

## Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	14/01/2012	12:00

05.573 14 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	14/01/2012	12:00

### 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 corresponent a la subrutina m\_add (suma de matrius,  $mr=m1+m2$ ).

```
m_add:
    push rbp
    mov rbp, rsp

    mov esi, 0
m_add_bucle:
    mov eax, [m1+esi*4]_
    add eax, [m2+esi*4]_
    mov [mr+esi*4], eax_
    inc esi
    cmp esi, 8
    jle m_add_bucle

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

## Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	14/01/2012	12:00

### Pregunta 1.2 (10%)

Completa el codi corresponent a la funció de C `m_cramer` (resoldre el sistema d'equacions  $Ax=b$  per Cramer).

```
void m_cramer(){
    int detA; //determinant de la matriu A
    int i;

    m_copy(A, mr);
    m_det_mr();
    detA=det;
    if ( _detA==0 ) {
        printf("El sistema no té solució\n");
    } else {
        printf("Solució del sistema\n");
        for (i=0;i<3;i++){
            col=i;
            m_copy_col();
            m_det_mr();
            //Mostrar el valor de les variables x1, x2, i x3
            printf("x%d=%d/%d = %.2f\t\n",i,det,detA,(float)
det/detA );
            m_copy(A, mr);
        }
        printf("\n");
    }
}
```

## Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	14/01/2012	12:00

### 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) = 00000000h	

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

a)
<pre>MOV  R2, 00001000h ADD  R2, [00001000h+R2] XOR  R1, R2</pre>
<pre>R2 = 00001000h R1 = 00000000h Z = 1,  S = 0 C = 0,  V = 0</pre>

b)
<pre>MOV  R1, 00001000h SUB  R1, [00000000h] ADD  [R1], R1</pre>
<pre>R1 = 0 M(00000000h) = 00001000h Z = 0,  S = 0 C = 0,  V = 0</pre>

## Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	14/01/2012	12:00

### Apartat 2.2 (15%)

En la memòria d'un computador CISCA tenim emmagatzemada una matriu de 10 x 100 elements (10 files per 100 columnes) a partir de l'adreça simbòlica M. Cada element és un nombre enter codificat en complement a 2 amb 32 bits.

Completeu els buits del fragment de codi CISCA per posar a 0 l'element  $M[i][j]$ , la qual cosa en C s'especificaria amb la sentència:

$M[i][j] = 0;$

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  $M[7][40]$  es troben emmagatzemats en les adreces de memòria M, M+4, M+400 i M+2960 respectivament.

Se sap que en R1 es troba emmagatzemat el valor de la variable i, i en R2 el de la j i que després d'executar-se el fragment de codi tots els registres han de mantenir els valors originals. El codi és correcte però no és tot l'eficient que podria ser.

```
PUSH R1
PUSH R2
MUL R1, 100
ADD R1, R2
MUL R1, 4; o també SLH R1, 2
MOV [M+R1], 0
POP R2
POP R1
```

## Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	14/01/2012	12:00

### Apartat 2.2 (10%)

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

```
MOV R3, A
MOV R3, [R3]
CMP R3, [B]
```

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 00FF0000h (que és el valor del PC abans de començar l'execució del fragment de codi). Supposeu també que les adreces simbòliques A i B valen 00001000h i 0000F000h 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
CMP	26h

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

Camp mode Bk<7..4>	Mode
0h	Immediat
1h	Registre
2h	Memòria
3h	Indirecte
4h	Relatiu
5h	Indexat
6h	A PC

## Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	14/01/2012	12:00

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
00FF0000	MOV R3, A	10	13	00	00	10	00	00				
00FF0007	MOV R3, [R3]	10	13	33								
00FF000A	CMP R3, [B]	26	13	20	00	F0	00	00				
00FF0011												

## Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	14/01/2012	12:00

### 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, 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	14/01/2012	12:00

### 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, 8, 9, 10, 11, 12, 30, 31, 32, 1, 2, 3, 8, 9, 10, 11, 12, 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, 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 2 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 ( $t_m$ ) ?

$$t_m = T_f \times t_f + (1 - T_f) \times t_e = 0,25 \times 24 \text{ ns} + 0,75 \times 2 \text{ ns} = 6 \text{ ns} + 1,5 \text{ ns} = 7,5 \text{ ns}$$

## Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	14/01/2012	12:00

### 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, 8, 9, 10, 11, 12, 30, 31, 32, 1, 2, 3, 8, 9, 10, 11, 12, 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:8
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 2 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 ( $t_m$ ) ?

$$t_m = T_f \times t_f + (1 - T_f) \times t_e = 0,3 \times 24 \text{ ns} + 0,7 \times 2 \text{ ns} = 7,2 \text{ ns} + 1,4 \text{ ns} = 8,6 \text{ ns}$$

## Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	14/01/2012	12:00

### 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 3 ms (  $t_{latència}$  )
- Velocitat de transferència del disc és de 20 MB/s (  $v_{transf}$  )
- Processador amb una freqüència de rellotge de 2 GHz
- 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, (  $t_{rec\_int}$  ).
- Temps de programació i finalització de la transferència de 20 ns (  $t_{prog} + t_{final}$  )
- Transferència de  $N_{dades} = 4.000$  dades de 4 Bytes

Quin és el temps total que dedica la CPU a la tasca d'Entrada/Sortida,  $t_{cpu}$  ?

temps de cicle de rellotge,  $t_{clock}$ ,  $1 / 2 \text{ GHz} = 0,5 \text{ nanosegons}$

temps per a executar una instrucció,  $t_{instr}$ ,  $3 \times 0,5 = 1,5 \text{ nanosegons.}$

Temps per atendre la interrupció,  $t_{rec\_int}$ :  $8 \text{ cicles} \times 0,5 \text{ ns} = 4 \text{ ns}$

Temps d'execució RSI,  $t_{rsi}$ :  $N_{rsi} \times t_{instr} = 10 \text{ instr.} \times 1,5 \text{ ns} = 15 \text{ ns}$

Temps consumit per CPU en cada interrupció,  $t_{transf\_dada}$ :

$$t_{transf\_dada} = t_{rec\_int} + t_{rsi} = 4 + 15 = 19 \text{ ns}$$

Nombre d'interrupcions produïdes (o nombre total de dades,  $N_{dades}$  ): 4.000 interrupcions

Temps consumit en total en TOTES les interrupcions:

$$t_{transf\_bloc} = t_{transf\_dada} \times N_{dades} = 19 \text{ ns} \times 4.000 \text{ interrupcions} = 76 \text{ us} \text{ (microsegons)}$$

$$t_{cpu} = ( t_{prog} + t_{final} ) + t_{transf\_bloc} = 20 \text{ ns} + 76 \text{ us} = 76,02 \text{ us.}$$

## Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	14/01/2012	12:00

### Pregunta 3.3 (10%)

#### 3.3.1 (5%)

El següent codi CISCA transfereix dades entre memòria i disc fent servir la tècnica d'E/S programada.

```

1.      MOV    R1, 200
2.      MOV    R2, DADES
3.      MOV    R3, 10000
4. Bucle:  IN    R4, [R3]
5.      AND    R4, 4
6.      JE     Bucle
7.      MOV    R0, [R2]
8.      OUT    R0, [R3+4]
9.      ADD    R2, 4
10.     DEC    R1
11.     JNE    Bucle
  
```

Indicar si la transferència és de sortida al disc o d'entrada des del disc, quantes dades es transfereixen en total entre la memòria i el disc, i quin bit del registre d'estat del controlador de disc verifica que es pot realitzar la transferència d'una dada.

Transferència de sortida de **200 dades** ( $200 \times 4 = 800$  Bytes).

El **bit 2** (o el tercer bit menys significatiu).

#### 3.3.2 (5%)

El sistema d'E/S per DMA d'un computador té les següents característiques:

- El **temps de cessió** i el **temps de recuperació** del bus (  $t_{\text{cessió}} + t_{\text{recup}}$  ) són 4 ns.
- El **temps de la transferència** pel bus (  $t_{\text{mem}}$  ) és de 2 ns.
- S'envien pel bus 2.000 dades de 32 bits cadascuna
- La memòria intermèdia fa servir ràfegues de 8 dades.

Calcular el temps total d'ocupació del bus per part del controlador de disc/DMA per a dur a terme la transferència fent servir ràfegues.

Temps ocupació Bus,  $t_{\text{transf\_dada}}$ :

Nombre de peticions del Bus,  $N_{\text{pet\_bus}}$ :

Temps total d'ocupació del Bus  $t_{\text{transf\_bloc}}$ :

$$t_{\text{cessió}} + 8 \times t_{\text{mem}} + t_{\text{recup}} = 4 + 8 \times 2 = 20 \text{ ns}$$

$$N_{\text{dades}} / N_{\text{ràfega}} = 2.000 / 8 = 250$$

$$t_{\text{transf\_dada}} \times N_{\text{pet\_bus}} = 20 \text{ ns} \times 250 = 5 \text{ us}$$