Computer Structure and Language

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

Half-Word Processing: RX Format

Load Half-Word

Mnemonic:
LH r1,S2(X2)
LH r1,S2
LH r1,D2(X2,B2)
LH r1,D2(X2)
LH r1,D2(,B2)
LH r1,D2

Operation: $r1 \leftarrow (M_{D2+(B2)+(X2)})_{HW}$; The 16-bit half-word is sign-extended to 32-bit word
OPCODE: 48h

Note 1: Storage address S2 = D2 + (B2). On top of it, we have (X2) indexed to S2 to generate the final address. Final address must be EVEN.
Note 2: If R0 is used as Base Register, B2, or Index Register, X2, it is ignored and NOT accounted for address generation.

Half-Word Processing: RX Format

Load Half-Word

Example 1:

Assembly instruction: LH 2,ARRAY(4)

Symbol ARRAY has an address formed by base register 12 and displacement value 123h.

Operation: R2 ← $(M_{(R12)+(R4)+123h})_{HW}$;

Machine code: 4824C123

Example 2:

Assembly instruction: LH 2,ARRAY

Operation: R2 ← $(M_{(R12)+123h})_{HW}$;

Machine code: 4820C123

Half-Word Processing: RX Format

Load Half-Word

Example 3:

Assembly instruction: LH 2,20(4,3)

Operation: R2 ← $(M_{(R3)+(R4)+20})_{HW}$;

Machine code: 48243014

Example 4:

Assembly instruction: LH 2,0(5)

Operation: R2 ← $(M_{(R5)})_{HW}$;

Machine code: 48250000

Example 5:

Assembly instruction: LH 2,10(,5)

Operation: R2 ← $(M_{(R5)+10})_{HW}$;

Machine code: 4820500A

Half-Word Processing: RX Format

Store Half-Word

Mnemonic:
STH r1,S2(X2)
STH r1,S2
STH r1,D2(X2,B2)
STH r1,D2(X2)
STH r1,D2(,B2)
STH r1,D2

Operation: $M_{D2+(B2)+(X2)} \leftarrow (r1)_{15-0}$; The least-significant half-word in r1 is used
OPCODE: 40h

Note 1: Storage address S2 = D2 + (B2). On top of it, we have (X2) indexed to S2 to generate the final address. Final address must be EVEN.
Note 2: If R0 is used as Base Register, B2, or Index Register, X2, it is ignored and NOT accounted for address generation.

Half-Word Processing: RX Format

Store Half-Word

Example 1:
Assembly instruction: STH 2,ARR(4) Symbol ARR has an address formed by base register 12 and displacement value 100h.
Operation: $M_{(R12)+(R4)+100h} \leftarrow (R2)_{15-0}$;
Machine code: 4024C100

Example 2:
Assembly instruction: STH 2,ARR
Operation: $M_{(R12)+100h} \leftarrow (R2)_{15-0}$;
Machine code: 4020C100

Half-Word Processing: RX Format

Store Half-Word

Example 3:
Assembly instruction: STH 2,20(4,3)
Operation: $M_{(R3)+(R4)+20} \leftarrow (R2)_{15-0}$;
Machine code: 40243014

Example 4:
Assembly instruction: STH 6,0(5)
Operation: $M_{(R5)} \leftarrow (R6)_{15-0}$;
Machine code: 40650000

Example 5:
Assembly instruction: STH 3,10(,5)
Operation: $M_{(R5)+10} \leftarrow (R3)_{15-0}$;
Machine code: 4030500A

Half-Word Processing: RX Format

Add Half-Word

Mnemonic: AH r1,S2(X2)
AH r1,S2
AH r1,D2(X2,B2)
AH r1,D2(X2)
AH r1,D2(,B2)
AH r1,D2

Operation: $r1 \leftarrow (r1)+(M_{D2+(B2)+(X2)})_{HW}$ and update CC with result.
OPCODE: 4Ah
The 16-bit half-word is sign-extended to 32-bit word before addition

Note 1: Storage address S2 = D2 + (B2). On top of it, we have (X2) indexed to S2 to generate the final address. Final address must be EVEN.
Note 2: If R0 is used as Base Register, B2, or Index Register, X2, it is ignored and NOT accounted for address generation.

Half-Word Processing: RX Format

Add Half-Word

Example 1:

Assembly instruction: AH 7,ABS(6)

Symbol ABS has an address formed by base register 11 and displacement value 234h. Suppose the content of half-word $M_{(R6)+(R11)+234h}$ is -123 and (R7) = 200.

Operation: R7 ← 77; CC ← 10;

Machine code: 4A76B234

Half-Word Processing: RX Format

Add Half-Word

Example 2:

Assembly instruction: AH 4,5(6,7)

Suppose the content of half-word $M_{(R6)+(R7)+5}$ is 333 and (R4) = -400.

Operation: R4 ← -67; CC ← 01;

Machine code: 4A467005

Example 3:

Assembly instruction: AH 6,0(10)

Suppose the content of half-word $M_{(R10)}$ is 25 and (R6) = -25.

Operation: R6 ← 0; CC ← 00;

Machine code: 4A6A0000

Half-Word Processing: RX Format

Subtract Half-Word

Mnemonic:
SH r1,S2(X2)
SH r1,S2
SH r1,D2(X2,B2)
SH r1,D2(X2)
SH r1,D2(,B2)
SH r1,D2

Operation: r1 ← (r1) - $(M_{D2+(B2)+(X2)})_{HW}$ and update CC with result.
OPCODE: 4Bh
The 16-bit half-word is sign-extended to 32-bit word before subtract

Note 1: Storage address S2 = D2 + (B2). On top of it, we have (X2) indexed to S2 to generate the final address. Final address must be EVEN.
Note 2: If R0 is used as Base Register, B2, or Index Register, X2, it is ignored and NOT accounted for address generation.

Half-Word Processing: RX Format

Subtract Half-Word

Example 1:
Assembly instruction: SH 3,AAA(6)
Symbol AAA has an address formed by base register 10 and displacement value 200.
Suppose the content of half-word $M_{(R6)+(R10)+200}$ is -1 and (R3) = 7FFFFFFFh.

Operation: R3 ← 80000000h; CC ← 11;
Machine code: 4B36A0C8

Half-Word Processing: RX Format

Subtract Half-Word

Example 2:

Assembly instruction: SH 2,33(7) Suppose the content of half-word $M_{(R7)+33}$ is 20 and (R2) = -10.

Operation: R2 ← -30; CC ← 01;

Machine code: 4B270021

Example 3:

Assembly instruction: SH 0,0(1) Suppose the content of half-word $M_{(R1)}$ is 0 and (R0) = 5.

Operation: R0 ← 5; CC ← 10;

Machine code: 4B010000

Half-Word Processing: RX Format

Multiply Half-Word

Mnemonic: MH r1,S2(X2) r1 can be any of 16 registers
MH r1,S2
MH r1,D2(X2,B2)
MH r1,D2(X2)
MH r1,D2(,B2)
MH r1,D2

Operation: $r1 \leftarrow (r1)_{15\text{-}0} * (M_{D2+(B2)+(X2)})_{HW}$ and update CC with result.

OPCODE: 4Ch

Note 1: Storage address S2 = D2 + (B2). On top of it, we have (X2) indexed to S2 to generate the final address. Final address must be EVEN.

Note 2: If R0 is used as Base Register, B2, or Index Register, X2, it is ignored and NOT accounted for address generation.

Half-Word Processing: RX Format

Multiply Half-Word

Example 1:

Assembly instruction: MH 2,HAM(2)

Symbol HAM has an address formed by base register 12 and displacement value 27Fh. Suppose the content of half-word $M_{(R2)+(R12)+27Fh}$ is -1 and (R2 = 2).

Operation: R2 ← FFFFFFFEh; CC ← 01;

Machine code: 4C22C27Fh

Half-Word Processing: RX Format

Multiply Half-Word

Example 2:

Assembly instruction: MH 3,33(7)

Suppose the content of half-word $M_{(R7)+33}$ is 20 and (R3) = -10.

Operation: R3 ← -200; CC ← 01;

Machine code: 4C370021

Example 3:

Assembly instruction: MH 0,0(10)

Suppose the content of half-word $M_{(R10)}$ is 21 and (R0) = 5.

Operation: R0 ← 105; CC ← 10;

Machine code: 4C0A0000

Half-Word Processing: RX Format

Compare Half-Word

Mnemonic: CH r1,S2(X2)
CH r1,S2
CH r1,D2(X2,B2)
CH r1,D2(X2)
CH r1,D2(,B2)
CH r1,D2

Operation: Realize (r1) - $(M_{D2+(B2)+(X2)})_{HW}$ and update CC with result.

OPCODE: 49h

The 16-bit half-word is sign-extended to 32-bit word before subtract

Note 1: Storage address S2 = D2 + (B2). On top of it, we have (X2) indexed to S2 to generate the final address. Final address must be EVEN.
Note 2: If R0 is used as Base Register, B2, or Index Register, X2, it is ignored and NOT accounted for address generation.

Half-Word Processing: RX Format

Compare half-Word

Example 1:
Assembly instruction: CH 2, HEY(11)

Symbol HEY has an address formed by base register 12 and displacement value 7.
Suppose the content of half-word $M_{(R12)+(R11)+7}$ is -100 and (R2)=-120.

Operation: CC ← 01;
Machine code: 492BC007

Half-Word Processing: RX Format

Compare Half-Word

Example 2:

Assembly instruction: CH 4,B'11'(B'10',X'A') Suppose the content of half-word $M_{(R2)+(R10)+3}$ is 33 and (R4) = 140.

Operation: CC ← 10;

Machine code: 4942A003

Example 3:

Assembly instruction: CH 2,0(1) Suppose the content of half-word $M_{(R1)}$ is 35 and (R2) = 35.

Operation: CC ← 00;

Machine code: 49210000

Example: Write an assembly program to sort a100-element half-Word array ARR in ascending order (Selection Sort).

```
SORTH  START  0

       Defining R12 as base register
       & initialize it to 6  → (R12) =6.

       LA     2,99
       XR     6,6        outer index
LOP2   LA     7,2(6)     inner index
       LR     3,2
LOP1   LH     4,ARR(6)
       CH     4,ARR(7)
       BL     OUT
       LH     5,ARR(7)
       STH    4,ARR(7)
       STH    5,ARR(6)
OUT    LA     7,2(7)
       BCT    3,LOP1
       LA     6,2(6)
       BCT    2,LOP2

       Returning to OS

ARR    DS     100H
       END    SORTH
```

Example: Write an assembly program to sort a100-element half-word array ARR in ascending order (Selection Sort).

Address	Machine Code	Assembly Code			
000000		SORTH	START	0	
		Defining R12 as base register & initialize it to 6 → (R12) =6.			Six Bytes
000006	41200063		LA	2,99	
00000A	1766		XR	6,6	outer index
00000C	41760002	LOP2	LA	7,2(6)	inner index
000010	1832		LR	3,2	
000012	4846C0▾A	LOP1	LH	4,ARR(6)	
000016	4947C0▾A		CH	4,ARR(7)	
00001A	4740C0▾4		BL	OUT	
0000▾E	4857C0▾A		LH	5,ARR(7)	
0000▾2	4047C0▾A		STH	4,ARR(7)	
0000▾6	4056C0▾A		STH	5,ARR(6)	
0000▾A	41770002	OUT	LA	7,2(7)	
0000▾E	4630C00C		BCT	3,LOP1	
0000▾2	41660002		LA	6,2(6)	
0000▾6	46▾0C006		BCT	2,LOP2	
0000▾A		Returning to OS			Six Bytes
0000▾0		ARR	DS	100H	
			END	SORTH	

Symbol Table

Symbol	B	Disp.
LOP2	C	006h
LOP1	C	00Ch
OUT	C	0▾4h
ARR	C	0▾Ah

End of Slides