

Puertas lógicas y álgebra de Boole

Puerta AND 2 entradas			puerta OR $X = A + B$	puerta AND $X = A * B$																														
Puerta NAND 2 entradas			<table border="1"><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1
A	B	X																																
0	0	0																																
0	1	1																																
1	0	1																																
1	1	1																																
A	B	X																																
0	0	0																																
0	1	0																																
1	0	0																																
1	1	1																																
Puerta OR 2 entradas			puerta NOR $X = \overline{A + B}$	puerta NAND $X = \overline{A * B}$																														
Puerta NOR 2 entradas			<table border="1"><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	0	<table border="1"><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	X	0	0	1	0	1	1	1	0	1	1	1	0
A	B	X																																
0	0	1																																
0	1	0																																
1	0	0																																
1	1	0																																
A	B	X																																
0	0	1																																
0	1	1																																
1	0	1																																
1	1	0																																
Puerta XOR 2 entradas			puerta NOT $X = \overline{A}$	<table border="1"><tr><th>A</th><th>X</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	X	0	1	1	0																								
A	X																																	
0	1																																	
1	0																																	
Puerta XNOR 2 entradas																																		

Álgebra de Boole: Propiedades →

Funciones a partir de la tabla de la verdad.

Expresiones

Miniterm: en forma de suma de productos

$S = ab+ac$

Maxiterm: en forma de producto de sumas

$S = (a+b) \cdot (a+c)$

Simplificación de funciones:

Numérica: por Quine-McKluskey

Manual: por Teoremas de Boole y de Morgan:

$\overline{a + b + c} = \overline{a} \overline{b} \overline{c} ; \overline{abc} = \overline{a} + \overline{b} + \overline{c}$

Por mapas de Karnaugh:

De la tabla de la verdad, se copian sólo, de las salidas con "1", los valores de entrada (negados si son 0, normales si son 1)

Ejemplo de tres variables: $S = \overline{C} \overline{B} A + \overline{C} B \overline{A} + \overline{C} B A + C \overline{B} A + C B \overline{A}$

a) Variables que NO cambian de valor: $S = \overline{B} \overline{A} + \overline{C} B + B \overline{A}$

b) Variables que NO cambian de valor: $F = A + B$

Ejemplo de 4 variables:

Variables que NO cambian de valor:

$S = \overline{D} \overline{B} + \overline{C} + \overline{B} A + D B \overline{A}$

	Adición	Producto
1	$a + \overline{a} = 1$	$a \cdot \overline{a} = 0$
2	$a + 0 = a$	$a \cdot 0 = 0$
3	$a + 1 = 1$	$a \cdot 1 = a$
4	$a + a = a$	$a \cdot a = a$
5	$a + b = b + a$	$a \cdot b = b \cdot a$
6	$a + (b + c) = (a + b) + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c$
7	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = a \cdot b + a \cdot c$
8	$a + a \cdot b = a$	$a \cdot (a + b) = a$
9	$\overline{(a + b)} = \overline{a} \cdot \overline{b}$	$\overline{(a \cdot b)} = \overline{a} + \overline{b}$

	BA	00	01	11	10
C	0		1	1	1
	1		1		1

$F = \overline{A} \overline{B} C + \overline{A} B C + A \overline{B} C + A B C + A \overline{B} \overline{C} + A B \overline{C} + \overline{A} B \overline{C} + \overline{A} \overline{B} \overline{C}$

	BC	00	01	11	10
A	0			1	1
	1	1	1	1	1

F = A + B

	BA	00	01	11	10
DC	00	1	1	1	1
	01	1	1		
	11		1		1
	10	1	1	1	1

Circuitos digitales combinacionales

Son aquellos en que sus salidas sólo dependen del estado de sus entradas

Circuitos de comunicación

- ▶ **Codificadores:** convierte datos digitales a código binario. Sin prioridad, sólo puede activarse una entrada cada vez. Con prioridad obtiene el código de entrada prioritario.
- ▶ **Descodificadores:** Convierte la información binaria en otro que suele ser combinación de sus entradas, por ejemplo el display de 7 segmentos.
- ▶ **Multiplexor:** Tiene un selector de canal para dejar pasar de varias entradas a una de ellas en la salida.
- ▶ **Demultiplexor:** El selectro de canal permite a una entrada pasar a diferentes canales de salida.

Circuitos aritméticos:

- ▶ **Comparadores:** indican en su salida si dos números binarios son iguales.
- ▶ **Sumador y restador binario:** suman o restan números binarios con o sin acarreo (carry/borrow)
- ▶ **Unidades aritméticas (ALU):** Son procesadores encargados de realizar operaciones aritméticas y lógicas complejas.

Circuitos digitales secuenciales

Memorizan y utilizan también la información de la salida anterior. Son circuitos con memoria

Biestables

Asíncronos: activados por nivel de tensión.

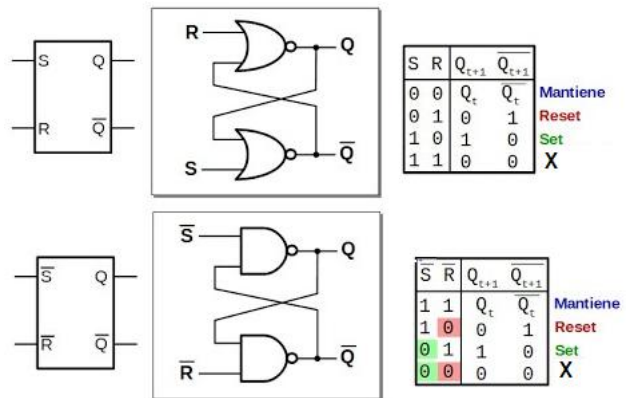
Biestable R-S:

Se construye con 2 puertas NOR o NAND

La salida Q_{t+1} dependerá de las entradas Set (1), Reset (0) o de su estado anterior Q_t :

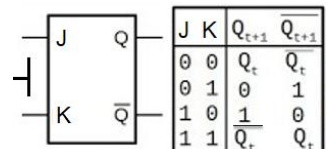
- ▶ **Set on:** Siempre pone a 1 la salida
- ▶ **Reset on:** Siempre pone a 0 la salida
- ▶ Sino, la salida mantiene el valor anterior

No se puede activar Set y Reset a la vez



Biestable J-K:

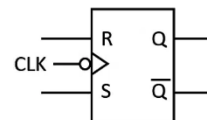
No tiene indeterminación sino que la salida invierte su estado anterior. Si se unen las entradas es un biestable T que cambia de estado cada vez que cambia la entrada.



Síncronos:

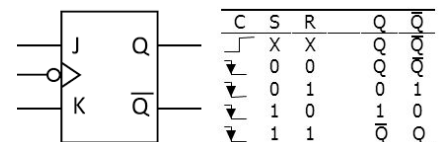
Síncrono por tensión:

Sólo lee los datos de entrada cuando se aplica un nivel alto en la entrada clock CLK



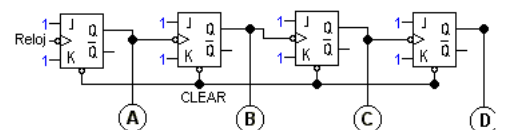
Síncrono por flanco:

Cambian de estado en el pulso de subida o de bajada (negada) de la señal de reloj.



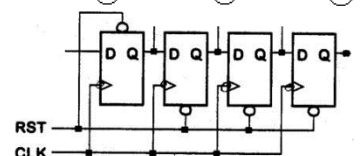
Contadores

Constituido por biestables interconectados en cascada, su salida dependerá del número de impulsos de la entrada.



Registros de desplazamiento

Conjunto de biestables en cascada que almacenan una palabra binaria. También pueden transmitir o cambiar datos de serie a paralelo.



Exercicis

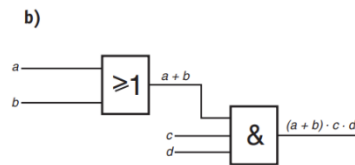
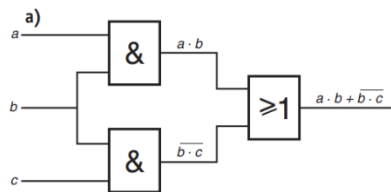
1. Efectua la conversió de sistema binari a decimal dels nombres següents:

a) 10110 b) 00100110 c) 101 d) 101111110 e) 011110110110

22, 38,5, 382, 1974

2. Representa els esquemes de portes lògiques que corresponen a les equacions següents:

a) $S = ab + \overline{b}c$ b) $S = (a + b)cd$



3. Obtén l'equació simplificada de la funció definida per la taula de veritat següent:

a	b	c	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

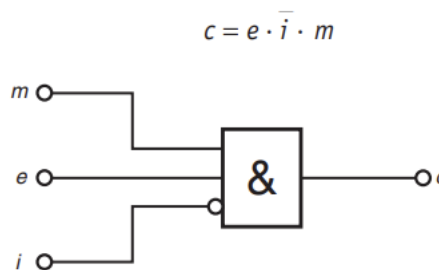
$$F = abc + \overline{a}bc + a\overline{b}c = a\overline{b}c + bc(\overline{a} + a) = a\overline{b}c + bc$$

4. Un radiador elèctric disposa d'un interruptor de posada en marxa i de dos termòstats: un que connecta els elements calefactors si la temperatura exterior és inferior a una de prefixada, t_a , i un de seguretat que els desconnecta si la temperatura interior supera els 90 °C. Utilitzant les variables d'estat:

- Termòstat exterior $e = 1$, $t_{ext} < t_a$; 0, $t_{ext} \geq t_a$.
- Termòstat interior $i = 1$, $t_{int} > 90$ °C; 0, $t_{int} \leq 90$ °C.
- Interruptor de posada en marxa $m = 1$, sí; 0, no.
- Funcionament calefactors $c = 1$, sí; 0, no.

- a) Determina la taula de veritat del sistema.
- b) Escribe la funció lògica entre les variables d'estat i, si escau, simplifica la.
- c) Dibuixa l'esquema de portes lògiques

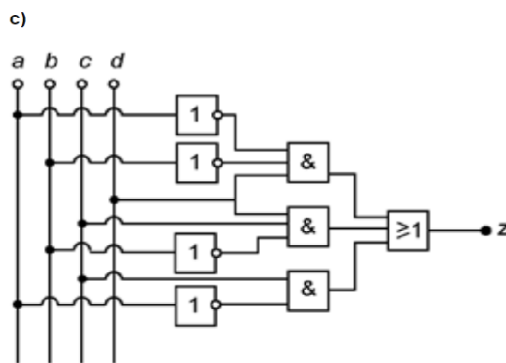
e	i	m	c
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0



PAU. 2022. El controlador d'un motor d'ascensor necessita un senyal que determini en quin sentit ha de posar-se en marxa l'ascensor (per a pujar o baixar). Per a aconseguir això, es dissenya un sistema digital on la sortida z pren valor 1 si l'ascensor ha de pujar i 0 en cas contrari. El sistema té com a entrades 4 variables digitals (a, b, c, d) per a codificar la planta on es troba l'ascensor i la planta on vol anar l'usuari de les quatre possibles: planta 0, planta 1, planta 2 i planta 3. Les dues primeres entrades codifiquen, en numeració binària, la planta en què es troba l'ascensor (per exemple, si es troba a la planta 3 els valors seran a = 1 i b = 1); les altres dues entrades (c, d) codifiquen, de la mateixa manera, el número de la planta que l'usuari selecciona. Utilitzant les variables d'estat descrites, dissenyeu el sistema digital que permeti determinar quan l'ascensor ha de moure's en sentit ascendent. Per fer-ho: a) Elaboreu la taula de veritat del sistema. b) Determineu la funció lògica entre aquestes variables i, si escau, simplifiqueu-la. c) Dibuixeu el diagrama de portes lògiques equivalent.

a	b	c	d	z
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

b) $z = (\bar{a}\bar{b}\bar{c}d) + (\bar{a}\bar{b}c\bar{d}) + (\bar{a}\bar{b}cd) + (\bar{a}b\bar{c}d) + (\bar{a}bcd) + (ab\bar{c}d)$
 simplificant: $z = \bar{a}\bar{b}d + \bar{b}cd + \bar{a}c$

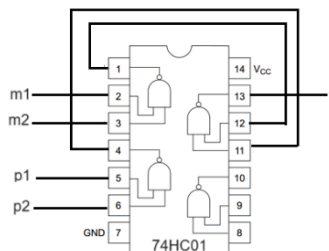
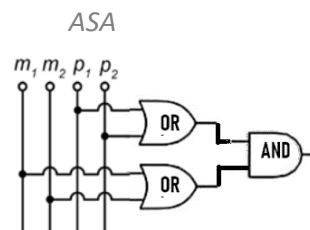
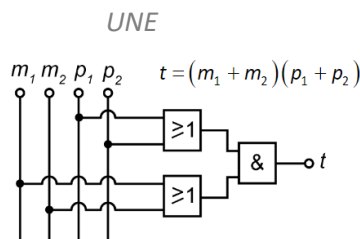


En una línia de producció hi ha una estació de treball on es realitzen operacions de tallat. El sistema de seguretat vol evitar que l'operari es trobi prop de l'eina de tall; amb aquesta finalitat, s'han instal·lat quatre polsadors: dos de situats a mitja altura (que s'accionen amb les mans) i dos de situats al terra (que s'accionen amb els peus). Per a realitzar l'operació de tallat, cal que l'operari premi a la vegada almenys un polsador de mitja altura amb la mà i un del terra amb el peu. Responeu a les qüestions que hi ha a continuació utilitzant les variables d'estat següents (veure).

- a) Escriviu la taula de veritat del sistema
- b) Determineu la funció lògica entre aquestes variables i, si escau, simplifiqueu-la
- c) Dibuixeu l'esquema de portes lògiques equivalent

Polsadors: m1: 1:activat, 0:No activat m2: 1:activat, 0: No activat
 Peu: p1: 1:activat, 0: No activat p2: 1: activat, 0: No activat
 Tallat t: 1: marcha 0: aturada

Sol:



P2019 El sistema de control d'obertura i tancament d'una porta automàtica de vidre és regulat per un sistema format per un detector de presència, un detector de moviment i un interruptor manual situat a la cabina de control. La porta, que en general està tancada, s'obre quan s'activa l'interruptor manual (independentment de l'estat de la resta d'elements de control) o quan s'activen els detectors de presència i de moviment a la vegada. Responen a les qüestions que hi ha a continuació utilitzant les variables d'estat següents:

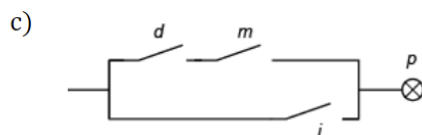
- detector de presència: $d = 1$: detecta presència 0: no detecta presència;
- detector de moviment: $m = 1$: detecta moviment 0: no detecta moviment;
- interruptor manual: $i = 1$: activat 0: no activat;
- porta: $p = 1$: oberta 0: tancada.

- a) Elaboreu la taula de veritat del sistema.
- b) Determineu la funció lògica entre aquestes variables i , si escau, simplifiqueu-la.
- c) Dibuixeu l'esquema de contactes equivalent.

d	m	i	p
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

b) $p = (d + m + i)(d + \bar{m} + i)(\bar{d} + m + i)$
 o també $p = (\bar{d} \cdot \bar{m} \cdot i) + (\bar{d} \cdot m \cdot i) + (d \cdot \bar{m} \cdot i) + (d \cdot m \cdot \bar{i}) + (d \cdot m \cdot i)$

$p = d \cdot m + i$



2023. En la defensa d'un projecte acadèmic, el tribunal que valora si el treball és apte o no apte està format per quatre membres: president, secretari, vocal 1 i vocal 2. La decisió es pren per majoria i , en cas d'empat, preval el vot de qualitat del president.

Responen a les qüestions que hi ha a continuació utilitzant les variables d'estat següents:

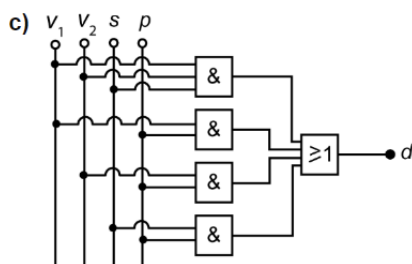
- vocal 1: $v_1 = 1$: vot favorable 0: vot en contra
- vocal 2: $v_2 = 1$: vot favorable; 0: vot en contra
- secretari: $s = 1$: vot favorable 0: vot en contra
- president: $p = 1$: vot favorable 0: vot en contra
- decisió: 1: treball apte 0: treball no apte

- a) Elaboreu la taula de veritat del sistema
- b) Determineu la funció lògica entre aquestes variables i , si escau, simplifiqueu-la

v_1	v_2	s	p	d
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Sol: b) $d = \bar{v}_1 \bar{v}_2 s p + \bar{v}_1 v_2 \bar{s} p + \bar{v}_1 v_2 s p + v_1 \bar{v}_2 \bar{s} p + v_1 \bar{v}_2 s p + v_1 v_2 \bar{s} p + v_1 v_2 s \bar{p} + v_1 v_2 s p$

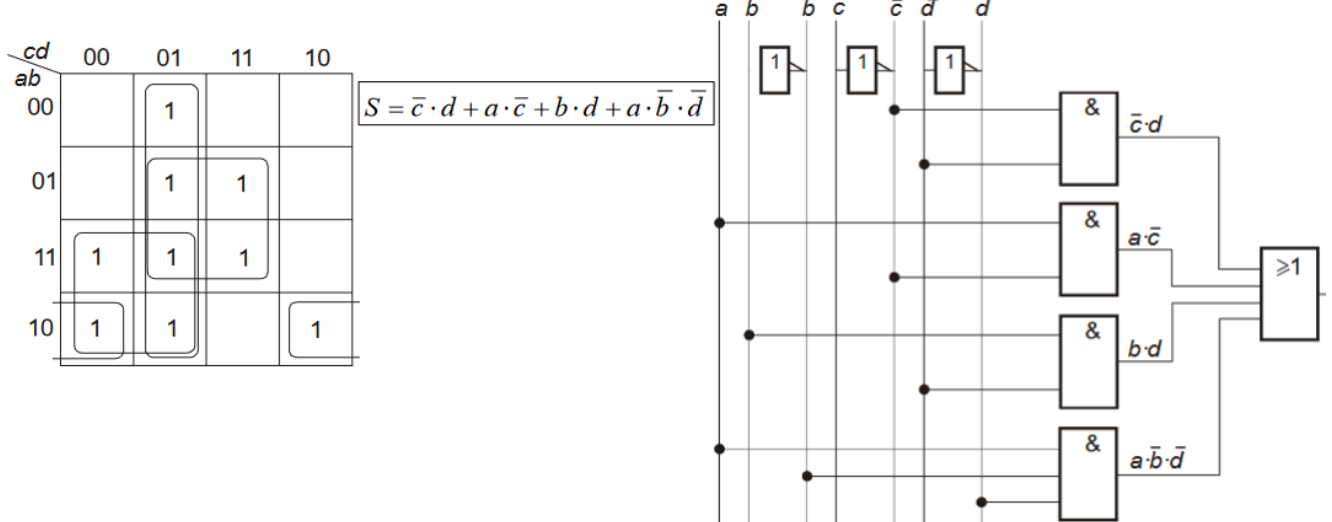
$d = v_1 v_2 s + s p + v_1 p + v_2 p$



Resuelto: Simplificar por el método de Karnaugh y dibujar el circuito de la siguiente expresión:

$$S = \bar{c} \cdot d \cdot (a + \bar{a}) \cdot (b + \bar{b}) + a \cdot \bar{b} \cdot c \cdot \bar{d} + a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} + a \cdot b \cdot \bar{c} \cdot \bar{d} + b \cdot c \cdot d \cdot (a + \bar{a})$$

$$S = a \cdot b \cdot \bar{c} \cdot d + \bar{a} \cdot b \cdot \bar{c} \cdot d + a \cdot \bar{b} \cdot \bar{c} \cdot d + \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d + a \cdot \bar{b} \cdot c \cdot \bar{d} + a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} + a \cdot b \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot b \cdot c \cdot d + a \cdot b \cdot c \cdot d$$



2024. L'accés a un edifici d'oficines està regulat per tres sistemes de control: una clau numèrica, una targeta magnètica i l'empremta dactilar. Es permet l'accés a l'edifici, en horari laboral, validant qualsevol dels tres sistemes de control. Fora de l'horari laboral, cal validar almenys dos dels tres sistemes. Responeu a les qüestions que hi ha a continuació utilitzant les variables d'estat següents:

a) Elaboreu la taula de veritat del sistema. b) Determineu la funció lògica entre aquestes variables i, si escau, simplifiqueu-la c) Dibuixeu el diagrama de portes lògiques equivalent.

$$\text{horari: } h = \begin{cases} 1: \text{laboral} \\ 0: \text{no laboral} \end{cases}; \text{ clau: } c = \begin{cases} 1: \text{vàlida} \\ 0: \text{no vàlida} \end{cases}; \text{ targeta: } t = \begin{cases} 1: \text{vàlida} \\ 0: \text{no vàlida} \end{cases}$$

$$\text{empremta: } e = \begin{cases} 1: \text{vàlida} \\ 0: \text{no vàlida} \end{cases}; \text{ accés: } a = \begin{cases} 1: \text{permès} \\ 0: \text{denegat} \end{cases}$$

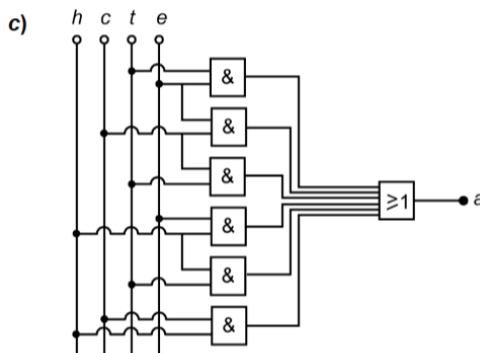
a)

h	c	t	e	a
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

b) $a = (h+c+t+e)(h+c+t+\bar{e})(h+c+\bar{t}+e)(h+\bar{c}+t+e)(\bar{h}+c+t+e)$

simplificant $a = te + ce + ct + he + ht + hc$

o també $a = c(e+h+t) + e(h+t) + ht$

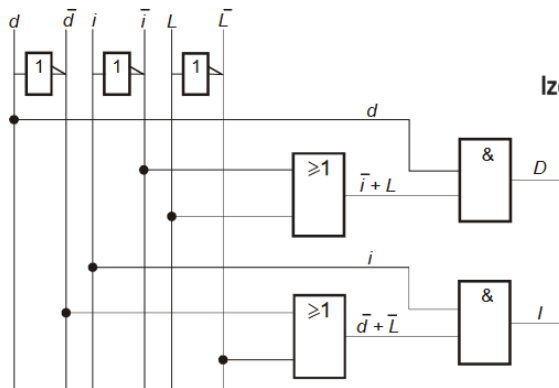


PAU Andalucía: Un motor eléctrico puede girar en ambos sentidos por medio de dos contactores: "D" para el giro a derecha y "I" para el giro a izquierda. Estos dos contactores son comandados por dos pulsadores de giro "d" (derecha) e "i" (izquierda) y un interruptor de selección "L" de acuerdo con las siguientes condiciones:

- Si sólo se pulsa uno de los dos botones de giro, el motor gira en el sentido correspondiente.
- Si se pulsan los dos botones de giro simultáneamente, el sentido de giro depende del estado del interruptor "L" de forma que, • Si "L" está activado, el motor gira a la derecha.
- Si "L" está en reposo, el motor gira a la izquierda.

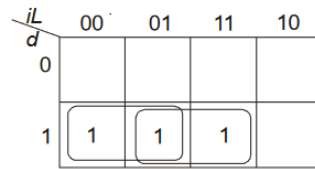
Establecer : a) La tabla de verdad. b) Las funciones lógicas D e I y simplificarlas. c) Su circuito lógico.

d	i	L	D	I
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0



Derecha

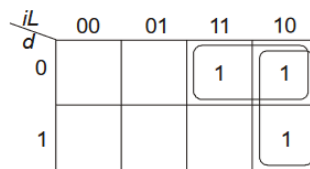
$$D = d \cdot \bar{i} \cdot \bar{L} + d \cdot \bar{i} \cdot L + d \cdot i \cdot L$$



$$D = d \cdot \bar{i} + d \cdot L \quad D = d \cdot (\bar{i} + L)$$

Izquierda

$$I = \bar{d} \cdot i \cdot \bar{L} + \bar{d} \cdot i \cdot L + d \cdot i \cdot \bar{L}$$

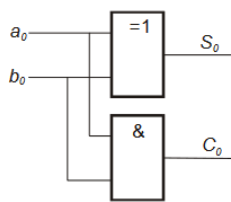


$$I = \bar{d} \cdot i + i \cdot \bar{L} \quad I = i \cdot (\bar{d} + \bar{L})$$

Combinacionales (And 98): Diseñe un circuito combinacional que realice la suma aritmética de dos números binarios, uno de un bit y otro de dos bits, y cuyo resultado también esté dado en binario. Represente el circuito mediante puertas lógicas.

La suma de los dos números $S = a_0 + b_1b_0$ órdenes iguales, $S_0 = a_0 + b_0$ acarreo C_0

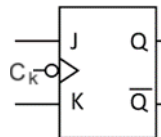
a_0	b_0	S_0	C_0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



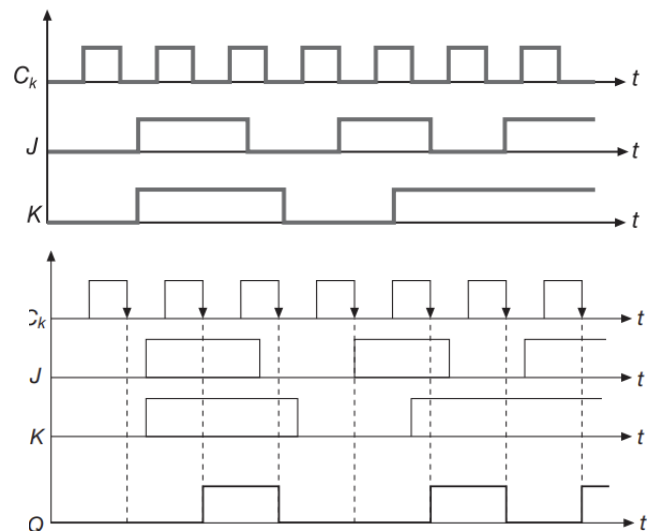
$$S_0 = \bar{a}_0 \cdot b_0 + a_0 \cdot \bar{b}_0 = a_0 \oplus b_0$$

$$C_0 = a_0 \cdot b_0$$

BG: Obtén el senyal de la sortida Q d'un biestable J-K master-slave activat per flanc de baixada, al qual s'apliquen els senyals següents:



Solució →



Circuito secuencial (asíncrono o síncrono) que sólo puede estar en uno de un número finito de estados.

Clasificación: Mealy: (la salida depende de su estado y la entrada)

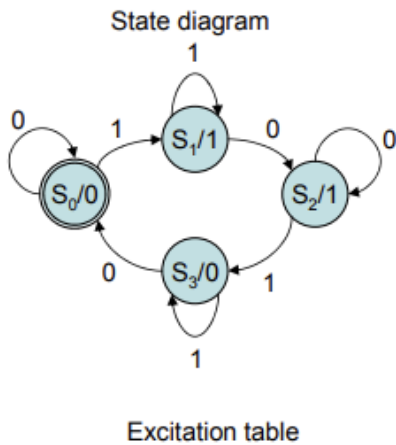
Moore: (la salida depende sólo de su estado).

SINTESIS DE FSM (implementar)

1º Identificar los estados y sus transiciones, con un diagrama de estados o con una tabla de estados

2º utilizar la sentencia switch (también con bucles for o if-else)

ejemplo Moore síncrona



State table

state	input		output
	0	1	
S0	S0	S1	0
S1	S2	S1	1
S2	S2	S3	1
S3	S0	S3	0

**State Codification
(gray codification)**

state	Q[1:0]
S0	0 0
S1	0 1
S2	1 1
S3	1 0

State table

state	input		output
	Q[1:0]	0	
00	00	01	0
01	11	01	1
11	11	10	1
10	00	10	0

Diseño de una FSM con Flip-Flop

Para implementar una FSM se utilizan flip-flops, generalmente del tipo D o JK, debido a su facilidad para almacenar y actualizar el estado.

Pasos del Diseño:

1. Definición del diagrama de estados:

- Enumere todos los estados posibles del sistema.
- Dibuje un diagrama de estados mostrando las transiciones entre estados basadas en las entradas.

2. Codificación de estados:

- Asigne un código binario a cada estado.
- El número de flip-flops necesarios se determina por la fórmula: $\log_2 n$, donde n es el número de estados.

3. Tabla de transiciones:

- Cree una tabla que muestre el estado actual, la entrada, el próximo estado y la salida.

4. Ecuaciones de transición de estado:

- Derive las ecuaciones lógicas que determinan los valores de las entradas de los flip-flops en función del estado actual y las entradas.

5. Ecuaciones de salida:

- Derive las ecuaciones lógicas para las salidas en función del estado actual y/o las entradas.

6. Implementación del circuito:

- Diseñe el circuito combinacional necesario para las entradas de los flip-flops y las salidas del sistema.
- Conecte los flip-flops y el circuito combinacional según las ecuaciones derivadas.

3. Ejemplo de FSM Simple

Supongamos una FSM con tres estados (A, B, C) y una entrada (X):

- **Estados:** A (00), B (01), C (10)
- **Entrada:** X (0 o 1)
- **Salidas:** Z (0 o 1)

Diagrama de estados:

- › Desde A: Con X=0, permanece en A. Con X=1, va a B.
- › Desde B: Con X=0, va a C. Con X=1, permanece en B.
- › Desde C: Con X=0, va a A. Con X=1, permanece en C.

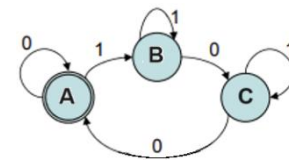


Tabla de transiciones y salidas:

Estado Actual	Entrada (X)	Próximo Estado	Salida (Z)
A (00)	0	A (00)	0
A (00)	1	B (01)	0
B (01)	0	C (10)	0
B (01)	1	B (01)	1
C (10)	0	A (00)	0
C (10)	1	C (10)	1

Ecuaciones de transición (con flip-flops tipo D):

- $D_1 = Q_0' \cdot X + Q_1 \cdot Q_0' \cdot X'$
- $D_0 = Q_0 \cdot X' + Q_1' \cdot Q_0 \cdot X$

Ecuaciones de salida: $Z = Q_1 \cdot Q_0'$

4. Implementación del Circuito:

El circuito final incluiría flip-flops D y lógica combinacional diseñada según las ecuaciones derivadas.

- **Flip-flops:** Se usarán dos flip-flops para representar los tres estados.
- **Lógica combinacional:** Se implementará utilizando puertas lógicas según las ecuaciones de transición y salida.

Diagrama de Bloques:

- **Flip-flops:** D1, D0
- **Entradas:** X
- **Salidas:** Z
- **Lógica Combinacional:** Implementación de ecuaciones de transición y salida.

El análisis de una FSM con flip-flops implica comprender la interacción entre los estados, las entradas y las salidas, así como diseñar adecuadamente el circuito lógico que gobierna las transiciones y las salidas del sistema.