

## Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	27/06/2015	15:30

05.573 27 06 15 EX  
05.573 27 06 15 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 (35%); Pregunta 3 (35%); Pregunta 4 (10%)
- 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 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	27/06/2015	15:30

### 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 (35%)

**2.1: 10%**

**2.2: 15%**

**2.3: 10%**

#### Pregunta 3 (35%)

**3.1: 15%**

**3.1.1: 10%**

**3.1.2: 5%**

**3.2: 20%**

**3.2.1: 10%**

**3.2.2: 5%**

**3.2.3: 5%**

#### Pregunta 4 (10%)

**4.1: 5%**

**4.2: 5%**

## Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	27/06/2015	15:30

### Pregunta 1

#### 1.1 Pràctica – Part obligatòria

Escriure un fragment de codi assemblador de la subrutina `calcIndexPl` que calculi l'índex per a accedir a la matriu `marks` i mines i el guardi a la variable `indexMat`. (No s'ha d'escriure el codi de tota la subrutina).

```

;;;;
; Calcular el valor de l'índex per a accedir a la matriu marks i mines
; (9x9) a partir dels valors de row i col (la posició actual del cursor).
; indexMat=(row*DimMatrix)+(col).
; Aquesta subrutina no té una funció en C equivalent.
;
; Variables utilitzades:
; row      : fila per accedir a les matrius mines i marks [0..(DimMatrix-1)]
; col      : columna per accedir a les matrius mines i marks [0..(DimMatrix-1)]
; indexMat: index per a accedir a les matrius mines i marks.
; Paràmetres d'entrada : Cap
; Paràmetres de sortida: Cap
;;;;
calcIndexPl:
    push rbp
    mov  rbp, rsp

    push rax
    push rbx
    push rdx

    mov rax, 0
    mov rbx, 0
    mov rdx, 0
    mov eax, [row]
    mov ebx, DimMatrix
    mul ebx          ;multipliquem per 9 (EDX:EAX = EAX * font)
    add eax, [col]
    mov [indexMat], eax
    ;indexMat=([row]*DimMatrix)+([col])

calcIndexPl_End:
    pop rdx
    pop rbx
    pop rax

    mov rsp, rbp
    pop rbp
    ret

```

## Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	27/06/2015	15:30

### 1.2 Pràctica – Part opcional

Completar el codi de la subrutina `checkEnd`, per a verificar si hem marcat totes les mines i hem obert totes les altres caselles

```

;;;;
; Verifica si hem marcat totes les mines (numMines=0) i hem obert
; totes les altres caselles (no hi ha cap posició de la matriu marks
; que tingui un espai en blanc), si és així, canviar l'estat a 2 (Guanya).
;
; Variables utilitzades:
; numMines: nombre de mines que queden per posar.
; state : Indica l'estat del joc. 0:sortir, 1:jugar, 2:guanya, 3:explosió
; Paràmetres d'entrada : Cap
; Paràmetres de sortida: Cap
;;;;
checkEndP2:
    push rbp
    mov rbp, rsp
    push rsi

    mov rsi, __0__ ;índex per accedir a la matriu marks

    ;Mirem si hem marcat totes les mines.
    cmp __DWORD[numMines]__, 0
    jg checkEndP2_End

    ;Iniciem el bucle pera mirar si hi ha espais en blanc
checkEndP2_Loop:

    cmp __BYTE[marks+esi]__, ' '
    je checkEndP2_End ;Si és un espai en blanc no hem acabat.

    inc __esi__

    cmp esi, __SizeMatrix__ ;DimMatrix*DimMatrix
    jl checkEndP2_Loop
    mov __DWORD[state]__, 2
    ;si hem mirat totes les posicions i no hi ha
    ;cap espai, vol dir que hem marcat totes les
    ;mines i obert tota la resta de posicions.

checkEndP2_End:
    pop rsi
    mov rsp, rbp
    pop rbp
    ret

```

## Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	27/06/2015	15:30

### Pregunta 2

#### 2.1

El estado inicial del computador CISCA justo antes de comenzar la ejecución de cada fragmento de código (de cada apartado) es el siguiente:

R0 = 00000A10h R1 = 00000B20h R2 = 00000C30h	M(00000A10h) = 0000F00Fh M(00000B20h) = 0000F000h M(00000C30h) = 00000FF0h M(0000F00Ah) = 00000000h M(000FF0A0h) = 0000000Ah	Z = 0, C = 0, S = 0, V = 0
--	--	----------------------------

L'adreça simbòlica W val 000FF0A0h. Quin serà l'estat del computador després d'executar cada fragment de codi? (només modificacions, excloent-hi el PC).

a)
<pre> MOV R0, 0 PLUS: CMP R0, R2       JG  END       DEC R2, 1       JMP PLUS END: </pre>
<p>R0:= 0 R2:= FFFFFFFFh</p> <p>Z = 0 , S = 1 , C = 1 , V = 0</p>

b)
<pre> MOV R0, [W] ADD [R1], R0 MOV R1, [R1] </pre>
<p>R0:= 0000000Ah [00000B20]:= 0000F00Ah R1:= 0000F00Ah</p> <p>Z = 0 , S = 0 , C = 0 , V = 0</p>

## Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	27/06/2015	15:30

### 2.2

Donat el següent codi en alt nivell:

```
i = 0;
while (i < MAX) { A[i] = 0; i = i + 1; };
```

A és un vector de 8 elements de 4 bytes cadascun. Es proposa la següent traducció a CISCA on hem deixat 6 llocs per omplir:

```

MOV R0, 0
PLUS:  CMP R0, MAX
      JE END    ○ JGE
MOV R1, R0
MUL R1, 4
MOV [A+R1], 0
INC R0
JMP PLUS
END:   MOV [I], 0
```

## Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	27/06/2015	15:30

### 2.3

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

```

CMP R0, 0000688Ah
JLE END
MOV R0, [R3]
END:

```

Traduïu-lo a llenguatge màquina i expresseu-lo en la següent taula. Supposeu que la primera instrucció del codi s'assembla a partir de l'adreça 0000600h (que és el valor del PC abans de començar l'execució del fragment de codi). En la següent taula useu una fila per a 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-lo 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 ajuda les taules de codis:

Taula de codis d'instrucció

B0	Instrucció
44h	JLE
26h	CMP
10h	MOV

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	
0000600h	CMP R0, 0000688Ah	26	10	00	8A	68	00	00					
0000607h	JLE END	44	60	03	00								
000060Bh	MOV R0, [R3]	10	10	33									
000060Eh													

## Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	27/06/2015	15:30

### Pregunta 3

#### 3.1. Memòria cau

##### Memòria cau completament associativa (LRU)

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 en una adreça múltiple de 8. Així, el bloc 0 conté les adreces 0, 1, 2, 3, 4, 5, 6, 7, el bloc 1, les adreces 8, 9, 10, 11, 12, 13, 14, 15, i el bloc N les adreces  $8*N$ ,  $8*N+1$ ,  $8*N+2$ ,  $8*N+3$ ,  $8*N+4$ ,  $8*N+5$ ,  $8*N+6$ ,  $8*N+7$ .

Suposem que el sistema també disposa d'una memòria cau de 4 línies (on cada línia té la mida d'un bloc, es 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, 7).

Suposem que el 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**.

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

7, 8, 24, 5, 23, 54, 55, 56, 17, 18, 32, 40, 4, 6, 63, 40, 48, 56, 42, 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.

Si és un encert escriurem E en la línia corresponent davant de les adreces del bloc, si és una fallada escriurem F i s'indicarà el nou bloc que es porta a la memòria cau en la línia que li correspongui, expressat de la forma b ( $a_0 - a_7$ ) on b: número de bloc, i ( $a_0 - a_7$ ) són les adreces del bloc, on  $a_0$  és la primera adreça del bloc i  $a_7$  és la vuitena (darrera) adreça del bloc.

Línia	Estat Inicial	7	8	24	5	23
0	0 (0 - 7)	E 0 (0 - 7)	0 (0 - 7)	0 (0 - 7)	E 0 (0 - 7)	0 (0 - 7)
1	1 (8 - 15)	1 (8 - 15)	E 1 (8 - 15)	1 (8 - 15)	1 (8 - 15)	1 (8 - 15)
2	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	E 2 (16 - 23)
3	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	E 3 (24 - 31)	3 (24 - 31)	3 (24 - 31)

Línia	54	55	56	17	18	32
0	0 (0 - 7)	0 (0 - 7)	0 (0 - 7)	0 (0 - 7)	0 (0 - 7)	F 4 (32 - 39)
1	F 6 (48 - 55)	E 6 (48 - 55)	6 (48 - 55)	6 (48 - 55)	6 (48 - 55)	6 (48 - 55)
2	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	E 2 (16 - 23)	E 2 (16 - 23)	2 (16 - 23)
3	3 (24 - 31)	3 (24 - 31)	F 7 (56 - 63)	7 (56 - 63)	7 (56 - 63)	7 (56 - 63)



## Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	27/06/2015	15:30

Línia	40	4	6	63	40	48
0	4 (32 - 39)	4 (32 - 39)	4 (32 - 39)	4 (32 - 39)	4 (32 - 39)	F 6 (48 - 55)
1	F 5 (40 - 47)	5 (40 - 47)	5 (40 - 47)	5 (40 - 47)	E 5 (40 - 47)	5 (40 - 47)
2	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	F 7 (56 - 63)	7 (56 - 63)	7 (56 - 63)
3	7 (56 - 63)	F 0 (0 - 7)	E 0 (0 - 7)	0 (0 - 7)	0 (0 - 7)	0 (0 - 7)

Línia	56	42	50			
0	6 (48 - 55)	6 (48 - 55)	E 6 (48 - 55)			
1	5 (40 - 47)	E 5 (40 - 47)	E 5 (40 - 47)			
2	E 7 (56 - 63)	7 (56 - 63)	7 (56 - 63)			
3	0 (0 - 7)	0 (0 - 7)	0 (0 - 7)			

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

$$T_e = 13 \text{ encerts} / 20 \text{ accessos} = 0,65$$

**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 40 ns. Considerant la taxa d'encerts obtinguda a la pregunta anterior, quin és el temps mitja d'accés a memòria ( $t_m$ ) ?

$$t_m = T_e \times t_e + (1 - T_e) \times t_f = 0,65 \times 5 \text{ ns} + 0,35 \times 40 \text{ ns} = 3,25 \text{ ns} + 14 \text{ ns} = 17,25 \text{ ns}$$

## Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	27/06/2015	15:30

### 3.2 Sistema d'E/S

#### 3.2.1 E/S programada

Es vol analitzar el rendiment de la comunicació de dades entre la memòria d'un processador i un port USB, que tenen les següents característiques:

- Velocitat de transferència del dispositiu d'E/S ( $v_{\text{transf}}$ ) = 500 KB/s
- Temps de latència mitjà del dispositiu ( $t_{\text{latència}}$ ) = 0
- Adreces dels **registres de dades i d'estat** del controlador d'E/S: 0E00h i 0E04h
- El bit del **registre d'estat** que indica que el controlador del port d'E/S està disponible és el bit 5, o el sisè bit menys significatiu (quan val 1 indica que està disponible)
- Processador amb una freqüència de rellotge de 2 GHz, el processador pot executar 2 instruccions per cicle de rellotge ( $t_{\text{instr}} = t_{\text{cicle}} / 2$ )
- Temps de programació i finalització de la transferència de 200 ns ( $t_{\text{prog}} + t_{\text{final}}$ )
- Transferència de **escriptura** des de memòria al port d'E/S
- Transferència de  $N_{\text{dades}}=400.000$  dades, es a dir,  $400.000 * 4$  Bytes = 1.600.000 Bytes
- Adreça inicial de memòria on resideixen les dades: 50000000h

Completeu el següent codi realitzat amb el repertori CISCA per a que realitzi la transferència descrita abans mitjançant la tècnica d'E/S programada.

```

1.      MOV      R3, _400000_
2.      MOV      R2, 50000000h
3. Bucle:  IN      R0, [E04]      ; llegir 4 bytes
4.      _AND_     R0, 00100000b
5.      _JE_      Bucle
6.      MOV      R0, _[R2]_      ; llegir 4 bytes
7.      ADD      R2, 4
8.      OUT _[E00h]_, R0      ; escriure 4 bytes
9.      SUB R3, 1
10.     JNE Bucle

```

## Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	27/06/2015	15:30

### 3.2.2 E/S per DMA

Suposarem que el controlador d'ES pot funcionar en mode DMA (Accés Directe a Memòria). La suma del **temps de cessió** del bus i del **temps de recuperació** del bus es de 10 ns (  $t_{\text{cessió}} + t_{\text{recup}} = 10 \text{ ns}$ ). El **temps de la transferència** pel bus es de 1 ns (  $t_{\text{mem}} = 1 \text{ ns}$ ).

Considerem que en la transferència per DMA, les dades s'envien entre la memòria i el controlador de DMA, en mode ràfega, i es disposa d'un buffer de mida  $m_{\text{buffer}} = 400$  bytes. Calcular el temps total d'ocupació del bus per part del controlador de DMA per a dur a terme la transferència que venim analitzant.

Mida de les ràfegues  $N_{\text{ràfega}} : m_{\text{buffer}} / m_{\text{dada}} = 400 / 4 = 100$

Temps ocupació Bus,  $t_{\text{transf_ràfega}} : t_{\text{cessió}} + 100 * t_{\text{mem}} + t_{\text{recup}} = 10 + 100 * 1 = 110 \text{ ns}$

Nombre de peticions del Bus,  $N_{\text{dades}} / N_{\text{ràfega}} : 400.000 / 100 = 4000$

Temps total d'ocupació del Bus  $t_{\text{transf_bloc}} : t_{\text{transf_dada}} * (N_{\text{dades}} / N_{\text{ràfega}}) = 110 * 4000 = 440000 \text{ ns} = 440 \text{ us}$   
= 0,44 ms

### 3.2.3 E/S per DMA

La CPU no pot fer cap tasca durant tot el temps en que el bus està ocupat per part del controlador de DMA. Quin percentatge de temps té disponible la CPU per a executar codi efectiu d'altres programes durant la transferència?

$t_{\text{transf_bloc}} = 0,44 \text{ ms}$

$t_{\text{bloc}} = 3200 \text{ ms}$

$\% \text{ocupació} = (t_{\text{transf_bloc}} * 100) / t_{\text{bloc}}$

Percentatge de temps disponible:  $100 - \% \text{ocupació} = 100 - (t_{\text{transf_bloc}} * 100) / t_{\text{bloc}} = 100 - (0,44 * 100) / 3200 = 100 - 0,01375 = 99,98625\%$

## Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	27/06/2015	15:30

### Pregunta 4

#### 4.1

El cicle d'execució complet d'una instrucció es divideix en diferents etapes: lectura de la instrucció, lectura dels operands font, execució...

Es aquestes fases, el processador utilitza diferents registres interns per realitzar la seva feina. Quina és la funció del registre MAR?

El registre MAR és el Memory Address Register, és a dir, el registre d'adreces de memòria. Conté l'adreça de memòria que es llegirà o s'escriurà, utilitzant el registre MBR per gestionar el contingut llegit o escrit.

#### 4.2

a) Quines són les tres polítiques d'assignació per a emmagatzemar dades dins d'una memòria cau? En que consisteixen?

1) Política d'assignació directa: un bloc de la memòria principal només potser en una única línia de la memòria cau. La memòria cau d'assignació directa és la que té la taxa de fallades més alta, però s'utilitza molt perquè és la més barata i fàcil de gestionar.

2) Política d'assignació completament associativa: un bloc de la memòria principal pot ser en qualsevol línia de la memòria cau. La memòria cau completament associativa és la que té la taxa de fallades més baixa. No obstant això, no se sol utilitzar perquè és la més cara i complexa de gestionar.

3) Política d'assignació associativa per conjunts: un bloc de la memòria principal pot ser en un subconjunt de les línies de la memòria cau, però dins del subconjunt pot trobar-se en qualsevol posició. La memòria cau associativa per conjunts és una combinació

b) En un sistema d'E/S gestionat per interrupcions. Quina funció tenen les senyals INT i INTA? Qui les activa i perquè?

En el moment en què el mòdul d'E/S demana l'atenció del processador (el mòdul activa la INT), es produeix una seqüència d'esdeveniments que el processador ha de gestionar per a atendre aquesta petició del mòdul d'E/S, garantint que després podrà tornar el control al programa al qual aturem l'execució per atendre la petició del mòdul d'E/S.

Anàlogament a la manera com es fa en la inhibició de les interrupcions, es pot avisar el mòdul d'E/S de dues maneres: la més habitual és utilitzant un maquinari específic que inclou una línia de reconeixement d'interrupció (INTA o INTACK, de l'anglès interrupt acknowledge) que activa el processador, igual que la INT és un senyal actiu a la baixa, o deixar que sigui responsabilitat de la mateixa RSI, que ha d'executar una o diverses instruccions per a accedir als registres del mòdul d'E/S i d'aquesta manera indicar-li que ja l'atén.