

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	25/01/2014	15:30

05.573 25 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 computadores	05.573	25/01/2014	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 (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	25/01/2014	15:30

Pregunta 1

1.1

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Escriure un caràcter en la fila (r) i columna (c) d'una matriu (m),
; rebuts com a paràmetre sobre els registres al, bh, bl i edx,
; respectivament.
; 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 assembleador cal calcular primer l'índex.
; index = r * dimMatrix + c i l'accés a la matriu és [m + índex]
;
; Variables utilitzades: Cap
; Paràmetres d'entrada :
;   rax: (al) valor que volem guardar en la matriu
;   rbx: (bh) fila(r), (bl) columna (c)
;   rdx: (edx) adreça de la matriu (m).
; Paràmetres de sortida: Cap
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
setMatrix:
    push rbp
    mov  rbp, rsp
    ; guardem els registres que modifiquem i que no són paràm. de sortida.
    push rax
    push rbx
    push rcx
    push rdx
    push rsi

```

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

```

    mov rcx, 0
    mov cl , al ; guardem el valor.
    mov rax, 0 ; Calculem index per accedir a la matriu.
    mov al, dimMatrix
    mul bh      ; ax=al*bh
    add al, bl
    mov esi, eax
    mov al, cl

```

```

    mov [edx+esi], al

```

```

setMatrix_end:
    pop rsi
    pop rdx
    pop rcx
    pop rbx
    pop rax
    mov rsp, rbp
    pop rbp
    ret

```

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	25/01/2014	15:30

1.2

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Cercar on està l'espai en blanc de la matriu pieces
; per fixar la posició de row i col a l'inici del joc.
; Si el nombre d'espais en blanc és 1, voldrà dir que la matriu és
; correcta i posarem state a 1 per jugar.
; Si el nombre d'espais en blanc és diferent de 1, voldrà dir
; que la matriu no és correcta i posarem state a 4 per sortir.
; Inicialitzar la matriu pieces amb els valors de la matriu piecesIni
; i cercar on està l'espai en blanc de la matriu pieces.
; Inicialitzar les variables row i col amb la fila i la columna on
; hi ha l'espai en blanc.
; Inicialitzar el moviments restants a 9 (moves=9) .
; Per a accedir a les matrius piecesIni i pieces utilitzar la subrutina
; getMatrix i setMatrix.
; Retornar l'estat del joc (state) sobre el registre eax.
;
; Variables utilitzades: piecesIni, pieces, moves, row i col.
; Paràmetres d'entrada: Cap
; Paràmetres de sortida: rax: (eax) state
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
getPosIni:
...
mov rax, 0
mov rcx, 0          ;cl: spaces.
mov rdx, 0
mov byte[moves], 9
mov bh, 0           ;files
getPosIni_for1:
    cmp bh, dimMatrix
    jge getPosIni_spaces
    mov bl, 0        ;columnes
getPosIni_for2:
    cmp bl, dimMatrix
    jge getPosIni_endfor2
    mov     __edx__, piecesIni
    ;Paràmetres d'entrada: (bh) fila, (bl) columna, (edx) adreça matriu
    ;Paràmetres de sortida: (al) element de la matriu.
    call    __getMatrix__    ;al=piecesIni[bh][bl]
    mov     edx, __pieces__
    ;Paràmetres d'entrada: (al)valor per guardar en la matriu,
    ; (bh)fila, (bl)columna, (edx)adreça de la matriu .
    ;Paràmetres de sortida: Cap
    call setMatrix
    cmp     __al__, ' '
    jne getPosIni_nextfor2
    mov [row], bh
    mov [col], bl
    inc cl
getPosIni_nextfor2:
    inc bl
    jmp getPosIni_for2
getPosIni_endfor2:
    inc bh
    jmp getPosIni_for1

getPosIni_spaces:
...
ret

```

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	25/01/2014	15:30

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 = 0000F00Fh R1 = 00000000h R2 = 0000AACh	M(00000000h) = 0000F00Fh M(0000AACh) = 0000F000h M(0000F00Ch) = 0000000Ah	Z = 0, C = 0, S = 1, V = 1
---------------------------------------------------	---------------------------------------------------------------------------------	----------------------------

L'adreça simbòlica W val 0000F00Ch. Quin serà l'estat del computador després d'executar cada fragment de codi? (només modificacions, excloent-hi el PC).

a)	b)
MOV R1,[R2] MOV [R2], 0h	AND [0000F00Ch], Ah SAL [W],1Dh
R1= 0000F000h M(0000AACh) = 00000000h Z = 0, S = 1, C = 0, V = 1	M(0000F00Ch) = 40000000h Z = 0, S = 0, C = 1, V = 0

2.2

Donat el següent codi en alt nivell:

```
X2= 0;
while (X1<=V[X2] && X2<100)
{
    X2++;
    X1 = X1 * 2;
}
```

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

```

MOV R1, [X1] ; Has d'especificar segon operand
MOV R2, 0
WHILE: CMP R2, 400 ; Has d'especificar segon operand
JE ENDWHILE
CMP R1, [V+R2] ; Has d'especificar segon operand
JG ENDWHILE ; Has d'especificar l'etiqueta destí.
MUL R1, 2
ADD R2, 4 ; Has d'especificar el segon operand
JMP WHILE ; Has d'especificar l'instrucció correcta.
```

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	25/01/2014	15:30

```
ENDWHILE:  MOV  [X1], R1
           MOV  [X2], R2
```

2.3

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

```
LOOP: SUB R3, 4
      CMP R1, [V+R3]
      JE LOOP
END:
```

Traduïu-ho a llenguatge màquina i expresseu-ho en la següent taula. Suposeu 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). Suposeu 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
21h	SUB
26h	CMP
41h	JE

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	Significat
-----------	------------

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	25/01/2014	15:30

Bk<3..0>	
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	SUB R3, 4	21	13	00	04	00	00	00					
0CC75BB7h	CMP R1, [V+R3]	26	11	53	10	AA	00	00					
0CC75BBEh	JE LOOP	41	60	EE	FF								
0CC75BC2h													

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	25/01/2014	15:30

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, 40, 54, 55, 12, 53, 39, 13, 21, 32, 22, 41

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: 40
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	40, 41, 42, 43, 44, 45, 46, 47
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: 54	Fallada: 12
Línia 0	0, 1, 2, 3, 4, 5, 6, 7	0, 1, 2, 3, 4, 5, 6, 7
Línia 1	40, 41, 42, 43, 44, 45, 46, 47	8, 9, 10, 11, 12, 13, 14, 15
Línia 2	48, 49, 50, 51, 52, 53, 54, 55	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	25/01/2014	15:30

	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: 41	Fallada:
Línia 0	32, 33, 34, 35, 36, 37, 38, 39	
Línia 1	40, 41, 42, 43, 44, 45, 46, 47	
Línia 2	16, 17, 18, 19, 20, 21, 22, 23	
Línia 3	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) ?

$$T_f = 6 \text{ fallades} / 20 \text{ accessos} = 0,3$$

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) ?

$$t_m = T_f \times t_f + (1 - T_f) \times t_e = 0,3 * 20 \text{ ns} + 0,7 * 2 \text{ ns} = 6 \text{ ns} + 1,4 \text{ ns} = 7,4 \text{ ns}$$

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, 40, 54, 55, 12, 53, 39, 13, 21, 32, 22, 41

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	25/01/2014	15:30

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: 40
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	40, 41, 42, 43, 44, 45, 46, 47

	Fallada: 54	Fallada: 39
Línia 0	48, 49, 50, 51, 52, 53, 54, 55	48, 49, 50, 51, 52, 53, 54, 55
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	32, 33, 34, 35, 36, 37, 38, 39
Línia 3	40, 41, 42, 43, 44, 45, 46, 47	40, 41, 42, 43, 44, 45, 46, 47

	Fallada: 21	Fallada: 41
Línia 0	48, 49, 50, 51, 52, 53, 54, 55	40, 41, 42, 43, 44, 45, 46, 47
Línia 1	8, 9, 10, 11, 12, 13, 14, 15	8, 9, 10, 11, 12, 13, 14, 15
Línia 2	32, 33, 34, 35, 36, 37, 38, 39	32, 33, 34, 35, 36, 37, 38, 39
Línia 3	16, 17, 18, 19, 20, 21, 22, 23	16, 17, 18, 19, 20, 21, 22, 23

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

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

$$T_f = 5 \text{ fallades} / 20 \text{ accessos} = 0,25$$

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 mitjà d'accés a memòria (t_m) ?

$$t_m = T_f \times t_f + (1-T_f) \times t_e = 0,25 * 20 \text{ ns} + 0,75 * 2 \text{ ns} = 5 \text{ ns} + 1,5 \text{ ns} = 6,5 \text{ ns}$$

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	25/01/2014	15:30

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:

- La CPU funciona amb un rellotge de 1GHz de freqüència i executa 1 instrucció per cada 2 cicles de rellotge
- Adreces dels **registres de dades i d'estat** del controlador d'E/S: 0A0h i 0A4h
- El bit del **registre d'estat** que indica que el controlador del port d'E/S està disponible és el bit 3, o el quart bit menys significatiu (quan val 1 indica que està disponible)
- Transferència de **escriptura** des de memòria al port d'E/S
- Transferència de $N_{\text{dades}}=200.000$ dades, és a dir, $200.000 \times 4 \text{ Bytes} = 800.000 \text{ Bytes}$
- Adreça inicial de memòria on resideixen les dades: 20000000h
- La velocitat de transferència el port és de 400.000 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, 20000000h
3. Bucle: IN     R0, [_0A4h_]      ; llegir 4 bytes
4.      AND    R0, _00001000b_
5.      _JE_   Bucle
6.      MOV    R0, [_R2_]         ; llegir 4 bytes
7.      ADD    _R2_, 4
8.      OUT    0A0h, _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

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	25/01/2014	15:30

1. `_CLI_`
2. `PUSH _R0_`
3. `_PUSH_ R1`
4. `MOV _R1_, [Dir]`
5. `MOV _R0_, [R1]`
6. `OUT _[0A0h]_, R0 ; escriure 4 bytes`
7. `ADD R1, _4_`
8. `MOV _[Dir]_, R1`
9. `POP _R1_`
10. `_POP_ R0`
11. `STI`
12. `IRET`

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

800.000 Bytes a transferir. 400.000 Bytes per segon. Això fa que el temps total de la transferència sigui de 2 segons.

Cada cicle de rellotge és de 1ns. Per tant, cada instrucció triga 2 ns.

Una interrupció necessita 12 instruccions, per tant són 24 ns.

Hi ha 200.000 interrupcions, per tant són 4.800.000 ns. o 4,8 ms.

Això representa un 0,24% del temps total de la transferència.

Examen 2013/14-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	25/01/2014	15:30