

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	23/06/2012	09:00

05.573 23 06 12 EX
05.573 23 06 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?
- 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? Quant?
- Indicacions específiques per a la realització d'aquest examen:

Enunciats

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	23/06/2012	09:00

Pregunta 1 (20%: 1.1: 10%, 1.2: 10%)

Només s'han de completar les instruccions marcades.

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.

Pregunta 1.1 (10%)

Completa el codi de la subrutina següent:

```
; Fer la derivada d'un polinomi. r(x)= p'(x).
; Per cada terme farem: nou coeficient=coeficient*exponent, nou exponent=exponent-1
; Esborrar el polinomi r(x) abans de fer la derivada cridant la subrutina p_delete.
; Paràmetres d'entrada : Cap
; Paràmetres de sortida: Cap
p_derive:
...
mov eax, r_coef
mov ebx, r_exp
call p_delete           ;esborrem el polinomi r(x).

mov esi, 0

p_derive_loop:
mov ecx, [p_exp+esi*4] ;Agafem l'exponent del polinomi.
cmp ecx, -1             ;Mirem si ja hem processat tots els termes del polinomi.
je p_derive_end

mov eax, [p_coef+esi*4] ;Agafem el coeficient del polinomi.
mul ecx                 ;Multipliquem [p_coef+esi*4] per [p_exp+esi*4]
dec ecx                 ;Decrementem l'exponent.
mov [r_coef+esi*4],eax  ;posem el nou coeficient a p
mov [r_exp+esi*4],ecx   ;posem el nou exponent a p
inc esi

cmp esi, 20
jge p_derive_end

jmp p_derive_loop

p_derive_end:
...
ret
```

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	23/06/2012	09:00

Pregunta 1.2 (10%)

Completa el codi de la subrutina següent:

```
; Sumar dos polinomis. r(x)=p(x)+q(x)
; Esborrar el polinomi r(x) abans de fer la suma cridant la subrutina p_delete.
; Inserir el termes de p(x) i de q(x) a r(x) segons l'exponent, de més gran a més petit.
; S'ha d'anar agafant termes de cada polinomi
; i inserir el terme que tingui l'exponent més gran.
; Si els dos termes tenen el mateix exponent s'han de sumar els coeficients abans d'inserir.
; Si la suma dels coeficients dóna 0, no inserim el terme al polinomi r(x).
; Si al fer la suma r(x) té més de 20 termes, esborrar r(x) i retornar -1.
; Paràmetres d'entrada : Cap
; Paràmetres de sortida: rax (0 si s'ha pogut fer la suma, -1 si no s'ha pogut fer la suma.)
p_add:
...
    p_add_loop:
        ...
        jmp p_add_loop

p_add_delete:
    mov eax, r_coef
    mov ebx, r_exp
    call p_delete                ;esborrem el polinomi r(x): r_coef, r_exp.
                                ;cridant la subrutina p_delete

    mov  eax, -1                ;no s'ha pogut fer la suma.
    jmp  p_add_end

p_add_ok:
    mov  eax, 0                  ;si s'ha pogut fer la suma

p_add_end:
    ...
    ret

; Esborrar els termes d'un vector de 20 posicions. El coeficient a 0 i l'exponent a -1.
; Paràmetres d'entrada : vec_coeficients(rax), vec_exponents(rbx)
; Paràmetres de sortida: Cap
p_delete:
    ... (Aquesta subrutina no s'ha de fer, és la rutina que es crida des de p_add).
    ret
```

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	23/06/2012	09:00

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

Apartat 2.1 (15%)

Suposeu el següent estat inicial de la CISCA (abans de cada apartat):

- Registres: $R_i = 2 * i$ per a $i = 0, 1, \dots, 15$.
- Memòria: $M(i) = (800h + i) \bmod 2^{32}$ per a $i = 0, 4, 8, \dots, 2^{32}-4$.

Completeu l'estat del computador després d'executar cada codi (indiqueu els valors dels registres en hexadecimal).

Suposeu que l'adreça simbòlica **V** val **600h**.

a)	b)
<pre>MOV R3, V ADD R7, R6 MOV [R3+4F0h], R7</pre>	<pre>SUB R5, R6 JLE S MOV R7, 1 JMP F S: MOV R7, 0 F:</pre>
<p>R3 = 600h</p> <p>R7 = 1Ah</p> <p>M(AF0h) = 1Ah</p> <p>Z = 0, C = 0, S = 0, V = 0</p>	<p>R5 = FFFFFFFEh</p> <p>R7 = 0h</p> <p>Z = 0, C = 1, S = 1, V = 0</p>

Apartat 2.2 (15%)

Completeu els buits en el fragment de codi CISCA perquè sigui una traducció eficient a l'assemblador que realitza la següent inicialització de la matriu M de 10 per 10 nombres enters (10 files per 10 columnes):

```
i = 0;
do {
M[i][9]=M[9][i];
i = i+1;
}while (i<10);
```

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 $M[5][4]$ es troben emmagatzemats en les adreces de memòria M, M+4, M+40 i M+216 respectivament.

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

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	23/06/2012	09:00

```

PUSH R1
PUSH R2
PUSH R3
MOV R1, 36
MOV R2, 360
DO: MOV R3, [M+R2]
MOV [M + R1], R3
ADD R1, 40
ADD R2, 4
CMP R2, 396
JLE DO
POP R3
POP R2
POP R1

```

Apartat 2.3 (10%)

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

```

                MUL   R3, [M+R5]
                DEC   R5
end_w:         SUB   [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 00002C0h (que és el valor del PC en l'estat inicial). Supposeu també que l'adreça simbòlica M val 00038A40h. 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<7..4>	B0<3..0>	Direccionament operands
MUL	2h	2h	Veure taula de modes
DEC	2h	5h	Veure taula de modes
SUB	2h	1h	Veure taula de modes

Taula de modes de direccionament (Bk<7..4>)

Camp mode Bk<7..4>	Mode
0h	Immediat

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	23/06/2012	09:00

1h	Registre
2h	Memòria
3h	Indirecte
4h	Relatiu
5h	Indexat

Taula de modes de direccionament (Bk<3..0>)

Camp mode Bk<3..0>	Significat
Nº registre	Si el mode especifica registre.
0	No s'especifica registre

@	Assemblador	Bk para k=0..10										
		0	1	2	3	4	5	6	7	8	9	10
000002C0h	MUL R3,[M+R5]	22	13	55	40	8A	03	00				
000002C7h	DEC R5	25	15									
000002C9h	SUB [M], R3	21	20	40	8A	03	00	13				
000002D0h												

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

Pregunta 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, és 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).

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.

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	23/06/2012	09:00

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, 21, 22, 23, 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:21	Fallada:12
Línia 0	0, 1, 2, 3		32, 33, 34, 35		
Línia 1	4, 5, 6, 7			20, 21, 22, 23	
Línia 2	8, 9, 10, 11				
Línia 3	12, 13, 14, 15	28, 29, 30, 31			12, 13, 14, 15

	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 5 ns i el temps total d'accés en cas de fallada (t_f) és de 25 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitja d'accés a memòria (t_m) ?

$$t_m = T_f \times t_f + (1 - T_f) \times t_e = 0,25 * 25 \text{ ns} + 0,75 * 5 \text{ ns} = 6,25 \text{ ns} + 3,75 \text{ ns} = 10 \text{ ns}$$

Apartat 3.1.2

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	23/06/2012	09:00

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, 21, 22, 23, 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 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 nou estat de la cau.

	Estat Inicial	Fallada:30	Fallada:32	Fallada:21	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			20, 21, 22, 23	
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 5 ns i el temps total d'accés en cas de fallada (t_f) és de 25 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitjà d'accés a memòria (t_m) ?

$$t_m = T_f \times t_f + (1 - T_f) \times t_e = 0,3 \times 25 \text{ ns} + 0,7 \times 5 \text{ ns} = 7,5 \text{ ns} + 3,5 \text{ ns} = 11 \text{ ns}$$

Pregunta 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 2 ms ($t_{\text{latència}}$)
- Velocitat de transferència del disc és de 200 MB/s (v_{transf})

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	23/06/2012	09:00

- La RSI executa 18 instruccions, cadascuna amb un temps de 5 cicles de rellotge
- La CPU requereix 10 cicles de rellotge addicionals des de que es detecta la interrupció del disc fins que es transfereix el control a la RSI, (t_{rec_int}).
- Temps de programació i finalització de la transferència de 30 ns ($t_{prog} + t_{final}$)
- Transferència de $N_{dades} = 10.000$ 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 200 MB/s, és a dir, 200 Bytes cada us, o 4 bytes cada 20 ns (per tant, tenim una interrupció cada 20 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 $10 + 5 \times 18 = 100$ cicles de rellotge (transferir control a RSI + executar RSI).

$20 \text{ ns} / 100 \text{ cicles de rellotge} = 0,20 \text{ ns} / \text{cicle de rellotge}$ (és el temps màxim)

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

$1 \text{ cicle de rellotge} / 0,20 \text{ ns} = 5.000 \text{ milions de cicles / segon} = 5 \text{ GHz}$

Pregunta 3.3 (10%)

3.3.1 (5%)

El temps de latència mitjà d'un disc, $t_{latència}$, és de 2 ms i la velocitat de transferència, v_{transf} , és de 200 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} ?

$t_{dades} = 1 \text{ MB} (N_{bytes}) / 200 \text{ MBytes/s} (v_{transf}) = 0,005 \text{ segons} = 5 \text{ ms}$.

$t_{bloc} = t_{latència} + t_{dades} = 2 \text{ ms} + 5 \text{ ms} = 7 \text{ ms}$.

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 MBytes per segon.

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

S'escriuen un total de $4 \times 8.000 = 32.000$ Bytes, amb un ritme de transferència de 100.000 Bytes / s $\rightarrow 32.000 / 100.000.000 = 0,320 \text{ us}$

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.

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	23/06/2012	09:00