

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	<div><div>opcode</div><div>4 bits</div></div>	<div><div>r1</div><div>4 bits</div></div>	<div><div>r2 / Qdata</div><div>4 bits</div></div>
Mnemonic	Opcode	Operation	
<code>mov r1,r2</code>	0000	$r1 \leftarrow (r2);$	
<code>mov r1,(r2)</code>	0001	$r1 \leftarrow (M_{(r2)});$	
<code>mov (r1),r2</code>	0010	$M_{(r1)} \leftarrow (r2);$	
<code>mov (r1),(r2)</code>	0011	$M_{(r1)} \leftarrow (M_{(r2)});$	
<code>add r1,r2</code>	0100	$r1 \leftarrow (r1) + (r2);$	
<code>add r1,(r2)</code>	0101	$r1 \leftarrow (r1) + (M_{(r2)});$	
<code>add (r1),r2</code>	0110	$M_{(r1)} \leftarrow (M_{(r1)}) + (r2);$	
<code>sub r1,r2</code>	0111	$r1 \leftarrow (r1) - (r2);$	
<code>sub r1,(r2)</code>	1000	$r1 \leftarrow (r1) - (M_{(r2)});$	
<code>sub (r1),r2</code>	1001	$M_{(r1)} \leftarrow (M_{(r1)}) - (r2);$	
<code>movq r1,qdata2</code>	1010	$r1 \leftarrow qdata2;$	
<code>addq r1,qdata2</code>	1011	$r1 \leftarrow (r1) + qdata2;$	
<code>subq r1,qdata2</code>	1100	$r1 \leftarrow (r1) - qdata2;$	
<code>swap r1,r2</code>	1101	$(r1) \Leftrightarrow (r2);$	
<code>swap (r1),(r2)</code>	1110	$(M_{(r1)}) \Leftrightarrow (M_{(r2)});$	

یک کامپیوتر دارای حافظه اصلی به گنجایش 2^{24} واحد آدرس پذیر ۴ بیتی، طول کلمه ۳۲ بیتی و ۱۶ ثبات همه منظوره R0 تا R15 می باشد. شیوه های نشانی دهی ماشین شامل ثباتی (مستقیم و غیر مستقیم)، بلافاصله و مستقیم حافظه ای، و شیوه نمایش اعداد مکمل ۲ است. دستورات در پنج قالب و طبق جداول زیر کد میشوند.

(c) Hamid Sarbazi-Azad			Computer Structure & Language -- Lecture#6: Answering Questions		3
------------------------	--	--	---	--	---

Format II			<div> <div>1111 opcode</div> <div>r</div> <div>address</div> </div> <div> <div>8 bits</div> <div>4 bits</div> <div>24 bits</div> </div>
Mnemonic	Opcode	Operation	
<u>mov</u> r, address	0000	$r \leftarrow (M_{address});$	
<u>mov</u> address, r	0001	$M_{address} \leftarrow (r);$	
<u>jnz</u> r, address	0010	if $(r) \neq 0$ then $PC \leftarrow address;$	
<u>jz</u> r, address	0011	if $(r) = 0$ then $PC \leftarrow address;$	
<u>ineg</u> r, address	0100	if $(r) < 0$ then $PC \leftarrow address;$	
<u>jpos</u> r, address	0101	if $(r) \geq 0$ then $PC \leftarrow address;$	
loop r, address	0110	$r \leftarrow (r) - 1;$ if $(r) \neq 0$ then $PC \leftarrow address;$	
movb r, address	0111	$r \leftarrow address;$	

Format III			<div> <div>1111 opcode</div> <div>r</div> <div>data</div> </div> <div> <div>8 bits</div> <div>4 bits</div> <div>32 bits</div> </div>
Mnemonic	Opcode	Operation	
<u>mov</u> r, #data	1000	$r \leftarrow data;$	
add r, #data	1001	$r \leftarrow (r) + data;$	
sub r, #data	1010	$r \leftarrow (r) - data;$	
and r, #data	1011	$r \leftarrow (r) \wedge data;$	
or r, #data	1100	$r \leftarrow (r) \vee data;$	

Format IV			<div> <div>1111 opcode</div> <div>address</div> </div> <div> <div>8 bits</div> <div>24 bits</div> </div>
Mnemonic	Opcode	Operation	
call address	1101	$R15 \leftarrow (PC); PC \leftarrow address;$	
<u>jmp</u> address	1110	$PC \leftarrow address;$	

Format V			<div> <div>1111 opcode</div> </div> <div> <div>8 bits</div> </div>
Mnemonic	Opcode	Operation	
ret	1111	$PC \leftarrow (R15);$	

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

۳- برنامه زیر چه می کند؟ (۳ نمره)

```

    org    0
L1:  mov   R0, L1+1
      mov   R3, #800
L2:  dw    40040040h
      dw    4004004h
      dw    400711h
L3:  mov   R2, L4
      add   R1, R2
      mov   R2, L3+2
      add   R2, R0
      or    R2, #3
      mov   L3+2, R2
      addq  R3, 8h
      jnz   R3, L3
      ret
L4:  org    1000h
      end

```

۴- برنامه سوال ۳ را به کد ماشین ترجمه کنید. (۲ نمره)

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

Address	Machine Code	Assembly Code
000000	F00000001 [1111 0000 0000 000001]	org 0
000009	F8300000320 [1111 1000 0011 00000320]	L1: mov R0,L1+1
000014	40040040 =add R0,R0→add R0,R0	mov R3,#800
00001C	04004004 =add R0,R0→add R0,R0→add R0,R0	L2: dw 40040040h
000024	00400711 =add R0,R0→add R0,R0→sub R1,R1	dw 4004004h
00002C	F02001000 [1111 0000 0010 001000]	dw 400711h
000035	412 [0100 0001 0010]	L3: mov R2,L4
000038	F0200002E [1111 0000 0010 00002E]	add R1,R2
000041	420 [0100 0010 0000]	mov R2,L3+2
000044	FC200000003 [1111 1100 0010 00000003]	add R2,R0
00004F	F1200002E [1111 0001 0010 00002E]	or R2,#3
000058	B38 [1011 0011 1000]	mov L3+2,R2
00005B	F2300002C [1111 0010 0011 00002C]	addq R3,8h
000064	FF [11111111]	jnz R3,L3
001000		ret
		L4: org 1000h
		end

Format I:

opcode	r1	r2/Qdata
4 bits	4 bits	4 bits

 III:

1111 opcode	r	data
8 bits	4 bits	32 bits

 II:

1111 opcode	r	address
8 bits	4 bits	24 bits

 IV:

1111 opcode	address
8 bits	24 bits

Symbol Table

Symbol	Address
L1	000000h
L2	000014h
L3	00002Ch
L4	001000h

END OF SLIDES