

PAC2

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 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 $8*N+7$. Una fórmula per calcular l'identificador numèric del bloc és la següent:

Bloc = adreça de memòria (adreça a 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 i 7).

Apartat 1.1 (2 punts) 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. 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, 4, 15, 16, 20, 21, 4, 22, 23, 24, 55, 56, 57, 16, 25, 26, 4, 58, 30, 31

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 (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 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: 55
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	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

	Fallada: 56	Fallada: 16
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	48, 49, 50, 51, 52, 53, 54, 55	16, 17, 18, 19, 20, 21, 22, 23
Línia 3	56, 57, 58, 59, 60, 61, 62, 63	56, 57, 58, 59, 60, 61, 62, 63

	Fallada: 25	Fallada: 58
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	56, 57, 58, 59, 60, 61, 62, 63

	Fallada: 30	
Línia 0	0, 1, 2, 3, 4, 5, 6, 7	
Línia 1	8, 9, 10, 11, 12, 13, 14, 15	
Línia 2	16, 17, 18, 19, 20, 21, 22, 23	
Línia 3	24, 25, 26, 27, 28, 29, 30, 31	

1.1.b) (0,4 punts) Quina és la taxa de fallades (T_f) ?

$$T_f = 6 \text{ fallades} / 20 \text{ accessos} = 0,30$$

1.1.c) (0,4 punts) Supposem 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) ?

$$t_m = T_f \times t_f + (1-T_f) \times t_e = 0,30 * 25 \text{ ns} + 0,7 * 5 \text{ ns} = 7,5 \text{ ns} + 3,5 \text{ ns} = 11 \text{ ns}$$

Apartat 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, 4, 15, 16, 20, 21, 4, 22, 23, 24, 55, 56, 57, 16, 25, 26, 4, 58, 30, 31

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.

	Estat Inicial	Fallada: 55
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	48, 49, 50, 51, 52, 53, 54, 55
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: 56	Fallada: 4
Línia 0	56, 57, 58, 59, 60, 61, 62, 63	56, 57, 58, 59, 60, 61, 62, 63
Línia 1	48, 49, 50, 51, 52, 53, 54, 55	0, 1, 2, 3, 4, 5, 6, 7
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

1.2.b) (0,4 punts) Quina és la taxa de fallades (T_f) ?

$$T_f = 3 \text{ fallades} / 20 \text{ accessos} = 0,15$$

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 6 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) ?

$$t_m = T_f \times t_f + (1-T_f) \times t_e = 0,15 * 20 \text{ ns} + 0,85 * 6 \text{ ns} = 3,0 \text{ ns} + 5,1 \text{ ns} = 8,1 \text{ ns}$$

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_{transf}) = 4 MB/s
- Temps de latència mitjà del dispositiu ($t_{\text{latència}}$) = 0
- 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 5, o el sisè bit menys significatiu (quan val 1 indica que està disponible)
- Processador amb una freqüència de rellotge de 1 GHz, i on totes les instruccions s'executen en 4 cicles de rellotge ($t_{\text{instr}} = 4$ cicles)
- 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 $N_{\text{dades}} = 100.000$ dades, es a dir, 100.000×4 Bytes = 400.000 Bytes
- Adreça inicial de memòria on resideixen les dades: 20000000h

Apartat 2.1 (2,5 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 = 100000

VALOR2 = 20000000h

VALOR3 = 0A4h

VALOR4 = 00100000b = 20h = 32d

VALOR5 = 0A0h

2.1.b) (1 punt) Quant temps dura la transferència?

$t_{\text{dades}} = 400.000 (N_{\text{bytes}}) / 4 \text{ MBytes/s } (v_{\text{transf}}) = 0,1 \text{ segons} = 100 \text{ ms}$

$t_{\text{bloc}} = t_{\text{latència}} + t_{\text{dades}} = 0 \text{ ms} + 100 \text{ ms} = 100 \text{ ms.}$

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?

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 $8 \times 4 (t_{instr}) = 32$ cicles.

Freqüència de rellotge = 1 GHz, implica un temps de cicle de 1 nanosegon ($1 / 10^9$)

Per tant, el temps mínim per a transferir una dada és: $32 \times 1 = 32$ ns

Es poden transferir 4 bytes cada 32 ns, es a dir: $4 / 32 \times 10^{-9} = 125 \text{ MByte/s}$

Apartat 2.2 (2,5 punts) E/S per Interrupcions

Suposeu que el següent codi CISCA 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 16 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 = 0A4h

VALOR2 = 32d

VALOR3 = Dir

VALOR4 = 0A0h

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 temps d'un cicle, t_{cicle} , és 1 nanosegon.

Temps per atendre la interrupció, t_{rec_int} : $16 \text{ cicles} \times 1 \text{ ns} (t_{cicle}) = 16 \text{ ns}$

Temps d'execució de una instrucció, t_{instr} : $4 \text{ cicles} \times 1 \text{ ns} (t_{cicle}) = 4 \text{ ns}$

Temps d'execució RSI, t_{rsi} : $N_{rsi} \times t_{instr} = 15 \text{ instr.} \times 4 \text{ ns} = 60 \text{ ns}$

Temps consumit per CPU en cada interrupció, $t_{\text{transf_dada}}$:

$$t_{\text{transf_dada}} = t_{\text{rec_int}} + t_{\text{rsi}} = 16 + 60 = 76 \text{ ns}$$

Nombre d'interrupcions produïdes (o nombre total de dades, N_{dades}): 100.000 interrupcions.

Temps consumit en total en TOTES les interrupcions:

$$t_{\text{transf_bloc}} = t_{\text{transf_dada}} \times N_{\text{dades}} = 76 \text{ ns} \times 100.000 \text{ interrupcions} = 7,6 \text{ ms}$$

(milisegons)

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

$$t_{\text{cpu}} = (t_{\text{prog}} + t_{\text{final}}) + t_{\text{transf_bloc}} = 200 \text{ ns} + 7,6 \text{ ms} = 7,6002 \text{ ms.}$$

Dels 100 ms de temps total per a realitzar la transferència (temps calculat abans), la CPU està dedicada a la tasca de E/S al voltant d'un $7,6002 / 100 = 0,076 \Rightarrow 7,6\%$ del temps.

2.2.c) (1 punt) 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 100.000 interrupcions durant 100 ms. (Aquesta última dada s'ha calculat abans)

Es a dir, tenim una interrupció cada $100.000 / 100.000 = 1 \text{ us}$ (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 consumit per la CPU en cada interrupció és 76 cicles de rellotge (transferir control a RSI + executar RSI).

$$1000 \text{ ns} / 76 \text{ cicles de rellotge} = 13,158 \text{ ns} / \text{cicle de rellotge} \text{ (és el temps màxim)}$$

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

$$1 \text{ cicle de rellotge} / 13,158 \text{ ns} = 76 \text{ milions de cicles / segon} = 76 \text{ MHz}$$

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 2 ns ($t_{\text{mem}} = 2 \text{ ns}$).

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 ràfegues de 100 dades (100×4 bytes cada vegada). Calcular el temps total d'ocupació del bus per part del controlador de DMA per a dur a terme la transferència que venim analitzant.

Temps ocupació Bus, $t_{\text{transf_ràfega}}$:

$$100 \times 2 = 210 \text{ ns}$$

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

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

$$210 \times 1.000 = 0,21 \text{ ms}$$

$$t_{\text{cessió}} + 100 \times t_{\text{mem}} + t_{\text{recup}} = 10 +$$

$$100.000 / 100 = 1.000$$

$$t_{\text{transf_ràfega}} \times N_{\text{dades}} / N_{\text{ràfega}} =$$

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?

$$t_{\text{cpu}} = (t_{\text{prog}} + t_{\text{final}}) + t_{\text{transf_bloc}} = 200 \text{ ns} + 0,21 \text{ ms} = 0,2102 \text{ ms}$$

Percentatge de temps disponible:

$$100 - (100 \times 0,2102 / 100) = 99,79 \%$$