

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	21/01/2012	09:00

05.573R21R01R12REEK€
05.573 21 01 12 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

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	21/01/2012	09:00

Pregunta 1 (20%: 2.1: 10%, 2.2: 10%)

Només s'han de completar les instruccions marcades.

Els punts suspensius indiquen que en aquesta subrutina 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.

Pregunta 1.1 (10%)

Completa el codi corresponent a la subrutina `m_copy_col` (copia el vector `b` de 3 posicions en la columna de la matriu `mr` 3x3 indicada per la variable `col`).

```
m_copy_col:
    push rbp
    mov rbp, rsp

    mov esi, [col]
    mov edi, 0
m_copy_col_bucle:
    mov eax, [b+edi*4]
    mov [mr+esi*4], eax
    add esi, 3
    inc edi
    cmp edi, 3
    jl  m_copy_col_bucle

m_copy_col_end:
    mov rsp, rbp
    pop rbp
    ret
```

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	21/01/2012	09:00

Pregunta 1.2 (10%)

Completa el codi corresponent a la subrutina que fa la transposada de la matriu mr.

```
m_trans_mr:
    push rbp
    mov rbp, rsp
    mov edx, 8
    m_trans_mr_row:
        mov esi, edx ; Valor de referència de la diagonal.
        mov edi, edx ; esi estarà per sota la diagonal i edi per
sobre.
        m_trans_mr_col:
            sub esi, 1 ; Restem primer per evitar transposar els
            sub edi, 3 ; valors de la diagonal que no s'han de moure
            jl m_trans_mr_next ; Hem acabat la columna
            mov eax, [mr+esi*4]; Fem l'intercanvi de valors.
            xchg eax, [mr+edi*4]
            mov [mr+esi*4], eax
            jmp m_trans_mr_col

m_trans_mr_next:
    sub edx, 4 ; Ens movem per la diagonal.
    jge m_trans_mr_row
m_trans_mr_end:
    mov rsp, rbp
    pop rbp
    ret
```

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	21/01/2012	09:00

Pregunta 2 (40%: 2.1: 15%, 2.2: 15%, 2.3: 10%)

Apartat 2.1 (15%)

L'estat inicial del computador CISCA just abans de començar l'execució de cada fragment de codi (de cada apartat) és el següent:

R0 = 00000000h	M(00000000h) = 00001000h	Z = 0 C = 0
R1 = 00001000h	M(00001000h) = 00002000h	S = 0 V = 0
R2 = 00002000h	M(00002000h) = 10000000h	

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

a) MOV R2, FFFFFFFFh SAR [R1+00001000h], 4 XOR R2,[00002000h]	b) ADD [R1], R1 MOV R2, 00002000h SUB [R1], R2
[00002000h] = 01000000h R2 = FFFFFFFFh Z = 0, S = 1 C = 0, V = 0	R2 = 00002000h M(00001000h) = 00001000h Z = 0, S = 0 C = 0, V = 0

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	21/01/2012	09:00

Apartat 2.2 (15%)

Completeu els buits en el fragment de codi CISCA perquè sigui una traducció eficient a l'assemblador del codi en C que posa a 0 la columna 20 de la matriu M de 7 per 100 nombres enters (7 files per 100 columnes):

```
i = 0;
do {
    M[i][20]=0;
    i = i+1;
} while (i<7);
```

La matriu està emmagatzemada per files en posicions consecutives de memòria, com és habitual quan es tradueix codi en C. Per exemple, els elements M[0][0], M[0][1], M[1][0] i M[5][40] es troben emmagatzemats en les adreces de memòria M, M+4, M+400 i M+2160 respectivament.

Després d'executar-se el fragment de codi tots els registres han de mantenir els valors originals.

```
        PUSH R1
        MOV  R1, 80
DO:     MOV  [M + R1], 0
        ADD  R1, 400
        CMP  R1, 2880      ; si aquí es posa 2480
        JL   DO            ; aquí s'ha de posar JLE
        POP  R1
```

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	21/01/2012	09:00

Apartat 2.3 (10%)

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

```

                MUL    R3, [X+R5]
                SUB    R5, 4
end_w:         MOV    [M], R3

```

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 00002027h (que és el valor del PC abans de començar l'execució del fragment de codi). Supposeu també que les adreces simbòliques X i M valen 00000004 i 00000008 respectivament. 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

Instrucció	B0
MOV	10h
SUB	21h
MUL	22h

Taula de codis d'adreçament (Bk<7..4>)

Camp mode Bk<7..4>	Mode
0h	Inmediato
1h	Registro
2h	Memoria
3h	Indirecto
4h	Relativo
5h	Indexado
6h	A PC

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	21/01/2012	09:00

Taula de codis 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.

Taula de resposta:

@	Ensamblador	Bk para k=0..10										
		0	1	2	3	4	5	6	7	8	9	10
00002027h	MUL R3, [X+R5]	22	13	55	04	00	00	00				
0000202Eh	SUB R5, 4	21	15	00	04	00	00	00				
00002035h	MOV [M], R3	10	20	08	00	00	00	13				
0000203Ch												

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	21/01/2012	09:00

Pregunta 3 (40%: 3.1: 15%, 3.2: 15%, 3.3: 10%)

Apartat 3.1 (15%)

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 4 paraules. Cada bloc comença a una adreça múltiple de 4. Així, el bloc 0 conté les adreces 0, 1, 2 i 3, el bloc 1, les adreces 4, 5, 6 i 7, i el bloc N les adreces $4*N$, $4*N+1$, $4*N+2$ i $4*N+3$. Una fórmula per calcular l'identificador numèric del bloc és la següent:

Bloc = adreça de memòria (adreça a paraula) DIV 4 (mida del bloc en paraules)

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, 4 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 i 3).

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	21/01/2012	09:00

Apartat 3.1.1

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. En aquest cas, l'identificador del bloc determina la línia específica on es pot guardar fent servir la següent fórmula (similar a la fórmula per determinar el bloc):

$$\text{Línia} = \text{identificador de bloc} \bmod 4 \text{ (mida de la cau en línies)}$$

L'execució d'un programa genera la següent llista de lectures a memòria:

1, 2, 3, 10, 11, 12, 13, 14, 30, 31, 32, 1, 2, 3, 10, 11, 12, 13, 14, 30

3.1.1.a)

La següent taula mostra l'estat inicial de la cau, que conté les primeres 16 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:30	Fallada:32	Fallada:1	Fallada:12
Línia 0	0, 1, 2, 3		32, 33, 34, 35	0, 1, 2, 3	
Línia 1	4, 5, 6, 7				
Línia 2	8, 9, 10, 11				
Línia 3	12, 13, 14, 15	28, 29, 30, 31			12, 13, 13, 14

	Fallada:30	Fallada:	Fallada:	Fallada:	Fallada:
Línia 0					
Línia 1					
Línia 2					
Línia 3	28, 29, 30, 31				

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

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

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 4 ns i el temps total d'accés en cas de fallada (t_f) és de 24 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitjà d'accés a memòria (t_m) ?

$$t_m = T_f \times t_f + (1 - T_f) \times t_e = 0,25 * 24 \text{ ns} + 0,75 * 4 \text{ ns} = 6 \text{ ns} + 3 \text{ ns} = 9 \text{ ns}$$

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	21/01/2012	09:00

Apartat 3.1.2

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 s'ha referenciat. Considerem la mateixa llista de lectures a memòria:

1, 2, 3, 10, 11, 12, 13, 14, 30, 31, 32, 1, 2, 3, 10, 11, 12, 13, 14, 30

3.1.2.a)

La següent taula mostra l'estat inicial de la cau, que conté les primeres 16 paraules de la memòria (organitzades en 4 blocs). Completar la taula (afegint les columnes que siguin necessàries) per a mostrar l'evolució de la cau durant l'execució del programa. Per a cada fallada en la cau omplir una nova columna indicant quina referència a memòria ha provocat la fallada i el nou estat de la cau.

	Estat Inicial	Fallada:30	Fallada:32	Fallada:1	Fallada:10
Línia 0	0, 1, 2, 3		32, 33, 34, 35		
Línia 1	4, 5, 6, 7	28, 29, 30, 31			
Línia 2	8, 9, 10, 11			0, 1, 2, 3	
Línia 3	12, 13, 14, 15				8, 9, 10, 11

	Fallada:12	Fallada:30	Fallada:	Fallada:	Fallada:
Línia 0		28, 29, 30, 31			
Línia 1	12, 13, 14, 15				
Línia 2					
Línia 3					

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

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

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 4 ns i el temps total d'accés en cas de fallada (t_f) és de 24 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitja d'accés a memòria (t_m) ?

$$t_m = T_f \times t_f + (1 - T_f) \times t_e = 0,3 \times 24 \text{ ns} + 0,7 \times 4 \text{ ns} = 7,2 \text{ ns} + 2,8 \text{ ns} = 10 \text{ ns}$$

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	21/01/2012	09:00

Apartat 3.2 (15%)

Una RSI amb repertori CISCA transfereix dades des del disc a la memòria del processador:

- Temps de latència mitjà del disc de 3 ms ($t_{\text{latència}}$)
- Velocitat de transferència del disc és de 20 MB/s (v_{transf})
- La RSI executa 10 instruccions, cadascuna amb un temps de 3 cicles de rellotge
- La CPU requereix 8 cicles de rellotge addicionals des de que es detecta la interrupció del disc fins que es transfereix el control a la RSI, ($t_{\text{rec_int}}$).
- Temps de programació i finalització de la transferència de 20 ns ($t_{\text{prog}} + t_{\text{final}}$)
- Transferència de $N_{\text{dades}} = 4.000$ dades de 4 Bytes

Fins a quina freqüència de rellotge mínima al processador es pot garantir que no es perdin dades en la transferència?

En la fase de transferència de dades, el disc proporciona 20 MB/s, és a dir, 20 Bytes cada us, o 4 bytes cada 200 ns (per tant, tenim una interrupció cada 200 ns). Aquest és el temps màxim que hauria de tardar la gestió de la interrupció, incloent el temps addicional per transferir el control a la RSI.

El temps consumit per la CPU en cada interrupció és $8 + 3 \times 10 = 38$ cicles de rellotge (transferir control a RSI + executar RSI).

$$200 \text{ ns} / 38 \text{ cicles de rellotge} = 5,26 \text{ ns / cicle de rellotge (és el temps màxim)}$$

Per tant, la freqüència mínima per no perdre dades seria de:

$$1 \text{ cicle de rellotge} / 5,26 \text{ ns} = 190 \text{ milions de cicles / segon} = \mathbf{190 \text{ MHz}}$$

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	21/01/2012	09:00

Pregunta 3.3 (10%)

3.3.1 (5%)

El temps de latència mitjà d'un disc, $t_{\text{latència}}$, és de 3 ms i la velocitat de transferència, v_{transf} , és de 40 MB/s. Si es vol llegir del disc un bloc consecutiu de 1 MB, N_{bytes} , quin és el temps de transferència de tot el bloc, t_{bloc} ?

$$t_{\text{dades}} = 1 \text{ MB } (N_{\text{bytes}}) / 40 \text{ MBytes/s } (v_{\text{transf}}) = 0,025 \text{ segons} = 25 \text{ ms.}$$

$$t_{\text{bloc}} = t_{\text{latència}} + t_{\text{dades}} = 3 \text{ ms} + 25 \text{ ms} = 28 \text{ ms.}$$

3.3.2 (5%)

Un programa fet amb el repertori CISCA envia un bloc de 8.000 dades de 4 bytes cada dada, des de la memòria al disc mitjançant la tècnica d'E/S programada. Cadascuna de les instruccions triga 4 cicles en executar-se, i el microprocessador funciona amb un rellotge de 2 GHz. La velocitat de transferència del disc és de 100.000 bytes per segon.

Quant temps dura la transferència i quin percentatge d'aquest temps dedica la CPU a la gestió de la transferència?

$$\text{S'escrueix un total de } 4 \times 8.000 = 32.000 \text{ Bytes, amb un ritme de transferència de } 100.000 \text{ Bytes/s} \rightarrow 32.000 / 100.000 = 320 \text{ ms}$$

La CPU està completament dedicada a la transferència (100%) i no fa altra cosa durant el temps en que es produeix la transferència amb el disc.