



# PAC2

## Estructura de computadores 2015 s1

Estudis d'informàtica, multimèdia i comunicació

## Presentació

La present PAC2 conté 2 preguntes amb diversos apartats cadascuna d'elles i representa el 50% de la nota de l'avaluació contínua.

Com podreu veure, els exercicis són molt semblats als quals heu fet durant aquests dies, en els quals a més heu pogut donar les solucions, comentar-les i plantejar dubtes en el fòrum. Aquesta PAC és **individual**, **avaluable** i per tant no pot comentar-se.

## Competències

Les competències específiques que persegueix la PAC2 són:

- [13] Capacitat per identificar els elements de l'estructura i els principis de funcionament d'un ordinador.
- [14] Capacitat per analitzar l'arquitectura i organització dels sistemes i aplicacions informàtics en xarxa.
- [15] Conèixer les tecnologies de comunicacions actuals i emergents i saber-les aplicar convenientment per dissenyar i desenvolupar solucions basades en sistemes i tecnologies de la informació.

## Objectius

Els objectius de la següent PAC són:

- Conèixer la organització del sistema de memòria d'un computador.
- Conèixer el funcionament de la memòria cau, així com dels algorismes de correspondència i reemplaçament.
- Conèixer la organització del sistema d'entrada/sortida.
- Comprendre les tècniques d'entrada/sortida (entrada/sortida programada, Interrupcions i DMA).

## Enunciat

Respondre cada pregunta o apartat en el requadre corresponent.

## Recursos

Podeu consultar els recursos disponibles a l'aula, però no fer ús del fòrum.

## Criteris de valoració

La **puntuació** de cada pregunta i els **criteris d'avaluació** els trobareu a cada pregunta.

## Format i data de lliurament

La PAC2 podeu lliurar-la a l'apartat de **Lliurament d'activitats** amb el nom **cognom1\_cognom2\_nom\_PAC2 (pdf / odt / doc / docx )**.

La data límit de lliurament és el **11/12/2015**.



## Enunciat

### Pregunta 1 (4 punts)

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 en una adreça múltiple de 8. Així, el bloc 0 conté les adreces 0, 1, 2, 3, 4, 5, 6, 7, el bloc 1, les adreces 8, 9, 10, 11, 12, 13, 14, 15, i el bloc N les adreces  $8*N$ ,  $8*N+1$ ,  $8*N+2$ ,  $8*N+3$ ,  $8*N+4$ ,  $8*N+5$ ,  $8*N+6$ ,  $8*N+7$ .

Una fórmula per calcular el identificador numèric del bloc és la següent:

Bloc = adreça de memòria (adreça paraula) DIV 8 (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, 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, 7).

#### Pregunta 1.1 (2 punts) Memòria cau d'assignació directa

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):

Línia = identificador de bloc MOD 4 (mida de la cau en línies)

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

1, 2, 3, 4, 8, 24, 15, 16, 2, 3, 54, 55, 56, 42, 12, 43, 48, 58, 31, 32

##### 1.1.a) (1,2 punts)

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 accés cal omplir una columna indicant si es tracta d'un encert o una fallada.

Si és un encert escriurem E en la línia corresponent davant de les adreces del bloc, si és una fallada escriurem F i en aquest cas s'indicarà el nou bloc que es porta a la memòria cau en la línia que li correspongui, expressat de la forma b:e ( $a_0 - a_7$ ) on b: número de bloc, e: etiqueta i ( $a_0 - a_7$ ) són les adreces del bloc, on  $a_0$  és la primera adreça del bloc i  $a_7$  és la vuitena (darrera) adreça del bloc.

Línia	Estat Inicial	1	2	3	4	8
0	0:0 (0 - 7)	E 0:0 (0 - 7)	E 0:0 (0 - 7)	E 0:0 (0 - 7)	E 0:0 (0 - 7)	0:0 (0 - 7)
1	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	E 1:0 (8 - 15)
2	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)
3	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)

Línia	24	15	16	2	3	54
0	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	E 0:0 (0 - 7)	E 0:0 (0 - 7)	0:0 (0 - 7)
1	1:0 (8 - 15)	E 1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)
2	2:0 (16 - 23)	2:0 (16 - 23)	E 2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	F 6:1 (48 - 55)
3	E 3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)

Línia	55	56	42	12	43	48
0	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)
1	1:0 (8 - 15)	1:0 (8 - 15)	F 5:1 (40 - 47)	F 1:0 (8 - 15)	F 5:1 (40 - 47)	5:1 (40 - 47)
2	E 6:1 (48 - 55)	6:1 (48 - 55)	6:1 (48 - 55)	6:1 (48 - 55)	6:1 (48 - 55)	E 6:1 (48 - 55)
3	3:0 (24 - 31)	F 7:1 (56 - 63)	7:1 (56 - 63)	7:1 (56 - 63)	7:1 (56 - 63)	7:1 (56 - 63)

Línia	58	31	32			
0	0:0 (0 - 7)	0:0 (0 - 7)	F 4:1 (32 - 39)			
1	5:1 (40 - 47)	5:1 (40 - 47)	5:1 (40 - 47)			
2	6:1 (48 - 55)	6:1 (48 - 55)	6:1 (48 - 55)			
3	E 7:1 (56 - 63)	F 3:0 (24 - 31)	3:0 (24 - 31)			

### 1.1.b) (0,4 punts)

Quina és la taxa de fallades ( $T_f$ ) ?

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

### 1.1.c) (0,4 punts)

Suposem que el temps d'accés a la memòria cau, o temps d'accés en cas d'encert ( $t_e$ ), és de 3 ns i el temps total d'accés en cas de fallada ( $t_f$ ) és de 30 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitjà d'accés a memòria ( $t_m$ ) ?

$$t_m = T_f \times t_f + (1 - T_f) \times t_e = 0,35 \times 30 \text{ ns} + 0,65 \times 3 \text{ ns} = 10,5 \text{ ns} + 1,95 \text{ ns} = 12,45 \text{ ns}$$

## Pregunta 1.2 (2 punts) 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:

1, 2, 3, 4, 8, 24, 15, 16, 2, 3, 54, 55, 56, 42, 12, 43, 48, 58, 31, 32

### 1.2.a) (1,2 punts)

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 accés cal omplir una columna indicant si es tracta d'un encert o una fallada.

Si és un encert escriurem E en la línia corresponent davant de les adreces del bloc, si és una fallada escriurem F i s'indicarà el nou bloc que es porta a la memòria cau en la línia que li correspongui, expressat de la forma b ( $a_0 - a_7$ ) on b: número de bloc, i ( $a_0 - a_7$ ) són les adreces del bloc, on  $a_0$  és la primera adreça del bloc i  $a_7$  és la vuitena (darrera) adreça del bloc.

Línia	Estat Inicial	1	2	3	4	8
0	0 (0 - 7)	E 0 (0 - 7)	E 0 (0 - 7)	E 0 (0 - 7)	E 0 (0 - 7)	0 (0 - 7)
1	1 (8 - 15)	1 (8 - 15)	1 (8 - 15)	1 (8 - 15)	1 (8 - 15)	E 1 (8 - 15)
2	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)
3	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)

Línia	24	15	16	2	3	54
0	0 (0 - 7)	0 (0 - 7)	0 (0 - 7)	E 0 (0 - 7)	E 0 (0 - 7)	0 (0 - 7)
1	1 (8 - 15)	E 1 (8 - 15)	1 (8 - 15)	1 (8 - 15)	1 (8 - 15)	1 (8 - 15)
2	2 (16 - 23)	2 (16 - 23)	E 2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)
3	E 3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	F 6 (48 - 55)

Línia	55	56	42	12	43	48
0	0 (0 - 7)	0 (0 - 7)	0 (0 - 7)	F 1 (8 - 15)	1 (8 - 15)	1 (8 - 15)
1	1 (8 - 15)	F 7 (56 - 63)	7 (56 - 63)	7 (56 - 63)	7 (56 - 63)	7 (56 - 63)
2	2 (16 - 23)	2 (16 - 23)	F 5 (40-47)	5 (40-47)	E 5 (40-47)	5 (40-47)
3	E 6 (48 - 55)	6 (48 - 55)	6 (48 - 55)	6 (48 - 55)	6 (48 - 55)	E 6 (48 - 55)

Línia	58	31	32			
0	1 (8 - 15)	F 3 (24 - 31)	3 (24 - 31)			
1	E 7 (56 - 63)	7 (56 - 63)	7 (56 - 63)			
2	5 (40-47)	5 (40-47)	F 4 (32-39)			
3	6 (48 - 55)	6 (48 - 55)	6 (48 - 55)			

### 1.2.b) (0,4 punts)

Quina és la taxa de fallades ( $T_f$ ) ?

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

**1.2.c) (0,4 punts)**

Suposem que el temps d'accés a la memòria cau, o temps d'accés en cas d'encert ( $t_e$ ), és de 3 ns i el temps total d'accés en cas de fallada ( $t_f$ ) és de 30 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitja d'accés a memòria ( $t_m$ ) ?

$$t_m = T_f \times t_f + (1 - T_f) \times t_e = 0,3 \times 30 \text{ ns} + 0,7 \times 3 \text{ ns} = 9 \text{ ns} + 2,1 \text{ ns} = 11,1 \text{ ns}$$

**Criteris de valoració.** Pels apartats 1.1.a i 1.2.a cada errada en les fallades o encerts de la memòria cau o en la col·locació d'un bloc a la cau resta 0,6. Els apartats restants es puntuaran amb els 0,4 punts cadascun d'ells si la solució és correcte i coherent amb la vostra resposta als apartats a) corresponents.



## Pregunta 2 (6 punts)

Es vol analitzar el rendiment de la comunicació de dades entre la memòria d'un processador i un port USB, que tenen les següents característiques:

- Velocitat de transferència del dispositiu d'E/S ( $v_{\text{transf}}$ ) = 500 KB/s
- Temps de latència mitjà del dispositiu ( $t_{\text{latència}}$ ) = 0
- Adreces dels **registres de dades i d'estat** del controlador d'E/S: 0E00h i 0E04h
- 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)
- Processador amb una freqüència de rellotge de 2 GHz, el processador pot executar 2 instruccions per cicle de rellotge ( $t_{\text{instr}} = t_{\text{cicle}} / 2$ )
- Temps de programació i finalització de la transferència de 200 ns ( $t_{\text{prog}} + t_{\text{final}}$ )
- Transferència de **escriptura** des de memòria al port d'E/S
- Transferència de  $N_{\text{dades}} = 400.000$  dades, es a dir,  $400.000 * 4 \text{ Bytes} = 1.600.000$  Bytes
- Adreça inicial de memòria on resideixen les dades: 50000000h

### Apartat 2.1 (2 punts) E/S programada

El següent codi realitzat amb el repertori CISCA realitza la transferència descrita abans mitjançant la tècnica d'E/S programada.

```

1.      MOV  R3, VALOR1
2.      MOV  R2, VALOR2
3. Bucle: IN   R0, [VALOR3]      ; llegir 4 bytes
4.      AND  R0, VALOR4
5.      JE   Bucle
6.      MOV  R0, [R2]           ; llegir 4 bytes
7.      ADD  R2, 4
8.      OUT  [VALOR5], R0      ; escriure 4 bytes
9.      SUB  R3, 1
10.     JNE  Bucle

```

#### 2.1.a) (0,5 punts)

Substituir pels valors adients:

**VALOR1**= 400000

**VALOR2**= 50000000h

**VALOR3**= E04h

**VALOR4**= 00100000b= 20h = 32d

**VALOR5**= E00h

### 2.1.b) (0,5 punts)

Quant temps dura la transferència  $t_{\text{transf\_bloc}}$ ?

Quin percentatge d'aquest temps dedica la CPU a la transferència?

$$t_{\text{transf\_bloc}} = t_{\text{latència}} + (N_{\text{dades}} * t_{\text{transf\_dada}})$$

$$t_{\text{latència}} = 0$$

$$N_{\text{dades}} = 400000$$

$$t_{\text{transf\_dada}} = m_{\text{dada}} / v_{\text{transf}} = 4 \text{ Bytes} / 500 \text{ Kbytes/s} = 0,008 \text{ ms}$$

$$t_{\text{transf\_bloc}} = 0 + (400000 * 0,008 \text{ ms}) = 3200 \text{ ms} = 3,2 \text{ s}$$

La CPU dedica el 100% del temps i per tant, el temps coincideix amb el temps dedicat pel perifèric  $t_{\text{bloc}}$

### 2.1.c) (1 punt)

Si volguéssim fer servir el mateix processador i el mateix programa però amb un dispositiu d'E/S més ràpid, quina és la màxima taxa o velocitat de transferència del nou dispositiu que es podria suportar sense que el dispositiu s'hagués d'esperar?

Freqüència de rellotge = 2 GHz, implica un temps de cicle de

$$t_{\text{cicle}} = 1 / 2 * 10^9 = 0,5 \text{ ns (nanosegons)}$$

$$t_{\text{instr}} = 0,5 \text{ ns} / 2 = 0,250 \text{ ns}$$

El mínim nombre d'instruccions que ha d'executar el programa per a cada dada transferida són les 8 instruccions: 3, 4, 5, 6, 7, 8, 9 i 10. Executar les 8 instruccions requereix  $8 * t_{\text{instr}} = 8 * 0,250 \text{ ns} = 2 \text{ ns}$

Per tant, el temps mínim per a transferir una dada és: 2 ns

Es poden transferir 4 bytes cada 2 ns, és a dir:  $4 / 2 * 10^{-9} = 2000 \text{ Mbyte/s} = 2 \text{ Gbytes/s}$





## Apartat 2.2 (2 punts) E/S per Interrupcions

Suposeu que el següent codi CISCÀ 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:

- El temps per atendre la interrupció ( $t_{rec\_int}$ ), o temps addicional des de que la CPU detecta la interrupció fins que es comença a executar la primera instrucció de la RSI és de 3 cicles de rellotge.
- 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.  IN    R0, [VALOR1] ; llegir 4 bytes
5.  AND   R0, VALOR2
6.  JE    Error ;salta a un codi de tractament de l'error no
                ;descrit, s'ha produït la petició per part del
                ;dispositiu però la dada no està disponible

7.  MOV   R1, [VALOR3]
8.  MOV   R0, [R1]
9.  OUT   [VALOR4],R0 ; escriure 4 bytes
10. ADD   R1, 4
11. MOV   [VALOR3], R1
12. POP   R1
13. POP   R0
14. STI
15. RETI

```

### 2.2.a) (0,5 punts)

Substituir pels valors adients:

VALOR1= E04h

VALOR2= 20h = 32d

VALOR3= Dir

VALOR4= E00h

### 2.2.b) (1 punt)

Quin és el temps total que dedica la CPU a la tasca d'Entrada/Sortida,  $t_{cpu}$ ? Quin percentatge del temps representa el tems de transferència del bloc  $t_{transf\_bloc}$  respecte al temps de transferència del bloc per part del perifèric  $t_{bloc}$  ?

El temps d'un cicle,  $t_{cicle} = 0,5 \text{ ns}$  (nanosegons)

Temps per atendre la interrupció,  $t_{rec\_int}$ :  $3 \text{ cicles} * 0,5 \text{ ns} = 1,5 \text{ ns}$

Temps d'execució de una instrucció,  $t_{instr}$ :  $t_{cicle} / 2 = 0,250 \text{ ns}$

Temps d'execució RSI,  $t_{rsi}$ :  $N_{rsi} * t_{instr} = 15 \text{ instr.} * 0,250 \text{ ns} = 3,75 \text{ ns}$

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

$$t_{transf\_dada} = t_{rec\_int} + t_{rsi} = 1,5 + 3,75 = 5,25 \text{ ns}$$

Nombre d'interrupcions produïdes (nombre total de dades,  $N_{dades}$ ): 400.000 interrupcions

Temps consumit en total en TOTES les interrupcions:

$$t_{transf\_bloc} = t_{transf\_dada} * N_{dades} = 5,25 \text{ ns} * 400.000 \text{ interrupcions} = 2100000 \text{ ns} = 2,1 \text{ ms (milisegons)}$$

El temps final d'ocupació de la CPU cal que inclogui el temps de programació i finalització de la transferència:

$$t_{cpu} = (t_{prog} + t_{final}) + t_{transf\_bloc} = 200 \text{ ns} + 2100000 \text{ ns} = 2100200 \text{ ns} = 2,1002 \text{ ms}$$

Dels  $3,2 \text{ s} = 3200 \text{ ms}$  de temps total per a realitzar la transferència (temps calculat a l'apartat 2.1 b), la CPU està dedicada a la tasca de E/S:

$$\%ocupació = t_{transf\_bloc} * 100 / t_{bloc} = 2,1 * 100 / 3200 \Rightarrow 0,065625 \% \text{ del temps.}$$

### 2.2c) (0,5 punts)

Si volguéssim reduir la freqüència de rellotge del processador per a reduir el seu consum energètic, fins a quina freqüència ho podríem fer sense reduir la velocitat de transferència amb el dispositiu d'E/S?

En la fase de transferència de dades, el controlador d'E/S genera 400.000 interrupcions durant 3,2 segons (dada calculada a l'apartat 2.1 b):

$$N_{dades} * t_{dada} = 3,2 \text{ s} = 3200 \text{ ms} = 3200000 \text{ us (microsegons)}$$

Es a dir, tenim una interrupció cada  $3200000 / 400000 = 8 \text{ us}$  (microsegons). 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 que pot consumir la CPU en una interrupció és

$$t_{transf\_dada}$$

El temps consumit per la CPU en cada interrupció és, com hem vist a l'apartat anterior, la suma del temps de transferir el control a la RSI + executar la RSI:

A l'enunciat es defineix que  $t_{rec\_int} = 3 \text{ cicles de rellotge} = 3 * t_{cicle}$  i per tant:

$$t_{transf\_dada} = t_{rec\_int} + t_{rsi} = t_{rec\_int} + (N_{rsi} * t_{instr}) = 3 * t_{cicle} + (15 * t_{instr})$$

Tal com hem vist a l'apartat anterior, i com diu l'enunciat, el temps d'una instrucció és:  $t_{instr} = t_{cicle} / 2$

$$\text{Per tant: } t_{transf\_dada} = 3 * t_{cicle} + (15 * t_{cicle} / 2) = 10,5 * t_{cicle}$$

Volem trobar el temps de cicle tal que el temps de transferència d'una dada sigui 8 ns:

$$8 \text{ ns} = 10,5 * t_{cicle} \Rightarrow t_{cicle} = 8 / 10,5 = 0,761 \text{ ns}$$

$$1 / 0,761 * 10^{-9} = 1,314 \text{ GHz} = 1314 \text{ Mhz}$$



### Apartat 2.3 (1 punt) E/S per DMA

Suposarem que el controlador d'ES pot funcionar en mode DMA (Accés Directe a Memòria). La suma del **temps de cessió** del bus i del **temps de recuperació** del bus es de 10 ns ( $t_{\text{cessió}} + t_{\text{recup}} = 10 \text{ ns}$ ). El **temps de la transferència** pel bus es de 1 ns ( $t_{\text{mem}} = 1 \text{ ns}$ ).

#### 2.3.a) (0,5 punts)

Considerem que en la transferència per DMA, les dades s'envien entre el controlador de DMA i la memòria, en mode ràfega, i es disposa d'un buffer de mida  $m_{\text{buffer}} = 400$  bytes. Calcular el temps total d'ocupació del bus per part del controlador de DMA per a dur a terme la transferència que venim analitzant.

Mida de les ràfegues  $N_{\text{ràfega}} : m_{\text{buffer}} / m_{\text{dada}} = 400 / 4 = 100$

Temps ocupació Bus,  $t_{\text{transf\_ràfega}} : t_{\text{cessió}} + 100 * t_{\text{mem}} + t_{\text{recup}} = 10 + 100 * 1 = 110 \text{ ns}$

Nombre de peticions del Bus,  $N_{\text{dades}} / N_{\text{ràfega}} : 400.000 / 100 = 4000$

Temps total d'ocupació del Bus  $t_{\text{transf\_bloc}} : t_{\text{transf\_dada}} * (N_{\text{dades}} / N_{\text{ràfega}}) = 110 * 4000 = 440000 \text{ ns} = 440 \text{ us} = 0,44 \text{ ms}$

**2.3.b) (0,5 punts)** La CPU no pot fer cap tasca durant tot el temps en que el bus està ocupat per part del controlador de DMA. Quin percentatge de temps té disponible la CPU per a executar codi efectiu d'altres programes durant la transferència?

$t_{\text{transf\_bloc}} = 0,44 \text{ ms}$

$t_{\text{bloc}} = 3200 \text{ ms}$

$\% \text{ocupació} = (t_{\text{transf\_bloc}} * 100) / t_{\text{bloc}}$

Percentatge de temps disponible:  $100 - \% \text{ocupació} = 100 - (t_{\text{transf\_bloc}} * 100) / t_{\text{bloc}} = 100 - (0,44 * 100) / 3200 = 100 - 0,01375 = 99,98625\%$

## Apartat 2.4 (1 punt) Qüestions d'E/S

### 2.4.a) (0,25 punts)

Quines són les dues tècniques bàsiques per establir prioritats a l'hora d'atendre els perifèrics d'un computador. Explica breument en que consisteixen.

Les polítiques de prioritats bàsiques són:

- **Prioritat fixa:** quan el processador està preparat per a atendre una petició, sempre comença la consulta pel perifèric que té més prioritat. El perifèric amb més prioritat sempre és atès el primer. Si hi ha moltes peticions de perifèrics prioritaris, els menys prioritaris pot ser que s'esperin molt temps per a ser atesos.
- **Prioritat rotativa:** quan el processador està preparat per a atendre una petició, consulta el perifèric que hi ha a continuació segons un número d'ordre preestablert. Als perifèrics s'hi assigna un ordre però tots tenen la mateixa prioritat. Normalment les polítiques de prioritat es defineixen segons les necessitats de cada dispositiu i les restriccions específiques que molts cops imposa el sistema d'E/S utilitzat en un computador. I es pot utilitzar diferents polítiques de prioritats en un mateix sistema.

### 2.4.b) (0,25 punts)

Un sistema que utilitza E/S programada i una política de prioritats rotativa, els temps que executa la rutina de prioritats formaria part de la fase de sincronització o de la transferència d'informació? Justifica la Resposta.

No és habitual gestionar operacions d'E/S amb múltiples dispositius utilitzant E/S programada. En cas de ser necessari cal tenir present que durant la sincronització s'ha de fer l'enquesta de tots els perifèrics implicats establint una política de prioritats implementada en forma de programa durant la sincronització. Aquest sistema d'identificar quin perifèric necessita ser atès s'anomena enquesta o polling.

### 2.4.c) (0,25 punts)

Quina tècnica es sol utilitzar per identificar quin perifèric ha sol·licitat l'atenció del processador en un sistema d'E/S amb connexió d'interrupcions per Daisy Chain ? Quin és el seu funcionament?

Vectorització. S'anomena vectorització la tècnica en què el processador identifica el mòdul d'E/S mitjançant la informació que envia el mateix mòdul d'E/S. Quan per a tractar les interrupcions utilitzem aquesta tècnica per a identificar qui fa la petició diem que tenim les interrupcions vectoritzades.

Un cop un mòdul d'E/S o més d'un han fet la petició al processador activant la INT i el processador accepta la petició, activa l'INTA i comença el procés per a saber quin s'ha d'atendre.

Aquest senyal INTA provinent del processador, actiu a la baixa, arriba al primer mòdul. Si aquest mòdul ha sol·licitat atenció bloqueja la propagació del senyal INTA i diposita al bus del sistema el vector d'interrupció, valor emmagatzemat al registre vector del mòdul d'E/S; si no ha sol·licitat atenció (no té activa la INT) deixa passar el senyal INTA al mòdul següent, que fa el mateix fins que arriba al darrer mòdul connectat a aquesta línia.

**2.4.d) (0,25 punts)**

En un sistema d'E/S gestionat per DMA, quina diferència hi ha entre el funcionament normal i el mode ràfega, quines avantatges té l'un respecte de l'altre?

Una manera d'optimitzar les operacions d'E/S per DMA consisteix a reduir el nombre de cessions i recuperacions del bus. Per a fer-ho, en lloc de sol·licitar i alliberar el bus per a cada dada que s'ha de transferir, es sol·licita i s'allibera el bus per a transferir un conjunt de dades de manera consecutiva. Aquesta modalitat de transferència s'anomena mode ràfega.

Per a fer la transferència d'aquest conjunt de dades, que anomenem ràfega, el controlador de DMA cal que disposi d'una memòria intermèdia (buffer), de manera que la transferència de dades entre la memòria i el controlador de DMA és pugui fer a la velocitat que permeti la memòria i no quedant limitada a la velocitat del perifèric.

Aquest mode de funcionament no afecta la programació ni la finalització de l'operació d'E/S descrita anteriorment, però sí que modifica la transferència de dades.

**Criteris de valoració.** En els apartats 2.1.a i 2.2.a cada valor erroni resta 0,25. La resta d'apartats estan bé o estan malament. No hi ha gradació.