

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.

--

- 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?

Cicle	1	2	3	4	5	6	7	8	9	10	11	12
Ordre												
Dades												
Adreces												

Cicle on pot començar una nova operació:

- 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?

- 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.

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?

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

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

d) Quantes instruccions s'executen?

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

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<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.

- b) Escriviu 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
Nombre de blocs de memòria vector z	3ínies de cau

- 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			
Vector z			
TOTAL			
Taxa d'encerts			

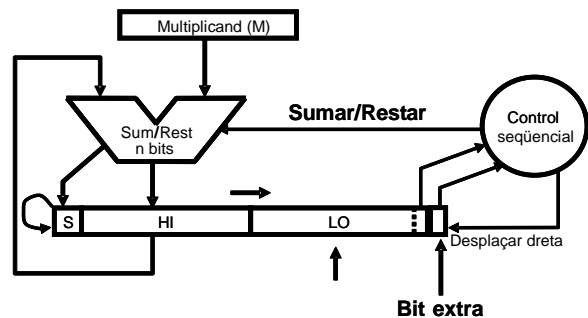
- d) Com afectarà la taxa d'encerts (H) l'ús d'una política de fallada en escriptura *write no-allocate*?

- 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.

- 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.

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.

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

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

- 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?

- 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

Recodificació per parelles de bits

5. (1 punt) Considereu el programa següent:

```

        .data 0x10000000
f:      .float 25.0
        .text 0x400000
        .globl __start
__start:
        lwcl $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.

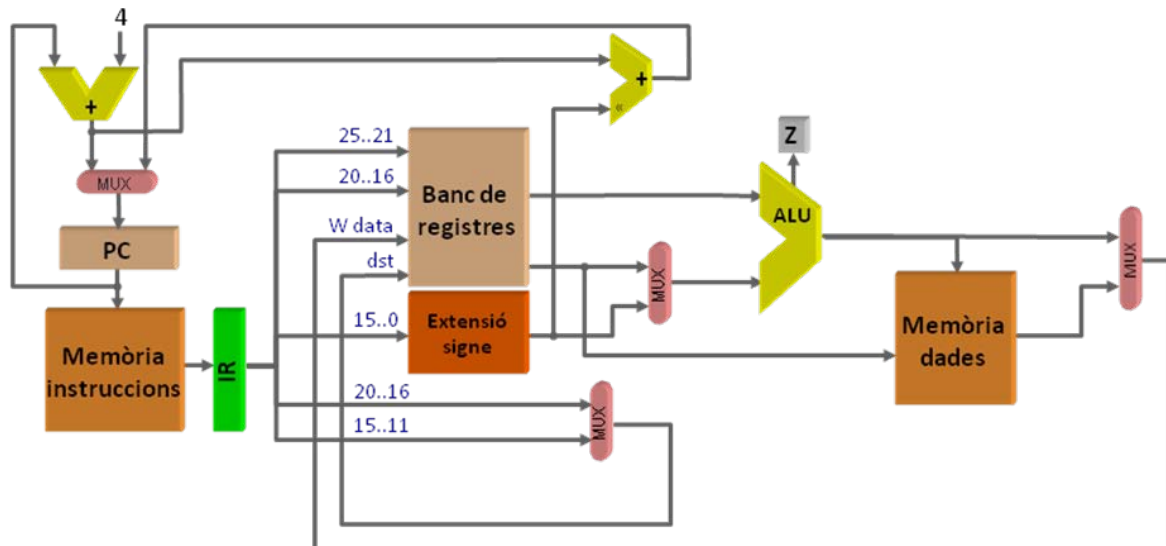
	1	8	23
Single precision	S	E	M
	1	11	52
Double precision	S	E	M
Special cases			
Subnormal	S	0000...00	M ≠ 0
± 0	S	0000...00	000000...00
± ∞	S	1111...11	000000...00
NaN	X	1111...11	M ≠ 0

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

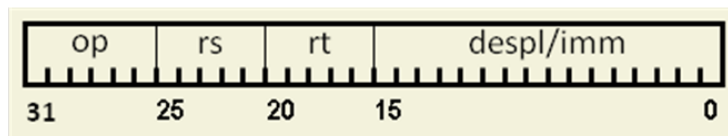
b) Quina és l'operació matemàtica equivalent a la tasca realitzada pel programa?

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

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