

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadores			

05.573 21 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 2

Assignatura	Codi	Data	Hora inici
Estructura de computadores			

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 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors			

Pregunta 1

1.1

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Guardar un valor de tipus char 'val', rebut com a paràmetre, en el vector V1
; utilitzant 'pos', també rebut com a paràmetre, com a índex del vector.
; Per a accedir al vector V1 (de tipus char), s'ha d'utilitzar
; adreçament indexat: [Adreça+reg]
; Variables utilitzades: V1
; Paràmetres d'entrada :
; rdi: (dil) val: valor que volem guardar al vector
; rsi: (esi) pos: posició on hem de guardar el valor (índex del vector).
; Paràmetres de sortida: ; Cap:
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
setV1:
    push rbp
    mov  rbp, rsp

    ;V1[pos] = val;
    mov [V1+esi], dil

    mov rsp, rbp
    pop rbp
    ret

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Llegir un valor de tipus char del vector V1 utilitzant 'pos',
; rebut com a paràmetre, com a índex del vector i retornar aquest valor
; com a paràmetre de sortida.
; Per a accedir al vector V1 (de tipus char), s'ha d'utilitzar
; adreçament indexat: [Adreça+reg]
; Variables utilitzades: V1
; Paràmetres d'entrada : rdi: (edi) pos: posició d'on hem de llegir el valor (índex del
vector).
; Paràmetres de sortida: rax: (al) val: valor que volem retornar del vector
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
getV1:
    push rbp
    mov  rbp, rsp

    ;V1[pos]
    mov rax, 0
    mov al,[V1+edi]

    mov rsp, rbp
    pop rbp
    ret

```

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors			

1.2

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Si operem amb variables de tipus BYTE (size=1), treure la variable del
; cim de la pila stack1, fer l'operació NEG i guardar el resultat a la pila.
; Si no hi ha, com a mínim, 1 element a la pila no es podrà fer
; l'operació i s'ha d'indicar posant la variable state=6.
; A la operació NEG 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.
; IMPORTANT: Per accedir a la pila, no es pot fer directament, s'ha de fer
; cridant les subrutines pushStack1, pushStack4, popStack1 i popStack4.
;
; Variables utilitzades: size, sp1, sp4, Z, S, C i V
; Paràmetres d'entrada : Cap
; Paràmetres de sortida: Cap
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
opNEG3:
    ...
    cmp BYTE[size], 1 ; if (size == 1) {
    jne opNEG3_size4
opNEG3_size1:
    cmp DWORD[sp1], 0; if (sp1 > 0 ) {
    jle opNEG3_fewOp
    call popStack1 ; Llegim de la pila

    neg al

    jmp opNEG3_FLAG
opNEG3_size4:
    ...
opNEG3_FLAG: ;mirem els bits de resultat.
    mov BYTE[Z], '0'
    mov BYTE[S], '0'
    mov BYTE[C], '0'
    mov BYTE[V], '0'
opNEG3_FLAGZ:
    jnz opNEG3_FLAGS
    mov BYTE[Z], '1'
opNEG3_FLAGS:
    jns opNEG3_FLAGC
    mov BYTE[S], '1'
opNEG3_FLAGC:
    ...
opNEG3_save:
    ...
opNEG3_fewOp:
    ...
opNEG3_end:
    ...
    ret

```

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadores			

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	M(00000400h) = 00000410h M(00000800h) = 00000810h	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 ν val 400h i l'adreça simbòlica B val 800h

a)

```
ADD [V],100h
MOV R1,[V]
SUB R1,[B]
```

$[V] = [V] + 100h = (400h + 10h) + 100h = 510h$
 $R1 = 510h$
 $R1 = 510h - [B] = 510h - 810h = FFFFD00h$

$[V] = 510h$
 $R1 = FFFFD00h$
 $S=1, C=1, V=0, Z=0$

b)

```
SAL [B],3
NOT [B]
```

$[B] = 0810h \ll 3 = 4080h$
 $[B] = FFFFBF7Fh$

$[B] = FFFFBF7Fh$
 $S=0, C=0, V=0, Z=0$

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors			

2.2

Donat el següent llenguatge d'alt nivell:

```

Desde i=0 fins 9 fer
    Si (V [i] > 10) llavors V[i] = V[i]*2
    sino V[i] = 10;
fdesde;
```

On V es tracta d'un vector de 10 enters de 32 bits. Es proposa el següent codi CISCA en el que heu d'omplir els forats per què sigui operatiu:

```

MOV R1, 0
cont: CMP R1, 40          ; Has d'especificar el segon operand
      JGE final
      CMP [V+R1], 10
      JLE m              ; Has d'especificar l'etiqueta destí
      SAL [V+R1], 1      ; Has d'especificar el primer operand
      JMP p              ; Has d'especificar la instrucció
m:    MOV [V+R1], 10
p:    ADD R1, 4          ; Has d'especificar el segon operand
      JMP cont
final:
```

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors			

2.3

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

```

        JE GO
        ADD R2,R3
GO:     SUB [X+R7],R2

```

Traduïu-ho a llenguatge màquina i expresseu-ho en la següent taula. Supposeu que la primera instrucció del codi es troba a partir de l'adreça 0004B00h (que és el valor del PC abans de començar l'execució del fragment de codi). Supposeu també que l'adreça simbòlica X 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
20h	ADD
21h	SUB
41h	JE

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	
00004B00	JE GO	41	60	03	00								
00004B04	ADD R2,R3	20	12	13									
00004B07	SUB [X+R7],R2	21	57	80	1D	00	00	12					
00004B0E													

Examen 2

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:

1, 2, 3, 31, 23, 32, 33, 8, 7, 6, 55, 12, 34, 46, 12, 13, 14, 2, 3, 4

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: 7	Fallada: 55
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	8, 9, 10, 11, 12, 13, 14, 15
Línia 2	16, 17, 18, 19, 20, 21, 22, 23	48, 49, 50, 51, 52, 53, 54, 55
Línia 3	24, 25, 26, 27, 28, 29, 30, 31	24, 25, 26, 27, 28, 29, 30, 31

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors			

	Fallada: 34	Fallada: 46
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	40, 41, 42, 43, 44, 45, 46, 47
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	24, 25, 26, 27, 28, 29, 30, 31

	Fallada: 12	Fallada: 2
Línia 0	32, 33, 34, 35, 36, 37, 38, 39	0, 1, 2, 3, 4, 5, 6, 7
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	24, 25, 26, 27, 28, 29, 30, 31

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

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

$$T_f = 7 \text{ fallades} / 20 \text{ accessos} = 0,35$$

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) ?

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

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:

1, 2, 3, 31, 23, 32, 33, 8, 7, 6, 55, 12, 34, 46, 12, 13, 14, 2, 3, 4

Examen 2

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: 7
Línia 0	8, 9, 10, 11, 12, 13, 14, 15	8, 9, 10, 11, 12, 13, 14, 15
Línia 1	32, 33, 34, 35, 36, 37, 38, 39	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	0, 1, 2, 3, 4, 5, 6, 7

	Fallada: 55	Fallada: 46
Línia 0	8, 9, 10, 11, 12, 13, 14, 15	8, 9, 10, 11, 12, 13, 14, 15
Línia 1	32, 33, 34, 35, 36, 37, 38, 39	32, 33, 34, 35, 36, 37, 38, 39
Línia 2	48, 49, 50, 51, 52, 53, 54, 55	48, 49, 50, 51, 52, 53, 54, 55
Línia 3	0, 1, 2, 3, 4, 5, 6, 7	40, 41, 42, 43, 44, 45, 46, 47

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

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

$$T_f = 6 \text{ fallades} / 20 \text{ accessos} = 0,30$$

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) ?

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

Examen 2

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 2GHz de freqüència i executa 1 instrucció cada 2 cicles de rellotge
- Adreces dels **registres de dades i d'estat** del controlador d'E/S: 0400h i 0404h
- 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 d'**escriptura** des de memòria al port d'E/S
- Transferència de $N_{\text{dades}}=1.600.000$ dades, és a dir, $1.600.000 \times 4 \text{ Bytes} = 6.400.000$ Bytes
- Adreça inicial de memòria on resideixen les dades: 010000000h
- La velocitat de transferència del port és de 100.000 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, 1600000
2.      MOV    R2, 010000000h
3. Bucle: IN     R0, _[0404h]_      ; llegir 4 bytes
4.      AND    R0, _00001000b_
5.      _JE_   Bucle
6.      MOV    R0, _[R2]_          ; llegir 4 bytes
7.      OUT    _[0400h]_, R0       ; escriure 4 bytes
8.      ADD    _R2_, 4
9.      _DEC_  R3
10.     _JNE_  Bucle

```

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

100%

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors			

3.2.2 E/S per Interrupcions

Completar el següent codi CISC 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  __[0400h]__, 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?

6.400.000 Bytes a transferir. 100.000 Bytes per segon. Això fa que el temps total de la transferència sigui de 64 segons.

Cada cicle de rellotge és de 0,5ns. Per tant, cada instrucció triga 1 ns.

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

Hi ha 1.600.000 interrupcions, per tant són 19.200.000 ns. o 19,2 ms.

Això representa un **0,03%** del temps total de la transferència.