

**Solución Examen Diseño Lógico**  
**Diciembre 2015**

**EJERCICIO 1**

a)  $A = 00100101$   
 $B = 10010111$

i) Binario natural:

$$A = 1 + 4 + 32 = 37$$

$$B = 1 + 2 + 4 + 16 + 128 = 151$$

ii) Enteros en Magnitud y Signo

$$A = 37$$

$$B = -23$$

iii) Enteros en complemento a 2

$$A = 37$$

$$-B = 01101000 + 1 = 01101001 = 1 + 8 + 32 + 64 = 105$$

$$\Rightarrow B = -105$$

iv) BCD empaquetado

$$A = 25$$

$$B = 97$$

b) B+A:

$$\begin{array}{r} 10010111 \\ 00100101 \\ \hline 10111100 \end{array}$$

no hay overflow dado que se suman dos números de signos opuestos.

B-A:

$$-A = 11011011$$

$$\begin{array}{r} 10010111 \\ 11011011 \\ \hline 01110010 \end{array}$$

hay overflow ya que la suma de dos números negativos no puede dar un número positivo.

## EJERCICIO 2

	00	01	11	10	00	01	11	10
q0	q1	<b>q0</b>	<b>q0</b>	q2		0	1	
q1	<b>q1</b>	q4	--	q3	0			
q2	q1	q0	<b>q2</b>	<b>q2</b>			0	0
q3	q4	<b>q3</b>	q0	<b>q3</b>		1		1
q4	<b>q4</b>	<b>q4</b>	q2	--	0	0		

a) Minimización de estados:

q1	0-4 2-3			
q2	X	0-4 2-3		
q3	X	1-4 3-4	X	
q4	1-4 0-2	V	1-4 0-4	X
	q0	q1	q2	q3

=> q1 es equivalente a q4.

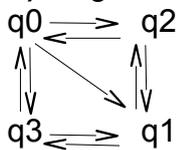
Tabla mínima:

	00	01	11	10	00	01	11	10
q0	q1	<b>q0</b>	<b>q0</b>	q2		0	1	
q1	<b>q1</b>	<b>q1</b>	q2	q3	0	0		
q2	q1	q0	<b>q2</b>	<b>q2</b>			0	0
q3	q1	<b>q3</b>	q0	<b>q3</b>		1		1

Conjuntos de destino:

00	01	11	10
(q1, q0, q2, q3)	(q0, q2) (q1) (q3)	(q0, q3) (q2, q1)	(q2, q0) (q3, q1)

b) Asignación: q0 = 00; q2 = 01; q1 = 11; q3 = 10



Eliminamos la carrera de q0 a q1 utilizando el ciclo por q2 (podría haberse utilizado también el ciclo q1-q3-q0)

La tabla con esta codificación queda entonces:

	00	01	11	10	00	01	11	10
00	01	<b>00</b>	<b>00</b>	01		0	1	
11	<b>11</b>	<b>11</b>	01	10	0	0		
01	11	00	<b>01</b>	<b>01</b>			0	0
10	11	<b>10</b>	00	<b>10</b>		1		1

**PROBLEMA 1:**

Diagrama de estados:

T L tfin / ON/OFF FRIO

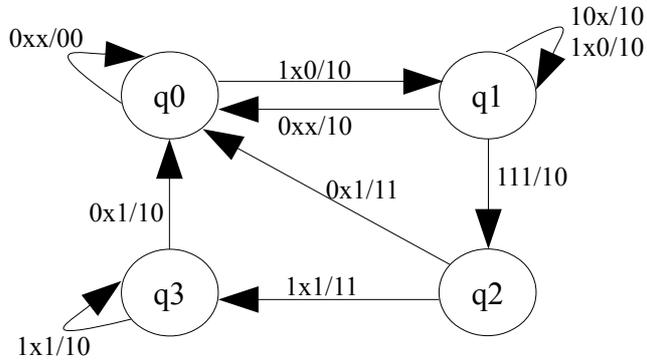
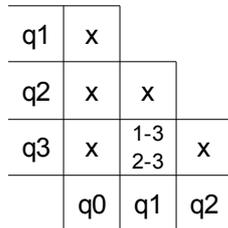


Tabla de estados:

	000	001	011	010	100	101	111	110	000	001	011	010	100	101	111	110
q0	q0	q0	q0	q0	q1	--	--	q1	00	00	00	00	10	--	--	10
q1	q0	q0	q0	q0	q1	q1	q2	q1	10	10	10	10	10	10	10	10
q2	--	q0	q0	--	--	q3	q3	--	--	11	11	--	--	11	11	--
q3	--	q0	q0	--	--	q3	q3	--	--	10	10	--	--	10	10	--

Minimización:



ya es mínimo

Codificación y tabla codificada:

- q0 – 00
- q1 – 01
- q2 – 11
- q3 – 10

	000	001	011	010	100	101	111	110	000	001	011	010	100	101	111	110
00	00	00	00	00	01	xx	xx	01	00	00	00	00	10	xx	xx	10
01	00	00	00	00	01	01	11	01	10	10	10	10	10	10	10	10
11	xx	00	00	xx	xx	10	10	xx	xx	11	11	xx	xx	11	11	xx
10	xx	00	00	xx	xx	10	10	xx	xx	10	10	xx	xx	10	10	xx

Mapas K:

D1:

Q1Q0\L.tfin	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	x	0	0	x
10	x	0	0	x

T=0

Q1Q0\L.tfin	00	01	11	10
00	0	x	x	0
01	0	0	1	0
11	x	1	1	x
10	x	1	1	x

T=1

$D1 = T.Q1 + T.L.tfin$

D0:

Q1Q0\L.tfin	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	x	0	0	x
10	x	0	0	x

T=0

Q1Q0\L.tfin	00	01	11	10
00	1	x	x	1
01	1	1	1	1
11	x	0	0	x
10	x	0	0	x

T=1

$D0 = T.!Q1$

Maxitérmino

ON/OFF:

Q1Q0\L.tfin	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	x	1	1	x
10	x	1	1	x

T=0

Q1Q0\L.tfin	00	01	11	10
00	1	x	x	1
01	1	1	1	1
11	x	1	1	x
10	x	1	1	x

T=1

$ON/OFF = T + Q1 + Q0$  (con una agrupación correcta de minitérminos se llegaría al mismo resultado a través de la suma de los mismos)

FRIO:

Q1Q0\L.tfin	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	x	1	1	x
10	x	0	0	x

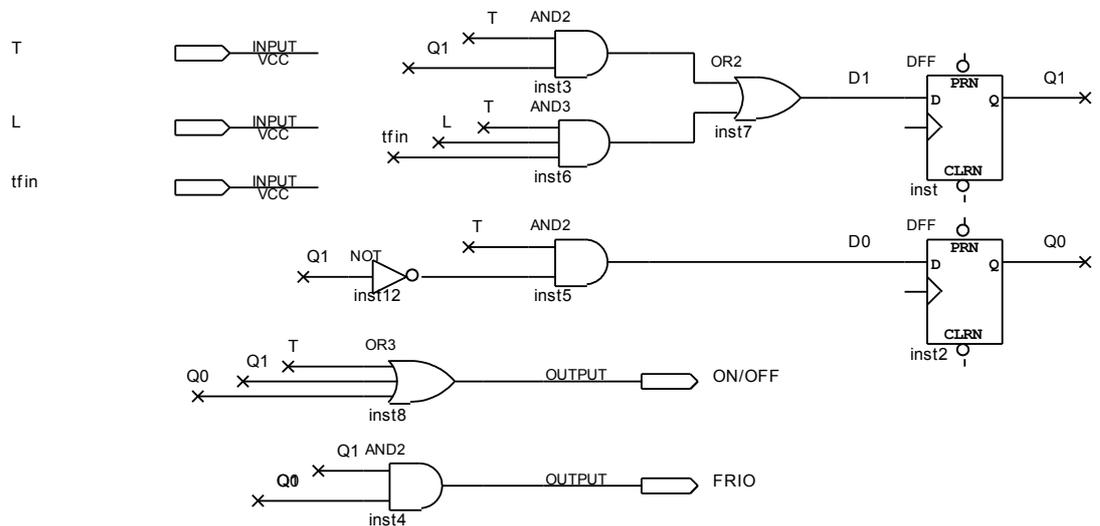
T=0

Q1Q0\L.tfin	00	01	11	10
00	0	x	x	0
01	0	0	0	0
11	x	1	1	x
10	x	0	0	x

T=1

$FRIO = Q1.Q0$

Circuito:



## PROBLEMA 2:

### SECUENCIA RTL:

Module: Solución  
 Input: Pin[3], liberar, ocupar, dout  
 Output: Pout[3], Dir[3], din, ena, valido, lleno  
 Memory: Cont[3], Mem[3]

0. Mem[] 000
1. din = 0  
 ena = liberar  
 Dir[] = Pin[]  
 Cont[] 000  
 (!ocupar; ocupar) / ( 1, 2 )
2. Dir[] = Mem[]  
 Mem [] INC (Mem[] )  
 valido = !dout  
 din = 1  
 ena = 1 ; si había un 1 se vuelve a escribir un 1.  
 Cont[] INC ( Cont[] )  
 lleno = (Cont[]=111) . dout  
 ( (Cont[]=111) or !dout ; ! ((Cont[]=111) or !dout) / ( 1, 2 )

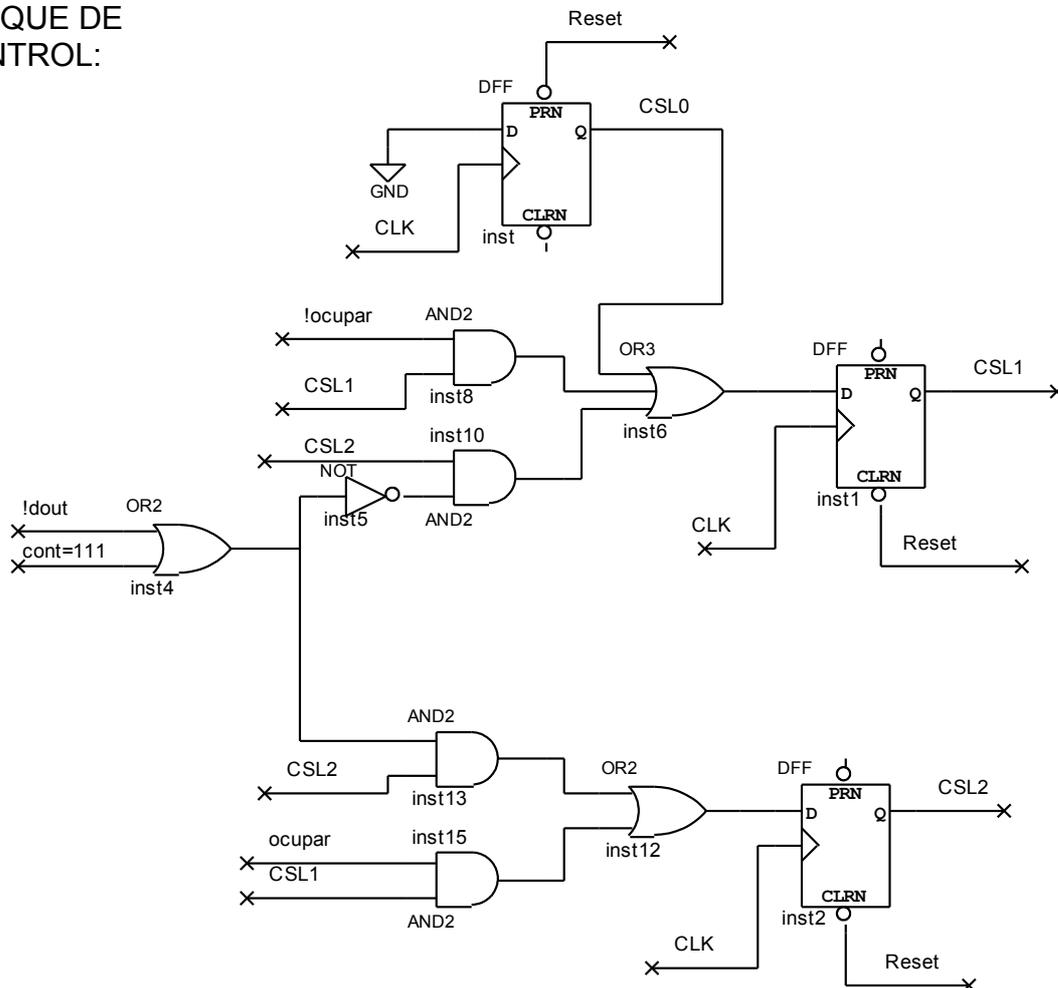
END SEQUENCE

Pout[] = Mem[]

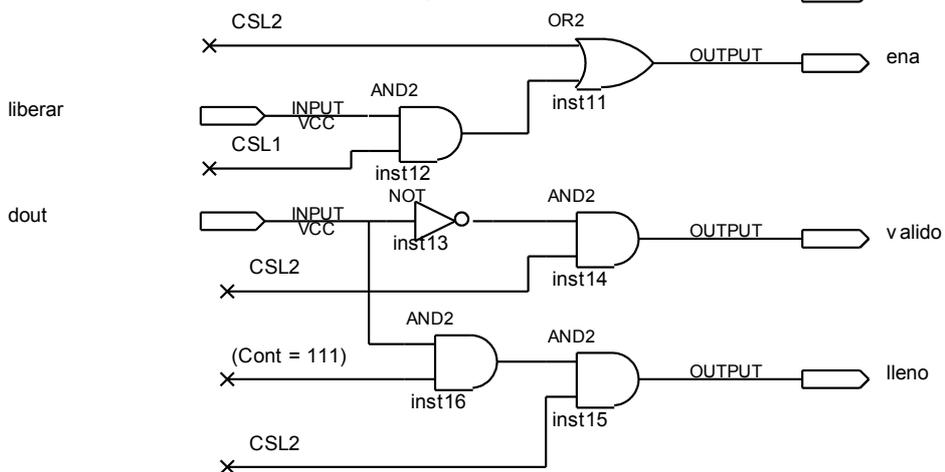
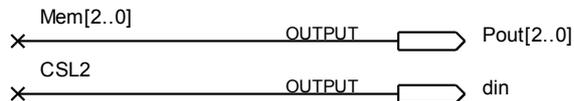
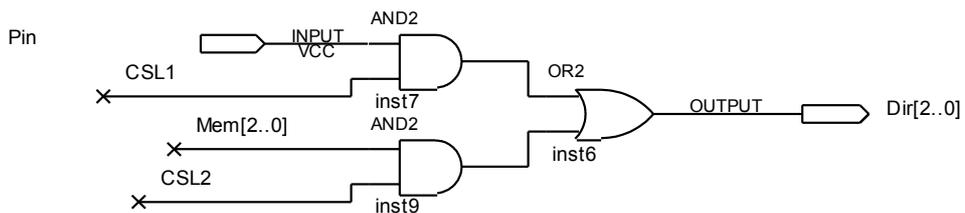
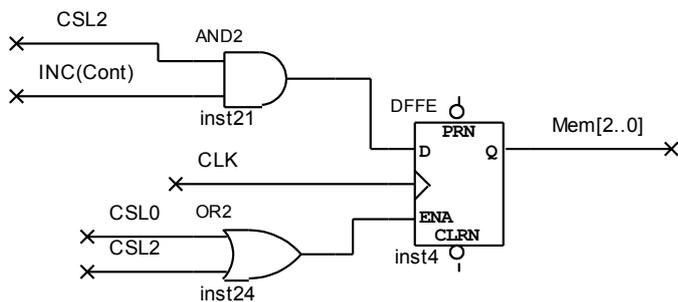
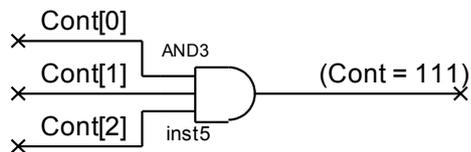
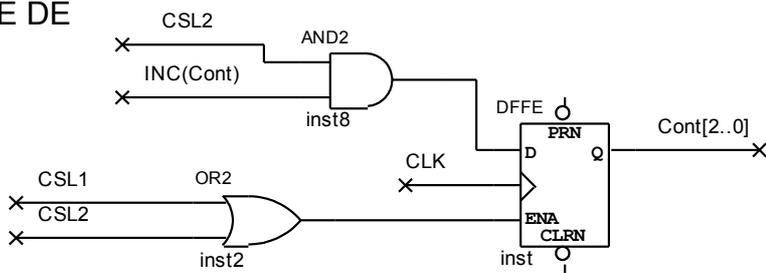
CONTROL RESET (0)

END

### BLOQUE DE CONTROL:



**BLOQUE DE DATOS:**



Nota: INC(Cont) es la salida de un incrementador circular de 3 bits que tiene a **Cont[2..0]** conectado a su entrada.