

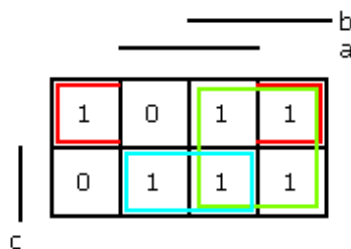
PREPA 1 Gabriel Rodríguez. Email: gabriel.rodriguez3103@gmail.com

1) Simplificar la siguiente expresión, realizar la tabla de la verdad y comprobar mediante el mapa de karnaugh .

a)

$$\begin{aligned}
 F &= ABC + AB'C + ABC' + A'B'C' + A'B \\
 &= AC(B + B') + ABC' + A'(B + B'C') \\
 &= AC + ABC' + A'(B + C') \\
 &= A(C + C'B) + A'B + A'C' \\
 &= A(C + B) + A'B + A'C' \\
 &= AC + AB + A'B + A'C' \\
 &= AC + B(A' + A) + A'C' \\
 &= AC + B + A'C'
 \end{aligned}$$

C	B	A	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

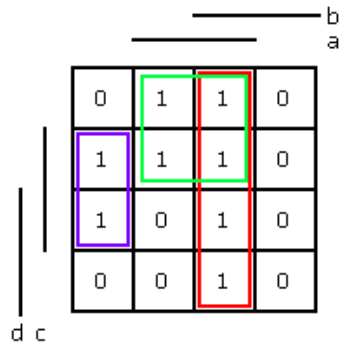


$$\rightarrow F = B + AC + A'C'$$

b)

$$\begin{aligned}
 F &= (A + C)(A + B')(A' + B + D') \\
 &= (A + AB' + AC + B'C)(A' + B + D') \\
 &= (A(1 + B' + C) + B'C)(A' + B + D') \\
 &= (A + B'C)(A' + B + D') \\
 &= AA' + AB + AD' + A'B'C + BB'C + B'CD' \\
 &= AB + AD' + A'B'C + B'CD'
 \end{aligned}$$

D	C	B	A	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
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	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1



$$\rightarrow F = AB + AD' + A'B'C$$

2) Diseño de circuitos

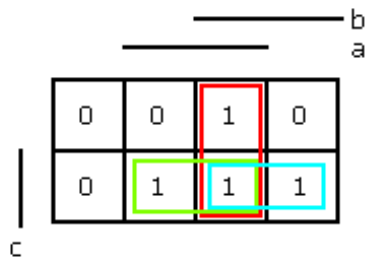
a) Diseñar el circuito del control de un motor mediante tres pulsadores, A,B,C, que cumpla las siguientes condiciones de funcionamiento:

- Si se pulsaran los tres pulsadores el motor se activa
- Si se pulsaran dos pulsadores cualesquiera, el motor se activa pero se enciende una lámpara de peligro.
- Si sólo se pulsa un pulsador, el motor no se activa pero si se enciende la lámpara indicadora de peligro.
- Si no se pulsa ningún pulsador, el motor y la lámpara están desactivados.

Al analizar el problema se puede observar que se tienen 3 entradas, correspondientes a los pulsadores, y dos salidas, el motor y la lámpara, y mediante las condiciones se construye la siguiente tabla:

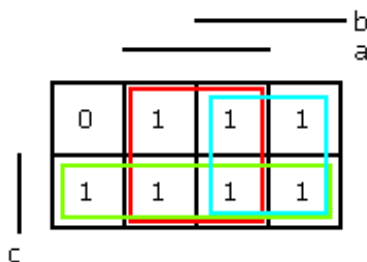
C	B	A	Motor	Lámpara
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	1
1	0	0	0	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Para el motor:



$$\text{Motor} = AB + AC + BC$$

Para la lámpara:



$$\text{Lámpara} = A + C + B$$

- b) Un codificador de posición de un eje proporciona un código de 4 bits que indica el ángulo de rotación del eje con una resolución de 30 grados, según la tabla adjunta. Diseñar un circuito lógico que indique, con un número de dos bits YZ, en cuál cuadrante se encuentra el eje (00= 1er cuadrante, 01= 2do cuadrante, 10= 3er cuadrante, 11= 4to cuadrante).

Ángulo del eje	Entradas			
	D	C	B	A
0° - 29°	0	0	1	1
30° - 59°	0	0	1	0
60° - 89°	0	1	1	0
90° - 119°	0	1	1	1
120° - 149°	0	1	0	1
150° - 179°	0	1	0	0
180° - 209°	1	1	0	0
210° - 239°	1	1	0	1
240° - 269°	1	1	1	1
270° - 299°	1	1	1	0
300° - 329°	1	0	1	0
330° - 359°	1	0	1	1

Se construye la tabla de la verdad, cuyas salidas corresponden a Y y Z, los casos no considerados tendrán la condición “no importa”.

Entradas				Salidas	
D	C	B	A	Y	Z
0	0	0	0	X	X
0	0	0	1	X	X
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	0
0	1	1	1	0	1
1	0	0	0	X	X
1	0	0	1	X	X
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	1	0

Se construye el mapa de karnaugh para ambas variables:

Para Y:

		b			
		a			
		X	X	0	0
		0	0	0	0
		1	1	1	1
		X	X	1	1
d	c				

$$Y = D$$

Para Z:

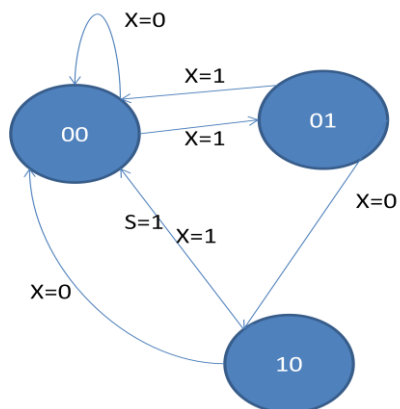
		b			
		a			
		X	X	0	0
		1	1	1	0
		0	0	0	1
		X	X	1	1
d	c				

$$Z = DBA' + D'B' + D'CA + DC'$$

3) Diseño de autómatas

- a) Un sistema de seguridad necesita un detector de secuencia que detecte la ocurrencia de la secuencia S=101. La secuencia no presenta solapamiento. Implemente utilizando flip-flop tipo J-K y maquina de estado tipo Mealy.

- Diagrama de estados:



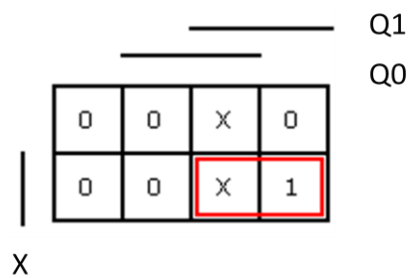
NOTA: En las transiciones donde no se define la salida es porque la misma es cero.

- Tabla de la Verdad

X	Q1	Q0	Q1*	Q0*	S	J1	K1	J0	K0
0	0	0	0	0	0	0	X	0	X
0	0	1	1	0	0	1	X	X	1
0	1	0	0	0	0	X	1	0	X
0	1	1	X	X	X	X	X	X	X
1	0	0	0	1	0	0	X	1	X
1	0	1	0	0	0	0	X	X	1
1	1	0	0	0	1	X	1	0	X
1	1	1	X	X	X	X	X	X	X

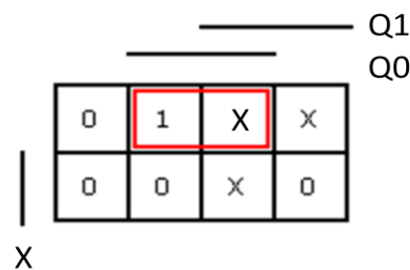
- Mapa de karnaugh y expresiones:

Para S:



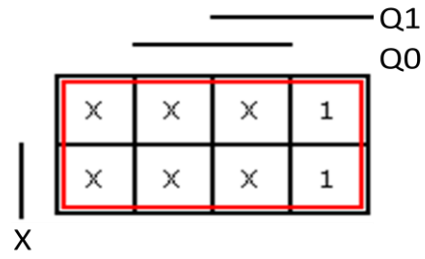
$$S = XQ1$$

- Para J1



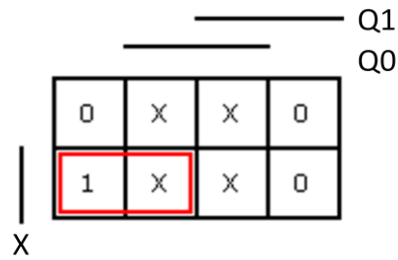
$$J1 = X'Q0$$

- Para K1



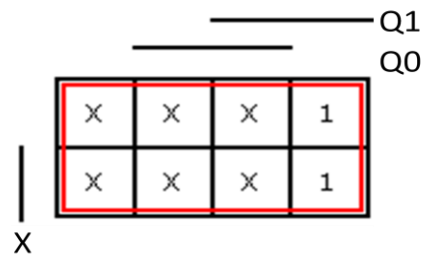
$$K1 = 1$$

- Para J0



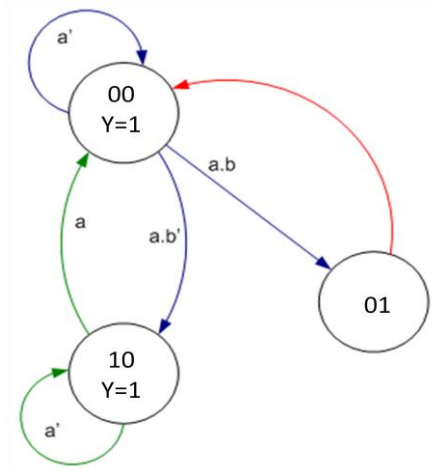
$$J0 = XQ1'$$

- Para K0



$$J0 = 1$$

- b) Diseñe usando flip flop tipo D, un circuito secuencial de dos entradas, a y b, y una salida tipo MOORE que implemente el diagrama de estado de la figura:

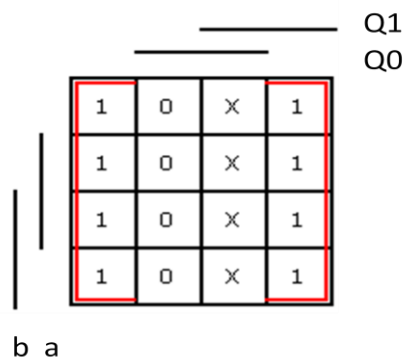


- Tabla de la Verdad

b	a	Q1	Q0	Q1*	Q0*	Y	D1	D0
0	0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0
0	0	1	0	1	0	1	1	0
0	0	1	1	X	X	X	X	X
0	1	0	0	1	0	1	1	0
0	1	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0
0	1	1	1	X	X	X	X	X
1	0	0	0	0	0	1	0	0
1	0	0	1	0	0	0	0	0
1	0	1	0	1	0	1	1	0
1	0	1	1	X	X	X	X	X
1	1	0	0	0	1	1	0	1
1	1	0	1	0	0	0	0	0
1	1	1	0	0	0	1	0	0
1	1	1	1	X	X	X	X	X

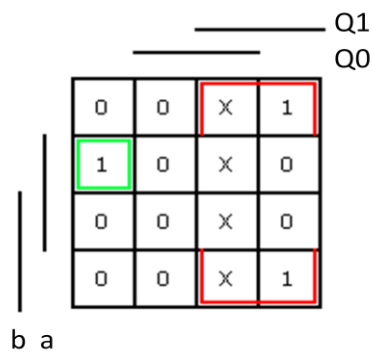
- Mapa de karnaugh y expresiones:

- Para Y



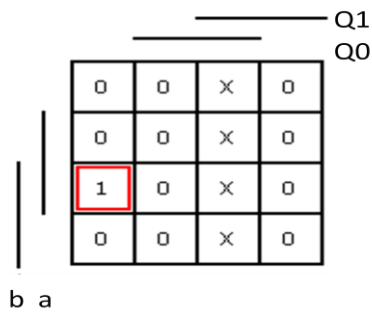
$$Y = Q0'$$

- Para D1



$$D1 = A'Q1 + B'AQ1'Q0'$$

- Para D0

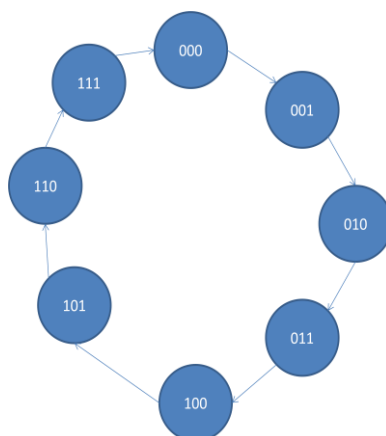


$$D0 = BAQ1'Q0'$$

4) Diseño de contadores

- Diseñe un CONTADOR binario de 3 bits que ascendente que vaya de 0 a 7 y vuelva a comenzar al terminar la secuencia. Realice el diseño utilizando flip flop tipo D.

- Diagrama de estados

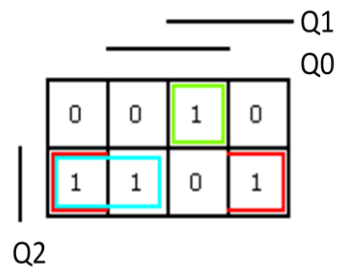


- Tabla de la verdad

Q2	Q1	Q0	Q2*	Q1*	Q0*	D2	D1	D0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0

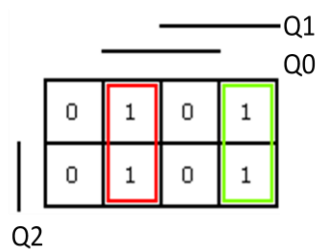
- Mapa de karnaugh y expresiones:

- Para D2:



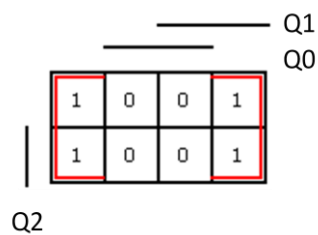
$$D2 = Q2Q0' + Q2Q1' + Q2'Q1Q0$$

- Para D1:



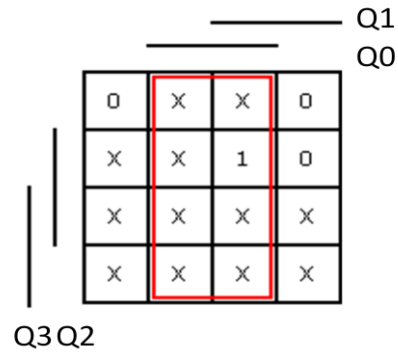
$$D1 = Q0Q1' + Q1Q0'$$

- Para D0:



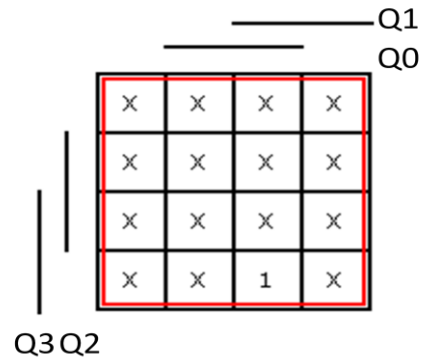
$$D0 = Q0'$$

- Para J3:



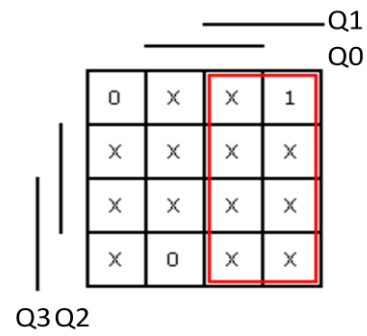
$$J3 = Q0$$

- Para K3:



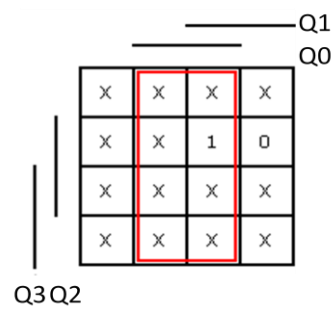
$$K3 = 1$$

- Para J2:



$$J2 = Q1$$

- Para K2:



$$K2 = Q0$$

- Para J1:

		Q1			
		Q0			
		1	x	x	x
		x	x	x	x
		x	x	x	x
		x	0	x	x
		Q3		Q2	

$$J1 = Q3'$$

- Para K1:

		Q1			
		Q0			
		x	x	x	0
		x	x	1	0
		x	x	x	x
		x	x	x	x
		Q3		Q2	

$$K1 = Q0$$

- Para J0:

		Q1			
		Q0			
		0	x	x	0
		x	x	x	1
		x	x	x	x
		x	x	x	x
		Q3		Q2	

$$J0 = Q2$$

- Para K0:

		Q1			
		Q0			
		x	x	x	x
		x	x	0	x
		x	x	x	x
		x	1	x	x
		Q3		Q2	

$$K0 = Q2'$$

5) Memorias:

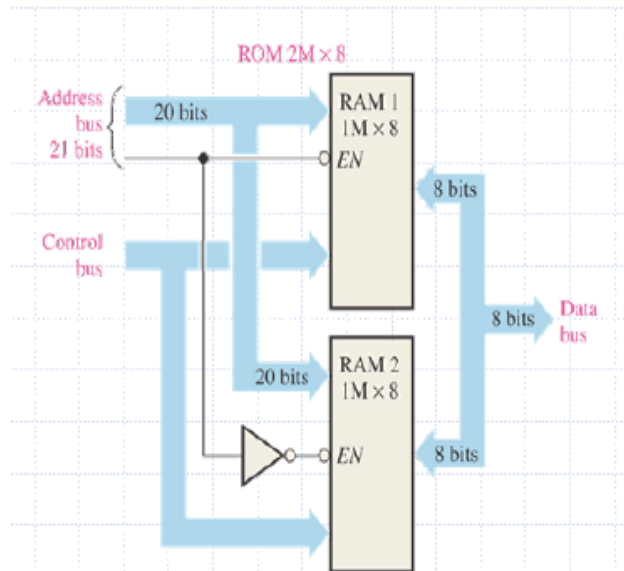
- a) Se tiene dos módulos de memoria de 1M por 8 bits y se quiere expandir la cantidad de memoria a 2M por 8 bits. (Expansión de capacidad)

Es importante conocer que la notificación de memoria viene dada por la relación: 2^m por n , donde m indica los bits de direcciones y n los bits de datos. Expandir la memoria implica aumentar el bus de direcciones, es decir, el valor de m . Entonces se tiene que:

$$2^m = 1[M] = 1000000 \rightarrow m = 19.9 \rightarrow m = 20$$

$$2^m = 2[M] = 2000000 \rightarrow m = 20.9 \rightarrow m = 21$$

Para colocar un bit más de dirección se realiza el siguiente montaje:



El bit EN permite aumentar la capacidad, ya que, mientras es 1, se habilita la memoria para la combinación 0XXXXXXXXXXXXXXXXXXXXX (RAM 1), mientras que al ser 0 se tiene 1XXXXXXXXXXXXXXXXXXXXX (RAM 2), teniendo finalmente 21 bits de dirección, que implican una memoria final de 2[M] por 8 bits.

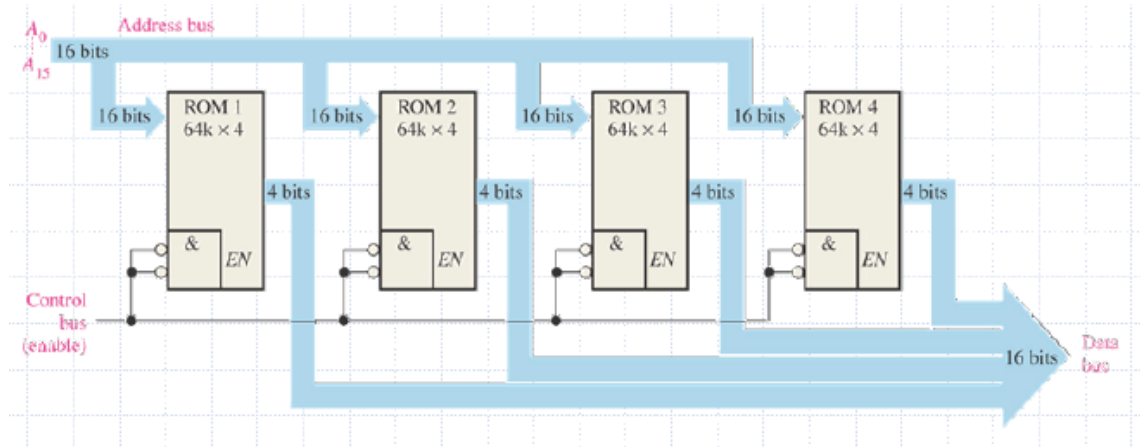
- b) Se tienen módulos de memoria ROM de 64k por 4 bits, y se requiere expandir el tamaño de la palabra a 64k por 16 bits.

Para este caso, si deseo 16 bits a la salida de la memoria, se tiene que:

$$n_{deseado} = n_{inicial} * (\text{cantidad de memorias}), \text{ teniendo que:}$$

$$\text{cantidad de memorias} = 4$$

Para este caso el bus de datos permanece igual y se realiza el siguiente montaje:



Se direcciona de igual forma en todas las memorias y los datos de salida se unen para formar el bus de 16 bits.