

## Problema 1 (2punts):

Es disposa de 4 xips de ROM:

- 2 xips 2K x 4 bits
- 1 xip 4K x 8 bits
- 1 xip 1K x 16 bits

Es vol implementar una única ROM de 8 bits i maximitzar la seva capacitat d'emmagatzematge, utilitzant només els 4 xips de què es disposa.

Es demana:

- a) Especifiqueu les entrades, sortides i grandària en kilobytes de la ROM de 8 bits a implementar.
- b) Dibuixeu el circuit complet incloent totes les entrades i sortides de la ROM. Es disposa de busos, comportes, multiplexors, buffers, decodificadors, etc. Justifiqueu la solució proposada.

Problema 2 (3 punts): Un ordinador té una unitat de memòria de 4 k paraules i una memòria cau de 256 paraules. La memòria cau és associativa per conjunts, amb 16 conjunts i 2 particions per conjunt. Quan es necessita una substitució a la memòria cau, en el conjunt corresponent es reemplaça la partició més antiga.

- a) Format de l'adreça

	Etiqueta	Conjunt	Paraula
Nº de bits			

- b) Suposant que inicialment la memòria cau és buida. Ompliu la taula senyalant les fallades i substitucions que es produiran a la memòria cau si es llegeixen consecutivament les adreces de la MP.

Adreça	Etiqueta		Conjunt	Fallada (si/no)	Bloc substituït
	L1	L2			
0000 0000 1111					
0000 0100 1111					
0001 0000 1111					
0001 0000 1100					
0010 1000 1111					
0000 0000 1101					

Problema 3 (5 punts): Sigui un computador capaç d'executar 10 MIPS ( $10^7$  instruccions per segon). Es vol connectar a l'ordinador, un sol perifèric amb una velocitat de transferència de 20.000 bytes / s. i sobre el qual es realitzen operacions de lectura de blocs de 1.024 bytes. Es pretén veure el comportament de la parella ordinador - perifèric davant les diferents tècniques d'entrada/sortida (programada, mitjançant interrupcions i per DMA)

Se sap que:

- La rutina de transferència d'E / S programada consta de 10 instruccions.
- La rutina de tractament d'interrupció en l'E/S mitjançant interrupcions consta de 20 instruccions.
- La rutina de inicialització del DMA consta de 8 instruccions. I en cada operació d'escriptura d'una dada a memòria el controlador ocupa els busos durant 500 ns.

Es demana que digueu el nombre d'instruccions d'altres processos que pot realitzar l'ordinador durant cadascun dels tipus d'E/S previstos.

Nom: .....

Problema 1: Si tenim les següents declaracions de variables globals, emmagatzemades a memòria a partir de l'adreça 0x10010000:

```
.data
.word 0x100100004
.byte 0xAA
.half 0x00BB
.word 0x88AACCEE
.half 0x0403
.byte 0x40, 0x50, 0x60, 0x70, 0x80, 0x90
```

- a) Indiqueu el contingut inicial de la memòria, en hexadecimal, segons el format que utilitza el simulador QtSpim si el processador MIPS està treballant en *little endian*. Podeu suposar que les posicions de memòria abans de la càrrega del programa valen zero.

Adreça	Valor (0)	Valor (4)	Valor (8)	Valor (c)
0x10010000				
0x10010010				

- b) Quin serà el valor dels registres modificats, en hexadecimal, després d'executar cadascuna de les línies de codi?

```
la $t0, a1      # Carrega al registre l'adreça de memòria      $t0: .....
lw $t1, 0($t0)  # Carrega al registre la paraula (32 bits)      $t1: .....
lb $t2, 2($t1)  # Carrega al registre el byte (fa l'extensió del signe) $t2: .....
sra $t2, $t2, 2 # Desplaçament lògic a la dreta (2)           $t2: .....
sh $t2, 4($t1)  # Emmagatzema una paraula de 16 bits a memòria  Mem(.....): .....
lw $t1, 4($t1)  # Carrega al registre la paraula (32 bits)      $t1: .....
```

Problema 2: El fragment de codi s'executa en un DLX amb segmentació. Supposeu que una dada es pot escriure en el banc de registres i llegir el seu nou valor en el mateix cicle. Recordeu que els cicles són: IF, ID, EX, MEM i WB

- a) Calculeu el nombre de cicles necessaris per executar aquest codi si no hi ha cap curtcircuit (anticipació d'operands).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
sub r5, r2, r3																								
add r4, r5, r6																								
sub r5, r4, r8																								
add r7, r2, r3																								
add r6, r7, r3																								
lw r1, 0(r6)																								
add r3, r1, r4																								
sub r6, r7, r8																								

- b) Calculeu el nombre de cicles si hi ha curtcircuits (anticipació d'operands).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
sub r5, r2, r3																								
add r4, r5, r6																								
sub r5, r4, r8																								
add r7, r2, r3																								
add r6, r7, r3																								
lw r1, 0(r6)																								
add r3, r1, r4																								
sub r6, r7, r8																								

- c) Tracteu de reordenar el codi per aconseguir que el nombre de cicles sigui mínim, si hi ha curtcircuit (anticipació d'operands). Quants cicles són necessaris en aquest cas?.

Problema 3: Representeu en l'estàndard IEEE 754 (puny flotant) de simple precisió els números 14.0 i -3.5

14.0:

-3.5:

Problema 4: Suposeu que a la memòria de la màquina senzilla hi ha els continguts següents:

@	Contingut	@	Contingut
0	add 100,101	100	1
1	mov 100, 102	101	65536
2	Cmp 100, 102	102	0

Suposant que el **PC** conté inicialment un **0**,

- a) Quin és el contingut final de les posicions de memòria involucrades en les instruccions?

@	Contingut
100	
101	
102	

- b) Com i quan varia el valor d'FZ?.

Descriviu el cicle d'execució de les següents instruccions suposant les dades inicials de l'apartat anterior i que el programa comença a l'adreça 0.

@	Contingut
0	add 100, 101
1	cmp 101, 102
2	beq 0

Quantes vegades s'executa la instrucció de l'adreça 0?

Problema 1: Si tenim les següents declaracions de variables globals, emmagatzemades a memòria a partir de l'adreça 0x10010000:

```
.data
.word 0x100100004
.byte 0xAA
.half 0x00BB
.word 0x88AACCEE
.half 0x0403
.byte 0x40, 0x50, 0x60, 0x70, 0x80, 0x90
.text
```

- a) Indiqueu el contingut inicial de la memòria, en hexadecimal, segons el format que utilitza el simulador QtSpim si el processador MIPS està treballant en *little endian*. Podeu suposar que les posicions de memòria abans de la càrrega del programa valen zero.

Adreça	Valor (0)	Valor (4)	Valor (8)	Valor (c)
0x10010000	10010004	00bb00aa	88aaccee	50400403
0x10010010	90807060	00000000	00000000	00000000

- c) Quin serà el valor dels registres modificats, en hexadecimal, després d'executar cadascuna de les línies de codi?

la \$t0, a1	# Carregar al registre l'adreça de memòria	\$t0: 0x10010000
lw \$t1, 0(\$t0)	# Carregar al registre la paraula (32 bits)	\$t1: 0x10010004
lb \$t2, 2(\$t1)	# Carregar al registre el byte (fa l'extensió del signe)	\$t2: 0xfffffbb
sra \$t2, \$t2, 2	# Desplaçament lògic a la dreta (2)	\$t2: 0xfffffee
sh \$t2, 4(\$t1)	# Emmagatzemar una paraula de 16 bits a memòria	Mem(10010008): 0x88aafce
lw \$t1, 4(\$t1)	# Carregar al registre la paraula (32 bits)	\$t1: 0x88aafce

Problema 2: El fragment de codi s'executa en un DLX amb segmentació. Supposeu que una dada es pot escriure en el banc de registres i llegir el seu nou valor en el mateix cicle. Recordeu que els cicles són: IF, ID, EX, MEM i WB

- a) Calculeu el nombre de cicles necessaris per executar aquest codi si no hi ha cap curtcircuit (anticipació d'operands).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
sub r5, r2, r3	IF	ID	EX	Mem	WR																			
add r4, r5, r6		IF	X	X	ID	EX	Mem	WR																
sub r5, r4, r8					IF	X	X	ID	EX	Mem	WR													
add r7, r2, r3								IF	ID	EX	Mem	WR												
add r6, r7, r3									IF	X	X	ID	EX	Mem	WR									
lw r1, 0(r6)												IF	X	X	ID	EX	Mem	WR						
add r3, r1, r4															IF	X	X	ID	EX	Mem	WR			
sub r6, r7, r8																		IF	ID	EX	Mem	WR		

- b) Calculeu el nombre de cicles si hi ha curtcircuits (anticipació d'operands).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
I1 sub r5, r2, r3	IF	ID	EX	Mem	WR																			
I2 add r4, r5, r6		IF	ID	EX	Mem	WR																		
I3 sub r5, r4, r8			IF	ID	EX	Mem	WR																	
I4 add r7, r2, r3				IF	ID	EX	Mem	WR																
I5 add r6, r7, r3					IF	ID	EX	Mem	WR															
I6 lw r1, 0(r6)						IF	ID	EX	Mem	WR														
I7 add r3, r1, r4							IF	ID	X	EX	Mem	WR												
I8 sub r6, r7, r8								IF	X	ID	EX	Mem	WR											

- c) Tracteu de reordenar el codi per aconseguir que el nombre de cicles sigui mínim, si hi ha curtcircuit (anticipació d'operands). Quants cicles són necessaris en aquest cas?

Només hi ha un cas a tractar. La I8 no depèn de l'anterior. Si la posem entre la I6 i la I7, aconseguirem executar el tros de codi amb només 12 cicles

Problema 3: Representeu en l'estàndard IEEE 754 (punt flotant) de simple precisió els números 14.0 i -3.5

14.0 : 41600000h: 0100 0001 0110 0000 0000 0000 0000 0000 b; S:0; Exp: 1000 0010 (130); Mant: 1.1100...

-3.5 : C0600000h: 1100 0000 0110 0000 0000 0000 0000 0000 b; S:1; Exp: 1000 0000 (128); Mant: 1.1100...

Problema 4: Suposeu que a la memòria de la màquina senzilla hi ha els continguts següents:

@	Contingut	@	Contingut
0	add 100,101	100	1
1	mov 100, 102	101	65536
2	cmp 100, 102	102	0

Suposant que el **PC** conté inicialment un **0**,

a) Quin és el contingut final de les posicions de memòria involucrades en les instruccions?

@	Contingut
100	1
101	1
102	1

b) Com i quan varia el valor d'**FZ**?

Varia després de l'execució de cadascuna de les operacions realitzades. Si el resultat de l'operació val zero es posa a 1

Add 100, 101 :  $M(101) = M(100) + M(101) = 1 + 0 = 1$ ; FZ = 0;

Mov 100, 102 :  $M(102) = M(100) = 1$ ; FZ = 0;

Cmp 100, 102 :  $M(100) - M(102) = 1 - 1 = 0$ ; FZ = 1;

Descriviu el cicle d'execució de les següents instruccions suposant les dades inicials de l'apartat anterior i que el programa comença a l'adreça 0.

@	Contingut
0	add 100, 101
1	cmp 101, 102
2	beq 0

c) Quantes vegades s'executa la instrucció de l'adreça **0**?

Add 100, 101:  $M(101) = M(100) + M(101) = 1 + 0 = 1$

Cmp 101, 102:  $M(101) - M(102) = 1 - 0 = 1$ ; FZ = 0

Beq 0: Salta si (FZ=1). No salta ja que val 0

La instrucció de l'adreça 0 s'executa només un cop.

## Problema 1:

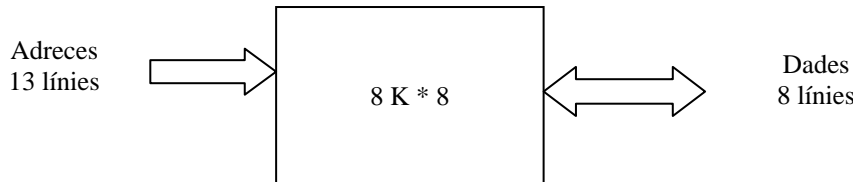
Es disposa de 4 xips de ROM:

- 2 xips 2K x 4 bits (11 bits)
- 1 xip 4K x 8 bits (12 bits)
- 1 xip 1K x 16 bits (10 bits)

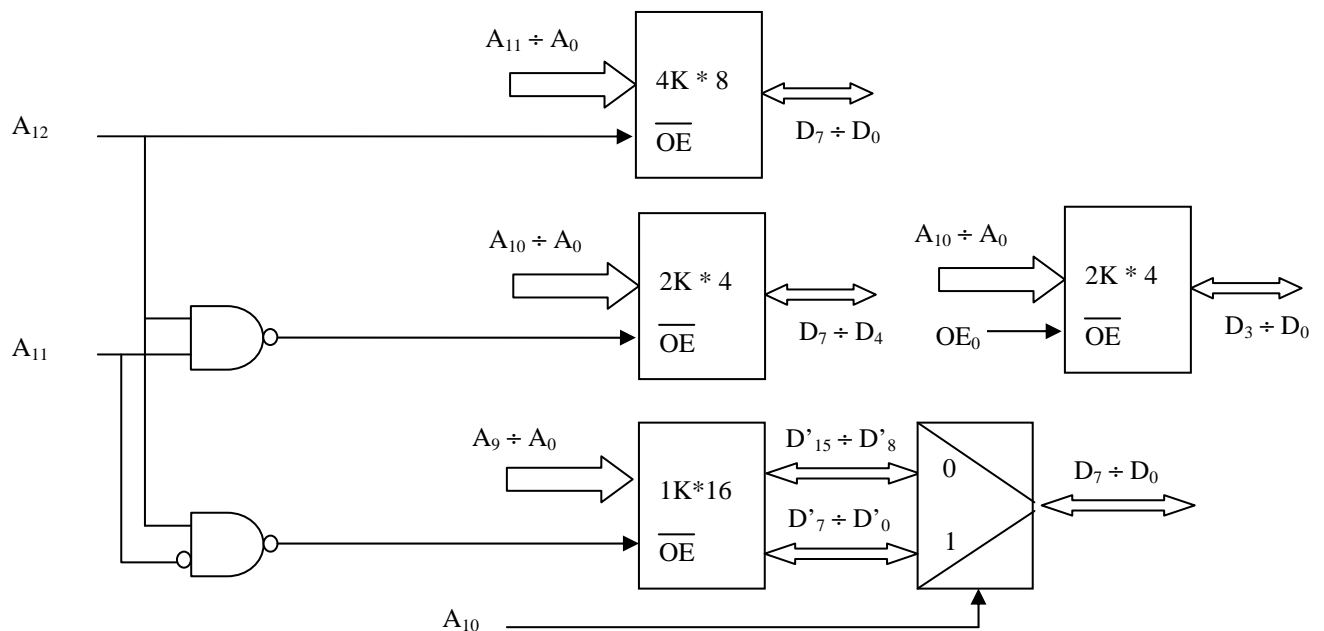
Es vol implementar una única ROM de 8 bits i maximitzar la seva capacitat d'emmagatzematge, utilitzant només els 4 xips de què es disposa.

Es demana:

- a) Especifiqueu les entrades, sortides i grandària en kilobytes de la ROM de 8 bits a implementar.



- b) Dibuixeu el circuit complet incloent totes les entrades i sortides de la ROM. Es disposa de busos, comportes, multiplexors, buffers, decodificadors, etc. Justifiqueu la solució proposada.



Problema 2: Un ordinador té una unitat de memòria de 4 k paraules i una memòria cau de 256 paraules. La memòria cau és associativa per conjunts, amb 16 conjunts i 2 particions per conjunt. Quan es necessita una substitució a la memòria cau, en el conjunt corresponent es reemplaça la partició més antiga.

- c) Format de l'adreça

	Etiqueta	Conjunt	Paraula
Nº de bits	5	4	3

- d) Suposant que inicialment la memòria cau és buida. Ompliu la taula senyalant les fallades i substitucions que es produiran a la memòria cau si es llegeixen consecutivament les adreces de la MP.

Adreça	Etiqueta		Conjunt	Fallada (si/no)	Bloc substituït
	L1	L2			
0000 0000 1111	0		1	Si	
0000 0100 1111	0		9	Si	
0001 0000 1111	2	0	1	Si	
0001 0000 1100	2	0	1	No	
0010 1000 1111	5	2	1	Si	0
0000 0000 1101	0	5	1	Si	2

Problema 3: Sigui un computador capaç d'executar 10 MIPS ( $10^7$  instruccions per segon). Es vol connectar a l'ordinador, un sol perifèric amb una velocitat de transferència de 20.000 bytes / s. i sobre el qual es realitzen operacions de lectura de blocs de 1.024 bytes. Es pretén veure el comportament de la parella ordinador - perifèric davant les diferents tècniques d'entrada/sortida (programada, mitjançant interrupcions i per DMA)

Se sap que:

- La rutina de transferència d'E / S programada consta de 10 instruccions.
- La rutina de tractament d'interrupció en l'E/S mitjançant interrupcions consta de 20 instruccions.
- La rutina de inicialització del DMA consta de 8 instruccions. I en cada operació d'escriptura d'una dada a memòria el controlador ocupa els busos durant 500 ns.

Es demana que digueu el nombre d'instruccions d'altres processos que pot realitzar l'ordinador durant cadascun dels tipus d'E/S previstos.

El perifèric subministra 20.000 bytes/s.  $\rightarrow 1/20000 = 50 \text{ us}$  (enviarà un byte cada 50 us)

El blocs son de 1024 bytes, la transferència de tot el bloc, tardarà:

$$1.024 \text{ bytes} * 50 \text{ us/bytes} = 51.200 \text{ us}$$

I es podran executar:  $51.200 \text{ us} * 10^7 \text{ instruccions/s} = 512.000 \text{ instruccions}$

1) Entrada i sortida per enquesta:

La CPU estarà tota l'estona preguntant si està apunt o no. No podrà fer res més.

2) Per interrupcions:

La RSI està formada per 20 instruccions, que s'executaran cada vegada que hi hagi una nova dada disponible. Al ser blocs de 1024 bytes, executarem un total de  $20 * 1024 \text{ bytes} = 20.480$  instruccions dedicades a la transferència de les dades.

El nombre d'instruccions d'altres processos que es podran executar son:

$$512.000 - 20.480 = 491.520 \text{ instruccions}$$

3) Per DMA

La CPU dedicarà el temps necessari per poder programar el controlador de DMA = 8 instruccions

El DMA utilitza el robatori de cicle, el perifèric ocupa els bussos durant 500 ns per a cadascuna de les transferències.

Per llegir els 1024 bytes tardarà:  $1024 \text{ bytes} * 500 \text{ ns} = 512 \text{ us}$ .

Amb aquest temps, la CPU podrà executar  $512 \text{ us} / 10 \text{ MIPS} = 5.120$  instruccions

En total, per fer la transferència gastarà  $5120 + 8 = 5.128$  instruccions

La diferència entre les instruccions serà:

$$512.000 - 5.128 = 506.872 \text{ instruccions que podrà executar un altre procés.}$$

Resum:

Tipus d'E/S	Instruccions que s'executen per fer la transferència	Instruccions per altres processos
Enquesta	512.000	0
Interrupció	20.480	491.520
DMA	5.128	506.872