

23 de gener de 2012

NOM: _____

SIGNATURA:

- La durada de l'examen és de 2 h 30 min.
- Escriviu el nom i els cognoms en lletres MAJÚSCULES i signeu en TOTS els fulls.
- Cal respondre a l'espai assignat per a això.
- No es permeten calculadores ni apunts.
- Heu de romandre en silenci durant la realització de l'examen.
- No es pot abandonar l'examen fins que el professor ho indique.
- Heu de tenir una identificació damunt la taula a la vista del professor (DNI, carnet UPV, targeta de resident, etc.)

The diagram illustrates a 2-bit counter implemented with two D flip-flops. The counter is driven by a common clock signal (CLK) and a count input (Cnt). The output of the counter is a 2-bit value, Y, consisting of bits Q_0 and Q_1 .

Flip-Flop 0 (Top):

- CLK:** Connected to the clock input of the flip-flop.
- D:** The D input is the logical OR of the Cnt signal and the output of a 3-input AND gate.
- Q:** The output of the flip-flop is Q_0 .
- Q₀:** The output of the flip-flop is Q_0 .

Flip-Flop 1 (Bottom):

- CLK:** Connected to the clock input of the flip-flop.
- D:** The D input is the logical OR of the outputs of three 3-input AND gates.
- Q:** The output of the flip-flop is Q_1 .
- Q₁:** The output of the flip-flop is Q_1 .

Logic Gates:

- 3-input AND gates:** Three 3-input AND gates are used to generate the D input of the second flip-flop. Their inputs are combinations of the Cnt signal and the outputs of the first flip-flop (Q_0).
- OR gates:** Two OR gates are used to generate the D inputs of the flip-flops. One OR gate takes the Cnt signal and the output of a 3-input AND gate. The other OR gate takes the outputs of the three 3-input AND gates.

Obeniu-ne les taules d'excitació i d'eixida, la taula d'estats i el diagrama d'estats, i calculeu la seqüència de valors de l'eixida **Y** per a la seqüència de valors en l'entrada **Cnt** 1 – 1 – 1 – 0 – 0 – 0 – 1 – 1 – 1 tenint en compte que l'estat inicial del sistema és **Q₁ Q₀ = 00**.

Taules d'excitació i d'eixida:

Funció d'excitació;

$$D_1 = Q_1 / \text{Cnt} + /Q_1 Q_0 \text{Cnt} + Q_1 / Q_0 \text{Cnt}$$

$$D_0 = \text{Cnt} \oplus Q_0$$

Funció d'eixida:

$$Y = Q_1 Q_0$$

Taula d'excitació:

$Q_1 Q_0 \text{Cnt}$	$D_1 D_0$
0 0 0	0 0
0 0 1	0 1
0 1 0	0 1
0 1 1	1 0
1 0 0	1 0
1 0 1	1 1
1 1 0	1 1
1 1 1	0 0

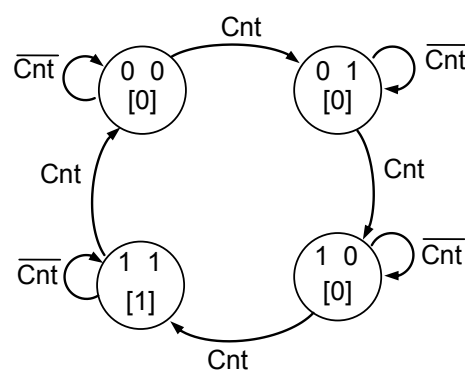
Taula d'eixida:

$Q_1 Q_0$	Y
0 0	0
0 1	0
1 0	0
1 1	1

Taula d'estats:

	<i>Estat següent</i>		
<i>Estat actual</i>	<i>Entrada Cnt</i>		<i>Eixida</i>
$Q_1 Q_0$	$\text{Cnt} = 0$	$\text{Cnt} = 1$	Y
0 0	0 0	0 1	0
0 1	0 1	1 0	0
1 0	1 0	1 1	0
1 1	1 1	0 0	1

Diagrama d'estats:



Examen parcial d'FCO – Temes del 5 al 7

23 de gener de 2011

COGNOMS: _____

NOM: _____

DNI: _____

SIGNATURA: _____

Eixida del sistema per a la seqüència de valors següents en l'entrada Cnt:

1 – 1 – 1 – 0 – 0 – 0 – 1 – 1 – 1

Considerem que l'estat inicial del sistema és 00, de manera que la seqüència de valors en l'eixida del sistema per a la seqüència d'entrada indicada és:

0 – 0 – 0 – 1 – 1 – 1 – 0 – 0 – 0

2. (2 punts) Dissenyau un SSS que implemente un comptador binari de 2 bits reversible governat per una entrada de selecció del sentit del compte anomenada E. Quan E=1, el comptador compta en sentit ascendent; quan E=0, el comptador compta en sentit descendent. Es demana:

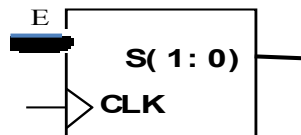
A) (0,25 punts) Interfície de l'SSS amb les entrades i eixides.

B) (1 punt) Diagrama d'estats.

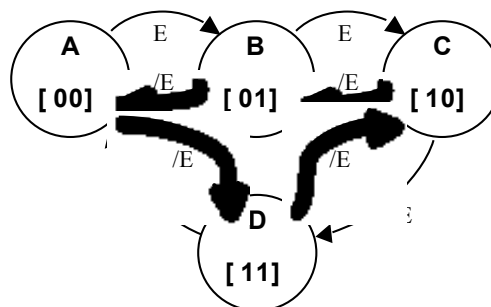
C) (0,75 punts) Taula d'estats amb la codificació d'estats.

D)

a) Interfície: el circuit és un comptador senzill que té com a entrades externes el senyal de rellotge i el senyal E per a indicar el sentit del compte. Com a eixides, té dues eixides d'un bit cadascuna.



b) Diagrama d'estats:



c) Taula d'estats amb els estats codificats:

Estat actual		Estat següent		Eixides
Sense codificar	Codificat	E		
	Q_1Q_0	0	1	S_1S_0
A	00	11	01	00
B	10	00	10	01
C	10	01	11	10
D	11	10	00	11

3. (2 punts) Donats els nombres decimals $A = 125$ i $B = -56$,

- A) (0,5 punts) Representeu-los per mitjà de 8 bits amb el conveni de representació Ca2, detallant totes les operacions que calguen per a obtenir-ne la representació.
- B) (0,75 punts) Realitzeu l'operació $A+B$ amb el conveni de representació Ca2 utilitzant 8 bits, i indiqueu clarament si hi ha desbordament, justificant-ho.
- C) (0,75 punts) Realitzeu l'operació $A-B$ amb el conveni de representació Ca2 utilitzant 8 bits, i indiqueu clarament si hi ha desbordament, justificant-ho.

A) Representació d'A i de B en Ca2 utilitzant 8 bits:

$$125_{10} = 01111101_2 = A (+125)$$

$$56_{10} = 00111000_2 \quad \text{Ca2 } (00111000) = 11001000_2 = B (-56)$$

B) Resultat de l'operació $R = A+B$ utilitzant 8 bits:

$$\boxed{11}111000$$

$$A \rightarrow 01111101_{\text{Ca2}} = 64+32+16+8+4+1=125_{10}$$

$$B \rightarrow + \underline{11001000}_{\text{Ca2}} = -128+64+8 = -56_{10}$$

$$R \rightarrow 01000101_{\text{Ca2}} = 64+4+1=69_{10}$$

Com que els dos últims bits de ròssec són iguals, no hi ha desbordament i el resultat és correcte.

C) Resultat de l'operació $R = A-B$ utilitzant 8 bits:

El primer pas és convertir la resta en una suma, la qual cosa es fa de la manera següent: $A-B = A + \text{Ca2}(B)$

$$\boxed{01}111000$$

$$A \rightarrow 01111101_{\text{Ca2}} = 64+32+16+8+4+1=125_{10}$$

$$\text{Ca2}(B) \rightarrow + \underline{00111000}_{\text{Ca2}} = 32+16+8=56_{10}$$

$$R \rightarrow 10110101_{\text{Ca2}} = 128+32+16+4+1 = 181 \text{ Fora de rang!}$$

Com que els dos últims bits de ròssec són DIFERENTS, sí que hi ha desbordament i el resultat NO és correcte.

Examen parcial d'FCO – Temes del 5 al 7

23 de gener de 2011

COGNOMS: _____

NOM: _____

DNI: _____

SIGNATURA: _____

4. (1,0 punts) Representeu el nombre -1024,3125 amb el format de simple precisió de l'estàndard IEEE754. Detalleu tots els passos realitzats i expresseu el resultat final en hexadecimal.

El primer pas és convertir la quantitat a binari, tant la part entera com la decimal.

La part entera pot convertir-se fent divisions successives, o simplement perquè sabem que 1024 és igual a 2^{10} .

$$1024_{10} = 10000000000_2$$

La part decimal s'obté fent multiplicacions successives:

0,3125	$\times 2 = 0,625$
0,625	$\times 2 = 1,25$
0,25	$\times 2 = 0,5$
0,5	$\times 2 = 1,0$

Per tant, $-1024,3125_{10} = -10000000000,0101_2$

En segon lloc expresseu la quantitat en coma flotant i normalitzada, és a dir, en la forma $\pm 1, M \times 2^E$:

$$-10000000000,0101_2 = -10000000000,0101_2 \times 2^0 = -1,00000000000101_2 \times 2^{10}$$

Una vegada normalitzada la magnitud, cal representar l'exponent en excés 127, és a dir, $E = e + 127 = 10 + 127 = 137 = 10001001$.

La representació de l'exponent en excés 127 no es correspon amb cap de les representacions reservades per a casos especials (0000000 i 11111111), per la qual cosa la quantitat es pot representar com a nombre normalitzat, assignant el bits corresponents als camps S, E i M:

Camp S (signe): 1 (negatiu)

Camp E (exponent): 10001001

Camp M (part fraccionària de la magnitud normalitzada i amb la tècnica del bit implícit. Per a obtenir els 23 bits s'ompli amb zeros per la dreta:

00000000000101000000000

Els 32 bits, tots junts, en l'ordre S, E, M:

1 10001001 00000000000101000000000

Finalment, agrupem els bits de quatre i els substituïm pel dígit hexadecimal equivalent per a obtenir la representació en hexadecimal:

1100 0100 1000 0000 0000 1010 0000 0000 0xC4800A00

El codi següent, escrit en llenguatge de l'assemblador del MIPS R2000, intercala els caràcters de dues cadenes i desa el resultat en una altra cadena.

```
.globl __start
.data 0x10000000
cad1:    .asciiz "AB"
cad2:    .asciiz "123"
.data 0x10001000
long:    .word 1
result:  .space 8

.text 0x00400000
__start:
    la $8, cad1
    la $9, cad2
    li $10, 1
    la $11, result
bucle:
    lb $12, 0($8)
    sb $12, 0($11)
    beq $12, $0, fin
    lb $12, 0($9)
    sb $12, 1($11)
    beq $12, $0, fin
    addi $8, $8, 1
    addi $9, $9, 1
    addi $10, $10, 2
    addi $11, $11, 2
    j bucle
fin:
    la $8, long
    sw $10, 0($8)
.end
```

Basant-vos en el codi anterior, responeu a les preguntes següents.

- 5. (0,5 punts)** Indiqueu el contingut del segment de dades abans d'iniciar-se l'execució, tenint en compte que les dades s'emmagatzemen en format *little-endian*. El contingut ha de especificar-se per a cada byte, amb hexadecimal per a les dades numèriques i amb els caràcters corresponents entre cometes en el cas de les cadenes.

31 ... 24	23 ... 16	15 ... 8	7 ... 0	Adreça
'1'	NULL	'B'	'A'	0x10000000
	NULL	'3'	'2'	0x10000004
				...

Examen parcial d'FCO – Temes del 5 al 7

23 de gener de 2011

COGNOMS: _____ NOM: _____

DNI: _____ SIGNATURA:

0x00	0x00	0x00	0x01	0x10001000
0x00	0x00	0x00	0x00	0x10001004
0x00	0x00	0x00	0x00	0x10001008

- 6. (0,5 punts)** Codifiqueu, mostrant el resultat en hexadecimal, la instrucció lb \$12,0(\$9). Detalleu tots els passos realitzats per obtenir el resultat final en hexadecimal.

La instrucció lb és de tipus I, per la qual cosa té quatre camps (CO de 6 bits, rs de 5 bits, rt de 5 bits i el camp Desp/Imm, de 16 bits).

La sintaxi és lb, rt, desp, (rs).

Com que és una instrucció lb, el codi d'operació corresponent és 0x20. En binari amb 6 bits, és 100000.

rs correspon al registre \$9 (lb \$12,0(\$9)). Expressat en binari amb 5 bits: 01001.

rt correspon al registre \$12 (lb \$12,0(\$9)). Expressat en binari amb 5 bits: 01100.

El desplaçament és 0 (lb \$12,0(\$9)). Expressat en binari amb 16 bits: 0000000000000000

La unió de tots els camps en l'ordre CO rs rt dep dóna la seqüència binària següent:

100000 01001 01100 0000000000000000

Aquesta seqüència es converteix a hexadecimal agrupant els bits de quatre en quatre i substituint-los pel dígit hexadecimal corresponent:

1000 0001 0010 1100 0000 0000 0000 0000

Resultat: 0x812c0000

- 7. (1 punt)** Indiqueu el contingut del segment de dades després de finalitzar l'execució, tenint en compte que les dades s'emmagatzemen en format "little endian". El contingut ha de especificar-se per cada byte, amb hexadecimal per a les dades numèriques i amb els caràcters corresponents entre cometes en el cas de les cadenes.

31	...	24	23	...	16	15	...	8	7	...	0	Adreça
		'1'			NULL			'B'			'A'	0x10000000
					NULL			'3'			'2'	0x10000004

				...
0x00	0x00	0x00	0x05	0x10001000
'2'	'B'	'1'	'A'	0x10001004
0x00	0x00	0x00	NULL	0x10001008

8. (1 punt) Determineu el contingut dels registres següents quan haja finalitzat l'execució del programa. Expresses el contingut en hexadecimal en el cas de les adreces, entre cometes en el cas dels caràcters i en decimal en la resta dels casos.

Registre	Contingut
\$8	0x10001000
\$9	0x10000005
\$10	5
\$11	0x10001008
\$12	0x00 (NULL)

9. (0,25 punts) De quin tipus és la cadena resultant "result"?
- De tipus ASCII, ja que no s'emmagatzema un caràcter 0x00 (NULL) al final del resultat.
 - De tipus ASCII, ja que les cadenes originals també són ASCII.
 - De tipus ASCIIZ, ja que s'emmagatzema un caràcter 0x00 (NULL) al final del resultat just abans d'eixir del bucle.
 - De tipus ASCIIZ, ja que es pren la precaució d'emmagatzemar sempre, una vegada finalitzat el bucle, un caràcter 0x00 (NULL).
 - Cap de les anteriors.

Resposta: C

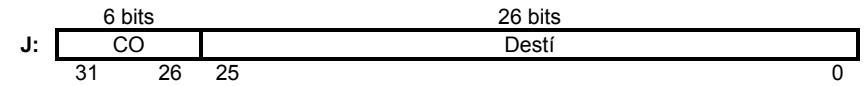
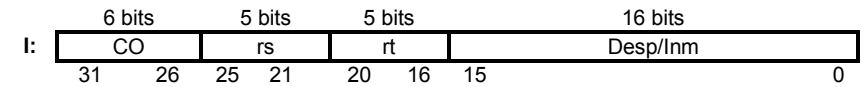
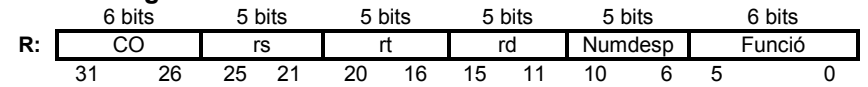
10. (0,25 punts) Quina és la condició d'eixida del bucle?
- Quan es localitza el caràcter 0x00 (NULL) de qualsevol de les dues cadenes ("cad1" i "cad2").
 - Quan es localitza el caràcter 0x00 (NULL) de la primera cadena ("cad1").
 - Quan es localitza el caràcter 0x00 (NULL) de la segona cadena ("cad2").
 - El bucle no té cap de condició d'eixida.
 - Cap de les anteriors.

Resposta: A

Sintaxi i tipus de les instruccions

Sintaxi	Format	Descripció
add rd, rs, rt	R	$rd \leftarrow rs + rt$
addi rt, rs, inm	I	$rt \leftarrow rs + \text{inm}$, la dada immediata és de 16 bits i s'estén el signe
sub rd, rs, rt	R	$rd \leftarrow rs - rt$
mult rs, rt	R	Multiplica rs per rt, i deixa els 32 bits de major pes en el registre HI i els 32 bits de menor pes en LO
div rs, rt	R	Divideix rs entre rt, i deixa el quocient en el registre LO i la resta en el registre HI
and rd, rs, rt	R	$rd \leftarrow rs \text{ and } rt$, l'operació lògica indicada es realitza bit a bit
nor rd, rs, rt	R	$rd \leftarrow rs \text{ nor } rt$, l'operació lògica indicada es realitza bit a bit
xor rd, rs, rt	R	$rd \leftarrow rs \text{ xor } rt$, l'operació lògica indicada es realitza bit a bit
or rd, rs, rt	R	$rd \leftarrow rs \text{ or } rt$, l'operació lògica indicada es realitza bit a bit
andi rt, rs, inm	I	$rt \leftarrow rs \text{ and } \text{inm}$, dada immediata de 16 bits que s'estén amb 16 zeros
ori rt, rs, inm	I	$rt \leftarrow rs \text{ or } \text{inm}$, dada immediata de 16 bits que s'estén amb 16 zeros
xori rt, rs, inm	I	$rt \leftarrow rs \text{ xor } \text{inm}$, dada immediata de 16 bits que s'estén amb 16 zeros
sll rd, rt, desp	R	$rd \leftarrow rt \ll \text{desp}$, desplaçament a esquerres, a mesura que desplaça s'ompli amb 0
srl rd, rt, desp	R	$rd \leftarrow rt \gg \text{desp}$, desplaçament a dretes, a mesura que desplaça s'ompli amb 0
sra rd, rt, desp	R	$rd \leftarrow rt \gg \text{desp}$, desplaçament a dretes, a mesura que desplaça s'estén el bit de signe
Sintaxi	Format	Descripció
lw rt, desp(rs)	I	$rt \leftarrow M[\text{desp} + rs]$, carrega una paraula de 32 bits. El desplaçament (desp) és de 16 bits i s'estén el signe
lh rt, desp(rs)	I	$rt \leftarrow M[\text{desp} + rs]$, carrega mitja paraula (16 bits) i estén el signe
lb rt, desp(rs)	I	$rt \leftarrow M[\text{desp} + rs]$, carrega un byte (8 bits) i estén el signe
sw rt, desp(rs)	I	$M[\text{desp} + rs] \leftarrow rt$
sh rt, desp(rs)	I	$M[\text{desp} + rs] \leftarrow rt$, emmagatzema la part baixa (16 bits) d'rt en memòria
sb rt, desp(rs)	I	$M[\text{desp} + rs] \leftarrow rt$, emmagatzema el byte menys significatiu d'rt en memòria
lui rt, inm	I	$rt31...16 \leftarrow \text{inm}$, $rt15...0 \leftarrow 0$
Sintaxi	Format	Descripció
mfhi rd	R	$rd \leftarrow HI$
mflo rd	R	$rd \leftarrow LO$
mthi rs	R	$HI \leftarrow rs$
mtlo rs	R	$LO \leftarrow rs$
Sintaxi	Format	Descripció
slt rd, rs, rt	R	Si $(rs < rt)$ llavors $rd \leftarrow 1$ si no $rd \leftarrow 0$
slti rt, rs, inm	I	Si $(rs < \text{inm})$ llavors $rt \leftarrow 1$ si no $rt \leftarrow 0$
Sintaxi	Format	Descripció
beq rs, rt, etiqueta	I	Si $(rs == rt)$ $PC \leftarrow \text{etiqueta}$. Si es compleix la condició salta a l'adreça etiqueta
bne rs, rt, etiqueta	I	Si $(rs != rt)$ $PC \leftarrow \text{etiqueta}$. Si es compleix la condició salta a l'adreça etiqueta
Sintaxi	Format	Descripció
j etiqueta	J	$PC \leftarrow \text{etiqueta}$, salta a l'adreça etiqueta
jal etiqueta	J	$\$31 \leftarrow PC + 4$, $PC \leftarrow \text{etiqueta}$, salta a l'adreça etiqueta i desa prèviament l'adreça de retorn en \$31
jr rs	R	$PC \leftarrow rs$, salta a l'adreça continguda en el registre rs

Codificació segons el format



Codis d'operació i funció

Instrucció	CO
addi	0x08
andi	0x0C
beq	0x04
bne	0x05
j	0x02
jal	0x03
lb	0x20
lh	0x21
lui	0x0F
lw	0x23
ori	0x0D
sb	0x28

6 bits

Instrucció	CO
sh	0x29
sw	0x2B
xori	0x0E

6 bits

Instrucció	CO	Funció
add	0x00	0x20
and	0x00	0x24
div	0x00	0x1A
jr	0x00	0x08

6 bits

Instrucció	CO	Funció
mfhi	0x00	0x10
mflo	0x00	0x12
mthi	0x00	0x11
mtlo	0x00	0x13
mult	0x00	0x18
nor	0x00	0x27
or	0x00	0x25
sll	0x00	0x00
slt	0x00	0x2A
srl	0x00	0x02
sub	0x00	0x22
xor	0x00	0x26

6 bits

6 bits

Conveni MIPS

Nom	Núm.
\$zero	0
\$at	1
\$v0, \$v1	2,3
\$a0, ..., \$a3	4,...,7

Nom	Núm.
\$t0, ..., \$t7	8, ..., 15
\$s0, ..., \$s7	16,...,23
\$t8, \$t, 0	24,25
\$k0, \$k1	26,27

Nom	Núm.
\$gp	28
\$sp	29
\$fp	30
\$ra	31

Pseudoinstruccions

Pseudoinstrucció (sintaxi)	Descripció
li rd, inm	$rd \leftarrow \text{inm}_{32 \text{ bits}}$
la rd, etiqueta	$rd \leftarrow \text{etiqueta}_{\text{adreça } 32 \text{ bits}}$