

# ESTRUCTURA DE COMPUTADORS

Segon examen parcial

23 de març de 2015

Cognoms i nom	DNI	Grup

**1 (1 punt)** Una CPU té una capacitat d'adreçament de 16 GB i fa servir paraules de 64 bits.

a) ¿Quants bits d'adreça cal per a adreçar tota la memòria? Raoneu-ne la resposta

34 bits, atès que  $2^{34} = 16 \text{ G}$

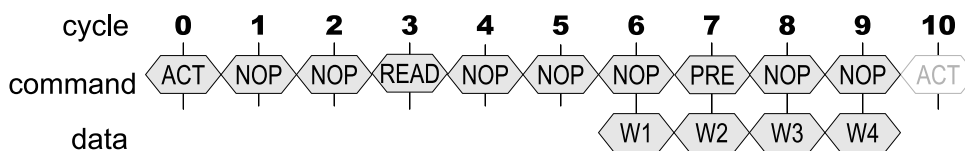
b) ¿Quantes línies de selecció d'octet té la CPU ( $BE_i^*$ )? Raoneu-ne la resposta

8 línies, des de  $BE_0^*$  fins a  $BE_7^*$ , ja que 64 bits = 8 bytes

c) ¿Com s'anomenen les línies físiques d'adreça de la CPU ( $A_i$ )? Raoneu-ne la resposta

Desde  $A_3$  hasta  $A_{33}$ , ya que los bits  $A_0..A_2$  están descodificados por  $BE_0^*..BE_7^*$

**2 (1 punt)** El cronograma de la figura correspon a una operació de lectura d'un bloc de 4 paraules d'un xip de memòria SDRAM que funciona a 100 MHz i té una amplada de paraula de 16 bits.



Indiqueu:

a) Quin és el **temps d'accés** d'aquesta memòria, expressat en nanosegons.

Com que les dades tarden en aparèixer 6 cicles de  $1 / 100\text{MHz} = 10 \text{ ns}$ , el temps d'accés és de 60 ns

b) Quina és l'**amplada de banda**.

1 paraula de 2 B (16 bits) a cada cicle de rellotge (10 ns)  $\rightarrow 2 \text{ B} / 10 \text{ ns} = 200 \text{ MB/s}$

**3 (1 punt)** Indiqueu en quins dels casos següents seria possible concatenar l'accés a dos blocs d'un xip de memòria RAM dinàmica:

- |  |                             |                             |      |
|--|-----------------------------|-----------------------------|------|
| a) Dos blocs d'una mateixa filera en el mateix banc.     | <input type="checkbox"/> Sí | <input type="checkbox"/> NO | (Sí) |
| b) Dos blocs en dues fileres distintes d'un mateix banc. | <input type="checkbox"/> Sí | <input type="checkbox"/> NO | (N)  |
| c) Dos blocs en dos bancs distintes.                     | <input type="checkbox"/> Sí | <input type="checkbox"/> NO | (Sí) |

**4 (1 punt)**

A) Considereu un MIPS al què s'ha connectat un mòdul de memòria M1 de 256 MB en l'adreça més alta del mapa (és a dir, que conté l'adreça 0xFFFFFFFF)

A.1 Quina és l'adreça menor de les contingudes en el mòdul?

0xF0000000

A.2 Quina serà la funció de selecció activa per nivell baix?

/A31 + /A30 + /A29 + /A28

B) Teniu un mòdul M2 de 128 MB i voleu instal·lar-lo de manera que no quedi espai lliure entre aquest i M1.

B.1 Quina és l'adreça menor de les contingudes en el mòdul?

0xE8000000

B.2 Quina serà la funció de selecció activa per nivell baix?

/A31 + /A30 + /A29 + A28 + /A27

C) Un dia trobeu instal·lat un altre mòdul M3 de memòria amb la funció de selecció SelM3 = A31 + /A30 + A29

C.1 Quina és l'adreça menor de les contingudes en el mòdul?

0x40000000

C.2 Quin és el llarg de l'espai lliure comprés entre aquesta adreça i l'adreça 0x00000000? Expressau-lo en MB

400h = 1024 MB

**5 (2 punts)** Es disposa d'una memòria cau de dades de 32 KB i memòria principal de 4 GB. Per limitacions de disseny, la grandària de la memòria de control associada a aquesta cau no pot superar el 10% de la grandària neta per a dades. Indiqueu quines de les configuracions de cau següents satisfan aquesta restricció i quines no. Justifiqueu les respostes indicant el nombre d'entrades de la memòria de control i el format de cada entrada en cadascun dels casos.

Grandària bloc (Bytes)	Correspondència	Política escriptura (encert)	Política escriptura (fallada)	Algoritme de reemplaçament	¿Acompleix?
16	Directa	Write through	No allocate	—	NO
16	4 vies	Write back	Allocate	LRU	NO
32	Directa	Write through	No allocate	—	SÍ
32	4 vies	Write back	Allocate	LRU	SÍ

La restricció del 10% supos que la memoria de control no debe superar el tamaño de 32 KB / 10 = 3276,8 Bytes (en la práctica, podemos truncar a 3276 B).

Caso 1: Hay 32 KB / 16 B = 2048 líneas. Las direcciones de memoria (de 32 bits, ya que el espacio direccionable total es de 4 GB) se estructuran en 4 bits para offset, 11 bits para etiqueta y 17 bits de etiqueta. Los bits de control necesarios para esta configuración se limitan a 1 (V) + 17 (Etiq) = 18 por cada línea, es decir:  $18 * 1024 = 36864 \text{ b} = 4608 \text{ B}$ , que es mayor que el 10 % de 32 KB.

Caso 2: Ahora tenemos  $2048 / 4 = 512$  conjuntos, luego hay 19 bits de etiqueta, 9 de conjunto y 4 de offset. Para el control hacen falta 1 (V) + 19 (Etiq) + 1 (Dirty) + 2 (LRU) = 23 bits por línea, es decir:  $23 * 2048 = 47104 \text{ b} = 5888 \text{ B}$ , por lo que tampoco cumple.

Caso 3: Ahora tenemos 32 KB / 32 B = 1024 líneas. Las direcciones tienen 5 bits de offset, 10 bits de línea y 17 de etiqueta. La memoria de control requiere 1 (V) + 17 (Etiq) = 18 bits, que por el número de líneas serán 18432 b, es decir, 2304 B. Esta configuración sí cumple la restricción.

Caso 4: Habrán 1024 líneas / 4 = 256 conjuntos. Direcciones de memoria: 5 bits offset, 8 para conjunto y los 19 restantes para etiqueta. La memoria de control requiere: 1 (V) + 19 (Etiq) + 1 (Dirty) + 2 (LRU) = 23 bits. Multiplicando por el número de líneas tendremos un tamaño total de  $23 * 1024 = 23552 \text{ b} = 2944 \text{ B}$ , por lo que esta configuración también cumple la restricción.

Resum:

Cas	Estructura adreça E+C+D	Bits control/línia V+E+D+LRU	Nre de línies	Total control en bits	Total control en bytes
1	17+11+4	1+17 = 18	2048	36864	4608
2	19+9+4	1+19+1+2 = 23	2048	47104	5888
3	17+10+5	1+17+0+0 = 18	1024	18432	2304
4	19+8+5	1+19+1+2 = 23	1024	23552	2944

## 6 (2.5 punts) Considereu el programa que fa els càlculs següents sobre un vector V de N components:

a) Calcula la suma de les components del vector  $f0 = \sum_{i=0}^{N-1} V[i]$

b) Fa la divisió de totes les components del vector pel valor obtingut en a)

Vegeu tot seguit el codi corresponent en un llenguatge d'alt nivell i en ensamblador del MIPS R2000. Noteu que s'hi fan dos recorreguts del vector en ordre creixent d'adreces. N és una constant que pren el valor indicat més avall. Cada pseudoinstrucció que apareix en el codi font en ensamblador es tradueix en només una instrucció màquina.

Alt nivell	Assemblador
<code>float V[N];</code>	<code>.data 0x10000000</code>
<code>float f0;</code>	<code>V: .float 2.0,... # N components</code>
<code>int t0,t1;</code>	
	<code>.text 0x00400000</code>
<code>f0 = 0.0; t0=0;</code>	<code>la \$t0,V</code>
<code>for (t1=N; t1&gt;0; t1--){</code>	<code>li \$t1,N</code>
<code>f0 = f0 + V[t0];</code>	<code>mtc1 \$zero,\$f0</code>
<code>t0 = t0 + 1;</code>	<code>for1: lwc1 \$f10,0(\$t0)</code>
<code>}</code>	<code>add.s \$f0,\$f0,\$f10</code>
<code>t0=0;</code>	<code>addi \$t1,\$t1,-1</code>
<code>for (t1=N; t1&gt;0; t1--){</code>	<code>addi \$t0,\$t0,4</code>
<code>V[t0] = V[t0]/f0;</code>	<code>bgtz \$t1,for1</code>
<code>t0 = t0 + 1;</code>	
<code>}</code>	<code>la \$t0,V</code>
	<code>li \$t1,N</code>
	<code>for2: lwc1 \$f10,0(\$t0)</code>
	<code>div.s \$f10,\$f10,\$f0</code>
	<code>swc1 \$f10,0(\$t0)</code>
	<code>addi \$t0,\$t0,4</code>
	<code>addi \$t1,\$t1,-1</code>
	<code>bgtz \$t1,for2</code>

El processador disposa de memòries cau separades per a instruccions i per a dades (1KB + 1KB) amb blocs de 32 bytes. En el moment en què comença l'execució del programa, totes les línies de la cau són invàlides.

A) ¿Quantes instruccions caben en un bloc? ¿Quants components del vector V caben en un bloc?

Instruccions / bloc de codi: 8

Components V / bloc de dades: 8

B) Supposeu que les memòries cau són de correspondència directa, escriptura posterior (*write back*) i sense ubicació en escriptura (*no-write-allocate*). Amb N=100, calculeu els valors següents a la fi de l'execució del codi:

Memòria cau	Paràmetre	Valor
Instruccions	Nombre d'accessos	$5 + 11 \times N = 1105$
	Nombre de blocs referenciats	$15 \text{ instr.} \times 4 \text{ bytes/inst} / 32 \text{ bytes/bloc} = 2 \text{ blocs}$
	Taxa d'encerts	$1 - 2/1105 \approx 100 \%$
Dades	Nombre d'accessos	$3 \times 100 = 300$
	Nombre de blocs referenciats	$N \times 4 / 32 = 13$
	Nombre de reemplaçaments	0
	Taxa d'encerts	$1 - 13/300$
	Nombre d'escriptures en memòria principal	0

C) Amb  $N = 260$ , amb una memòria cau de les mateixes característiques, calculeu:

Memòria cau	Paràmetre	Valor
Dades	Nombre d'accessos	$3 \times 260 = 780$
	Nombre de blocs referenciats	$260 \times 4 / 32 = 33$
	Nombre de reemplaçaments	3
	Taxa d'encerts	$1 - 35/780$
	Nombre d'escriptures de bloc en memòria principal	1
	Nombre de blocs en cau no coherents amb memòria	32

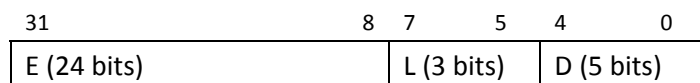
D) Què canviarà en l'apartat B (amb  $N=100$ ) si la memòria cau de dades aplica les polítiques d'escriptura directa (*write-through*) i amb ubicació (*write-allocate*)

Cadascuna de les escriptures **swc1 \$f10,0(\$t0)** segueix a una lectura **lwc1 \$f10,0(\$t0)** en la mateixa adreça. Per tant, no hi ha cap fallada d'escriptura i la política d'ubicació és irrellevant. Es a dir: totes les escriptures són encerts i el canvi no afecta a la taxa de fallades.

Amb la política *write-through*, les escriptures d'una paraula en un bloc de la cau es transmeten directament a la memòria principal, i mai no hi ha incoherència entre ambdós nivells de la jerarquia de memòria.

Si  $N=100$ , cadascuna de les 100 escriptures en la memòria cau haguera provocat l'escriptura de la paraula implicada en la memòria principal. Al final del codi tots els blocs de dades que queden a la cau serien coherents amb la memòria principal.

**7 (1.5 punts)** Un processador semblant al MIPS té connectada una memòria cau de dades de correspondència directa i escriptura posterior (*write-back*) amb ubicació (*write-allocate*) formada per 8 línies que contenen blocs de 32 bytes. L'estructura de l'adreça que interpreta la memòria cau és



La taula de la dreta mostra l'estat inicial de la memòria cau

A) Quin és el rang d'adreces del bloc contingut en la línia 6?

De 0x87654fc0 a 0x87654fdf

Línia	V	M	E (hex)
0	1	1	100000
1	0	–	-----
2	1	0	000200
3	1	0	1af002
4	0	–	-----
5	0	–	-----
6	1	0	87654f
7	1	1	000666

B) Heu d'explicar com afecta a l'estat de la cau cadascuna de les instruccions que teniu més avall. En cada cas heu de contestar quina és la línia afectada, si es tracta d'un cas d'encert (E) o de fallada (F) i l'estat en què queden els bits de vàlid (V), modificat (M) i l'etiqueta (E). Considereu que \$t0 = 0x10000000. Heu de partir sempre de l'estat inicial.

	Línia	E/F	Estat resultant		
			V	M	E
<b>lw \$t1, 0(\$t0)</b>	0	E	1	1	1000000
<b>lw \$t1, 0x120(\$t0)</b>	1	F	1	0	1000001
<b>sw \$t1, 0x120(\$t0)</b>	1	F	1	1	1000001