



Examen 2021 Estructura de Computadores

Estructura de Computadores (Universitat Oberta de Catalunya)

Examen 2020/21-1

Assignatura Codi Data Hora inici

Estructura de computadores 05.573 10/1/2021 12:00

Pregunta 2

2.1

L'estat inicial del computador CISCA just abans de començar l'execució de cada fragment de codi (a cada apartat) és el següent:

R2 = 00000020h	M(00000290h) = 77007700h	Z = 0, C = 0, S = 0, V = 0
R4 = 00000040h	M(00000060h) = 00000810h	
R8 = 00000080h	M(00000250h) = 00001810h	

Completeu l'estat del computador després d'executar cada codi (indiqueu els valors dels registres en hexadecimal). Suposeu que l'adreça simbòlica A val 250h.

a)

ADD R2, R4
MOV R8, [R2]
SUB R8, [A]

R2 = 20h+40h = 60h

Z = 0, C = 0, S = 0, V = 0

R8 = 00000810h

Z = 0, C = 0, S = 0, V = 0

R8 = R8-[A] = 00000810h-00001810h = FFFFF000h

Z = 0, C = 1, S = 1, V = 0

R2 = 60h

R8=FFFFFF00h

Z = 0, C = 1, S = 1, V = 0

b)

MOV R2,[A+R4]
CMP [A],R2
JNE EXIT1
JMP EXIT2
EXIT1: ADD R2,R2
EXIT2:

A+R4 = 250h+40h = 290h

R2 = [290h] = 77007700h

[250h]-77007700h = 00001810h-77007700h = 88FFA110h

Z = 0, C = 1, S = 1, V = 1

R2 = R2+R2 = 77007700h+77007700h = EE00 EE00h

Z = 0, C = 0, S = 1, V = 1

R2 = EE00EE00h

Z = 0, C = 0, S = 1, V = 1

2.2

Donat el següent codi d'alt nivell:

```
if (A>=B) {  
    if (C=B) A=C;  
    else C= B;  
}  
else A=B;
```

Es proposa la següent traducció a CISCA on hem deixat 7 espais per omplir

```
INI: MOV R0, [A]  
MOV R1, [B]  
MOV R2, [C]  
CMP R0, R1  
JL LSE1  
CMP R2, R1  
JNE LSE2  
MOV R0, R2  
JMP END  
LSE2: MOV R2, R1  
JMP END  
LSE1: MOV R0, R1  
END: MOV [A], R0  
MOV [B], R1  
MOV [C], R2
```

2.3

Donat el següent fragment de codi d'un programa en llenguatge ensamblador de CISCA:

```
CONT: CMP R2, [A+R4]
      JE END
      ADD R10, [A]
      JMP CONT
```

END:

Traduiu-ho a llenguatge màquina i expresseu-ho en la següent taula. Supposeu que la primera instrucció del codi s'assembla a partir de l'adreça 003FC000h (que és el valor del PC abans de començar l'execució del fragment de codi). Supposeu que l'adreça simbòlica A val 00000040h. En la següent taula useu una fila per codificar cada instrucció. Si suposem que la instrucció comença en l'adreça @, el valor Bk de cadascun dels bytes de la instrucció amb adreces @+k per a k=0, 1,... s'ha d'indicar en la taula en hexadecimal en la columna corresponent (recordeu que els camps que codifiquen un desplaçament en 2 bytes o un immediat o una adreça en 4 bytes ho fan en format little endian, això cal tenir-ho en compte escrivint els bytes de menor pes, d'adreça més petita, a l'esquerra i els de major pes, adreça major, a la dreta). Completeu també la columna @ que indica per a cada fila l'adreça de memòria del byte B0 de la instrucció que es codifica en aquesta fila de la taula.

A continuació us donem com a ajuda les taules de codis:

Taula de codis d'instrucció

B0	Instrucció
43h	JE
26h	CMP
20h	ADD
40h	JMP

Taula de modes d'adreçament (Bk<7..4>)

Camp mode Bk<7..4>	Mode
0h	Immediat
1h	Registre
2h	Memòria
3h	Indirecte
4h	Relatiu
5h	Indexat
6h	Relatiu a PC

Taula de modes d'adreçament (Bk<3..0>)

Camp mode Bk<3..0>	Significat
Num. registre	Si el mode ha d'especificar un registre
0	No s'especifica registre.

@	Assemblador	0	1	2	3	4	5	6	7	8	9	10
003FC000h	CONT: CMP R2, [A+R4]	26	12	54	40	00	00	00				
003FC007h	JE END	43	60	0D	00							
003FC00Bh	ADD R10, [A]	20	1A	20	40	00	00	00				
003FC012h	JMP CONT	40	00	00	C0	3F	00					
003FC018h	END:											

Pregunta 3

3.1. Memòria cau

Memòria cau d'assignació directa

0, 18, 2, 22, 15, 23, 24, 50, 17, 3, 18, 19, 32, 4, 6, 65, 51, 20, 56, 50

3.1.1 La següent taula mostra l'estat inicial de la cau, que conté les primeres 32 paraules de la memòria (organitzades en 4 blocs).

Completar la taula per a mostrar l'evolució de la cau durant l'execució del programa. Per a cada accés cal omplir una columna indicant si es tracta d'un encert o una fallada.

Línia	Estat Inicial	0	18	2	22	15
0	0:0 (0 - 7)	E 0:0 (0 - 7)	0:0 (0 - 7)	E 0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)
1	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	E 1:0 (8 - 15)
2	2:0 (16 - 23)	2:0 (16 - 23)	E 2:0 (16 - 23)	2:0 (16 - 23)	E 2:0 (16 - 23)	2:0 (16 - 23)
3	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)

Línia	23	24	50	17	3	18
0	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	E 0:0 (0 - 7)	0:0 (0 - 7)
1	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)
2	E 2:0 (16 - 23)	2:0 (16 - 23)	F 6:1 (48 - 55)	F 2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)
3	3:0 (24 - 31)	E 3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)

Línia	19	32	4	6	65	51
0	0:0 (0 - 7)	F 4:1 (32 - 39)	F 0:0 (0 - 7)	E 0:0 (0 - 7)	F 8:2 (64 - 71)	8:2 (64 - 71)
1	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)
2	E 2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	6:1 (48 - 55)
3	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)

Línia	20	56	50			
0	8:2 (64 - 71)	8:2 (64 - 71)	8:2 (64 - 71)			
1	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)			
2	F 2:0 (16 - 23)	2:0 (16 - 23)	F 6:1 (48 - 55)			
3	3:0 (24 - 31)	F 7:1 (56 - 63)	7:1 (56 - 63)			

3.1.2 a) Quina és la taxa d'encerts (T_e) ?

$$11/20 = 0.55$$

3.1.2 b) Suposem que el temps d'accés a la memòria cau, o temps d'accés en cas d'encert (t_e), és de 5 ns i el temps total d'accés en cas de fallada (t_f) és de 25 ns. Considerant la taxa d'encerts obtinguda a la pregunta anterior, quin és el temps mitjà d'accés a memòria (t_m) ?

$$t_m = (T_e \times t_e) + (1 - T_e) \times t_f = (0.55 \cdot 5) + (1 - 0.55) \cdot 25 = 14 \text{ ns}$$

3.2 Sistema d'E/S

E/S programada

Es vol analitzar el rendiment de la comunicació de dades entre la memòria d'un processador i un port USB, utilitzant E/S programada, amb les següents característiques:

- Velocitat de transferència del dispositiu d'E/S $v_{\text{transf}} = 20 \text{ MBytes/s} = 20000 \text{ Kbytes/s}$
- Temps de latència mitjà del dispositiu $t_{\text{latència}} = 0$
- Adreces dels **registres d'estat** i **dades** del controlador d'E/S: 0A10h i 0A14h, respectivament
- El bit del **registre d'estat** que indica que el controlador del port d'E/S està disponible és el bit 4, o sigui el cinquè bit menys significatiu (quan val 1 indica que està disponible)
- Processador amb una freqüència de rellotge de 2 GHz, el temps de cicle $t_{\text{cicle}} = 0,5 \text{ ns}$.
- El processador executa una instrucció cada dos cicles de rellotge
- Transferència de **escriptura** des de memòria al port d'E/S
- Transferència de $N_{\text{dades}} = 2.500.000$ dades
- La mida d'una dada és $m_{\text{dada}} = 4 \text{ bytes}$
- Adreça inicial de memòria on resideixen les dades: A0000000h

3.2.1 El següent codi realitzat amb el joc d'instruccions CISCA realitza la transferència descrita abans mitjançant la tècnica d'E/S programada. Completeu el codi.

```
1. MOV R3, 2500000
2. MOV R2, A0000000h
3. Bucle: IN R0, [0A10h]; llegir 4 bytes
4. AND R0, 00010000b
5. JE Bucle
6. MOV R0, [R2] ; llegir 4 bytes
7. ADD R2, 4
8. OUT [0A14h], R0 ; escriure 4 bytes
9. SUB R3
10. JNE Bucle
```

3.2.2 Quant temps dura la transferència del bloc de dades $t_{\text{transf_bloc}}$?

$$t_{\text{transf_bloc}} = t_{\text{latència}} + (N_{\text{dades}} \cdot t_{\text{transf_dato}})$$

$$t_{\text{latència}} = 0$$

$$t_{\text{transf_dato}} = m_{\text{dato}} / v_{\text{transf}} = 4 \text{ Bytes} / 20000000 \text{ Bytes/s} = 0.0000002 \text{ s}$$

$$t_{\text{transf_bloc}} = 0 + (2500000 \cdot 0.0000002) = 0.5 \text{ s}$$

3.2.3 Si volguéssim fer servir el mateix processador i el mateix programa però amb un dispositiu d'E/S més ràpid, quina és la màxima taxa o velocitat de transferència del nou dispositiu que es podria suportar sense que el dispositiu s'hagués d'esperar?

$$t_{\text{ciclo}} = 0.5 \text{ ns}$$

$$t_{\text{instr}} = 2 \cdot 0.5 = 1 \text{ ns para ejecutar una instrucción}$$

Para cada $m_{\text{dato}} = 4 \text{ bytes}$ se necesita ejecutar 8 instrucciones de la rutina

$$1 \cdot 8 = 8 \text{ ns}$$

La velocidad máxima del dispositivo será:

$$4 \text{ bytes} / 8 \text{ ns} = 0.5 \text{ bytes/ns}$$