

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	20/06/2015	12:00

05.573R20R06R15REEΔ€
05.573 20 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	20/06/2015	12:00

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 plantejareu 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: 10%

Pregunta 4 (10%)

4.1: 5%

4.2: 5%

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	20/06/2015	12:00

Pregunta 1

1.1 Pràctica – Part obligatòria

Escriure un fragment de codi assemblador de la subrutina `showMinesP1` que obté el valor de les unitats i de les desenes de la variable `numMines` i els deixa als registres `al` i `dl` respectivament. (No s'ha d'escriure el codi de tota la subrutina).

```

; ; ; ;
; Converteix el valor de NumMines (entre 0 i 99) en dos caràcters ASCII.
; S'ha de dividir el valor entre 10, el quocient representarà les
; desenes i el residu les unitats, i després s'han de convertir
; a ASCII sumant 48.
; Mostra els dígit (caràcter ASCII) de les desenes a la fila 22,
; columna 10 de la pantalla i les unitats a la fila 22, columna 12,
; (la posició s'indica a través de les variables rowCur i colCur).
; Per a posicionar el cursor cridar a la subrutina gotoxyP1 i per a mostrar
; els caràcters a la subrutina printchP1.
; Variables utilitzades:
; numMines: nombre de mines que queden per posar
; charac   : caràcter que volem mostrar
; rowCur  : fila per a posicionar el cursor a la pantalla
; colCur  : columna per a posicionar el cursor a la pantalla
; Paràmetres d'entrada : Cap
; Paràmetres de sortida: Cap
; ; ; ;
showMinesP1:
    push rbp
    mov  rbp, rsp
    push rax
    push rbx
    push rdx
    mov  rax, 0
    mov  eax, [numMines]
    mov  edx, 0

    ; calcular unitats i desenes
    mov  ebx, 10
    div  ebx          ; EAX=EDX:EAX/EBX EDX=EDX:EAX mod EBX

    add  al, '0'      ; convertim unitats a caràcters ASCII
    add  dl, '0'      ; convertim desenes a caràcters ASCII

    ; Posicionar el cursor i mostrar dígit
    ; AQUESTA PART DEL CODI NO S'HA D'IMPLEMENTAR

showMinesP1_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	20/06/2015	12:00

1.2 Pràctica – Part opcional

Completar el codi de la subrutina `updateBoardP2`, per a que mostri el contingut de la matriu `marks`.

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Aquesta subrutina es dóna feta. NO LA PODEU MODIFICAR.
; Mostrar un caràcter, rebut com a paràmetre al registre dil,
; en la pantalla en la posició on està el cursor,
; cridant a la funció printch_C.
; Variables utilitzades: Cap
; Paràmetres d'entrada : rdi (dil): Caràcter que volem mostrar a la pantalla
; Paràmetres de sortida: Cap
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
printchP2:

; ; ; ;
; Actualitza el contingut del Tauler de Joc amb les dades de la matriu
; marks i el nombre de mines que queden per marcar.
; S'ha de recorre tot la matriu marks, i per a cada element de la matriu
; posicionar el cursor a la pantalla i mostrar els caràcter de la matriu,
; després mostra el valor de NumMines a la part inferior del tauler
; Per posicionar el cursor cridar a la subrutina gotoxyP2, per mostrar
; els caràcters a la subrutina printchP2 i per mostrar numMines a la
; subrutina ShowMinesP2 implementant correctament el pas de paràmetres.
; Variables utilitzades: marks : matriu 9x9 on hi han les mines marcades
; i el nombre de mines de les caselles obertes.
; Paràmetres d'entrada : Cap
; Paràmetres de sortida: Cap
; ; ; ;
updateBoardP2:
    ...
    mov edi, RowScreenIni+3 ;fila inicial del cursor
    mov esi, ColScreenIni+4 ;columna inicial del cursor
    mov ebx, 0 ;fila (0-8)
    mov ecx, 0 ;columna (0-8)
    mov edx, 0 ;índex per a accedir a la matriu marks. (0-80)
    ;índex=(fila*DimMatrix)+(columna)

    ;Iniciem el bucle per a mostrar la matriu.
updateBoardP2_bucle:
    call gotoxyP2 ;posiciona el cursor
    __push__ rdi
    ;caràcter de la matriu que volem mostrar
    mov dil, __[marks+edx]__
    call __printchP2__ ;mostrar caràcter
    pop rdi
    inc __edx__ ;incrementem l'índex per accedir a la matriu
    add esi, 4 ;Actualitzem la columna del cursor
    inc ecx ;Actualitzem la columna.
    cmp ecx, __DimMatrix__
    jl updateBoardP2_bucle
    mov ecx, 0 ;columna inicial
    mov esi, ColScreenIni+4 ;columna inicial del cursor
    add edi, 2 ;Actualitzem la fila del cursor
    inc __ebx__ ;Actualitzem la fila.
    cmp ebx, DimMatrix
    jl updateBoardP2_bucle
    call showMinesP2 ;Actualitzem el valor de numMines a pantalla
updateBoardP2_end:
    ...
    ret

```

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	20/06/2015	12:00

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:

R0 = 00000A10h R1 = 00000B20h R2 = 00000C30h	M(00000A10h) = 0000F00Fh M(00000B20h) = 0000F000h M(00000C30h) = 00000FF0h M(000000F0h) = 00000001h M(000FF0A0h) = 0000000Ah	Z = 0, C = 0, S = 0, V = 0
--	--	----------------------------

Quin serà l'estat del computador després d'executar cada fragment de codi? (només modificacions, excloent-hi el PC).

a)	XOR R0, R0 JE END ADD [R0+B20h], R2 END:
R0 := 0	
Z = 1 , S = 0 , C = 0 , V = 0	

b)	DEC [000000A10h] ADD [000000A10h], R2
[00000A10] = 0000F00Eh [00000A10] = 0F00Eh + C30h = 0000FC3Eh	
Z = 0 , S = 0 , C = 0 , V = 0	

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	20/06/2015	12:00

2.2

Donat el següent codi en alt nivell:

Si $(A[j] > A[i])$ $A[i] = A[i] + A[j]$;

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, [j]
MUL R0, 4
MOV R2, [A+R0]
MOV R1, [i]
MUL R1, 4
CMP R2, [A+R1]
JLE END
ADD [A+R1], R2

```

END :

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	20/06/2015	12:00

2.3

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

```
MOV R1, R0
MUL R1, 4
ADD [A+R1], 10
```

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 00006880h (que és el valor del PC abans de començar l'execució del fragment de codi). Supposeu que l'adreça simbòlica A val 00004000h. 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
10h	MOV
22h	MUL
20h	ADD

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	
00006880h	MOV R1, R0	10	11	10									
00006883h	MUL R1, 4	22	11	00	04	00	00	00					
0000688Ah	ADD [A+R1], 10	20	51	00	40	00	00	00	0A	00	00	00	
00006895h													

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	20/06/2015	12:00

Pregunta 3

3.1. Memòria cau

Memòria cau completament associativa (FIFO)

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 FIFO**.

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

Inicialment la memòria cau és buida i s'omple seqüencialment començant per la línia 0.

3.1.1 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	-	F 0 (0 - 7)	0 (0 - 7)	0 (0 - 7)	E 0 (0 - 7)	0 (0 - 7)
1	-	-	F 1 (8 - 15)	1 (8 - 15)	1 (8 - 15)	1 (8 - 15)
2	-	-	-	F 3 (24 - 31)	3 (24 - 31)	3 (24 - 31)
3	-	-	-	-	-	F 2 (16 - 23)

Línia	54	55	56	17	18	32
0	F 6 (48 - 55)	E 6 (48 - 55)	6 (48 - 55)	6 (48 - 55)	6 (48 - 55)	6 (48 - 55)
1	1 (8 - 15)	1 (8 - 15)	F 7 (56 - 63)	7 (56 - 63)	7 (56 - 63)	7 (56 - 63)
2	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	F 4 (32 - 39)
3	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	E 2 (16 - 23)	E 2 (16 - 23)	2 (16 - 23)

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	20/06/2015	12:00

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

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

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

$$T_e = 9 \text{ encerts} / 20 \text{ accessos} = 0,45$$

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 2 ns i el temps total d'accés en cas de fallada (t_f) és de 30 ns. Considerant la taxa d'encerts obtinguda a la pregunta anterior, quin és el temps mitjà d'accés a memòria (t_m) ?

$$t_m = T_e \times t_e + (1 - T_e) \times t_f = 0,45 \times 2 \text{ ns} + 0,55 \times 30 \text{ ns} = 0,9 \text{ ns} + 16,5 \text{ ns} = 17,4 \text{ ns}$$

3.2.1 E/S per interrupcions

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

- Velocitat de transferència del dispositiu d'E/S $v_{\text{transf}} = 1 \text{ MBytes/s} = 1000 \text{ Kbytes/s}$
- Temps de latència mitjà del dispositiu $t_{\text{latència}} = 0$
- Adreces dels **registres d'estat** i **dades** del controlador d'E/S: 0C00h i 0C04h
- El bit del **registre d'estat** que indica que el controlador del port d'E/S està disponible és el bit 4, o 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 pot executar 2 instruccions per cicle de rellotge
- Transferència de **lectura** des de memòria al port d'E/S
- Transferència de **$N_{\text{dades}} = 200000$** dades
- La mida d'una dada és **$m_{\text{dada}} = 4$** bytes
- Adreça inicial de memòria on resideixen les dades: A0000000h

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	20/06/2015	12:00

- El temps per atendre la interrupció (t_{rec_int}) és de 2 cicles de rellotge

a) Completeu 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, mitjançant la tècnica de E/S per interrupcions.

Es fa servir una variable global que es representa amb l'etiqueta **Addr**, i que al principi del programa conté l'adreça inicial de memòria on emmagatzemar les dades rebudes.

```

1.  CLI
2.  PUSH R0
3.  PUSH R1
4.  IN  R0,  _[0C04h]_
5.  MOV  R1,  [Addr]
6.  MOV  _[R1]_, R0
7.  ADD  R1,  _4_
8.  MOV  _[Addr]_, R1
9.  POP  R1
10. POP  R0
11. STI
12. _IRET_

```

b) Quant temps dedica la CPU a la transferència del bloc de dades t_{transf_bloc} ?

El temps d'un cicle, $t_{cicle} = 0,5$ ns (nanosegons)

Temps per atendre la interrupció, t_{rec_int} : 2 cicles * 0,5 ns = 1 ns

Temps d'execució de una instrucció, t_{instr} : $t_{cicle} / 2 = 0,250$ ns

Temps d'execució RSI, t_{rsi} : $N_{rsi} * t_{instr} = 12 \text{ instr.} * 0,250 \text{ ns} = 3 \text{ ns}$

Temps consumit per CPU en cada interrupció, t_{transf_dada} :

$$t_{transf_dada} = t_{rec_int} + t_{rsi} = 1 + 3 = 4 \text{ ns}$$

Nombre d'interrupcions produïdes (nombre total de dades, N_{dades}): 200000 interrupcions

Temps consumit en total en TOTES les interrupcions:

$$t_{transf_bloc} = t_{transf_dada} * N_{dades} = 4 \text{ ns} * 200000 \text{ interrupcions} = 800000 \text{ ns} = 0,8 \text{ ms (milisegons)}$$

c) Quin és el percentatge d'ocupació del processador? Percentatge que representa el temps de transferència del bloc t_{transf_bloc} respecte al temps de transferència del bloc per part del perifèric t_{bloc}

$$t_{bloc} = t_{atència} + (N_{dades} * t_{dada})$$

$$t_{atència} = 0$$

$$N_{dades} = 200000$$

$$t_{dada} = m_{dada} / v_{transf} = 4 / 1000 \text{ Kbytes/s} = 0,004 \text{ ms}$$

$$t_{bloc} = 0 + (200000 * 0,004) \text{ ms} = 800 \text{ ms}$$

$$\% \text{ ocupació} = (t_{transf_bloc} * 100 / t_{bloc}) = (0,8 * 100) / 800 = 0,1\%$$

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	20/06/2015	12:00

Pregunta 4

4.1

Els processadors tenen diferents registres interns per poder fer la seva feina. Enumera els registres d'instrucció que coneguis i comenta la seva utilitat.

- PC (Program Counter): conté l'adreça de la instrucció següent que hi ha que llegir a memòria.
- IR (Instruction Register): conté la instrucció que s'executa.

4.2

a) Quins tipus de proximitat referencial podem distingir? En que consisteixen?

Distingim dos tipus de proximitat referencial:

1) Proximitat temporal. És quan, en un interval de temps determinat, la probabilitat que un programa accedeixi de manera repetida a les mateixes posicions de memòria és molt gran. La proximitat temporal és deguda principalment a les estructures iteratives; un bucle executa les mateixes instruccions repetidament, de la mateixa manera que les crides repetitives a subrutines.

2) Proximitat espacial. És quan, en un interval de temps determinat, la probabilitat que un programa accedeixi a posicions de memòria properes és molt gran. La proximitat espacial és deguda principalment al fet que l'execució dels programes és seqüencial – s'executa una instrucció darrere l'altra llevat de les bifurcacions –, i també a la utilització d'estructures de dades que estan emmagatzemades en posicions de memòria contigües.

b) En un sistema d'E/S gestionat per DMA. Quina funció tenen les senyals BUSREQ i BUSACK? Qui les activa i perquè?

Per a controlar l'accés al bus són necessaris dos senyals, BUSREQ i BUSACK (semblants als senyals INT i INTA utilitzats en interrupcions). Amb el senyal BUSREQ el controlador de DMA sol·licita el control del bus i el processador cedeix el bus activant el senyal BUSACK.