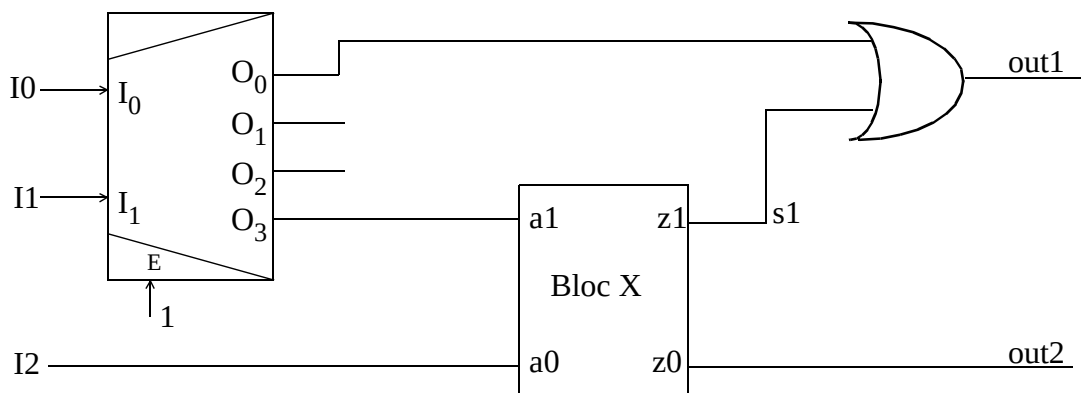
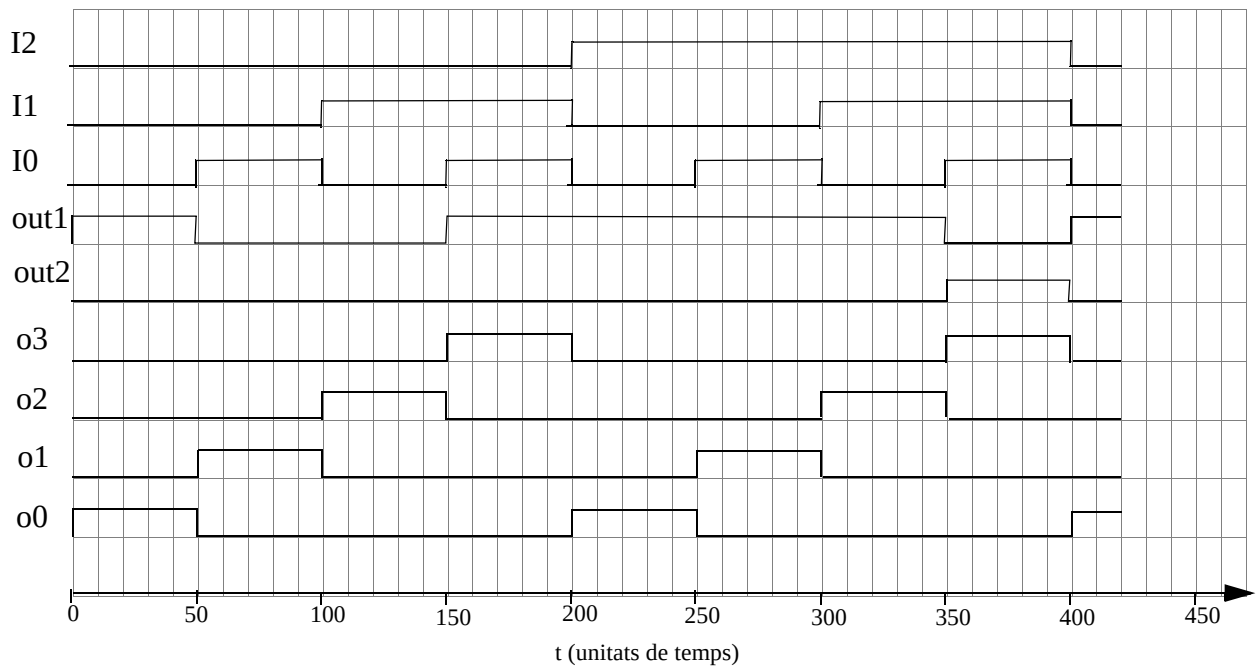


Examen Final d'IC

Curs 2000-01/Q1 - 22 de gener de 2.001

Problema 1 (2 punts)

Donat el circuit de la figura i el cronograma que descriu el seu comportament :



Nota important:

En tots els apartats excepte el d suposeu que els blocs i portes no introdueixen cap retard.

a) (objectiu 6.2.b) Omple els senyals o0, o1, o2 i o3 en el cronograma adjunt. (Resolt en el cronograma)

b) (objectiu 6.2.a) Omple la taula de veritat del circuit sencer

I2	I1	I0	out2	out1
0	0	0	0	1
0	0	1	0	0
0	1	0	0	0
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

c) (objectiu 6.4.a.3) Dóna l'expressió algebraica d'out1 minimitzada per Karnaugh.

I2	I1 I0			
	00	01	11	10
0	1	0	1	0
1	1	1	0	1

$$\text{out1} = \overline{I2} I1 I0 + I2 \overline{I1} + I2 \overline{I0} + \overline{I1} \overline{I0}$$

d) (objectiu 6.2.c) Només per a aquest apartat considera que els retards introduïts per les portes i els blocs són els següents:

Retard introduït pel bloc decodificador = 10 unitats

Retard de la porta OR = 10 unitats

Retard del bloc X = 20 unitats

Escriu la seqüència de blocs i senyals que formen el camí crític (Nota: El camí crític és el camí amb el retard més gran).

(I1, I0) -> Decodificador -> o3 -> a1 -> Bloc X -> s1 -> OR -> out1

Quant val el retard màxim del circuit sencer? $T_{\max} = T_{\text{deco}} + T_x + T_{\text{or}} = 10 + 20 + 10 = 40$ unitats de temps

e) (objectiu 6.4.a.5) Dissenya una ROM que faci la mateixa funció que el circuit sencer. No es demana la implementació sinó el contingut de la ROM.

Quin és el seu tamany? ROM $2^3 * 2$

		0	1
I2	+	0	0
I1		0	0
I0	-	0	1
		0	1
		0	1
		0	1
		1	0
		out2	out1

Problema 2 (2 punts)

Dado el circuito secuencial de la figura, donde SLS es un bloque secuencial que se caracteriza por la siguiente tabla de transición.

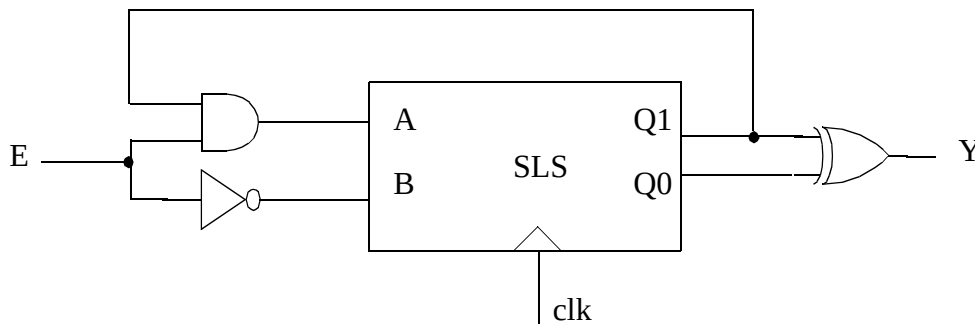


Tabla de transición del bloque SLS:

Q1	Q0	A	B	Q1 ⁺	Q0 ⁺
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	1
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	1	1	1
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	1	1
1	1	1	1	1	0

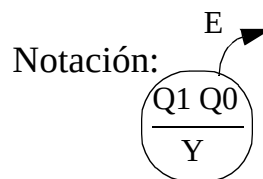
a) Escribe la tabla de verdad
(Objectiu 6.6)

Q1	E	A	B
----	---	---	---

b) Escribe la tabla de transición
(Objectiu 6.6)

Q1	Q0	E	Q1 ⁺	Q0 ⁺
----	----	---	-----------------	-----------------

c) Dibuja el grafo de estados del circuito utilizando la siguiente notación.
(Objectiu 6.6)



Problema 3 (3 punts)

a) (Objectiu 7.3) Tradueix a llenguatge màquina les tres últimes instruccions del següent fragment de programa

```

SEGUENT:      LOAD      vector(R1), R2
               BLE       INCREMENT
               ADD       R3, R2, R3
INCREMENT:    SUBI      R1, #1, R1
               BG        SEGUENT
               STORE     R3, positiu(R0)
    
```

amb aquesta taula de símbols

Nom	Valor
SEGUENT	30h
INCREMENT	33h
vector	01h
positius	00h

- b) (Objectiu 7.1) Completeu les columnes que calgui del cronograma corresponent a les instruccions SUBI i BG.
- c) (Objectiu 7.1) Tot considerant el cronograma proporcionat, quina és la instrucció que s'executa després de BG ?

Problema 4 (3 punts)

Contestar totes les preguntes al full de respostes

Donat el següent codi ensamblador de la MR:

```
                .begin inici
long:           .dw 10
vector:         .dw 1,2,3,4,5,6,7,8,9,10

inici:          LOAD long(R0), R1
                ADD R0, R0, R2
loop:           LOAD long(R1), R3
                ADD R2, R3, R2
                SUBI R1, #1, R1
                BG loop

                .end
```

Continuació del problema 4

(a) [objectiu 8.2] Quants cicles de rellotge fan falta per executar tot el codi usant el graf d'estats del xuletari? (*justifica la resposta*)

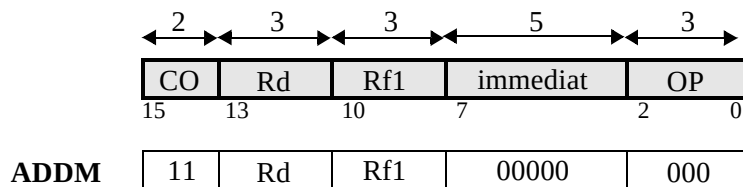
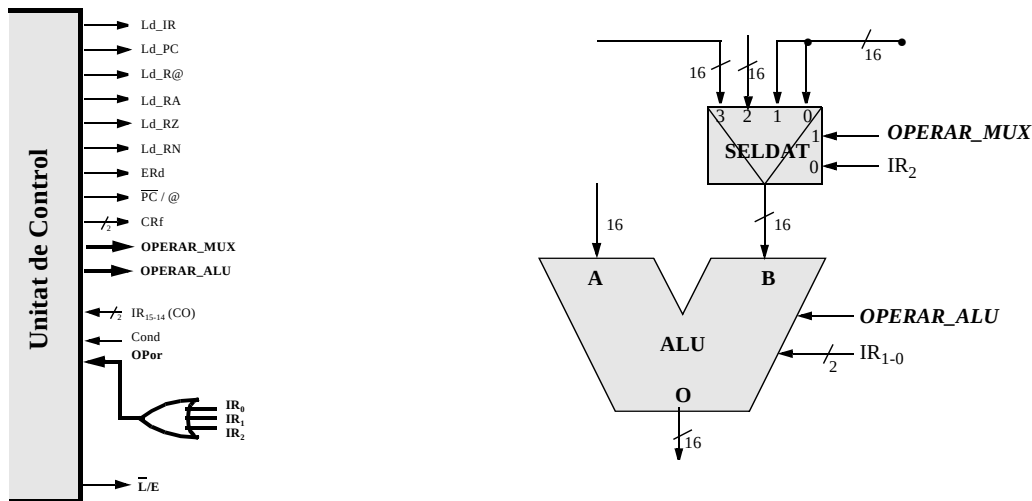
Eliminem la instrucció ADDI de la MR.

(b) [objectiu 7] Es pot fer el mateix que feia la instrucció ADDI amb una de les instruccions restants? (*raona la resposta*).

Afegim una nova instrucció que aprofita la codificació deixada lliure per la ADDI sense fer cap canvi a la Unitat de Proces de la MR. La nova instrucció es diu **ADDM** i a la figura es mostra la seva codificació i funcionalitat. Aquesta instrucció deixa al registre Rd el resultat de sumar Rd amb el contingut de la posició de

memoria indicada a Rf1. Els flags Z i N queden modificats segons el resultat de la operació.

A la Unitat de Control li entra un nou bit que es la OR dels tres bits baixos de l'IR i la sortida *OPERAR* es converteix en dues noves sortides: *OPERAR_MUX* i *OPERAR_ALU* per diferenciar els accessos de lectura a memòria que no fan operació aritmètica (LOAD) dels accessos que sí que en fan (la nova instrucció).



Notació: ADDM (Rf1), Rd

Semàntica: $Rd = Rd + Mem[Rf1]$

(c) [objectiu 9.1] Completar el nou graf d'estats i la taula de sortides associada per que permeti fer les mateixes instruccions que la MR original (sense la ADDI) i la nova ADDM. *Noteu que els únics canvis en els estats originals estan indicats en negreta.*

(d) [objectiu 8.2] Ara volem executar un nou codi que fa el mateix que el de l'apartat (a). Quants cicles de rellotge fan falta? (*justificar la resposta*)

```

                .begin inici
long            .dw 10
vector:         .dw 1,2,3,4,5,6,7,8,9,10

```

```
inici:    LOAD long(R0), R1
          ADD R0, R0, R2
loop:     ADDM (R1), R2
          SUBI R1, #1, R1
          BG loop

          .end
```

Problema 3

a)

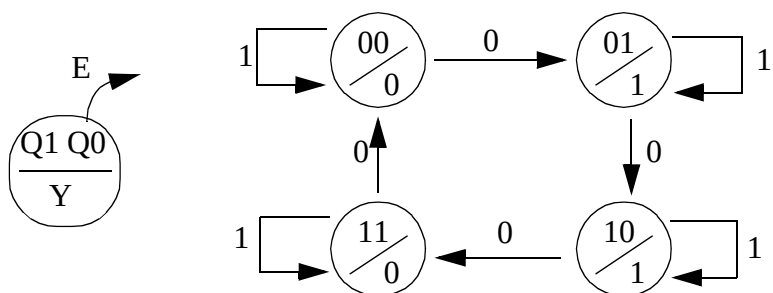
Q1	E	A	B
0	0	0	1
0	1	0	0
1	0	0	1
1	1	1	0

b)

Q1	Q0	E	Q1 ⁺	Q0 ⁺
0	0	0	0	1
0	0	1	0	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

c)

Q1	Q0	Y
0	0	0
0	1	1
1	0	1
1	1	0



Problema 3

a)

	Binari	Hexadecimal
SUBI R1, #1, R1	1100 1001 0000 1001	C909h
BG SEGUENT	1011 1000 0011 0000	B830h
STORE R3, positiu(R0)	0101 1000 0000 0000	5800h

b)

	ADD R3, R2, R3			SUBI R1, #1, R1		BG SEGUENT			
Estat	FETCH	DECO	ARIT	DECO	ARIT	DECO			
Ld_IR			1	0	1	0			
Ld_PC			1	0	1	0			
Ld_R@			0	1	0	1			
Ld_RA			0	1	0	1			
Ld_RZ			1	0	1	0			
Ld_RN			1	0	1	0			
ERd			1	0	1	0			
$\overline{L}/E$			0	0	0	0			
$\overline{PC}/@$			0	x	0	x			
CRf			2	1	2	1			
OPERAR			1	x	1	x			
M@			33h	x	34h	x			
Mout			C909h	x	B830h	x			
Min			0004h	0001h	0000h	0000h			
Din			0103h	x	0000h	x			
PC			33h	34h	34h	35h			
IR			DB44h	C909h	C909h	B830h			
R@			43h	43h	0Ah	0Ah			
RA			00FFh	00FFh	0001h	0001h			
RZ			0	0	0	1			
RN			0	0	0	0			
R1			0001h	0001h	0001h	0000h			
R2			0004h	0004h	0004h	0004h			
R3			00FFh	0103h	0103h	0103h			

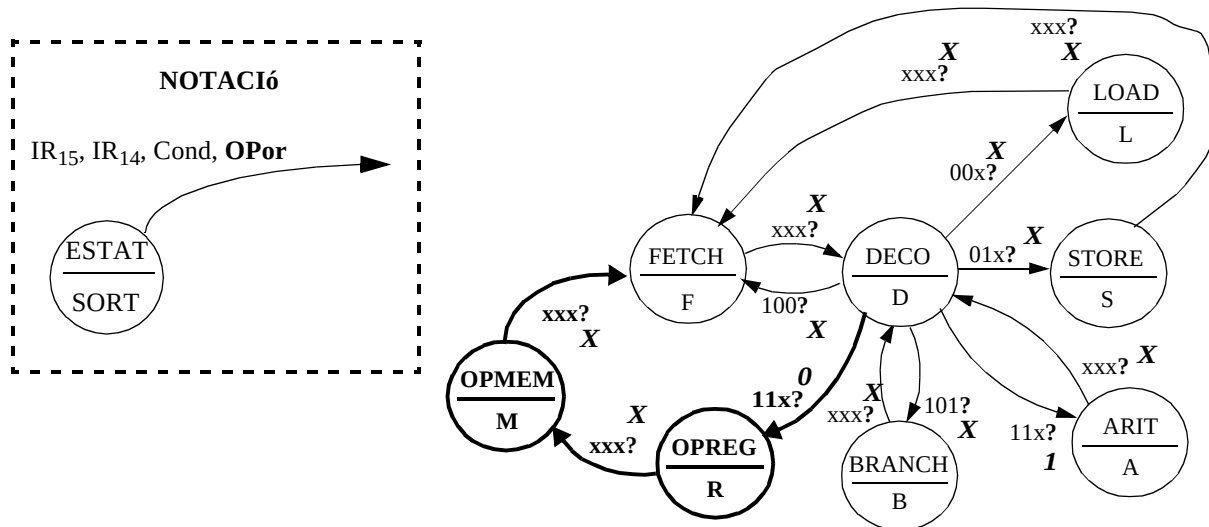
c) La instrucció STORE, ja que el salt no es pren.

Full de respostes problema 4

(a) Cicles: 96 (6 + 9x10)

(b) *SUBI Rf, #-i, Rd fa el mateix que ADDI Rf, #i, Rd excepte quan i = -16*

(c) En el graf d'estats Indicar al costat dels interrogants el valor que haurien de tenir i omplir les columnes R i M de la taula de sortides.



Sortides	F	D	A	L	S	B	R	M
Ld_IR	1	0	1	0	0	1	0	x
Ld_PC	1	0	1	0	0	1	0	0
Ld_R@	0	1	0	0	0	0	0	x
Ld_RA	0	1	0	0	0	0	1	x
Ld_RZ	0	0	1	1	0	0	x	1
Ld_RN	0	0	1	1	0	0	x	1
ERd	0	0	1	1	0	0	0	1
$\overline{L}/E$	0	0	0	0	1	0	0	0
$\overline{PC}/@$	0	x	0	1	1	1	x	1
CRf	x	1	2	x	0	x	0	x
OPERAR_MUX	x	x	1	0	x	x	x	0
OPERAR_ALU	x	x	1	0	x	x	x	1

(d) Cicles: 86 (6 + 8x10)