow

Machine code: 1351

(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #7: IBM360 Machine

5

RR Instructions:

OPCODE

r1

r2

Load Negative Register

Mnemonic:

LNR

r1,r2

Operation:

$r1 \leftarrow -|(r2)|$; and update CC accordingly;

OPCODE:

11h

Example 1:

Assembly instruction:

LNR

2,4

assume (R4) = -123

Operation:

$R2 \leftarrow -123$; and $CC \leftarrow 01$;

Machine code:

1124

Example 2:

Assembly instruction:

LNR

5,5

assume (R5) = 13

Operation:

$R5 \leftarrow -13$; and $CC \leftarrow 01$;

Machine code:

1155

(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #7: IBM360 Machine

6

RR Instructions:

OPCODE

r1

r2

Load Positive Register

Mnemonic:

LPR

r1,r2

Operation:

$r1 \leftarrow |(r2)|$; and update CC accordingly;

OPCODE:

10h

Example 1:

Assembly instruction:

LPR

B'010',7

assume (R7) = -123

Operation:

$R2 \leftarrow 123$; and $CC \leftarrow 10$;

Machine code:

1027

Example 2:

Assembly instruction:

LPR

5,B'101'

assume (R5) = 13

Operation:

$R5 \leftarrow 13$; and $CC \leftarrow 10$;

Machine code:

1055

RR Instructions:

OPCODE	r1	r2

Add Register

Mnemonic: AR r1,r2

Operation: $r1 \leftarrow (r1) + (r2)$; and update CC accordingly;

OPCODE: 1Ah

Example 1:

Assembly instruction: AR 2,7 assume (R2) = -123, (R7) = 5

Operation: $R2 \leftarrow -118$; and $CC \leftarrow 01$;

Machine code: 1A27

Example 2:

Assembly instruction: AR 5,6 assume (R5) = -13, (R6) = 20

Operation: $R5 \leftarrow 7$; and $CC \leftarrow 10$;

Machine code: 1A56

RR Instructions:

OPCODE	r1	r2

Subtract Register

Mnemonic: SR r1,r2

Operation: $r1 \leftarrow (r1) - (r2)$; and update CC accordingly;

OPCODE: 1Bh

Example 1:

Assembly instruction: SR 8,8 assume (R8) = -50000

Operation: $R8 \leftarrow 0$; and $CC \leftarrow 00$;

Machine code: 1B88

Example 2:

Assembly instruction: SR 5,6 assume (R5) = -13, (R6) = -22

Operation: $R5 \leftarrow 9$; and $CC \leftarrow 10$;

Machine code: 1B56

(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #7: IBM360 Machine

9

RR Instructions:

OPCODE

r1

r2

Multiply Register

Mnemonic:

MR

r1,r2

r1 must be an even numbered register

Operation:

$r1:r1+1 \leftarrow (r1+1)*(r2)$; and update CC accordingly;

OPCODE:

1Ch

Example 1:

Assembly instruction:

MR

8,4

assume (R8)=-5, (R9)=5, (R4)=-7

Operation:

$R8:R9 \leftarrow -35$; and $CC \leftarrow 01$; $\rightarrow (R8)=-1, (R9)=-35$

Machine code:

1C84

Example 2:

Assembly instruction:

MR

4,5

assume (R4)=-10, (R5)=-2

Operation:

$R4:R5 \leftarrow 4$; and $CC \leftarrow 10$; $\rightarrow (R4)=0, (R5)=4$

Machine code:

1C45

(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #7: IBM360 Machine

10

RR Instructions:

OPCODE

r1

r2

Divide Register

Mnemonic:

DR

r1,r2

r1 must be an even numbered register

Operation:

$r1+1 \leftarrow \frac{(r1):(r1+1)}{(r2)}$; $r1 \leftarrow \text{remainder}$; update CC accordingly;

OPCODE:

1Dh

Example 1:

Assembly instruction:

DR

8,4

assume (R8):(R9)=503, (R4)=5

Operation:

$R9 \leftarrow 100$; $R8 \leftarrow 3$; and $CC \leftarrow 10$;

Machine code:

1D84

Example 2:

Assembly instruction:

DR

4,6

assume (R4):(R5) =-11, (R6)=3

Operation:

$R5 \leftarrow -3$; $R4 \leftarrow -2$; and $CC \leftarrow 01$;

Machine code:

1D46

RR Instructions:

OPCODE	r1	r2

Add Logical Register

Mnemonic: ALR r1,r2 Pure binary addition
 Operation: $r1 \leftarrow (r1) + (r2)$; and update CC accordingly;
 OPCODE: 1Eh

Example 1:

Assembly instruction: ALR 2,7 assume (R2) = FFFFFFFEh, (R7) = 5
 Operation: $R2 \leftarrow 3$; and $CC \leftarrow 11$;
 Machine code: 1E27

Example 2:

Assembly instruction: ALR 5,6 assume (R5) = 13, (R6) = 20
 Operation: $R5 \leftarrow 33$; and $CC \leftarrow 10$;
 Machine code: 1E56

RR Instructions:

OPCODE	r1	r2

Subtract Logical Register

Mnemonic: SLR r1,r2 Pure binary subtract
 Operation: $r1 \leftarrow (r1) - (r2)$; and update CC accordingly;
 OPCODE: 1Fh

Example 1:

Assembly instruction: SLR 8,8 assume (R8) = FFF78955h
 Operation: $R8 \leftarrow 0$; and $CC \leftarrow 00$;
 Machine code: 1F88

Example 2:

Assembly instruction: SLR 5,6 assume (R5) = 20, (R6) = 32
 Operation: $R5 \leftarrow \text{FFFFFFF4h}$; and $CC \leftarrow 11$;
 Machine code: 1F56

(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #7: IBM360 Machine

13

RR Instructions:

OPCODE

r1

r2

And Register

Mnemonic:

NR

r1,r2

Bitwise Logical And

Operation:

$r1 \leftarrow (r1) \wedge (r2)$; and update CC accordingly;

OPCODE:

14h

Example 1:

Assembly instruction:

NR

4,9

assume (R4)=FFFFFFFEh, (R9) = 1

Operation:

$R4 \leftarrow 0$; and $CC \leftarrow 00$;

Machine code:

1449

$00...00 = 11...110 \wedge 00...01$

Example 2:

Assembly instruction:

NR

5,6

assume (R5) = 20, (R6) = 36

Operation:

$R5 \leftarrow 4$; and $CC \leftarrow 10$;

Machine code:

1456

$00...00100 = 00...010100 \wedge 00...0100100$

(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #7: IBM360 Machine

14

RR Instructions:

OPCODE

r1

r2

Or Register

Mnemonic:

OR

r1,r2

Bitwise Logical Or

Operation:

$r1 \leftarrow (r1) \vee (r2)$; and update CC accordingly;

OPCODE:

16h

Example 1:

Assembly instruction:

OR

4,6

assume (R4)=FFFFFFFEh, (R6)=1

Operation:

$R4 \leftarrow -1$; and $CC \leftarrow 01$;

Machine code:

1646

$11...11 = 11...110 \vee 00...01$

Example 2:

Assembly instruction:

OR

5,6

assume (R5) = 20, (R6) = 36

Operation:

$R5 \leftarrow 52$; and $CC \leftarrow 10$;

Machine code:

1656

$00...0110100 = 00...010100 \vee 00...0100100$

(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #7: IBM360 Machine

15

RR Instructions:

OPCODE

r1

r2

Exclusive-Or Register

Mnemonic:

XR

r1,r2

Bitwise Logical Exclusive-Or

Operation:

$r1 \leftarrow (r1) \text{ xor } (r2); \text{ and update CC accordingly;}$

OPCODE:

17h

Example 1:

Assembly instruction:

XR

4,6

assume (R4)=FFFFFFFEh, (R6)=11

Operation:

$R4 \leftarrow -11; \text{ and } CC \leftarrow 01;$

Machine code:

1746

$11...10101 = 11...110 \text{ xor } 00...01011$

Example 2:

Assembly instruction:

XR

5,6

assume (R5) = -1, (R6) = -10

Operation:

$R5 \leftarrow 9; \text{ and } CC \leftarrow 10;$

Machine code:

1756

$00...001001 = 11...11111 \text{ xor } 11...10110$

(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #7: IBM360 Machine

16

RR Instructions:

OPCODE

r1

r2

Compare Register

Mnemonic:

CR

r1,r2

Operation:

Realize (r1)-(r2), and update CC accordingly;

OPCODE:

19h

CC \leftarrow 00; if (r1)=(r2)

CC \leftarrow 01; if (r1)<(r2)

CC \leftarrow 10; if (r1)>(r2)

Example 1:

Assembly instruction:

CR

8,8

assume (R8) = -50780

Operation:

$CC \leftarrow 00;$

Machine code:

1988

Example 2:

Assembly instruction:

CR

5,6

assume (R5) = -13, (R6) = -22

Operation:

$CC \leftarrow 10;$

Machine code:

1956

(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #7: IBM360 Machine

17

RR Instructions:

OPCODE

r1

r2

CC ← 00; if (r1)=(r2)

CC ← 01; if (r1)<(r2)

CC ← 10; if (r1)>(r2)

Compare Logical Register

Mnemonic:

CLR

r1,r2

Pure binary comparison

Operation:

Realize (r1)-(r2) in pure binary, and update CC accordingly;

OPCODE:

15h

Example 1:

Assembly instruction:

CLR

8,8

assume (R8) = FFF78901h

Operation:

CC ← 00;

Machine code:

1588

Example 2:

Assembly instruction:

CLR

5,6

assume (R5) = -1, (R6) = 31

Operation:

CC ← 10;

Machine code:

1556

11...11111 > 00...011111

(c) Hamid Sarbazi-Azad

Computer Structure & Language -- Lecture #7: IBM360 Machine

18

RR Instructions:

OPCODE

r1

r2

Branch on Count Register

Mnemonic:

BCTR

r1,r2

Operation:

$r1 \leftarrow (r1) - 1$; if (r1)≠0 then PC ← (r2);

OPCODE:

06h

Example 1:

Assembly instruction:

BCTR

8,7

Assume (R8) = 2, (R7) = 011FFFF0h

Operation:

R8 ← 1; PC← (R7);

Machine code:

0687

Example 2:

Assembly instruction:

BCTR

5,0

Assume (R5) = 23, (R0) = -22

Operation:

R5 ← 22;

Machine code:

0650

This is the cheapest way to decrease the content of a register.

Tricky Facts:

All opcodes are selected carefully and accurately and NOT accidentally, either for **reducing complexity** or for the sake of **design beauty** 😊

AR 1Ah

SR 1Bh

MR 1Ch

DR 1Dh

4 NR 14h

6 OR 16h

7 XR 17h

End of Slides