Espai grapa

Examen 3

Assignatura	Codi	Data	Hora inici
Estructura de computadors			

⊂05.573ℜ28ℜ06ℜ14ℜEΞ}∈
05.573 28 06 14 EX

Enganxeu en aquest espai una etiqueta identificativa amb el vostre codi personal
Examen

Fitxa tècnica de l'examen

- **Comprova que el codi i el nom de l'assignatura corresponen a l'assignatura en la qual estàs matriculat.**
- **Només has d'enganxar una etiqueta d'estudiant a l'espai corresponent d'aquest full.**
- **No es poden adjuntar fulls addicionals.**
- **No es pot realitzar la prova en llapis ni en retolador gruixut.**
- **Temps total:** 2 **h.**
- **En cas que els estudiants puguin consultar algun material durant l'examen, quin o quins materials poden consultar?**
  No es pot utilitzar calculadora, ni material auxiliar.
- **Valor de cada pregunta:** Pregunta 1 (20%); Pregunta 2 (40%); Pregunta 3 (40%)
- **En cas que hi hagi preguntes tipus test: Descompten les respostes errònies?** NO **Quant?**

- **Indicacions específiques per a la realització d'aquest examen:**

Enunciats

No es pot utilitzar calculadora. Cal saber interpretar un valor en binari, decimal o hexadecimal per a realitzar l'operació que es demani. I el resultat s'ha d'expressar en el format corresponent.

Examen 3

Assignatura	Codi	Data	Hora inici
Estructura de computadors			

Valoració de les preguntes de l'examen

Pregunta 1 (20%)

Pregunta sobre la pràctica.

Cal completar les instruccions marcades o afegir el codi que es demana.
Els punts suspensius indiquen que hi ha més codi però no l'heu de completar.

NOTA: En cas que el codi proposat en cada pregunta no es correspongui amb la forma que vosaltres plantejaríeu la resposta, podeu reescriure el codi o part del codi segons el vostre plantejament.

1.1: 10%

1.2: 10%

Pregunta 2 (40%)

2.1: 15%

2.2: 15%

2.3: 10%

Pregunta 3 (40%)

3.1: 20%

3.1.1: 10%

3.1.2: 10%

3.2: 20%

3.2.1: 10%

3.2.2: 10%

Examen 3

Assignatura	Codi	Data	Hora inici
Estructura de computadors			

Pregunta 1

1.1

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Treure un valor de tipus char del vector stack1.
; Com sp1 indica la primera posició lliure del vector stack1, primer
; decrementem l'índex sp1 i després agafem el valor del vector
; utilitzant sp1 com índex i el retornem.
; No cal esborrar el valor del vector, ara ja no el considerarem com
; un element de la pila.
; Per a accedir al vector stack1 (de tipus byte), s'ha d'utilitzar
; adreçament indexat: [Adreça+reg]
; Variables utilitzades: stack1, sp1
; Paràmetres d'entrada : Cap.
; Paràmetres de sortida: rax: (al) val: valor que volem retornar del vector
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
popStack1:
    push rbp
    mov  rbp, rsp

    ;guardem els registres que modifiquem i que no són paràm. de sortida.
    push rsi
    mov esi, [sp1]
    dec esi                     ;sp1--
    mov al, [stack1+esi]        ;val = stack1[sp1];
    mov [sp1], esi
    pop rsi

    mov rsp, rbp
    pop rbp
    ret
```

1.2

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Si operem amb dades de tipus BYTE (size=1), sumar els valors
; V1[0] i V1[1] i guardar el resultat a V1[2].
; IMPORTANT: Per accedir els vectors V1 i V4, no es pot fer directament,
; s'ha de fer cridant les subrutines getV1, setV1, getV4 i setV4.
; A la operació ADD considerem tots els bits de resultat: Z, S, C i V.
; Els bits de resultat s'han de consultar amb els salts condicionals
; just després de fer l'operació i abans de que es modifiquin per
; una altra instrucció. En el codi C no considerem els bits de resultat.
;
; Variables utilitzades: size, Z, S, C i V
; Paràmetres d'entrada : Cap
; Paràmetres de sortida: Cap
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
opADD2:
    ...
    cmp BYTE[size], 1
    jne opADD2_size4
  opADD2_size1:
    mov  edi, 1      ; Llegim V1[1]
    call getV1
    mov  bl, al
    mov  edi, 0       ; Llegim V1[0]
    call getV1
     add  al, bl
    jmp  opADD2_FLAG

  opADD2_size4:
    ...
  opADD2_FLAG:      ;mirem els bits de resultat. En C no es fa.
    mov BYTE[Z], '0'
    mov BYTE[S], '0'
    mov BYTE[C], '0'
    mov BYTE[V], '0'
  opADD2_FLAGZ:
     jnz opADD2_FLAGS
    mov BYTE[Z], '1'
  opADD2_FLAGS:
     jns opADD2_FLAGC
    mov BYTE[S], '1'
  opADD2_FLAGC:
    ...
  opADD2_save:
    ...
  opADD2_end:
    ...
    ret
```

Examen 3

Assignatura	Codi	Data	Hora inici
Estructura de computadors			

Pregunta 2

2.1

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

R1 = 00000008h R2 = 0000000Fh R10 = 00000050h	M(00000400h) = 00000410h M(00000800h) = 00000810h M(000000EFh) = 00000100h	Z = 0, C = 0, S = 0, V = 0

Completeu l'estat del computador després d'executar cada codi (indiqueu els valors dels registres en hexadecimal).
Suposeu que l'adreça simbòlica v val 400h i l'adreça simbòlica B val 800h

a)

```
ADD  R1, 10h
XOR  R10,[R2+E0h]
MOV  [B+R2],R10
```

R1= 8h+ 10h= 18h
R2+E0=0Fh+E0h=EFh
[R2+E0]=F0h+10h=100h
R10= 50h XOR 100h = 150h
[B+R2]= [80FH]= 150h

R1= 18h
R10 = 150h
[80Fh]= 150h

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

b)

```
MOV [V], 10h
SAL [V], 10h
```

[V] = 10h
[V]= 00100000h

[V]= 00100000h
Z = 0, C = 0 , S = 0 , V = 0

2.2

Donat el següent codi d'alt nivell:

```
for (i=0,j=8; j>0; i++, j--)
    {
        V1[i]= V2[j];
    }
```

V1 i V2 són vectors de 9 elements de 4 bytes cadascun. Es proposa la següent traducció a CISCA on hem deixat 5 llocs per omplir:

Examen 3

Assignatura	Codi	Data	Hora inici
Estructura de computadors			

```
INI:    MOV R1, 32          ; Has d'especificar el segon operand.
        MOV R3, V1
REP:    MOV R4, [V2+R1]     ; Has d'especificar el segon operand.
        MOV [R3], R4        ; Has d'especificar el primer operand.
        ADD R3, 4
        SUB R1, 4           ; Has d'especificar la instrucció.
        CMP R1, 0
        JG REP              ; Has d'especificar la instrucció.
```

Examen 3

Assignatura	Codi	Data	Hora inici
Estructura de computadors			

2.3

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

```
MOV R2, [V]
SAR R1, 1
AND R2, [V+R1]
```

Traduïu-ho a llenguatge màquina i expresseu-ho en la següent taula. Suposeu que la primera instrucció del codi es troba a partir de l'adreça 00FF0010h (que és el valor del PC abans de començar l'execució del fragment de codi). Suposeu també que l'adreça simbòlica V val 00001D80h. En la taula de resultats 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
10h	MOV
30h	AND
36h	SAR

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
Nº registre	Si el mode ha d'especificar un registre
0	No s'especifica registre.

		Bk per a k=0..10										
@	**Assemblador**	0	1	2	3	4	5	6	7	8	9	10
00FF0010	MOV R2, [V]	10	12	20	80	1D	00	00				
00FF0017	SAR R1, 1	36	11	00	01	00	00	00				
00FF001E	AND R2, [V+R1]	30	12	51	80	1D	00	00				
00FF0025												

Examen 3

Assignatura	Codi	Data	Hora inici
Estructura de computadors			

Pregunta 3

3.1

Tenim un sistema de memòria en el que tots els accessos es fan a paraula (no ens importa quina és la mida d'una paraula). Suposarem que l'espai d'adreces de memòria es descompon en blocs de 8 paraules. Cada bloc comença a una adreça múltiple de 8. Així, el bloc 0 conté les adreces 0, 1, 2, 3, 4, 5, 6 i 7; el bloc 1, les adreces 8, 9, 10, 11, 12, 13, 14 i 15, i el bloc N les adreces 8*N, 8*N+1, 8*N+2, 8*N+3, 8*N+4, 8*N+5, 8*N+6 i 8*N+7.

Suposem que el sistema també disposa d'una memòria cau de 4 línies (on cada línia té la mida d'un bloc, és a dir, 8 paraules). Aquestes línies s'identifiquen com a línies 0, 1, 2 i 3. Quan es fa referència a una adreça de memòria principal, si aquesta adreça no es troba a la memòria cau, es porta tot el bloc corresponent des de la memòria principal a una línia de la memòria cau (així si fem referència a l'adreça 2 de memòria principal portarem el bloc format per les paraules 0, 1, 2, 3, 4, 5, 6 i 7).

3.1.1 Memòria Cau d'Accés Directe

Suposem que el sistema fa servir una **política d'assignació directa**, de manera que cada bloc de la memòria principal només es pot portar a una línia determinada de la memòria cau.

L'execució d'un programa genera la següent llista de lectures a memòria:

14, 23, 24, 25, 26, 7, 32, 33, 34, 8, 55, 56, 6, 5, 12, 13, 14, 32, 40, 33

3.1.1.a) 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 fallada en la cau cal omplir una nova columna indicant quina referència a memòria ha provocat la fallada i el canvi que es produeix en l'estat de la memòria cau (la línia que es modifica).

	Estat Inicial	Fallada: 32
Línia 0	0, 1, 2, 3, 4, 5, 6, 7	32, 33, 34, 35, 36, 37, 38, 39
Línia 1	8, 9, 10, 11, 12, 13, 14, 15	8, 9, 10, 11, 12, 13, 14, 15
Línia 2	16, 17, 18, 19, 20, 21, 22, 23	16, 17, 18, 19, 20, 21, 22, 23
Línia 3	24, 25, 26, 27, 28, 29, 30, 31	24, 25, 26, 27, 28, 29, 30, 31

	Fallada: 55	Fallada: 56
Línia 0	32, 33, 34, 35, 36, 37, 38, 39	32, 33, 34, 35, 36, 37, 38, 39
Línia 1	8, 9, 10, 11, 12, 13, 14, 15	8, 9, 10, 11, 12, 13, 14, 15
Línia 2	48, 49, 50, 51, 52, 53, 54, 55	48, 49, 50, 51, 52, 53, 54, 55
Línia 3	24, 25, 26, 27, 28, 29, 30, 31	56, 57, 58, 59, 60, 61, 62, 63

Examen 3

Assignatura	Codi	Data	Hora inici
Estructura de computadors			

	Fallada: 6	Fallada: 32
Línia 0	0, 1, 2, 3, 4, 5, 6, 7	32, 33, 34, 35, 36, 37, 38, 39
Línia 1	8, 9, 10, 11, 12, 13, 14, 15	8, 9, 10, 11, 12, 13, 14, 15
Línia 2	48, 49, 50, 51, 52, 53, 54, 55	48, 49, 50, 51, 52, 53, 54, 55
Línia 3	56, 57, 58, 59, 60, 61, 62, 63	56, 57, 58, 59, 60, 61, 62, 63

	Fallada: 40	Fallada:
Línia 0	32, 33, 34, 35, 36, 37, 38, 39	
Línia 1	40, 41, 42, 43, 44, 45, 46, 47	
Línia 2	48, 49, 50, 51, 52, 53, 54, 55	
Línia 3	56, 57, 58, 59, 60, 61, 62, 63	

	Fallada:	Fallada:
Línia 0		
Línia 1		
Línia 2		
Línia 3		

3.1.1.b) Quina és la taxa de fallades (T_f) ?

T_f = 6 fallades / 20 accessos = 0,3

3.1.1.c) Suposem que el temps d'accés a la memòria cau, o temps d'accés en cas d'encert (t_e), és de 2 ns i el temps total d'accés en cas de fallada (t_f) és de 20 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitja d'accés a memòria (t_m) ?

$$t_m = T_f \times t_f + (1\text{-}T_f) \times t_e = 0{,}3 * 20 \text{ ns} + 0{,}7 * 2 \text{ ns} = 6 \text{ ns} + 1{,}4 \text{ ns} = 7{,}4 \text{ ns}$$

3.1.2 Memòria Cau d'Accés Completament Associatiu

Ara suposem que el mateix sistema fa servir una política d'emplaçament completament associativa, de manera que qualsevol bloc de la memòria principal es pot portar a qualsevol bloc de la memòria cau.
Si trobem que la cau ja està plena, es fa servir un algorisme de reemplaçament LRU, de manera que traurem de la memòria cau aquell bloc que fa més temps que no es referència.

Considerem la mateixa llista de lectures a memòria:

14, 23, 24, 25, 26, 7, 32, 33, 34, 8, 55, 56, 6, 5, 12, 13, 14, 32, 40, 33

Examen 3

Assignatura	Codi	Data	Hora inici
Estructura de computadors			

3.1.2.a) 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.

	Estat Inicial	Fallada: 32
Línia 0	0, 1, 2, 3, 4, 5, 6, 7	0, 1, 2, 3, 4, 5, 6, 7
Línia 1	8, 9, 10, 11, 12, 13, 14, 15	32, 33, 34, 35, 36, 37, 38, 39
Línia 2	16, 17, 18, 19, 20, 21, 22, 23	16, 17, 18, 19, 20, 21, 22, 23
Línia 3	24, 25, 26, 27, 28, 29, 30, 31	24, 25, 26, 27, 28, 29, 30, 31

	Fallada: 8	Fallada: 55
Línia 0	0, 1, 2, 3, 4, 5, 6, 7	0, 1, 2, 3, 4, 5, 6, 7
Línia 1	32, 33, 34, 35, 36, 37, 38, 39	32, 33, 34, 35, 36, 37, 38, 39
Línia 2	8, 9, 10, 11, 12, 13, 14, 15	8, 9, 10, 11, 12, 13, 14, 15
Línia 3	24, 25, 26, 27, 28, 29, 30, 31	48, 49, 50, 51, 52, 53, 54, 55

	Fallada: 56	Fallada: 6
Línia 0	56, 57, 58, 59, 60, 61, 62, 63	56, 57, 58, 59, 60, 61, 62, 63
Línia 1	32, 33, 34, 35, 36, 37, 38, 39	0, 1, 2, 3, 4, 5, 6, 7
Línia 2	8, 9, 10, 11, 12, 13, 14, 15	8, 9, 10, 11, 12, 13, 14, 15
Línia 3	48, 49, 50, 51, 52, 53, 54, 55	48, 49, 50, 51, 52, 53, 54, 55

	Fallada: 32	Fallada: 40
Línia 0	56, 57, 58, 59, 60, 61, 62, 63	40, 41, 42, 43, 44, 45, 46, 47
Línia 1	0, 1, 2, 3, 4, 5, 6, 7	0, 1, 2, 3, 4, 5, 6, 7
Línia 2	8, 9, 10, 11, 12, 13, 14, 15	8, 9, 10, 11, 12, 13, 14, 15
Línia 3	32, 33, 34, 35, 36, 37, 38, 39	32, 33, 34, 35, 36, 37, 38, 39

3.1.2.b) Quina és la taxa de fallades (T_f) ?

T_f = 7 fallades / 20 accessos = 0,35

3.1.2.c) Suposem que el temps d'accés a la memòria cau, o temps d'accés en cas d'encert (t_e), és de 2 ns i el temps total d'accés en cas de fallada (t_f) és de 20 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitja d'accés a memòria (t_m) ?

$$t_m = T_f \times t_f + (1\text{-}T_f) \times t_e = 0{,}35 * 20 \text{ ns} + 0{,}65 * 2 \text{ ns} = 7 \text{ ns} + 1{,}3 \text{ ns} = 8{,}3 \text{ ns}$$

Examen 3

Assignatura	Codi	Data	Hora inici
Estructura de computadors			

3.2

Es vol realitzar la següent comunicació de dades entre la memòria d'un computador i un port USB, que tenen les següents característiques:

- La CPU funciona amb un rellotge de 1GHz de freqüència i executa 1 instrucció per cada 2 cicles de rellotge
- Adreces dels **registres de dades** i **d'estat** del controlador d'E/S: A00h i A04h
- El bit del **registre d'estat** que indica que el controlador del port d'E/S està disponible és el bit 3, o el quart bit menys significatiu (quan val 1 indica que està disponible)
- Transferència de **escriptura** des de memòria al port d'E/S
- Transferència de N_{dades}=100.000 dades, és a dir, 100.000 × 4 Bytes = 400.000 Bytes
- Adreça inicial de memòria on resideixen les dades: 20000000h
- La velocitat de transferència el port és de 200.000 Bytes per segon

3.2.1 E/S programada

Completar el següent codi realitzat amb el repertori CISCA que realitza la transferència descrita abans mitjançant la tècnica d'E/S programada.

```
1.              MOV   R3, 100000
2.              MOV   R2, 20000000h
3. Bucle:       IN    R0, _[A04h]_        ; llegir 4 bytes
4.              AND   R0, _00001000b_
5.              _JE__ Bucle
6.              MOV   R0,_[R2]_           ; llegir 4 bytes
7.              ADD   _R2_, 4
8.              OUT   [A00h], _R0_        ; escriure 4 bytes
9.              SUB   R3, __1__
10.             _JNE_ Bucle
```

Quin és el percentatge de temps que dedica la CPU a la tasca d'Entrada/Sortida?

100%

Examen 3

Assignatura	Codi	Data	Hora inici
Estructura de computadors			

3.2.2 E/S per Interrupcions

Completar el següent codi CISCA que és una rutina de servei a les interrupcions (RSI) per a transferir a través del dispositiu d'E/S anterior, el mateix nombre de dades que abans amb E/S programada, però ara mitjançant la tècnica de E/S per interrupcions. Suposeu:

- Es fa servir una variable global que es representa amb l'etiqueta **Dir,** i que al principi del programa conté l'adreça inicial de memòria on resideixen les dades a transferir

```
1.      _CLI__
2.      PUSH  _R0___
3.            _PUSH_ R1
4.      MOV   _R1_, [Dir]
5.            MOV   _R0_, [R1]          ; llegir 4 bytes
6.      OUT   _[A00h]_,R0         ; escriure 4 bytes
7.      ADD   R1, _4_
8.      MOV   _[Dir]__, R1
9.      POP   _R1__
10.     _POP_ R0
11.     STI
12.     IRET
```

Quin és el percentatge de temps que dedica la CPU a la tasca d'Entrada/Sortida?

400.000 Bytes a transferir. 200.000 Bytes per segon. Això fa que el temps total de la transferència sigui de 2 segons.

Cada cicle de rellotge és de 1ns. Per tant, cada instrucció triga 2 ns.

Una interrupció necessita 12 instruccions, per tant són 24 ns.

Hi ha 100.000 interrupcions, per tant són 2.400.000 ns. o 2,4 ms.

Això representa un 0,12% del temps total de la transferència.