

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: Answering Questions
2

Format I

4bits	4bits	32bits
OPCODE	<i>r</i>	<i>data</i>

OPCODE	Mnemonic	Operation
0000	mov <i>r</i> , # <i>data</i>	$r \leftarrow \text{data};$

Format II

4bits	4bits	4bits
OPCODE	<i>r1</i>	<i>r2</i>

OPCODE	Mnemonic	Operation
0001	mov <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r2);$
0010	add <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) + (r2);$
0011	sub <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) - (r2);$
0100	and <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) \text{ and } (r2);$
0101	or <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) \text{ or } (r2);$
0110	xor <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) \text{ xor } (r2);$
----- FLOATING-POINT INSTRUCTIONS -----		
0111	ctf <i>r1</i> , <i>r2</i>	$r1 \leftarrow \text{float } (r2);$ // Convert integer to float
1000	cti <i>r1</i> , <i>r2</i>	$r1 \leftarrow \text{integer } (r2);$ // Convert float to integer
1001	efp <i>r1</i> , <i>r2</i>	$r1 \leftarrow \text{fraction } (r2);$ // Move fraction part of (<i>r2</i>) into <i>r1</i>
1010	fadd <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) + (r2);$
1011	fsub <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) - (r2);$
1100	fmul <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) * (r2);$
1101	fdiv <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) / (r2);$
1110	fsqrt <i>r1</i> , <i>r2</i>	$r1 \leftarrow \text{SQRT } (r2);$ // Calculate square-root of (<i>r2</i>)

یک کامپیوتر دارای حافظه اصلی به گنجایش 2^{29} کلمه 32 بیتی، طول واحد آدرس پذیر 4 بیتی، و 16 بیت همه منظوره $R0$ تا $R15$ است (محتویات بیت $R0$ همیشه صفر است و قابل تغییر نیست). شیوه‌های نشاندهی مورد استفاده ثباتی-مستقیم، بلافاصله و شاخص‌بندی، و شیوه نمایش اعداد ممیز ثابت مکمل 2 و ممیز شناور با استاندارد IEEE 754 Single Precision Floating-point می‌باشند. دستورات کامپیوتر در سه قالب طبق جدول زیر کد می‌شوند:

(c) Hamid Sarbazi-Azad		Computer Structure & Language -- Lecture#6: Answering Questions			3
		8 bits	4 bits	4 bits	32 bits
Format III		1111 OPCODE	r1	r2	addr

OPCODE	Mnemonic	Operation
1111 0000	mov r1, <u>addr</u> (r2)	$r1 \leftarrow (M_{\text{addr}+(r2)});$
1111 0001	mov <u>addr</u> (r2), r1	$M_{\text{addr}+(r2)} \leftarrow (r1);$
1111 0010	loop- r1, r2, <u>addr</u>	$r1 \leftarrow (r1) - (r2);$ if $(r1) > (r2)$ then PC \leftarrow <u>addr</u> ;
1111 0011	loop+ r1, r2, <u>addr</u>	$r1 \leftarrow (r1) + 1;$ if $(r1) < (r2)$ then PC \leftarrow <u>addr</u> ;
1111 0100	jl r1, r2, <u>addr</u>	if $(r1) < (r2)$ then PC \leftarrow <u>addr</u> ;
1111 0101	jh r1, r2, <u>addr</u>	if $(r1) > (r2)$ then PC \leftarrow <u>addr</u> ;
1111 0110	je r1, r2, <u>addr</u>	if $(r1) = (r2)$ then PC \leftarrow <u>addr</u> ;
1111 0111	jne r1, r2, <u>addr</u>	if $(r1) \neq (r2)$ then PC \leftarrow <u>addr</u> ;

----- FLOATING-POINT INSTRUCTIONS -----

1111 1000	fjl r1, r2, <u>addr</u>	if $(r1) < (r2)$ then PC \leftarrow <u>addr</u> ;
1111 1001	fjh r1, r2, <u>addr</u>	if $(r1) > (r2)$ then PC \leftarrow <u>addr</u> ;
1111 1010	fjnan1 r1, r2, <u>addr</u>	if $(r1) = \text{NaN}$ or $(r2) = \text{NaN}$ then PC \leftarrow <u>addr</u> ;
1111 1011	fjnan2 r1, r2, <u>addr</u>	if $(r1) = \text{NaN}$ and $(r2) = \text{NaN}$ then PC \leftarrow <u>addr</u> ;
1111 1100	fjden1 r1, r2, <u>addr</u>	if $(r1)$ or $(r2)$ is denormalized then PC \leftarrow <u>addr</u> ;
1111 1101	fjden2 r1, r2, <u>addr</u>	if $(r1)$ and $(r2)$ are denormalized then PC \leftarrow <u>addr</u> ;
1111 1110	fjinf1 r1, r2, <u>addr</u>	if $(r1) = \pm\infty$ or $(r2) = \pm\infty$ then PC \leftarrow <u>addr</u> ;
1111 1111	fjinf2 r1, r2, <u>addr</u>	if $(r1) = \pm\infty$ and $(r2) = \pm\infty$ then PC \leftarrow <u>addr</u> ;

(c) Hamid Sarbazi-Azad		Computer Structure & Language -- Lecture#6: Answering Questions			4
------------------------	--	---	--	--	---

۱- طول تمامی ثباتهای ماشین را تعیین کنید. (۱ نمره)

$L_{\text{MAR}} = 32 \text{ bits};$
 $L_{\text{MBR}} = 32 \text{ bits};$
 $L_{\text{R0-R15}} = 32 \text{ bits};$
 $L_{\text{IR}} = 48 \text{ bits};$
 $L_{\text{PC}} = 32 \text{ bits};$

(c) Hamid Sarbazi-Azad Computer Structure & Language -- Lecture#6: Answering Questions 5

۲- برنامه‌ای به زبان اسمبلی بنویسید که برای عدد ممیزشناور x مقدار $e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$ را با دقت 2^{-10} محاسبه کند. (۳ نمره)

```

org 0

mov R1,#1.0      // 1.0
mov R15,R1       // EXP(X)
mov R2,X(R0)     // X
mov R3,R1        // counter i
mov R10,R1       // X^i
mov R4,#1.0e-10  // epsilon

loop:  fmul R10,R2 // X^i-1 * X
      fdiv R10,R3  // X^i / i
      fadd R15,R10 // Sigma
      fadd R3,R1   // i=i+1
      fjh R10,R4,loop
      mov EXP(R0),R15
      je R0,R0,OS_Control_Point

EXP:   dw 1.0
X:     dw 3.56e-7
end

```

(c) Hamid Sarbazi-Azad Computer Structure & Language -- Lecture#6: Answering Questions 6

۳- برنامه زیر چه می‌کند؟ مقدار اولیه تمامی ثباتها مثبت است. (۳ نمره)

۴- برنامه اسمبلی سوال ۳ را به کد ماشین ترجمه کنید. (۳ نمره)

```

org 0

float1: dw -1.0e+109 ; e = +109
int1:   dw 3300000h
array:  dw 100 dup(?),
        dw 5000000h,1040000h
        dw 80100h, 3200Ah, 0

mov R11,R0
loop: mov R2,array-8(R1)
      dw -40.5e+112 ; e = +112
      dw 38BFC22h,39Ah
      loop- R1,R4,loop
out1: add R10,R5
      loop- R1,R4,loop
out2: add R11,R5
      loop- R1,R4,loop

je R0,R0,OS_control_point

end

```

(c) Hamid Sarbazi-AzadComputer Structure & Language -- Lecture#6: Answering Questions7

AddressMachine CodeAssembly Code

00000000mov R5,#1 → mov R4,#81111011000000000000000000000000

00000000F6000000 = je R0,R0,330h float1: dw -1.0e+109 // e = +109

0000000803300000 0330 used by je int1: dw 3300000h

00000010??????? ?????? array: dw 100 dup(?),

0000033005000000 01040000 dw 5000000h,1040000h

0000034000080100 0003200A 00000000 dw 80100h, 3200Ah, 0

mov R1,#800 → mov R10,#01111010001000100000000000000000

000003581B0 mov R11,R0

0000035BF02100000008 loop: mov R2,array-8(R1)

00000367FA220000 = fjan1 R2,R2,out1 dw -40.5e+112 // e = +112

0000036F038BFC22 0000039A = fjden1 R2,R2,out1 dw 38BFC22h,39Ah

0000037FF2140000035B loop- R1,R4,loop

0000038B2A5 out1: add R10,R5

0000038EF2140000035B loop- R1,R4,loop

0000039A2B5 out2: add R11,R5

0000039DF2140000035B loop- R1,R4,loop

000003A9F600???????? je R0,R0,OS_control_point

end

Symbol Table

SymbolAddress

float100000000h

int100000008h

array00000010h

loop0000035Bh

out10000038Bh

out20000039Ah

Format I

4bits4bits32bits

OPCODErdata

Format II

4bits4bits4bits

OPCODEr1r2

Format III

8bits4bits4bits32bits

LLLL.OPCODEr1r2addr

END OF SLIDES

4