

Examen 1

| Assignatura | Codi | Data | Hora inici |
|----------------------------|------|------|------------|
| Estructura de computadores | | | |

05.573180614EX
05.573 18 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 1

| 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 1

| Assignatura | Codi | Data | Hora inici |
|---------------------------|------|------|------------|
| Estructura de computadors | | | |

Pregunta 1

1.1

```

////////////////////////////////////
; Posar un valor de tipus char al vector stack1.
; Com sp1 indica la primera posició lliure del vector stack1, primer
; posem el valor al vector utilitzant sp1 com index
; i després incrementem l'índex sp1.
; Per a accedir al vector stack1 (de tipus byte), s'ha d'utilitzar
; adreçament indexat: [Adreça+reg]
; Variables utilitzades: stack1, sp1
; Paràmetres d'entrada : rdi: (dil) val: valor que volem guardar al vector
; Paràmetres de sortida: Cap.
////////////////////////////////////
pushStack1:
    push rbp
    mov  rbp, rsp

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

    mov  rsp, rbp
    pop  rbp
    ret

```

Examen 1

| Assignatura | Codi | Data | Hora inici |
|---------------------------|------|------|------------|
| Estructura de computadors | | | |

1.2

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Convertir el valor d'una variable de tipus char (size=1), rebut com
; a paràmetre, en una cadena de caràcters (digits) que guardarem en el
; vector 'vecCharValue' segons la base (base) amb què estem treballant.
;
; ...
;
; Variables utilitzades:
; base, size, baseMaxDigits, vecCharValue
;
; Paràmetres d'entrada :
; rdi: (dil) valor que volem convertir.
;
; Paràmetres de sortida:
; Cap.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
charToVecChars:
    ...

    mov bl , dil                ; bl:val = dil:v
    cmp BYTE[base], 10        ; if (base==10 && v<0)
    jne charToVecChars_NObase10neg
    cmp dil, 0
    jge charToVecChars_NObase10neg
    neg bl                    ; val = -v;

charToVecChars_NObase10neg:
    ;for (i = 0; i < (baseMaxDigits*size); i++) {
    mov ah, 0

    mov al, [baseMaxDigits]

    mov dl, [size]

    mul dl ; (baseMaxDigits * size)

    mov r9d, 0
    mov r9w, ax                ;r9w: (baseMaxDigits*size)
    mov esi, 0                 ;esi: i

    ...

charToVecChars_end:
    ...

    ret

```

Examen 1

| 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 R5 = 00000028h R10 = 00000050h | M(00007E40h) = 00007E50h 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 `v` val 800h.

a)

```
MOV R5, FFFF0000h
ADD R5, [00007E40h]
NOT R5
```

```
R5= FFFF0000h
R5=FFFF0000h + 00007E50h =
FFFF7E50h
NOT R5 = 000081AFh
R5= 000081AFh
C=0, V=0, S=0, Z=0
```

b)

```
XOR [V], R10
SAR [V], 1h
JE F
S: SUB R1, R1
F:
```

```
[V]= 0810h XOR 50H = 840h
[800]= 420h
R1 = 0
[800]= 420h
R1= 0
Z = 1, C = 0, S = 0, V = 0
```

Examen 1

| Assignatura | Codi | Data | Hora inici |
|---------------------------|------|------|------------|
| Estructura de computadors | | | |

2.2

En la memòria d'un computador CISCA tenim emmagatzemades dos vectors A i B de 10 i 20 elements respectivament. Cada element és un nombre enter codificat en complement a 2 amb 32 bits.

Completeu els buits del fragment de codi CISCA realitzar l'equivalent a la següent sentència en C:

$$A[i] = A[i] + B[j]$$

Els vectors estan emmagatzemats en posicions consecutives de memòria, com és habitual quan es tradueix codi en C. Per exemple, els elements A[0], A[1], A[2] i M[7] es troben emmagatzemats en les adreces de memòria A, A+4, A+8 i A+28 respectivament. El mateix per al vector B.

Se sap que en R1 es troba emmagatzemat el valor de la variable "i", i en R2 el de la "j" i que després d'executar-se el fragment de codi tots els registres han de mantenir els valors originals.

```
PUSH R2
PUSH R1
MUL R1, 4           ; Heu d'especificar la instrucció
MUL R2, 4           ; Heu d'especificar el primer operand
MOV R3, [B+R2]      ; Heu d'especificar el segon operand
ADD [A+R1], R3
POP R1              ; Heu d'especificar l'operand
POP R2              ; Heu d'especificar l'operand
```

2.3

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

```
SUB [100h], ABh
ADD R1, [R2]
XOR R1, R2
```

Traduiu-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 00FF0010h (que és el valor del PC abans de començar l'execució del fragment de codi). 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.

Examen 1

| Assignatura | Codi | Data | Hora inici |
|---------------------------|------|------|------------|
| Estructura de computadors | | | |

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

Taula de codis d'instrucció

| B0 | Instrucció |
|-----|------------|
| 20h | ADD |
| 21h | SUB |
| 32h | XOR |

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 | SUB [100h], ABh | 21 | 20 | 00 | 01 | 00 | 00 | 00 | AB | 00 | 00 | 00 | |
| 00FF001B | ADD R1, [R2] | 20 | 11 | 32 | | | | | | | | | |
| 00FF001E | XOR R1, R2 | 32 | 11 | 12 | | | | | | | | | |
| 00FF0021 | | | | | | | | | | | | | |

Examen 1

| 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:

0, 1, 2, 23, 24, 12, 4, 32, 33, 5, 6, 54, 55, 7, 8, 9, 12, 22, 34, 1

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: 5 | Fallada: 54 |
|---------|--------------------------------|--------------------------------|
| 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 1

| Assignatura | Codi | Data | Hora inici |
|---------------------------|------|------|------------|
| Estructura de computadors | | | |

| | | |
|---------|--------------------------------|--------------------------------|
| | Fallada: 22 | Fallada: 34 |
| 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: 1 | Fallada: |
| Línia 0 | 0, 1, 2, 3, 4, 5, 6, 7 | |
| Línia 1 | 8, 9, 10, 11, 12, 13, 14, 15 | |
| Línia 2 | 16, 17, 18, 19, 20, 21, 22, 23 | |
| Línia 3 | 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) ?

$$T_f = 6 \text{ fallades} / 20 \text{ accessos} = 0,3$$

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) ?

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

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:

0, 1, 2, 23, 24, 12, 4, 32, 33, 5, 6, 54, 55, 7, 8, 9, 12, 22, 34, 1

Examen 1

| Assignatura | Codi | Data | Hora inici |
|----------------------------|------|------|------------|
| Estructura de computadores | | | |

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 | 8, 9, 10, 11, 12, 13, 14, 15 |
| Línia 2 | 16, 17, 18, 19, 20, 21, 22, 23 | 32, 33, 34, 35, 36, 37, 38, 39 |
| Línia 3 | 24, 25, 26, 27, 28, 29, 30, 31 | 24, 25, 26, 27, 28, 29, 30, 31 |

| | Fallada: 54 | Fallada: 22 |
|---------|--------------------------------|--------------------------------|
| 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 | 32, 33, 34, 35, 36, 37, 38, 39 | 16, 17, 18, 19, 20, 21, 22, 23 |
| Línia 3 | 48, 49, 50, 51, 52, 53, 54, 55 | 48, 49, 50, 51, 52, 53, 54, 55 |

| | Fallada: 34 | Fallada: |
|---------|--------------------------------|----------|
| Línia 0 | 0, 1, 2, 3, 4, 5, 6, 7 | |
| Línia 1 | 8, 9, 10, 11, 12, 13, 14, 15 | |
| Línia 2 | 16, 17, 18, 19, 20, 21, 22, 23 | |
| Línia 3 | 32, 33, 34, 35, 36, 37, 38, 39 | |

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

$$T_f = 4 \text{ fallades} / 20 \text{ accessos} = 0,2$$

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) ?

$$t_m = T_f \times t_f + (1-T_f) \times t_e = 0,2 * 20 \text{ ns} + 0,8 * 2 \text{ ns} = 4 \text{ ns} + 1,6 \text{ ns} = 5,6 \text{ ns}$$

Examen 1

| 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 cicle de rellotge.
- Adreces dels **registres de dades** i **d'estat** del controlador d'E/S: 0200h i 0204h respectivament.
- El bit del **registre d'estat** que indica que el controlador del port d'E/S està disponible és el bit 5, o el sisé bit menys significatiu (quan val 1 indica que està disponible).
- Transferència d'**escriptura** des de memòria al port d'E/S.
- Transferència de $N_{dades}=400.000$ dades, és a dir, 400.000×4 Bytes = 1.600.000 Bytes
- Adreça inicial de memòria on resideixen les dades: 40000000h
- La velocitat de transferència el port és de 5.000 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, __400000__
2.      MOV   R2, 40000000h
3. Bucle: IN    R0, [0204h]          ; llegir 4 bytes
4.      _AND_ R0, 00100000b
5.      _JE_   Bucle
6.      MOV   R0, _[R2]_            ; llegir 4 bytes
7.      _OUT_ [0200h], R0          ; escriure 4 bytes
8.      ADD   R2, __4__
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 1

| 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        _[0200h]_, 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?

1.600.000 Bytes a transferir. 5.000 Bytes per segon. Això fa que el temps total de la transferència sigui de 320 segons.

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

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

Hi ha 400.000 interrupcions, per tant són 4.800.000 ns. o 4,8 ms.

Això representa un 0,0015% del temps total de la transferència.