



EXAMEN PARCIAL DE FUNDAMENTOS DE COMPUTADORES

CURSO 2015-16, PRIMER PARCIAL (CONV. DE JUNIO), 12 DE FEBRERO DE 2016

1. **(1 punto)** Dados los siguientes números: $A = +(27)_{10}$, $B = -(127)_8$, $C = +(74)_{16}$ y $D = +(1111010)_2$
 - a) **(0,5 puntos)** Expréselos en representación en complemento a 2 con 8 bits.
 - b) **(0,5 puntos)** Efectúe las operaciones (A-B) y (-C-D) indicando en cada caso si hay desbordamiento y/o acarreo y el por qué.

2. **(2,5 puntos)** Un depósito de agua cuenta para su llenado con dos electroválvulas (E1 y E2) que suministran un caudal de 50 y 10 litros/minuto respectivamente. El nivel del agua en su interior puede conocerse mediante la lectura de 3 sensores (S1, S2 y S3) tales que:
 - La salida del sensor S1 vale 1, cuando se ha llegado al 100% de la capacidad del depósito.
 - La salida del sensor S2 vale 1, si la cantidad de agua en el depósito supera el 90%.
 - La salida del sensor S3 vale 1, si el nivel se encuentra por encima del 60%.

Se desea diseñar un sistema combinacional que controle la activación de las electroválvulas en función del valor de los sensores según las siguientes especificaciones:

 - Cuando la cantidad de agua existente en el depósito se encuentre entre el 0 y el 60% de la capacidad total, se deberá llenar el depósito a razón de 60 litros/minuto.
 - Si el nivel de agua se encuentra entre el 60 y el 90%, la velocidad de llenado será de 50 litros/minuto.
 - Si el nivel se encuentra entre el 90 y el 100%, se procederá a terminar de llenar el depósito con un caudal de 10 litros/minuto.
 - Si el nivel ha llegado a su punto máximo, es decir al 100%, se deben cerrar ambas electroválvulas.

Se pide:

 - a) **(0,5 puntos)** Obtener la tabla de verdad del sistema.
 - b) **(1 punto)** Utilizando el menor número de puertas NAND, implementar un circuito con el comportamiento anteriormente especificado.
 - c) **(0,5 puntos)** Implementar el control de E1 utilizando un decodificador 3 a 8 y el menor número de puertas lógicas.
 - d) **(0,5 puntos)** Implementar el control de E2 utilizando un multiplexor 4 a 1 y el menor número de puertas lógicas.

3. **(1 punto)** Usando un sumador binario de 8 bits así como las puertas y/o multiplexores que necesite, diseñe un sumador saturado de 8 bits en C2. Su comportamiento es el siguiente:
 - Si al realizar la suma se produce un desbordamiento positivo (es decir, el resultado de la suma sería positivo pero requeriría más de 8 bits para ser representado correctamente en C2), el sumador devuelve el máximo valor representable en C2 con 8 bits.
 - Si al realizar la suma se produce un desbordamiento negativo (es decir, el resultado de la suma sería negativo pero requeriría más de 8 bits para ser representado correctamente en C2), el sumador devuelve el mínimo valor representable en C2 con 8 bits.
 - Si al realizar la suma no se produce desbordamiento alguno (es decir, el resultado de la suma puede representarse correctamente en C2 con 8 bits), el sumador devuelve el resultado de dicha suma.

4. **(2,5 puntos)** Sea un sistema secuencial con una entrada de 2 bits (X), una salida de un bit (Z) y que se comporta según la siguiente expresión:

$$Z(t) = \begin{cases} 1 & \text{si } \underline{X}(t-2) = (01) \text{ y } \underline{X}(t-1) = (10) \\ 0 & \text{en caso contrario} \end{cases}$$

Se pide:

- (1 punto)** Especificar el sistema mediante un diagrama de estados de tipo Moore.
 - (0,5 puntos)** Obtener las tablas de transición y de salida del sistema.
 - (1,5 puntos)** Implementar el sistema con biestables D y una ROM.
5. **(3 puntos)** Utilizando un contador módulo 4 con carga en paralelo y el mínimo número de puertas lógicas, implemente un sistema secuencial cuya salida de 2 bits (\underline{Z}) en función de una entrada de 2 bits (\underline{X}) siga repetidamente las siguientes secuencias:
- Si $\underline{X} = 0$, la salida del sistema seguirá la secuencia **0,1,2,3...**
 - Si $\underline{X} = 1$, la salida del sistema seguirá la secuencia **0,2,1,3....**
 - Si $\underline{X} = 2$, la salida del sistema seguirá la secuencia **0,3,1,2....**

Si el valor de entrada cambia en un cierto ciclo, el sistema seguirá la nueva secuencia a partir del dígito que esté en la salida en dicho ciclo (véase la figura).

X(t)	0	0	0	0	0	2	2	2	2	2	2	1	1	1	1
Z(t)	0	1	2	3	0	1	2	0	3	1	2	0	2	1	3
Secuencia "0123"						Secuencia "0312"						Secuencia "0213"			

Diagrama de Estados 15

tabla 5 up - 0'45

resto - 0'75

11 Febrero 2016

1

① $A = +27$

$|+27| \rightarrow$ divido recursivamente por 2 para encontrar la magnitud representada en binario puro

$$\begin{array}{r} 27/2 \\ 1 \ 13/2 \\ 1 \ 6/2 \\ 0 \ 3/2 \\ 1 \ 1 \end{array} \Rightarrow |27| = \Delta 10416_p$$

$+27 = 011011_{c_2}$ Extensión Sign. 00011011_{c_2}

$B = -(127)_8$

$|-127_8| \rightarrow$ cada dígito se sustituye por la representación binaria con 3 bits

$|-127_8| = 00101011_p = 00101011_{c_2}$

para encontrar -127 , cambiamos el signo:

$$\begin{array}{r} \cancel{10101000} \\ \Delta 10101000 \\ + \\ \hline 11010100_{c_2} \end{array}$$

con 8 bits

$\Delta 0101000_{c_2}$

$C = +24_{16}$

se sustituye cada dígito por su representación binaria con 4 bits

$24_{16} = \boxed{01110100_{c_2}}$

$D = 01111010$

(a)

$$A = 00011011c_2$$

$$B = 10101001c_2$$

$$C = 01110100c_2$$

$$D = 01111010c_2$$

(b)

$$\boxed{A-B} = A + (-B)$$

$$B = 10101001$$

$$\begin{array}{r} -B \Rightarrow 01010110 \\ + 1 \\ \hline 01010111 \end{array}$$

$$\begin{array}{r} + 00011011 \\ 01010111 \\ \hline 01110010 \end{array}$$

desbordamiento, la suma de 2 positivos da un positivo

acarreo → el n.º de bits del resultado y de los operandos es el mismo.

$$\boxed{-C-D} = -C + (-D)$$

$$\begin{array}{r} C = 01110100 \\ -C \Rightarrow 10001011 \\ + 1 \\ \hline 10001100 \end{array}$$

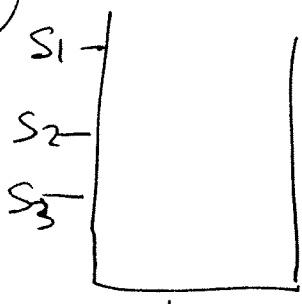
$$\begin{array}{r} D = 01111010 \\ -C \Rightarrow 10001011 \\ + 1 \\ \hline 10000110 \end{array}$$

$$\begin{array}{r} 10001100 \\ 10000110 \\ \hline 00001010 \end{array}$$

desbordamiento → la suma de 2 n.ºs ~~positivos~~ negativos no puede dar un n.º positivo

acarreo → el resultado tiene más bits q. los operandos

2



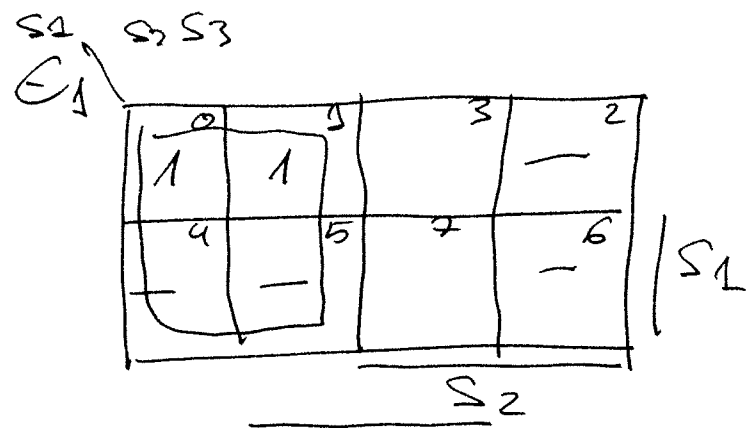
- Es imposible q. se active el S_1 sin q. se hayan activado el S_2 y S_3
 - Es imposible q. se active el S_2 si no se activa el S_3

\Rightarrow hay un conjunto de entradas q. NO se pueden dar \Rightarrow las salidas don't care

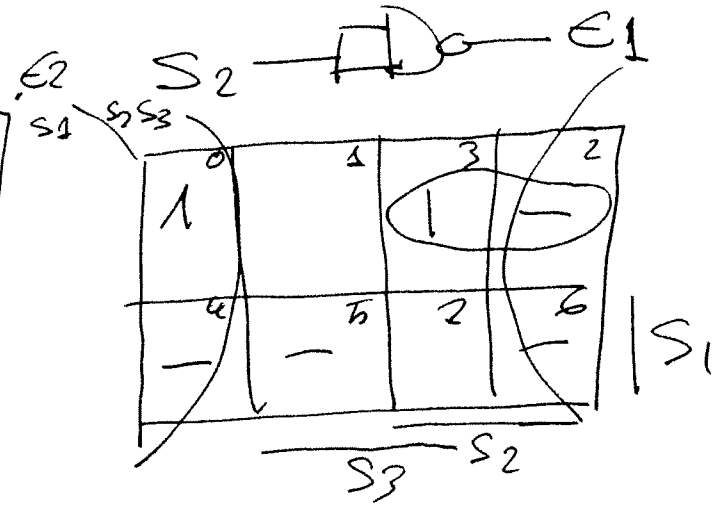
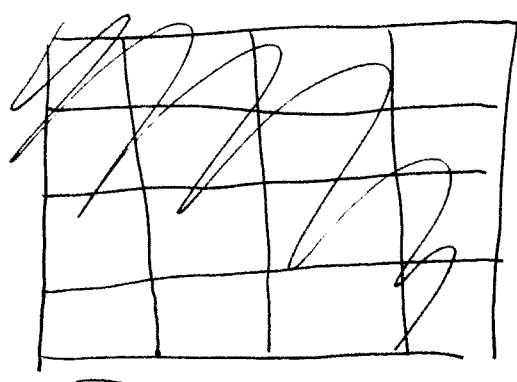
4

S_1	S_2	S_3	E_1	E_2
0	0	0	1	1
0	0	1	1	0
0	1	0	-	-
0	1	1	0	1
1	0	0	-	-
1	0	1	-	-
1	1	0	-	-
1	1	1	0	0

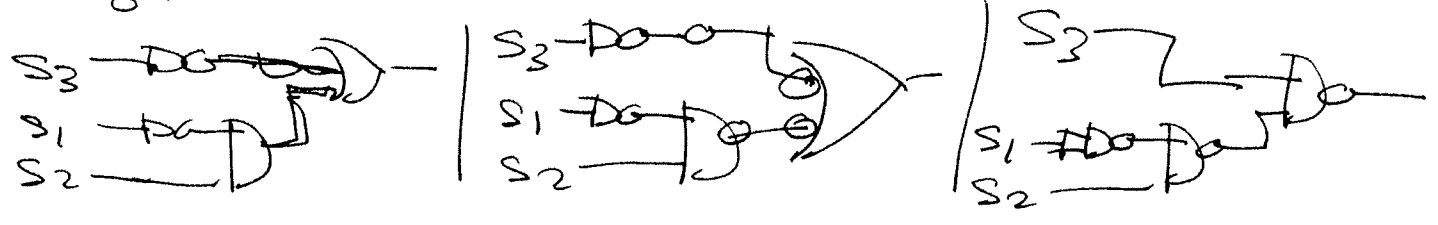
para E_1 y E_2 el valor activo indica la salida abierta



$$E_1 = \frac{S_3}{S_2}$$

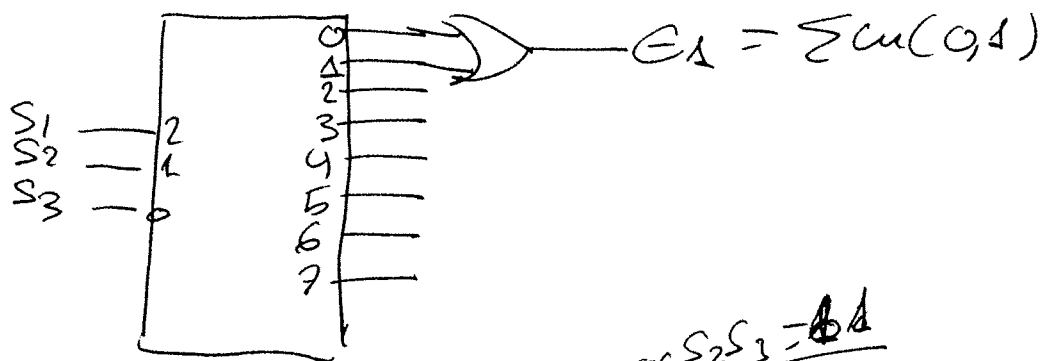


$$E_2 = \overline{S_3} + \overline{S_1} S_2$$

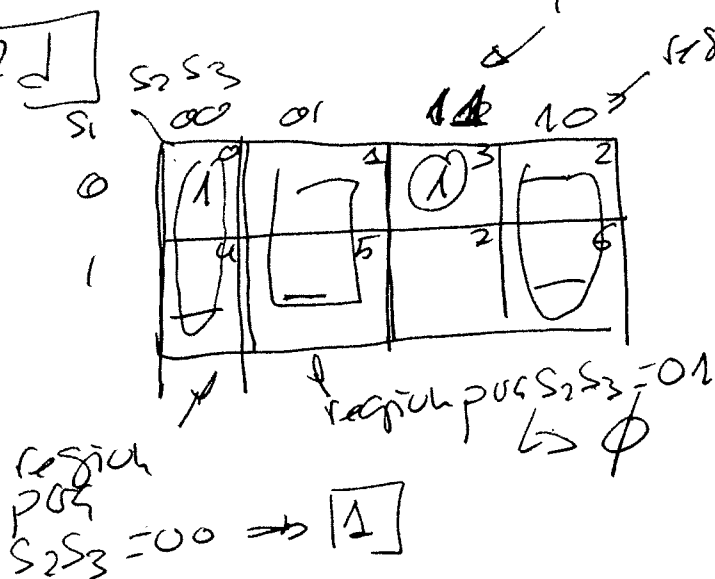


2c

9

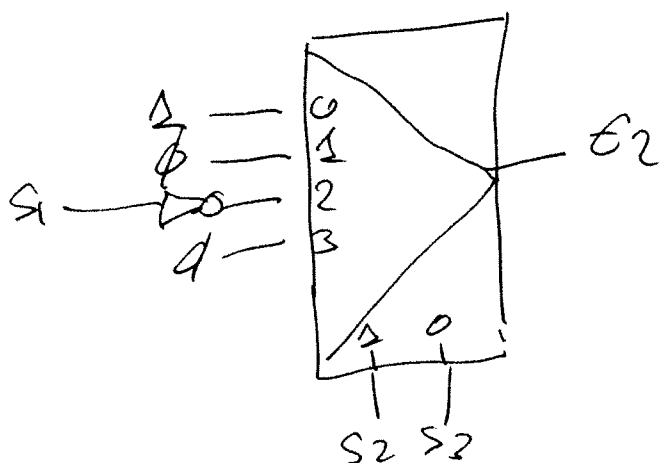


2d



tanto como señales de control del multiplexor $S_2 S_3$

calculo las subregiones del mapa de K. y correspondo con cada una de las posibles combinaciones de S_2 y S_3

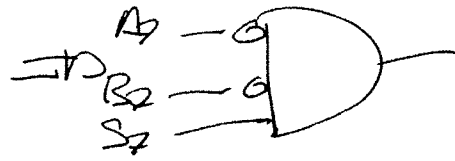


(3) → tenemos q. detecto cuando hay desbordamiento en un sumador de complemento a 2.

→ Desbordamiento ⇒ 2 n.º positivos no pueden dar un negativo $A_7 = 0$

$$B_7 = 0$$

$$S_7 = 1$$



⇒ 2 n.º negativos no pueden dar un n.º positivo $A_7 = 1$

$$B_7 = 1$$

$$S_7 = 0$$

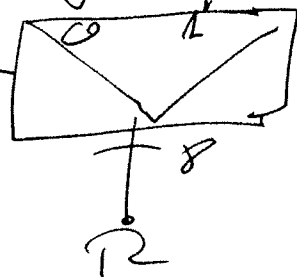
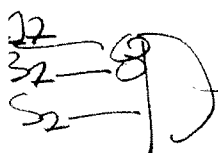
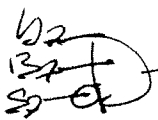
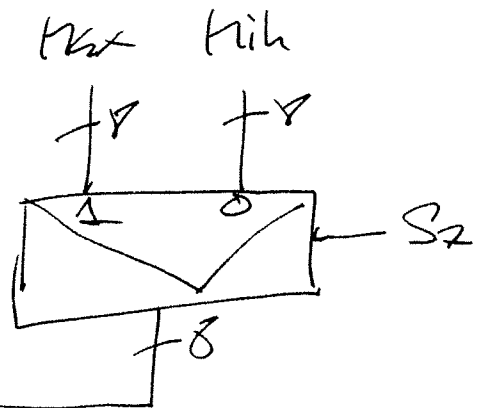
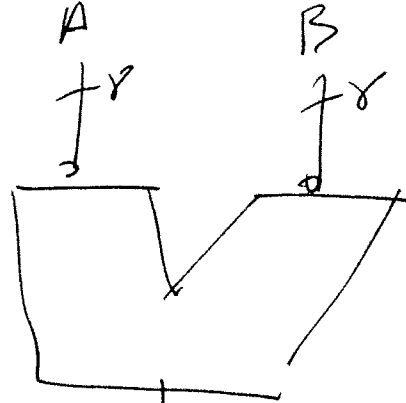


⊗ Max n.º positivo representable

$$Max = 01111111$$

⊗ Max negativo representable

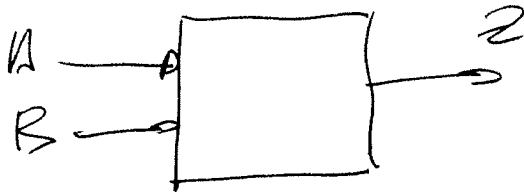
$$Min = 10000000$$



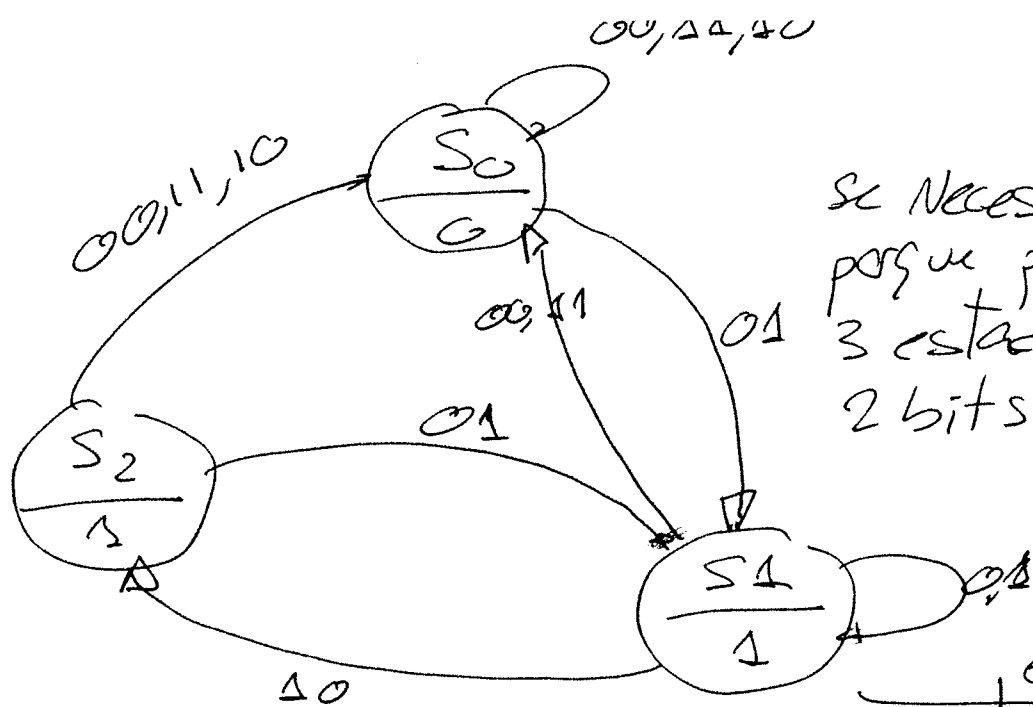
6

so → no ha llegado ninguna combinación de
entradas q. forme parte del patrón
→ señal de Reset.

S_2 - ha llegado la segunda parte del patron $\begin{cases} A = \underline{1} \\ B = \phi \end{cases}$



44 ~~estos~~ tres casos reduce el ~~partes~~



Se Necesita 2 bits
porque para codificar
3 estados necesito
2 bits

Tabla de verdad

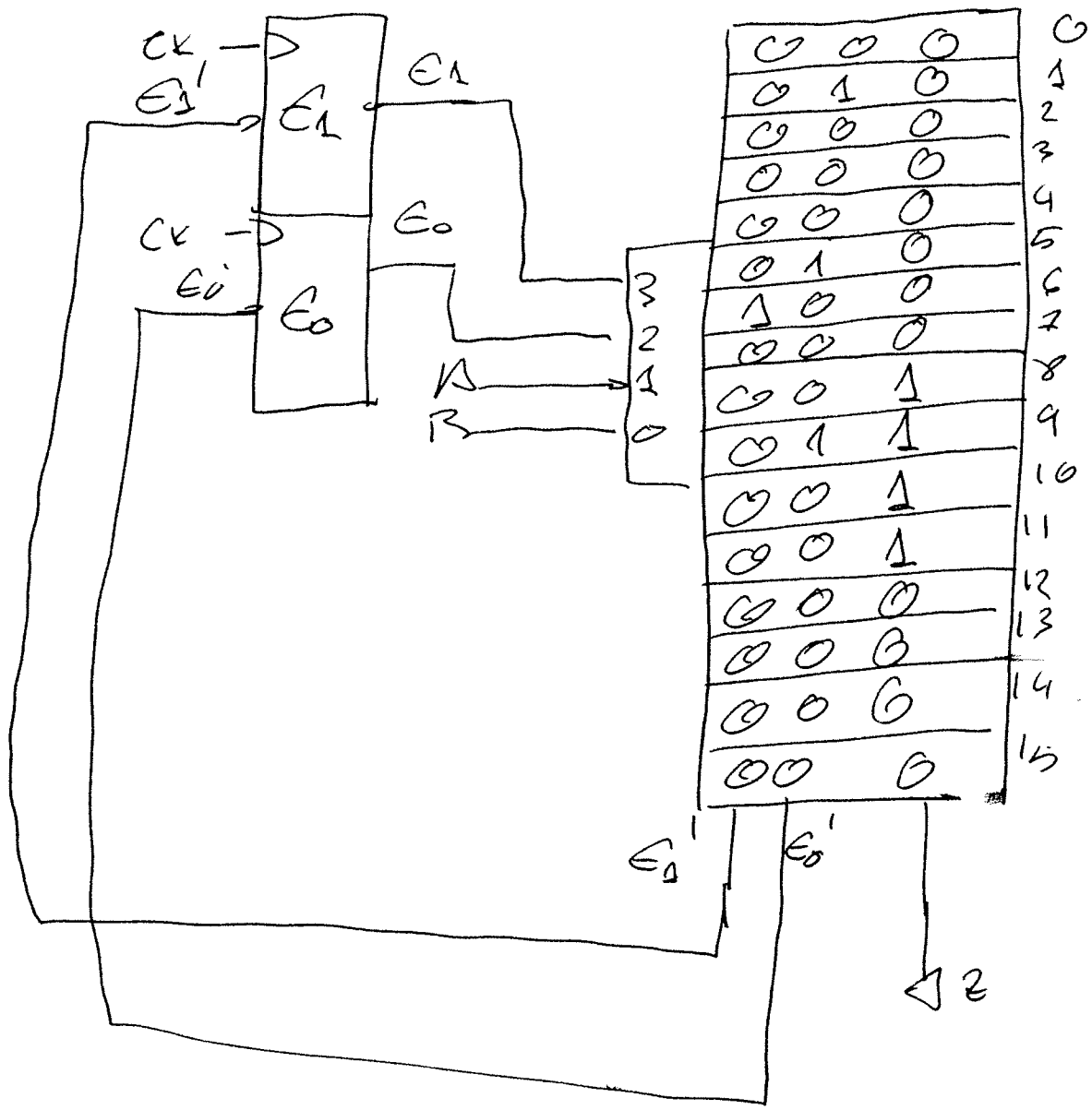
E_1	E_0	A	B	E_1'	E_0'
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

	E_1	E_0
S_0	0	0
S_1	0	1
S_2	1	0

→ la codificación
Δ & no se utiliza
⇒ sus salidas
serán don't cares

⇒ como es Karnaugh
la salida solo
depende del estado

E_1	E_0	Z
0	0	0
0	1	0
1	0	1
1	1	—



(5) Un contador modulo 4 tiene 4 estados

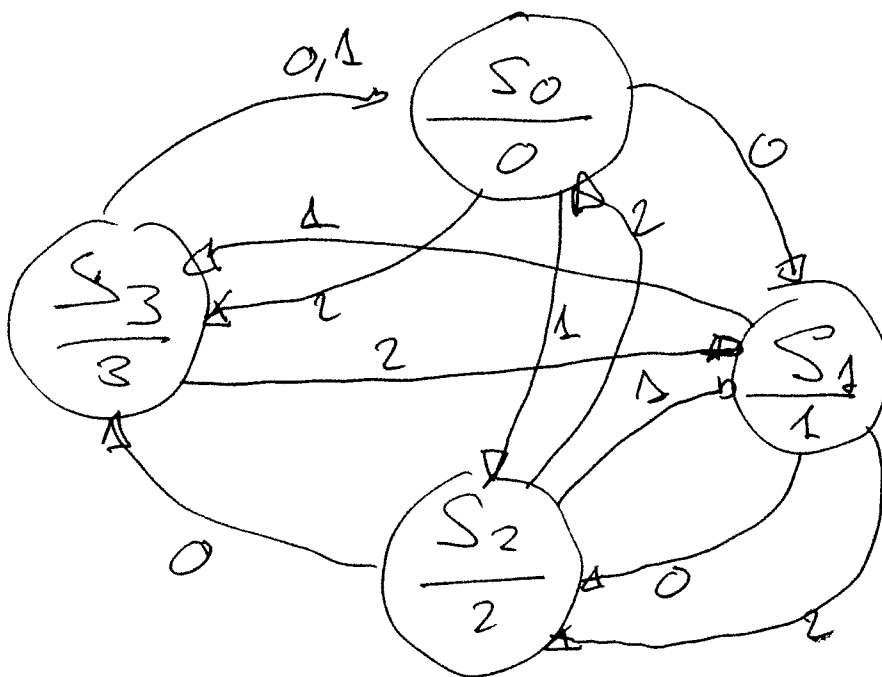
$S_0 \rightarrow \text{salida} = 0$

$S_1 \rightarrow \text{salida} = 1$

$S_2 \rightarrow \text{salida} = 2$

$S_3 \rightarrow \text{salida} = 3$

aprovechando las capacidades de carga paralela (cd) y cuenta (c) hay q. recorrer los estados del contador segun el valor de la entrada X



Se necesitan 2 bits para codificar los estados

(10)

	$E_1 E_0$
S_0	0 0
S_1	0 1
S_2	1 0
S_3	1 1

$E_1 E_0 \rightarrow$ el siguiente estado

$P_1 P_0 \rightarrow$ es la entrada paralela q. hay q. esperar por implementos de saltos en el diagrama de estado

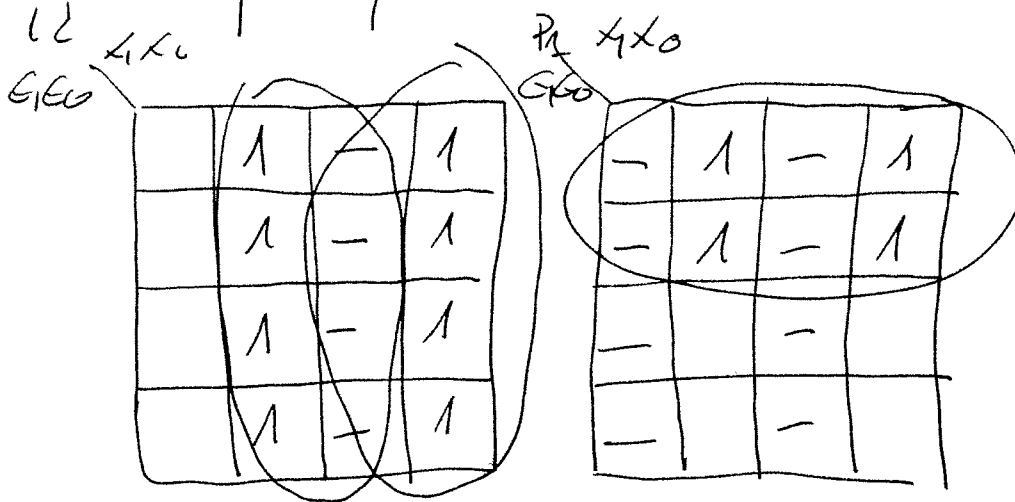
Recuerda q. cuando el siguiente estado corresponde a una cuenta natural (p.e. después del S_2 va S_3 o después del S_1 va el S_2) se activa la cuenta y la señal load se deja a 0

cuando el siguiente estado no corresponde a una cuenta natural hay q. realizar una carga paralela $ld = 1$; $c = d$

$E_1 E_0$	$x_1 x_0$	$E_1 E_0$	ld	c	$P_1 P_0$
0 0	0 0	0 1	0	1	- -
0 0	0 1	1 0	1	-	1 0
0 0	1 0	1 1	1	-	1 1
0 0	1 1	- -	-	-	- -
0 1	0 0	1 0	0	1	- -
0 1	0 1	1 1	1	-	1 1
0 1	1 0	1 0	1	-	1 0
0 1	1 1	- -	-	-	- -
1 0	0 0	1 1	0	1	- -
1 0	0 1	0 1	1	-	0 1
1 0	1 0	0 0	1	-	0 0
1 0	1 1	- -	-	-	- -
1 1	0 0	0 0	0	1	- -
1 1	0 1	0 0	1	-	0 0
1 1	1 0	0 1	1	-	0 1
1 1	1 1	- -	-	-	- -

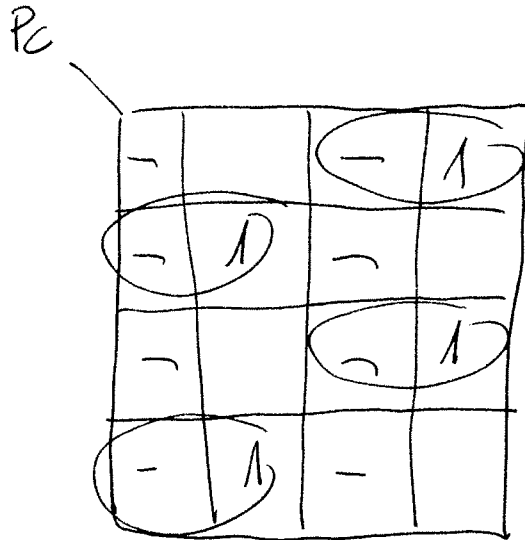
claramente se ve q. la señal de evento c
siempre puede estar a 1.

(11)



$$I_0 = X_0 + X_1$$

$$I_1 = E_1$$



$$P_0 = \bar{E}_1 \bar{E}_0 \bar{X}_1 + \bar{E}_1 \bar{E}_0 X_1 + \bar{E}_1 E_0 \bar{X}_1 + \bar{E}_1 E_0 X_1$$

Esta expresion se puede modificar por q. utilice puertas xor.

sacando factor comun:

$$\bar{E}_1 (\bar{E}_1 \bar{E}_0 + \bar{E}_1 E_0) + X_1 (\bar{E}_1 \bar{E}_0 + \bar{E}_1 E_0) =$$

$$\bar{E}_1 (E_0 \oplus \bar{E}_0) + X_1 (E_0 \oplus \bar{E}_0)$$

