

Examen 2015/16-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/01/2016	12:00

05.573R16R01R16REEE€
05.573 16 01 16 EX

Enganxeu en aquest espai una etiqueta identificativa
amb el vostre codi personal
Examen

Aquest enunciat correspon també a les assignatures següents:

- 05.096 - Ampliació d'estructura i tecnologia de computadors

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 2015/16-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	16/01/2016	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 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 2015/16-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/01/2016	12:00

Pregunta 1

1.1 Pràctica – Part obligatòria

Escriure un fragment de codi assemblador de la subrutina `copyMatrixP1` que copia la matriu (`mRotated`) sobre la matriu (`m`), que són matrius de tipus `WORD`. (No s'ha d'escriure el codi de tota la subrutina).

```

;;;;
; Copiar els valors de la matriu (mRotated) a la matriu (m).
; Variables utilitzades:
; m      : matriu 4x4 on hi han el números del tauler de joc.
; mRotated: matriu 4x4 per fer la rotació.
; Paràmetres d'entrada : Cap.
; Paràmetres de sortida: Cap.
;;;;
copyMatrixP1:
    push rbp
    mov  rbp, rsp
    ...
    mov  eax, 0
    mov  r8d, 0                ;i = r8d
copyMatrixP1_Rows:
    mov  r9d, 0                ;j = r9d
copyMatrixP1_Cols:

    mov  bx, WORD[mRotated+eax] ;mRotated[i][j]
    mov  WORD[m+eax], bx        ;m[i][j] = mRotated[i][j]
    add  eax, 2                 ;incrementem l'índex
    inc  r9d                    ;
    cmp  r9d, DimMatrix        ;

    jl  copyMatrixP1_Cols
    inc  r8d
    cmp  r8d, DimMatrix
    jl  copyMatrixP1_Rows

copyMatrixP1_End:
    ...
    mov  rsp, rbp
    pop  rbp
    ret

```

Examen 2015/16-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	16/01/2016	12:00

1.2 Pràctica – Part opcional

Completar el codi de la subrutina rotateMatrixRP2. (Només completar els espais marcats, no es poden afegir o modificar altres instruccions).

```

; ; ; ;
; Calcular el valor de l'index per a accedir a una matriu (4x4) que guardarem al
; registre (eax) a partir de la fila (edi) i la columna (esi) rebuts com a paràmetre.
; eax=((edi*DimMatrix)+(esi))*2
; multipliquem per 2 perquè és una matriu de tipus short (WORD) 2 bytes.
; Aquesta subrutina no té una funció en C equivalent.
; Variables utilitzades: Cap.
; Paràmetres d'entrada : rdi(edi) : fila per a accedir a la matriu (4x4).
;                               rsi(esi) : columna per a accedir a la matriu (4x4).
; Paràmetres de sortida: rax(eax) : index per a accedir a la matriu (4x4) de tipus WORD.
; ; ; ;
calcIndexP2:

; ; ; ;
; Rotar a la dreta la matriu, rebuda com a paràmetre (edi), sobre
; la matriu (mRotated).
; La primera fila passa a ser la quarta columna, la segona fila passa
; a ser la tercera columna, la tercera fila passa a ser la segona
; columna i la quarta fila passa a ser la primera columna.
; A l'enunciat s'explica en més detall com fer la rotació.
; NOTA: NO és el mateix que fer la matriu transposada.
; La matriu rebuda com a paràmetre no s'ha de modificar,
; els canvis s'han de fer a la matriu (mRotated).
; Per a accedir a les matrius des d'assemblador cal calcular l'index
; a partir de la fila i la columna cridant la subrutina calcIndexP2.
; m[row][col], en C, és equivalent a WORD[m+eax], en assemblador, si
; eax = ((row*DimMatrix)+(col))*2. m[1][2] és [m+12].
; No s'ha de mostrar la matriu.
; Variables utilitzades: Cap.
; Paràmetres d'entrada : rdi(edi): Adreça de la matriu que volem rotar.
; Paràmetres de sortida: Cap.
; ; ; ;
rotateMatrixRP2:
    ...
    mov edx, edi                ;Adreça de la matriu que volem rotar
    mov r8d, 0                  ;i = r8d
    rotateMatrixRP2_Rows:
        mov r9d, 0              ;j = r9d
        rotateMatrixRP2_Cols:
            mov edi, r8d
            mov esi, r9d
            call calcIndexP2     ;index per a accedir a la matriu[i][j];
            mov bx, __WORD__[__edx+eax__]
            mov r10d, DimMatrix
            dec r10d
            sub r10d, __r8d__    ;DimMatrix-1-i = r10d
            mov edi, __r9d__
            mov __esi__, r10d
            call calcIndexP2
            ;mRotated[j][DimMatrix-1-i] = matriu[i][j]
            mov __WORD__[__mRotated+eax__], bx
            inc r9d
            cmp r9d, DimMatrix
            jl rotateMatrixRP2_Cols
        inc r8d
        cmp r8d, DimMatrix
        jl rotateMatrixRP2_Rows
    rotateMatrixRP2_End:
    ...
    ret

```

Examen 2015/16-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/01/2016	12:00

Pregunta 2

2.1

L'estat inicial del computador CISCA just abans de començar l'execució de cada fragment de codi (a 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)

```
MUL R5, 10h
MUL R10, 16
SUB R10, R5
```

```
R5= 00000280h
R10= 00000500h
R10= 00000280h
```

```
C=0, V=0, S=0, Z=0
```

b)

```
MOV R1, [V]
XOR R1, [00007E40h]
DEC R1
```

```
R1= 00000810h
R1= 00007640h
R1= 0000763Fh
```

```
R1= 0000763Fh
C=0, V=0, S=0, Z=0
```

Examen 2015/16-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	16/01/2016	12:00

2.2

Donat el següent codi d'alt nivell:

```

if (A<B) {
    if (C>=B) A =C;
    else B= C;
}
else A=B;

```

Es proposa la següent traducció a CISCA on hem deixat 6 espais per omplir.

```

INI: MOV R0, [A]
     MOV R1, [B]
     MOV R2, [C]
     CMP R0, R1
     JGE LSE1
     CMP R2, R1
     JL  LSE2
     MOV R0, R2
     JMP END
LSE2: MOV R1, R2
     JMP END
LSE1: MOV R0, R1
END:  MOV [A], R0
     MOV [B], R1
     MOV [C], R2

```

Examen 2015/16-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/01/2016	12:00

2.3

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

```
MOV R1, R10
MUL R12, 4
SUB [A+R12], 2048
```

Tradueix-ho a llenguatge màquina i expresseu-ho en la següent taula. Supposeu que la primera instrucció del codi s'assembla a partir de l'adreça 00004600h (que és el valor del PC abans de començar l'execució del fragment de codi). Supposeu que l'adreça simbòlica A val 02A0 0600h. En la següent taula 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
21h	SUB
22h	MUL
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 registro	Si el mode ha d'especificar un registre
0	No s'especifica registre.

Examen 2015/16-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	16/01/2016	12:00

		Bk per a k=0..10											
@	Assemblador	0	1	2	3	4	5	6	7	8	9	10	
00004600h	MOV R1,R10	10	11	1A									
00004603h	MUL R12, 4	22	1C	00	04	00	00	00					
0000460Ah	SUB [A+R12], 2048	21	5C	00	06	A0	02	00	00	08	00	00	
00004615h													

Examen 2015/16-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/01/2016	12:00

Pregunta 3

3.1

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:

0, 1, 2, 15, 23, 16, 55, 56, 17, 18, 28, 30, 40, 5, 63, 25, 43, 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	0	1	2	15	23
0	-	F 0 (0 - 7)	E 0 (0 - 7)	E 0 (0 - 7)	0 (0 - 7)	0 (0 - 7)
1	-	-	-	-	F 1 (8 - 15)	1 (8 - 15)
2	-	-	-	-	-	F 2 (16 - 23)
3	-	-	-	-	-	-

Línia	16	55	56	17	18	28
0	0 (0 - 7)	0 (0 - 7)	F 7 (56 - 63)	7 (56 - 63)	7 (56 - 63)	7 (56 - 63)
1	1 (8 - 15)	1 (8 - 15)	1 (8 - 15)	1 (8 - 15)	1 (8 - 15)	F 3 (24 - 31)
2	E 2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	E 2 (16 - 23)	E 2 (16 - 23)	2 (16 - 23)
3	-	F 6 (48 - 55)	6 (48 - 55)	6 (48 - 55)	6 (48 - 55)	6 (48 - 55)

Examen 2015/16-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/01/2016	12:00

Línia	30	40	5	63	25	43
0	7 (56 - 63)	7 (56 - 63)	7 (56 - 63)	E 7 (56 - 63)	7 (56 - 63)	7 (56 - 63)
1	E 3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	E 3 (24 - 31)	3 (24 - 31)
2	2 (16 - 23)	F 5 (40 - 47)	5 (40 - 47)	5 (40 - 47)	5 (40 - 47)	E 5 (40 - 47)
3	6 (48 - 55)	6 (48 - 55)	F 0 (0 - 7)	0 (0 - 7)	0 (0 - 7)	0 (0 - 7)

Línia	56	42	50			
0	E 7 (56 - 63)	7 (56 - 63)	F 6 (48 - 55)			
1	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)			
2	5 (40 - 47)	E 5 (40 - 47)	5 (40 - 47)			
3	0 (0 - 7)	0 (0 - 7)	0 (0 - 7)			

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

$$T_e = 11 \text{ encerts} / 20 \text{ accessos} = 0,55$$

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 de fallades obtinguda a la pregunta anterior, quin és el temps mitjà d'accés a memòria (t_m) ?

$$t_m = T_e \times t_e + (1 - T_e) \times t_f = 0,55 \times 2 \text{ ns} + 0,45 \times 30 \text{ ns} = 1,1 \text{ ns} + 13,5 \text{ ns} = 14,6 \text{ ns}$$

Examen 2015/16-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/01/2016	12:00

3.2 Sistema d'E/S

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 $v_{\text{transf}} = 10 \text{ MBytes/s} = 10000 \text{ Kbytes/s}$
- Temps de latència mitjà del dispositiu $t_{\text{latència}} = 0$
- Adreces dels **registres d'estat i dades** del controlador d'E/S: 0BF0h i 0BF4h
- 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 1 instrucció per cicle de rellotge
- Transferència de **lectura** des de memòria al port d'E/S
- Transferència de **$N_{\text{dades}} = 400000$ dades**
- La mida d'una dada és **$m_{\text{dada}} = 4 \text{ bytes}$**
- Adreça inicial de memòria on resideixen les dades: A0000000h
- El temps per atendre la interrupció ($t_{\text{rec_int}}$) és de 2 cicles de rellotge

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, [0BF4h]
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

```

3.2.2) Quant temps dedica la CPU a la transferència del bloc de dades $t_{\text{transf_bloc}}$?

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

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

Temps d'execució de una instrucció, $t_{\text{instr}}: t_{\text{cicle}} = 0,5 \text{ ns}$

Temps d'execució RSI, $t_{\text{rsi}}: N_{\text{rsi}} * t_{\text{instr}} = 12 \text{ instr.} * 0,5 \text{ ns} = 6 \text{ ns}$

Examen 2015/16-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/01/2016	12:00

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

$$t_{\text{transf_dada}} = t_{\text{rec_int}} + t_{\text{rsi}} = 1 + 6 = 7 \text{ ns}$$

Nombre d'interrupcions produïdes (nombre total de dades, N_{dades}): 400000 interrupcions

Temps consumit en total en TOTES les interrupcions:

$$t_{\text{transf_bloc}} = t_{\text{transf_dada}} * N_{\text{dades}} = 7 \text{ ns} * 400000 \text{ interrupcions} = 2800000 \text{ ns} = 2,8 \text{ ms (milisegons)}$$

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

$$t_{\text{bloc}} = t_{\text{latència}} + (N_{\text{dades}} * t_{\text{dada}})$$

$$t_{\text{latència}} = 0$$

$$N_{\text{dades}} = 400000$$

$$t_{\text{dada}} = m_{\text{dada}} / v_{\text{transf}} = 4 / 10000 \text{ Kbytes/s} = 0,0004 \text{ ms}$$

$$t_{\text{bloc}} = 0 + (400000 * 0,0004) \text{ ms} = 160 \text{ ms}$$

$$\% \text{ ocupació} = (t_{\text{transf_bloc}} * 100 / t_{\text{bloc}}) = (2,8 * 100) / 160 = 1,75\%$$

Examen 2015/16-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/01/2016	12:00

Pregunta 4

4.1

Quan es dissenya una arquitectura, un dels aspectes importants és la grandària de les instruccions. Què dues alternatives hi ha i quins són els seus avantatges i inconvenients?

- **Instruccions de grandària fixa:** totes les instruccions ocuparan el mateix nombre de bits. Aquesta alternativa simplifica el disseny del processador i l'execució de les instruccions pot ser més ràpida.
- **Instruccions de grandària variable:** la grandària de les instruccions dependrà del nombre de bits necessari per a cadascuna. Aquesta alternativa permet dissenyar un conjunt ampli de codis d'operació, l'adreçament pot ser més flexible i permet posar referències a registres i memòria. Com a contrapartida, augmenta la complexitat del processador.

4.2

a) En la memòria cau, quines polítiques d'assignació es defineixen? Descriure-les breument.

- 1) **Política d'assignació directa:** un bloc de la memòria principal només pot ser 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.

b) Quins són els passos bàsics per a la gestió d'una interrupció en un sistema amb una única línia d'interrupció i un únic mòdul d'E/S?

- 1.- Petició del mòdul d'entrada/Sortida
- 2.- Cicle de reconeixement de la interrupció
 - 2.a.- Reconeixement de la interrupció
 - 2.b.- Salvaguarda de l'estat del processador
 - 2.c.- Crida a la RSI
- 3.- Execució de la rutina de servei d'interrupció
 - 3.a.- Inici de l'execució de la RSI
 - 3.b.- Intercanvi de la dada
 - 3.c Finalització de l'execució de la RSI
 - 3.d Retorn d'interrupció: Restaurar l'estat del processador.