

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	15/01/2014	18:30

05.573R15R01R14REED
05.573 15 01 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 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	15/01/2014	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 (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 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	15/01/2014	18:30

Pregunta 1

1.1

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Llegir l'element de la fila (r) i la columna (c) d'una matriu (m) rebuts
; com a paràmetre sobre els registres bh, bl i edx, respectivament, i
; retornar el caràcter llegit de la matriu sobre el registre al.
; La matriu ha de ser una matriu quadrada de dimensió 'dimMatrix'
; de tipus byte (char) i la fila i la columna valors positius i
; menors que dimMatrix.
; Per accedir a m[r][c], en ensamblador cal calcular primer l'índex.
; índex = r * dimMatrix + c i per accedir a la matriu hem de fer [m + índex]
;
; Variables utilitzades:
; Cap
;
; Paràmetres d'entrada :
; rbx: (bh) fila(r), (bl) columna (c)
; rdx: (edx) adreça de la matriu (m).
;
; Paràmetres de sortida:
; rax: (al) element de la matriu. m[r][c]

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
getMatrix:
    push rbp
    mov  rbp, rsp
    ; guardem els registres que modifiquem i que no són paràm. de sortida.
    push rbx
    push rdx
    push rsi

```

; escriure el codi per assignar al registre AL
m[r][c], per accedir a m[r][c] hem de fer:
[m + índex] on índex = r * dimMatrix + c

```

    mov rax, 0
    mov al, dimMatrix ; Calculem índex per accedir a la matriu.
    mul bh             ; ax = al * bh
    add al, bl
    mov esi, eax

```

```

    mov rax, 0
    mov al, [edx+esi]

```

```

getMatrix_end:
    pop rsi
    pop rdx
    pop rbx
    mov rsp, rbp
    pop rbp
    ret

```

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	15/01/2014	18:30

1.2

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Validar si totes les peces de la matriu pieces estan ordenades
; i l'espai en blanc està a la darrera posició (inferior-dreta).
; Si les peces no estan ordenades no modifiquem l'estat (state) rebut
; com a paràmetre sobre el registre edi. Si les peces estan ordenades
; posem state a 2. Retornar l'estat actualitzat (state) sobre el registre eax
; Per accedir a la matriu pieces utilitzar la subrutina getMatrix.
;
; Variables utilitzades: pieces
; Paràmetres d'entrada: rdi: (edi) state
; Paràmetres de sortida: rax: (eax) state
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
checkPieces:
...
; Fragment de codi per veure si la matriu està ordenada
mov bh, 0 ;files
mov ch, 0
mov cl, 0 ;posicions que accedim de pieces
checkPieces_for1:
    cmp bh, dimMatrix
    jge checkPieces_sorted
    mov bl, 0 ;columnes
    checkPieces_for2:
        cmp bl, dimMatrix
        jge checkPieces_endfor2
        cmp cl, ((dimMatrix*dimMatrix)-1)
        jge checkPieces_sorted
        mov     ___edx___, pieces
        ;Subrutina getMatrix
        ;Paràmetres entrada: (bh) fila, (bl) columna, (edx) adreça de la matriu
        ;Paràmetres sortida: (al) element de la matriu.
        call getMatrix
        cmp     ___al___, ch
        jl      checkPieces_unsorted
        mov     ___ch___, al
        inc     ___bl___
        inc cl
        jmp     checkPieces_for2
    checkPieces_endfor2:
        inc     ___bh___
        jmp     checkPieces_for1
    checkPieces_sorted:
        mov eax, 2 ;state=2. Matriu ordenada
        jmp     checkPieces_end
    checkPieces_unsorted:
        mov eax, edi
checkPieces_end:
...
ret

```

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	15/01/2014	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 = 00001010h R1 = 00002020h R2 = 00003030h	M(00002020h) = 0000F00Fh M(00003030h) = 0000F000h M(0000F0A0h) = 0000000Ah	Z = 0, C = 1, S = 1, V = 0
--	--	----------------------------

L'adreça simbòlica W val 0000F0A0h. Quin serà l'estat del computador després d'executar cada fragment de codi? (només modificacions, excloent-hi el PC).

a)	b)
ADD R0, R1 SUB [R1], R0	XOR R2, [W] MOV [R1], R0
R0 = 00003030h M(00003030h) = 0000BFDh Z = 0, S = 0, C = 0, V = 0	R2 = 0000303Ah M(00002020h) = 00001010h Z = 0, S = 0, C = 0, V = 0

2.2

Donat el següent codi en alt nivell:

```
i = 0;
suma = 0;
do
{
  suma = suma + v[i];
  i = i + 2;
} while (i < 100) && (suma < 1000)
```

Es defineix V com un vector de 100 elements de 4 bytes cada un. Es proposa la següent traducció a CISCA on hem deixat 6 llocs per omplir:

```
INI:  MOV R0, [i]           ; R0 implementa la variable 'i' del bucle.
      MOV R0, 0
      MOV R2, [SUMA]
      XOR R2, R2           ; has d'especificar l'instrucció.
LOOP: ADD R2, [V+R0]       ; has d'especificar el mode de direccionament.
      ADD R0, 8            ; has d'especificar el segon operand
      CMP R0, 400
```

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	15/01/2014	18:30

JGE END ; Has d'especificar l' instrucció de salt correcta
 CMP R2, 1000 ; has d'especificar el primer operand.
 JLE LOOP ; Has d'especificar la instrucció de salt correcta.
 END: MOV [SUMA], R2
 MOV [i], R0

2.3

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

```

INI:    ADD    [V], R4
        SUB    R0, R4
        JLE    INI
  
```

Traduïu-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 **0CC75BB0h** (que és el valor del PC abans de començar l'execució del fragment de codi). Supposeu també que l'adreça simbòlica V val **0000AA10h**. 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.

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

Taula de codis d'instrucció

B0	Instrucció
20h	ADD
21h	SUB
44h	JLE

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

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	15/01/2014	18:30

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	
0CC75BB0h	ADD [V], R4	20	20	10	AA	00	00	14					
0CC75BB7h	SUB R0, R4	21	10	14									
0CC75BBAh	JLE INI	44	60	F2	FF								
0CC75BBEh													

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	15/01/2014	18:30

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:

30, 22, 28, 5, 6, 7, 8, 9, 17, 32, 4, 55, 22, 53, 39, 12, 21, 32, 22, 13

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: 4	Fallada: 55
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 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	15/01/2014	18:30

	Fallada: 22	Fallada: 53
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

	Fallada: 39	Fallada: 21
Línia 0	32, 33, 34, 35, 36, 37, 38, 39	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	48, 49, 50, 51, 52, 53, 54, 55	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:	Fallada:
Línia 0		
Línia 1		
Línia 2		
Línia 3		

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

$$T_f = 7 \text{ fallades} / 20 \text{ accessos} = 0,35$$

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) ?

$$t_m = T_f \times t_f + (1 - T_f) \times t_e = 0,35 \times 20 \text{ ns} + 0,65 \times 2 \text{ ns} = 7 \text{ ns} + 1,3 \text{ ns} = 8,3 \text{ ns}$$

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:

30, 22, 28, 5, 6, 7, 8, 9, 17, 32, 4, 55, 22, 53, 39, 12, 21, 32, 22, 13

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	15/01/2014	18:30

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	16, 17, 18, 19, 20, 21, 22, 23
Línia 3	24, 25, 26, 27, 28, 29, 30, 31	32, 33, 34, 35, 36, 37, 38, 39

	Fallada: 55	Fallada: 12
Línia 0	0, 1, 2, 3, 4, 5, 6, 7	8, 9, 10, 11, 12, 13, 14, 15
Línia 1	48, 49, 50, 51, 52, 53, 54, 55	48, 49, 50, 51, 52, 53, 54, 55
Línia 2	16, 17, 18, 19, 20, 21, 22, 23	16, 17, 18, 19, 20, 21, 22, 23
Línia 3	32, 33, 34, 35, 36, 37, 38, 39	32, 33, 34, 35, 36, 37, 38, 39

	Fallada:	Fallada:
Línia 0		
Línia 1		
Línia 2		
Línia 3		

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

$$T_f = 3 \text{ fallades} / 20 \text{ accessos} = 0,15$$

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) ?

$$t_m = T_f \times t_f + (1 - T_f) \times t_e = 0,15 * 20 \text{ ns} + 0,85 * 2 \text{ ns} = 3 \text{ ns} + 1,7 \text{ ns} = 4,7 \text{ ns}$$

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:

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	15/01/2014	18:30

- La CPU funciona amb un rellotge de 2GHz 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 sisè bit menys significatiu (quan val 1 indica que està disponible).
- Transferència d'**escriptura** des de memòria al port d'E/S.
- Transferència de $N_{dades}=200.000$ dades, és a dir, 200.000×4 Bytes = 800.000 Bytes
- Adreça inicial de memòria on resideixen les dades: 10000000h
- 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, __200000__
2.          MOV  R2, 10000000h
3. Bucle:   IN    R0, [0204h]          ; llegir 4 bytes
4.          _AND_ R0, 00010000b
5.          _JE_  Bucle
6.          MOV  R0, [_R2_]           ; llegir 4 bytes
7.          ADD  R2, __4__
8.          _OUT_ 0200h, R0           ; escriure 4 bytes
9.          SUB  R3, __1__
10.         _JNE_ Bucle

```

Quin és el percentatge de temps que dedica la CPU a la tasca d'Entrada/Sortida?

100%

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]
6.          OUT  [_0200h]_, R0        ; escriure 4 bytes
7.          _ADD_ R1, 4
8.          MOV  __[Dir]__, R1
9.          POP  __R1__
10.         POP  R0
11.         STI

```

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	15/01/2014	18:30

12. IRET

Quin és el percentatge de temps que dedica la CPU a la tasca d'Entrada/Sortida?

800.000 Bytes a transferir. 5.000 Bytes per segon. Això fa que el temps total de la transferència sigui de 160 segons.

Cada cicle de rellotge és de 0,5ns. Per tant, cada instrucció triga 0,5 ns.

Una interrupció necessita 12 instruccions, per tant són 6 ns.

Hi ha 200.000 interrupcions, per tant són 1.200.000 ns. o 1,2 ms.

Això representa un 0,00075% del temps total de la transferència.

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	15/01/2014	18:30