

# ESTRUCTURA DE COMPUTADORS

Examen Primer Parcial

Gener 2012

Cognoms i Nom

DNI

Grup

--	--	--

1. (2 punts) Un xip de memòria dinàmica SDRAM DDR de 1 Gb ( $2^{30}$  bits) de capacitat es configura per a treballar amb paraules de 8 bits (x8). El xip s'organitza internament en 4 bancs, cadascun d'ells amb 16384 fileres. La grandària de bloc és de 8 paraules. Els paràmetres temporals més importants, expressats en ns, són:  $t_{RC} = 55$  (ACTIVE-to-ACTIVE),  $t_{RCD} = 15$  (ACTIVE-to-READ),  $t_{RAS} = 40$  (ACTIVE-to-PRECHARGE) i  $t_{RP} = 15$  (PRECHARGE). Considereu que el xip funciona a la freqüència més alta, és a dir, 200 MHz, i que la latència de CAS associada és de 3 cicles (CL = 3).
- a) Indiqueu a quin banc, filera i columna del xip fa referència l'adreça **0x7008A03**. Tingueu en compte que l'accés al xip és lineal, és a dir, les adreces s'interpreten espacialment com banc-filera-columna.

Cal calcular el nombre de columnes que hi ha en cada banc. La forma més senzilla consisteix a fer  $27-2-14=11$  bits, és a dir, els 27 bits d'adreçament del xip es reparteixen entre banc (n'hi ha 4), filera dintre del banc (n'hi ha 16384) i, finalment, columna. Així doncs, hi ha  $2^{11}=2048$  columnes de paraula (byte) per cada banc.

L'adreça 0x7008A03, expressada en binari 111 0000 0000 1000 1010 0000 0011 s'interpreta segons els camps de banc ( $11_2$ ), filera ( $10000000010001_2$ ) i columna ( $01000000011_2$ ).

En resum, s'adreça el banc  $3_{10}$ , la filera  $8209_{10}=2011_{16}$  i la columna  $515_{10}=203_{16}$ .

- b) Feu la temporització de les ordres per a una operació de lectura d'un bloc de 8 paraules en eixa SDRAM DDR. Tingueu en compte que la precàrrega no és automàtica. En quin cicle es pot començar una nova operació de lectura?

Primer expressem els temps en cicles de rellotge tot dividint per 5 ns:  $t_{RC}=11$ ,  $t_{RCD}=3$ ,  $t_{RAS}=8$  i  $t_{RP}=3$ . L'accés transcorre així: les dades apareixen després de  $t_{RCD}+t_{CL}=3+3=6$  cicles; la ràfega s'estén durant 4 cicles (en ser DDR es transmeten dues dades per cada cicle); així doncs, la ràfega acaba després de 10 cicles de rellotge. Ara bé, l'ordre de precàrrega no es pot emetre abans de  $t_{RAS}=8$  cicles, per la qual cosa l'operació de lectura acaba als  $8+3=11$  cicles, això és, un cicle després de transmetre la darrera dada de la ràfega. La seqüència d'ordres és:

Cicle	1	2	3	4	5	6	7	8	9	10	11	12
Ordre	ACT	NOP	NOP	READ	NOP	NOP	NOP	NOP	PRE	NOP	NOP	
Dades							D1/D2	D3/D4	D5/D6	D7/D8		
Adreces	Filera			Columna								

Cicle on pot començar una nova operació: **Cicle número 12**

- c) Calculeu el nombre de xips necessaris per tal de construir un mòdul estàndard de 1 Gbyte de capacitat. Quina és la màxima amplada de banda que pot assolir el mòdul?

El xip té una capacitat de 1 Gb = 128 MB. Per tant, en un mòdul de 1 GB haurem d'incloure  $2^{30}/2^7=8$  xips.

L'amplada màxima del mòdul es pot calcular com:  $200 \times 8 \times 2 = 3200 \text{ MB/s} = 3.2 \text{ GB/s}$

- d) Tingueu en compte que dos d'aquests mòduls, M1 i M2, s'utilitzen en un mapa de memòria per al processador MIPS R2000. Calculeu les funcions de selecció actives a nivell baix de cadascun dels mòduls si s'ubiquen a partir de les adreces  $0x80000000$  i  $0xC0000000$ , respectivament.

Les adreces del mòdul M1 tenen la forma  $10xx...x$  i les del mòdul M2  $11xx...x$ .

Per tant, les funcions de selecció són:

$$/SEL_{M1} = /A_{31} + A_{30}$$

$$/SEL_{M2} = /A_{31} + /A_{30}$$

2. (1 punt) Considereu el programa següent:

```
.data 0x10005000
v:    .word 0,1,2,3,4,5,6,7
      .text 0x00400000
      lui $t0, 0x1000
      ori $t0, $t0, 0x5000
      addi $t1, zero, 8
bucle: lw $t2, 0($t0)
      bgez $t2, salta
      sw $zero, 0($t0)
salta: addiu $t0, $t0, 4
      addi $t1, $t1, -1
      bgtz $t1, bucle
```

Contesteu les qüestions següents, tot **justificant-ne** les respostes:

- a) Quants bytes ocupen els segments de dades i codi, respectivament?

El segment de dades ocupa  $8 \times 4 = 32$  bytes.  
El segment de codi ocupa  $9 \times 4 = 36$  bytes.

- b) Quina és la instrucció que es troba en l'adreça  $0x00400018$ ?

Es tracta de la instrucció **addiu** (adreça base més un desplaçament de  $6 \times 4 = 24 = 18_{16}$ )

- c) En quina adreça de memòria està l'element **v[7]**?

Es tracta de l'adreça  $0x1000501C$  (adreça base més un desplaçament de  $7 \times 4 = 28 = 1C_{16}$ ).

d) Quantes instruccions s'executen?

S'hi executen  $3 + 8 \cdot 5 = 43$  instruccions; noteu que la instrucció `sw` mai s'executa perquè tots els elements del vector són positius.

e) Quants accessos i de quin tipus (lectura o escriptura) es fan al segment de dades i codi, respectivament?

Accessos a codi (lectures): 43 (coincideix amb el nombre d'instruccions executades).

Accessos a dades: 8 lectures (lectura de tots els elements del vector).

3. (3 punts) Disposem d'un processador MIPS R2000 amb una cau L1 de dades amb capacitat de 128 bytes, grandària de bloc de 16 bytes i correspondència directa. La política de fallada en escriptura és *write allocate* i la política d'actualització és *write back*. Es vol accedir a dos vectors, `V` i `Z`, cadascun de 12 elements enters (32 bits), emmagatzemats en memòria en les adreces `0x100C0120` i `0x100E00A0`, respectivament. L'accés als vectors es fa amb el programa següent (indicat en C i en ensamblador); se suposa que la variable `i` s'emmagatzema en registre.

<pre>for (i=0;i&lt;12;i++)     Z[i]= -V[i];</pre>	<pre>V:      .data 0x100C0120         .word 0,1,2,3,4,5,6,7,8,9,10,11         .data 0x100E00A0 Z:      .space 48          .text 0x00400000         .globl __start  __start: lui \$t1, 0x100C         ori \$t1, \$t1, 0x0120         lui \$t2, 0x100E         ori \$t2, \$t2, 0x00A0         ori \$t3, \$zero, 12 bucle:  lw \$t0, 0(\$t1)         li \$t4, -1         xor \$t0, \$t0, \$t4         addi \$t0, \$t0, 1         sw \$t0, 0(\$t2)         addiu \$t1, \$t1, 4         addiu \$t2, \$t2, 4         addi \$t3, \$t3, -1         bne \$t3, \$zero, bucle         .end</pre>
---	---

- a) Indiqueu el nom i la grandària dels camps en què s'interpreta l'adreça de memòria per a gestionar la cau L1.

Direcció es  
de 32 bits  
(MIPS R2000)

25 bits

3 bits

4 bits

Etiqueta

Línea

Desplaz. Bloque

Desplazamiento Bloque (4 bits) porque el tamaño del bloque es de 16 bytes ( $2^4$  bytes)  
Línea (3 bits) porque la cache está formada por 8 conjuntos ( $2^3$  conj.). El número de bloques de la cache es de 8 (128 bytes/16 bytes). Como la correspondencia es directa, el número de vías es 1, por lo que el número de conjuntos será igual a 8 (número\_bloques/numero\_vías). El resto de bits corresponden a la etiqueta.

- b) Escribiu els nombres de bloc de memòria en què s'emmagatzemen els vectors  $v$  i  $z$ , així com les línies de cau en què s'ubiquen (mapegen).

Nombre de blocs de memòria vector $v$		Línies de cau
Corresponden a los 28 bits más significativos de la dirección	0x100C012	2
	0x100C013	3
	0x100C014	4
Nombre de blocs de memòria vector $z$		3ínies de cau
	0x100E00A	2
	0x100E00B	3
	0x100E00C	4

Corresponden a los 3 bits del campo línea

- c) Calculeu la taxa d'encerts ( $H$ ) en l'accés a la memòria cau de dades després de l'execució del programa.

	Nombre de fallades	Nombre d'encerts	Nombre d'accessos
Vector $v$	12	0	12
Vector $z$	12	0	12
TOTAL	24	0	24
Taxa d'encerts	Total_Aciertos/Total_Accesos= 0/24= 0		

- d) Com afectarà la taxa d'encerts ( $H$ ) l'ús d'una política de fallada en escriptura *write no-allocate*?

El empleo de *write no-allocate* elimina los fallos de conflicto entre los vectores  $V$  y  $Z$ , dado que el vector  $Z$  no se ubica en la cache. Sin embargo, tiene como consecuencia que todos los accesos al vector  $Z$  serán fallos.

El balance de aciertos y fallos es el siguiente:

Vector  $V$ : 3 F + 9 A; Vector  $Z$ : 12 F + 0 A; el total de accesos sigue siendo el mismo

La tasa de aciertos será:  $9/24 = 0,375$

- e) Calculeu la grandària en nombre de bits de la memòria de control (etiquetes i bits de control). Indiqueu-ne clarament el nombre d'entrades i la descomposició en camps i grandària de cadascuna d'aquestes entrades.

Tamaño memoria de control= numero\_bloques x tamaño\_entrada\_memoria\_control

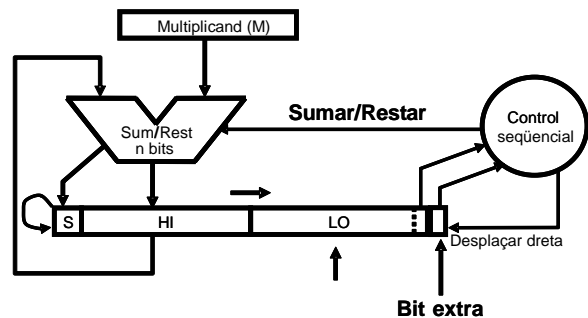
$8 \times [\text{etiqueta (25 bits)} + V (1 \text{ bit}) + M (1 \text{ bit})] = 8 \times 27 \text{ bits} = 216 \text{ bits}$

- f) En el context anterior, tenint en compte una taxa d'encerts de 0,9 i temps d'accés a la cau L1 i a memòria principal de 5 i 200 ns, respectivament, calculeu el temps mitjà d'accés a memòria.

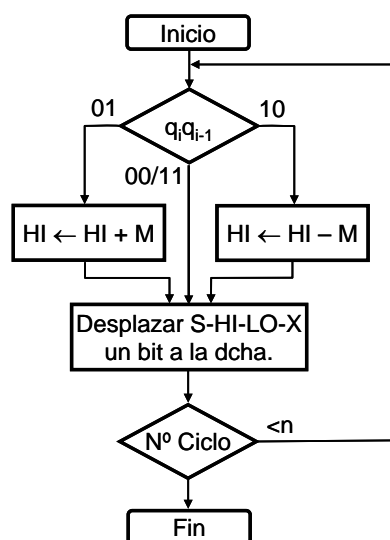
Tiempo medio de acceso a memoria=  $5\text{ns} \times 0,9 + 200\text{ns} \times 0,1 = 24,5\text{ns}$

4. (2 punts) Considereu l'operador de multiplicació seqüencial per a enters amb signe de 32 bits de la figura. El circuit disposa del control per a multiplicar amb l'algorisme de Booth. El cost de cadascuna de les operacions és el següent:

Inicialitzar registres i circuit de control	6 ns
Avaluar 2 bits de Q	2 ns
Sumar/Restar	12 ns
Desplaçar S-HI-LO 1 bit	2 ns
Escriure registre HI	2 ns
Avaluar el cicle actual	2 ns



- a) Quin és l'algorisme que el control seqüencial hauria d'aplicar per a fer la multiplicació per Booth?



- b) Quin és el temps mínim de cicle de rellotge per aquest circuit? Teniu en compte que en un cicle ha d'executar-se tota una iteració. Justifiqueu-ne la resposta.

Tiempo para una iteración =  $2 + 12 + 2 + 2 + 2 = 20$  ns

T ciclo = 20 ns

- c) Quants cicles de rellotge es necessiten per a fer una operació de multiplicació? Justifiqueu-ne la resposta.

33 ciclos = 1 ciclo para inicialización + 32 ciclos de procesamiento

- d) Quina és la màxima productivitat que es podria assolir amb aquest circuit?

T para una operación =  $33 \cdot 20$  ns = 660 ns

Productividad máxima =  $1 / 660$  ns  $\approx 1,5$  MOPS (1.151 MOPS)

- e) Supposeu que modifiquem el circuit anterior a fi de multiplicar segons l'algorisme de recodificació per parelles de bits tot mantenint el mateix temps de cicle. Quina serà la productivitat en aquest cas?

Se necesitarían solo 17 ciclos (1+16)

T operación = 340 ns

Productividad  $\approx 3$  MOPS (2.94 MOPS)

- f) Recodifiqueu per l'algorisme de Booth i per l'algorisme de recodificació per parelles de bits el nombre  $0xC2$  expressat com a enter amb signe de 8 bits.

Recodificació per Booth                      0 -1 0 0 0 +1 -1 0

Recodificació per parelles de bits        -1 0 +1 -2

5. (1 punt) Considereu el programa següent:

```

        .data 0x10000000
f:      .float 25.0
        .text 0x400000
        .globl __start
__start:
        lwc1 $f12,f
        mfc1 $t0,$f12
        li $t1,0x7F800000
        and $t2,$t0,$t1
        li $t3,0x01000000
        addu $t2,$t2,$t3
        and $t2,$t2,$t1
        li $t1,0x807FFFFFFF
        and $t0,$t0,$t1
        or $t0,$t0,$t2
        mtc1 $t0,$f0      # Resultat en $f0

        li $v0,10         # Fi d'execució
        syscall

```

Com a recordatori, la figura següent mostra els formats en simple i doble precisió de l'estàndard IEEE 754, així com els casos especials.

	1	8	23
Single precision	S	E	M
	1	11	52
Double precision	S	E	M
Special cases			
Subnormal	S	0000...00	M ≠ 0
± 0	S	0000...00	000000...00
± ∞	S	1111...11	000000...00
NaN	X	1111...11	M ≠ 0

a) Quin és el tipus de dades empleat tant per als valors d'entrada com per al resultat?

The program takes a single-precision floating-point value and gives the result in register \$f0, of the same type.

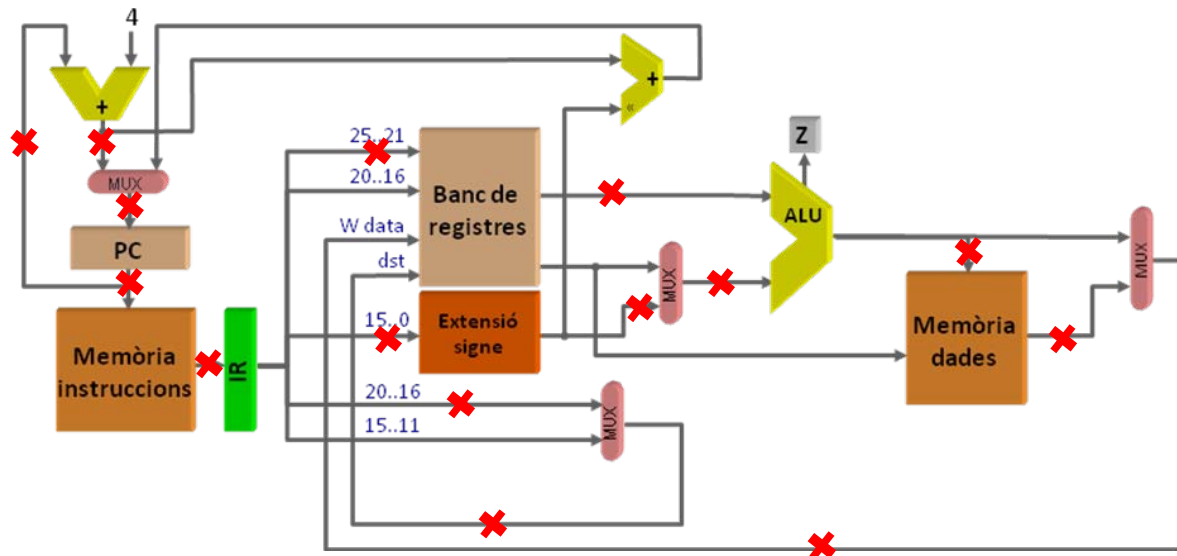
b) Quina és l'operació matemàtica equivalent a la tasca realitzada pel programa?

The program adds 2 to the exponent of the input value, effectively calculating the operation  $\times 4$ .

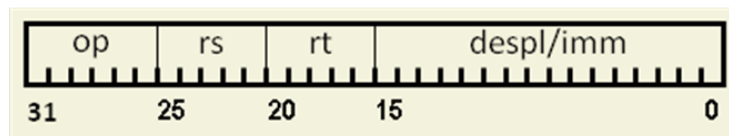
c) Aquesta operació, pot produir desbordament? En quin cas? El programa tracta el possible desbordament o simplement l'ignora?

Overflow is possible since we are adding 2 to the exponent. This may produce an exponent of value 111...1 or 000...0, which could lead to a subnormal, zero, infinity or a NaN. The program does NOT care for that situation; it simply clears the potential carry.

6. (1 punt) En la figura es mostra la ruta de dades vista a classe.



- a) Marqueu sobre la figura les línies actives per a l'execució de la instrucció **lw** (load word) que té el format mostrat a continuació.



- b) Tenint en compte que **lw** (load word) és la instrucció que requereix més temps per a executar-se, i tenint en compte els retards indicats més avall, calculeu la duració mínima del període de rellotge per aquest processador (en ns), la freqüència màxima del rellotge (en MHz) i el nombre d'instruccions que pot executar per segon (MIPS, *million instructions per second*).

Accés al banc de registres: 3 ns  
 Operació de l'ALU: 5 ns  
 Accés a memòries d'instruccions i de dades: 7 ns  
 Resta de retards menyspreable

The delay of the load word instruction is the time to access the instruction memory (7 ns) plus access to the register file (3 ns) plus the ALU delay (5 ns) plus access to the data memory (7 ns) plus access again to the register file for writing (3ns).

The total is hence  $7 + 3 + 5 + 7 + 3 = 25$  ns.

The minimum clock period is therefore 25 ns, since load word is the slowest instruction. This corresponds with a maximum frequency of  $1/25 \text{ ns} = 40 \text{ MHz}$ . Since the datapath requires one clock cycle per instruction, the maximum productivity of this datapath is 40 MIPS.