

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	17/06/2015	18:30

05.573R17R06R15REX{€
05.573 17 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	17/06/2015	18:30

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 computadors	05.573	17/06/2015	18:30

Pregunta 1

1.1 Pràctica – Part obligatòria

Escriure un fragment de codi ensamblador de la subrutina posCurScreenP1 que calcula el valor colCur. (No s'ha d'escriure el codi de tota la subrutina)

```

; ; ; ;
; Posiciona el cursor a la pantalla dins el tauler, en funció de les
; variables row i col i les constants RowScreenIni i ColScreenIni.
; Per a calcular la posició del cursor a pantalla (rowCur i colCur)
; utilitzar aquestes fórmules:
; rowCur = (row*2) + rowScreenIni + 3;
; colCur = (col*4) + colScreenIni + 4;
; Per posicionar el cursor cridar a la subrutina gotoxyP1.
;
; 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)]
; rowCur : fila per posicionar el cursor a la pantalla
; colCur : columna per posicionar el cursor a la pantalla
;
; Paràmetres d'entrada : Cap
;
; Paràmetres de sortida: Cap
; ; ; ;
posCurScreenP1:
    push rbp
    mov rbp, rsp

    push rdi
    push rsi

    mov esi, [col]
    shl esi, 2                ; (col*4)
    add esi, ColScreenIni    ; (col*4)+colScreenIni
    add esi, 4                ; (col*4)+colScreenIni+4

    mov [colCur], esi
    call gotoxyP1

posCurScreenP1_End:
    pop rsi
    pop rdi

    mov rsp, rbp
    pop rbp
    ret

```

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	17/06/2015	18:30

1.2 Pràctica – Part opcional

Completar el codi de la subrutina `showMinesP2` per a que mostri `numMines` a la part inferior del tauler de joc.

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Aquesta subrutina es dona feta. NO LA PODEU MODIFICAR.
; Situar el cursor en una fila i una columna de la pantalla
; en funció de la fila i columna indicats pels paràmetres esi i edi
; cridant a la funció gotoxy_C.
; Variables utilitzades: Cap
; Paràmetres d'entrada : edi: Fila, esi: Columna
; Paràmetres de sortida: Cap
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
gotoxyP2:

; ; ; ;
; 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 dels registres edi i esi).
; Per a posicionar el cursor cridar a la subrutina gotoxyP2 i per a mostrar
; els caràcters a la subrutina printchP2 implementant correctament el pas
; de paràmetres
; Variables utilitzades: numMines: nombre de mines que queden per posar.
; Paràmetres d'entrada : Cap
; Paràmetres de sortida: Cap
; ; ; ;
showMinesP2:
    push rbp
    mov  rbp, rsp
    ...
    ; dividir numMines entre 10, el quocient representarà les desenes
    ; i el tenim en el registre al i el residu representarà les unitats
    ; i el tenim en el registre dl.

    add al,'0'          ;convertim els valors a caràcters ASCII
    add dl,'0'
    ; Posicionar el cursor i mostrar dígit
    mov  __edi__, 22 ;RowScreenIni+DimMatrix+DimMatrix+3=22
    mov  __esi__, 10 ;ColScreenIni+5 = 10
    call __gotoxyP2__
    push rdi
    mov  dil, al
    call printchP2
    __pop__ rdi
    add  __esi__, 2
    call __gotoxyP2__
    mov  dil, dl
    call printchP2

showMinesP2_End:
    ...
    mov  rsp, rbp
    pop  rbp
    ret

```

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	17/06/2015	18:30

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(0000010Fh) = 00000000h M(0000020Fh) = FFFFFFFFh	Z = 0, C = 1, S = 1, V = 0
--	--	----------------------------

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

<p>a)</p> <pre> SUB R1, R0 ADD R0, [R2] END: MOV R0, [R2] </pre>	<p>b)</p> <pre> SAR [R0], 4h ADD R1, FFFFFFF448h </pre>
<p>R1:= 110h R0:= 1A00h R0:= FF0h</p> <p>Z = 0 , S = 0 , C = 0 , V = 0</p>	<p>[R0]:= 00000F00h R1:= FFFFFFF68h</p> <p>Z = 0 , S = 1 , C = 0 , V = 0</p>

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	17/06/2015	18:30

2.2

Donat el següent codi en alt nivell:

```
i = -8;
do {
    A[8+i] = 0;
    i = i + 1;
} while (i < 0);
```

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 [I], F8h

MOV R0, [I]

PLUS:

MOV R1, R0

ADD R1, 8

SAL R1, 2

MOV [A+R1], 0

ADD R0, 1

CMP R0, 0

JL PLUS

MOV [I], R0

END:

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	17/06/2015	18:30

2.3

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

```

CMP [R6+16h], R1
JNE END
MOV [A], R10
END:

```

Traduiu-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 000000C0h (que és el valor del PC abans de començar l'execució del fragment de codi). Supposeu que l'adreça simbòlica A val 00000400h. 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
42h	JNE
10h	MOV
26h	CMP

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	
000000C0h	CMP [R6+16h], R1	26	42	16	00	11							
000000C5h	JNE END	42	60	07	00								
000000C9h	MOV [A], R10	10	20	00	04	00	00	1A					
000000D0h													

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	17/06/2015	18:30

Pregunta 3

Memòria cau d'assignació directa

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'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:

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 en aquest cas s'indicarà el nou bloc que es porta a la memòria cau en la línia que li correspongui, expressat de la forma b:e ($a_0 - a_7$) on b: número de bloc, e:etiqueta 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 (0 - 7)	E 0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	E 0:0 (0 - 7)	0:0 (0 - 7)
1	1:0 (8 - 15)	1:0 (8 - 15)	E 1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)
2	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	E 2:0 (16 - 23)
3	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	E 3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)

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

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	17/06/2015	18:30

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

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

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 4 ns i el temps total d'accés en cas de fallada (t_f) és de 20 ns. Considerant la taxa d'encerts obtinguda a la pregunta anterior, quin és el temps mitjà d'accés a memòria (t_m) ?

$$t_m = T_e \times t_e + (1 - T_e) \times t_f = 0,65 \times 4 \text{ ns} + 0,35 \times 20 \text{ ns} = 2,6 \text{ ns} + 7 \text{ ns} = 9,6 \text{ ns}$$

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	17/06/2015	18:30

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, utilitzant E/S programada, 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: 0E00h i 0E04h
- El bit del **registre d'estat** que indica que el controlador del port d'E/S està disponible és el bit 2, o el tercer 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 **escriptura** des de memòria al port d'E/S
- Transferència de $N_{\text{dades}} = 100000$ dades
- La mida d'una dada és $m_{\text{dada}} = 4 \text{ bytes}$
- Adreça inicial de memòria on resideixen les dades: A0000000h

El següent codi realitzat amb el joc d'instruccions CISCA realitza la transferència descrita abans mitjançant la tècnica d'E/S programada. Completeu el codi.

```

1.          MOV      R3, _100000_
2.          MOV R2, A0000000h
3. Bucle:   IN  R0, _ [0E00h] _ ; llegir 4 bytes
4.          AND R0, 00000100b
5.          JE  Bucle
6.          MOV R0, [R2]          ; llegir 4 bytes
7.          ADD  R2, _4_
8.          OUT [0E04h], 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	17/06/2015	18:30

3.2.2 E/S programada

Quant temps dura la transferència del bloc de dades $t_{\text{transf_bloc}}$?

$$t_{\text{transf_bloc}} = t_{\text{atència}} + (N_{\text{dades}} * t_{\text{transf_dada}})$$

$$t_{\text{atència}} = 0$$

$$N_{\text{dades}} = 100000$$

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

$$t_{\text{transf_bloc}} = 0 + (100000 * 0,004 \text{ ms}) = 400 \text{ ms} = 0,4 \text{ s}$$

3.2.3 E/S programada

Si volguéssim fer servir el mateix processador i el mateix programa però amb un dispositiu d'E/S més ràpid, quina és la màxima taxa o velocitat de transferència del nou dispositiu que es podria suportar sense que el dispositiu s'hagués d'esperar?

$$t_{\text{cicle}} = 0,5 \text{ ns (nanosegons)}$$

$$t_{\text{instr}} = 0,5 \text{ ns} / 2 = 0,250 \text{ ns}$$

El mínim nombre d'instruccions que ha d'executar el programa per a cada dada transferida són les 8 instruccions: 3, 4, 5, 6, 7, 8, 9 i 10. Executar les 8 instruccions requereix $8 * t_{\text{instr}} = 8 * 0,250 \text{ ns} = 2 \text{ ns}$

Per tant, el temps mínim per a transferir una dada és: 2 ns

Es poden transferir 4 bytes cada 2 ns, es a dir: $4 / 2 * 10^{-9} = 2000 \text{ Mbyte/s} = 2 \text{ Gbytes/s}$

Examen 2014/15-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	17/06/2015	18:30

Pregunta 4

4.1

Què és un DSP? En què es diferencia d'un processador?

DSP: Digital Signal Processor és un dispositiu capaç de processar senyals en temps real de diferents fonts (so, video...)

Les principals diferències amb els processadors són:

- Els DSP són molt ràpids i implementen per hardware funcions que altres processadors fan per software
- Disposen de convertidors DAC o ADC
- Són escalables.

4.2

a) Quins factors fan que l'esquema de jerarquia de memòria funcioni satisfactòriament en un computador convencional? En què consisteixen?

Els dos factors que bàsicament fan que l'esquema de jerarquia de memòries funcioni satisfactòriament en un computador són els següents:

- El flux de dades entre els nivells de la jerarquia de memòries es pot fer en paral·lel amb el funcionament normal del processador.
- El principi de proximitat referencial dels programes. El codi dels programes s'organitza en subrutines, té estructures iteratives i treballa amb conjunts de dades agrupades.

b) En un sistema d'E/S gestionat per DMA, quina diferència hi ha entre el funcionament normal i el mode ràfega, quines avantatges té l'un respecte de l'altre?

Una manera d'optimitzar les operacions d'E/S per DMA consisteix a reduir el nombre de cessions i recuperacions del bus. Per a fer-ho, en lloc de sol·licitar i alliberar el bus per a cada dada que s'ha de transferir, es sol·licita i s'allibera el bus per a transferir un conjunt de dades de manera consecutiva. Aquesta modalitat de transferència s'anomena mode ràfega.

Per a fer la transferència d'aquest conjunt de dades, que anomenem ràfega, el controlador de DMA cal que disposi d'una memòria intermèdia (buffer), de manera que la transferència de dades entre la memòria i el controlador de DMA és pugui fer a la velocitat que permeti la memòria i no quedant limitada a la velocitat del perifèric.

Aquest mode de funcionament no afecta la programació ni la finalització de l'operació d'E/S descrita anteriorment, però sí que modifica la transferència de dades.