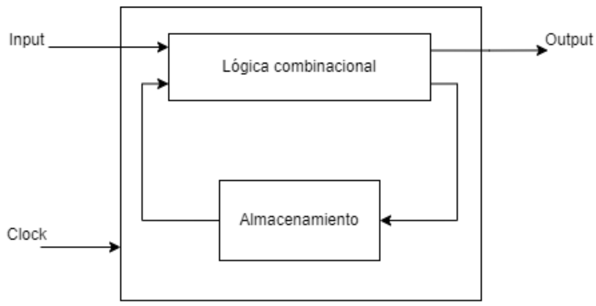
Circuito Lógico Combinacional cuya salida depende de los valores actuales y pasados de las señales de entrada.

Componente:

- Señal de entrada y salida(binaria).
- Señal de reloj (binaria periodica).
- Lógica combinacional(determina la salida y el próximo estado)
- Almacenamiento (mantiene información sobre el estado actual)



0.1 Máquina de estados

Solo hablaremos de los Circuitos Secuenciales Síncronos.
Red de combinacionales y biestables conectados entre si. Puede haber caminos cíclicos, pero tienen que atravesar al menos un biestable. Todos los biestables usan la misma señal reloj. Las señales de entrada se sincronizan con el mismo reloj.
Tiempo del reloj debe ser mayor o igual al tiempo de propagación del flip flop más el tiempo de propagación del CLC[1](ecuación 1).

$$T_c \geq T_p(FF) + T_p(CLC) \tag{1}$$

Existen dos tipos de maquinas de estados:**Mealy** y **Moore**. Son parecidas pero no iguales. Veremos la de Mealy puesto que la de Moore será más fácil. Notesé la diferencia entre ambas, la única

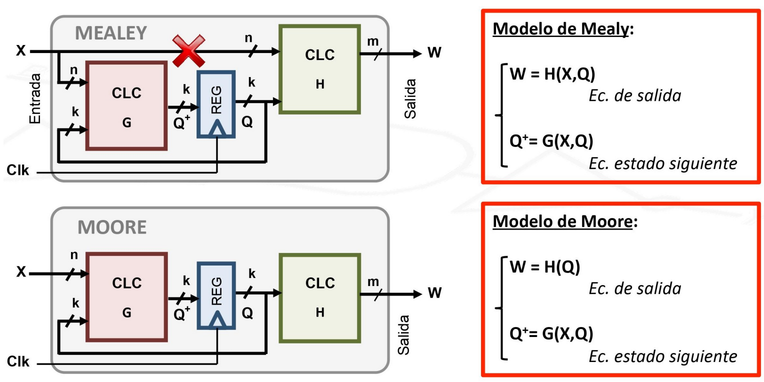


Figura 1: Tipos de maquina de estado

diferencia es la entrada, en Moore solo se dirige a CLC G mientras que en el de Mealy de dirige tanto al CLC G como al CLC H.

[1] Abreviatura de Circuito Lógico Combinacional

0.1.1 Modelo de Mealy

Cualquier CLS[2] puede expresarse como: Agrupando todos los circuitos en un único CLC(en la figura 1, solo contariamos con REG y CLC H) y los biestables(flip flop) en único REG. Es posible que existan más CLC(en la figura 1 se presenta dos CLC,G y H).

Datos que queremos caracterizar

- Número de entradas: X=$(x_{n-1}, x_{n-2}, \ldots, x_1, x_0)$
- Número de salidas W=$(W_{n-1}, W_{n-2}, \ldots, W_1, W_0)$
- Número de estados Q=$(q_{n-1}, q_{n-2}, \ldots, q_1, q_0)$

Por eso necesitamos caracterizar:

- Tabla de verdad de W.
- Tabla de verdad de Q^+.
- Estado inicial de Q

Puede ser un poco complicado de entender, vamos a hacer un ejercicio para entender mejor.

Ejercicio 0.1 Con:

- X=(X_0)
- W=(W_1, W_0)
- Q=(q_1, q_0)
- Estado inicial=(0,0)

Antes de todo, analicemos: Tenemos X que representa las entradas, W que representa las salidas, Q es el número de estados y el estado inicial, que representa el estado en el que el circuito siempre va empezar. Primero, mirando la forma de un CLC de Mealy (1), poseemos dos CLC: G y H. Primero empecemos con G: Este CLC G es un circuito de transición, puesto que no es la salida, sino que las salidas se almacenan para que el siguiente CLC funcione. ¿Como armamos la tabla de verdad?

Primero, como siempre identificamos las entradas: Tenemos X como entrada y Q el número de estados, podemos concluir que nuestras tablan deben tener 3 entradas: $q_1, q_0 y x_0$; escribimos todas las combinaciones pero ¿Cómo sabes que salidas habrá? Las salidas serán las mismas que el número de estados, se les diferencia con un simbolo de suma encima ellos que significa que el estado ha evolucionado. La **tabla de transición** se vería de la siguiente manera:

q_1	q_0	x_0	q_1^+	q_0^+
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	0	0
1	1	0	x	x
1	1	1	x	x

Las x significa que puede tomar cualquier valor puesto que esos valores no nos importan. En este caso, las salidas q^+ serán puestas con esos números a manera de ejemplo pero pueden ser distintos valores, depende de tu proposito. Con esto ya hemos terminado nuestra **tabla de transición Q^+ G**. Antes de pasar el CLC H entendamos el CLC G:

¿Qué nos quiere decir?

Nos da como estado inicial (0,0) teniendo la forma de Q(q_1, q_0). Nuestro punto de partida será (0,0). Ahora ese estado inicial (0,0) evolucionará, cambiará a otro estado (el que queramos, ya

[2] Circuito Lógico Secuencial

depende de nuestro proposito), en este ejemplo el estado inicial tiene dos posibles evoluciones dependiendo si x_0 es 0 o 1. Si x_0=0, el estado inicial (0,0) evoluciona a (0,0) mientras que si x_0=1, el estado (0,0) evoluciona a (0,1). Listo, ahora recordemos que el CLC G tiene en secuencia el CLC H que lo veremos luego y tiene al REGistro; es decir, las salidas (q_1^+, q_0^+) serán las entradas (q_1, q_0). Expliquemos este paso, si tenemos (0,0) y x_0=0, es estado de evolución será (0,0), ese mismo estado será las entradas (q_1, q_0), y entraremos en un bucle infinito, si x_0=1, las entradas (0,0) evolucionan a (0,1), vemos que aquí podemos avanzar en la tabla, ahora salida (0,1) será nuestra entrada y estaremos antes otros dos casos; ahora tenemos como estado inicial (0,1), si x_0=0 la salida será (1,0) que significa que avanzaremos en la tabla pero si x_0=1 no avanzaremos, nos quedaremos en ese estado infinitamente. En otras palabras, las salidas al ser las mismas que las entradas, van avanzando en su evolución, eso depende de nuestra entrada x(no siempre, depende de nuestro propósito). Si seguimos bajo esta lógica, llegaremos hasta el penúltimo estado, donde la entrada (1,0) permanecerá en el mismo estado si x_0=0, y estará en ese bucle hasta que x_0=1, donde evolucionará a (0,0) y se volverá a repetir el ciclo, con esto se asegurar un ciclo evolutivo cerrado.**¿Y el CLC H?**

Es posible añadir más CLC,si añadimos un CLC que vaya después del REG, las entradas del CLC H(el que viene despues del REG) serán las salidas del CLC G. Podemos decir que las salidas del CLC G serán las entradas del mismo CLC G(se autoalimenta con respecto a las entradas) y a su vez serán las entradas del CLC H, pero el CLC H no tiene retroalimentación, solo depende de las entradas y los estados almacenados(salidas del CLC G que se almacenan en REG).

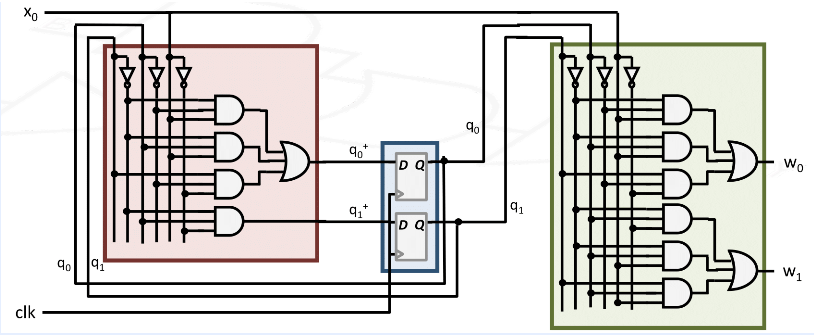
Dada la **Tabla de verdad de la salida**, en este ejemplo las definiremos de la siguiente manera[a]

q_1	q_0	x_0	w_1	w_0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	0

Si te fijas bien, las salidas de la tabla de transición Q^+(CLC G) están contenidas en las entradas de la tabla de verdad de W(Tabla de verdad de la salida), y las salidas son independientes, puesto que estas serán las salidas de nuestra maquina de estado.

¿Y como armo el circuito?

Para armar el circuito se tiene que simplificar ambas tablas a SOP o POS, según nos sea mejor, una vez que hayamos logrado simplificar lo más que podamos procedemos a implementar el circuito, el circuito de nuestro ejemplo sería:



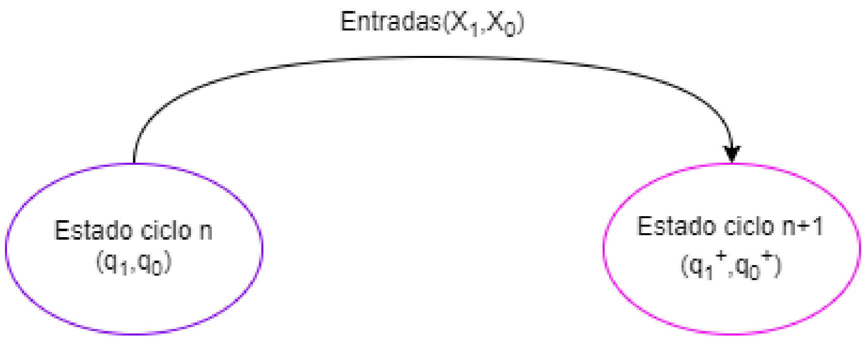
Según lo visto se puede completar la siguiente tabla sabiendo el estado inicial es (0,0):

q_0	0	0	0	1	1	0	1
q_1	0	0	0	0	0	1	0
Ciclo		n	n+1	n+2	n+3	n+4	n+5
X_0		0	1	1	0	1	1
w_0		0	1	1	0	0	1
w_1		0	0	1	1	0	1
q_0^+		0	1	1	0	1	1
q_1^+		0	0	0	1	0	0

[a] Ojo, no siempre saldrá así, ya depende de nuestro proyecto o nuestras necesidades.

0.2 Grafos de estados

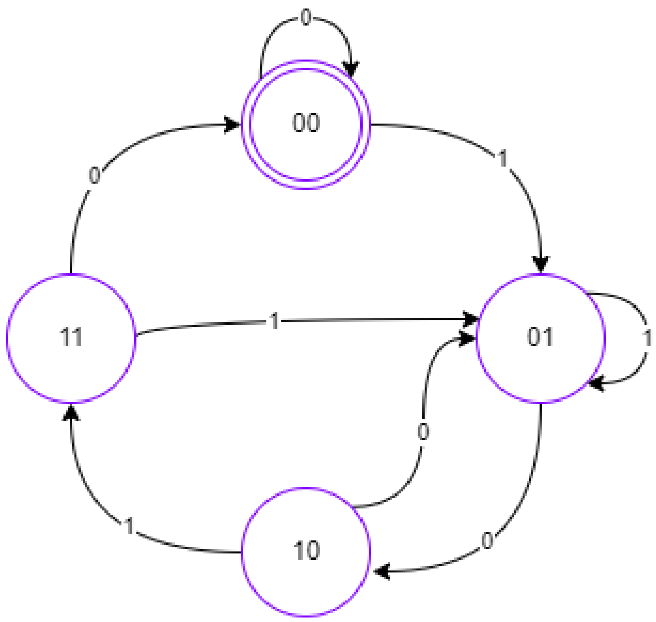
Grafos de estados o diagramas de estado son una forma de expresar las tablas de transición, el formato es el siguiente:



Dada la tabla de transición:

q_1	q_0	x_0	q_1^+	q_0^+
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	1	1
1	1	0	0	0
1	1	1	0	1

Su grafo correspondiente es:

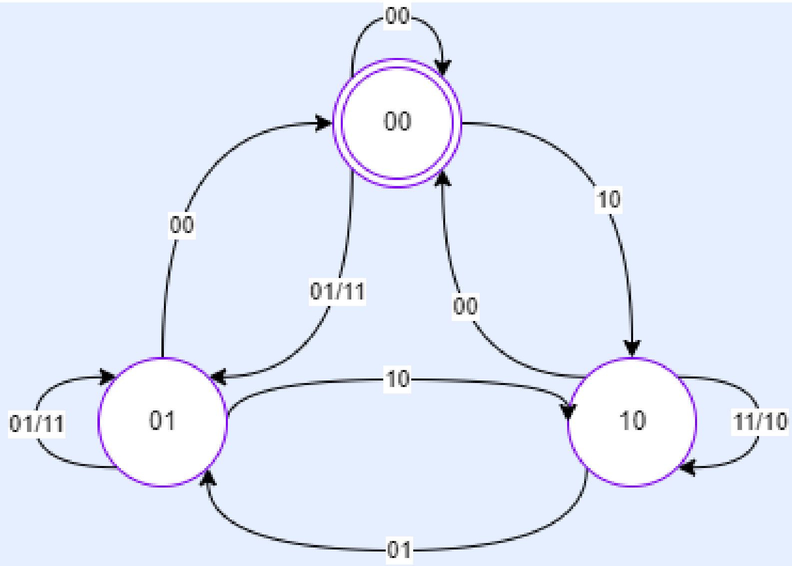


El **estado inicial** es representado con el dobe circulo.

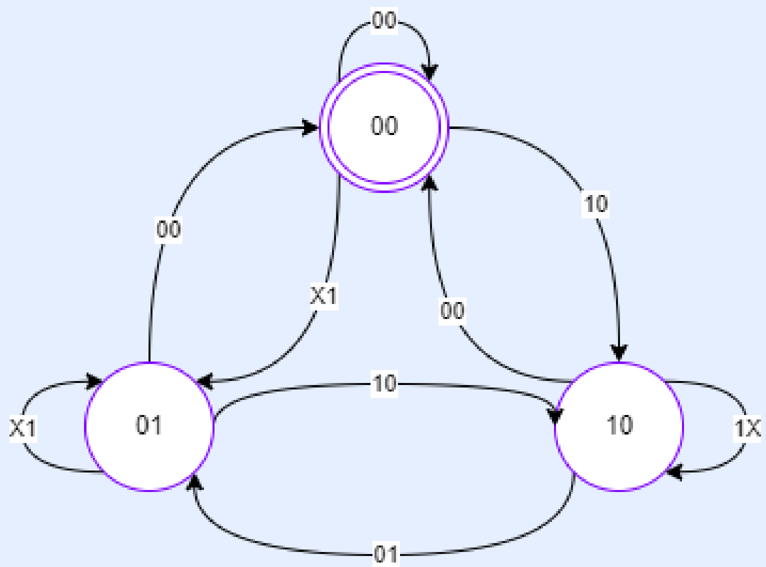
Ejercicio 0.2 Dada la tabla de transición de un CLS, dibuja su grafo de estados:

q_1	q_0	x_1	x_0	q_1^+	q_0^+
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	1	0	1
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

Solución:
Nos fijamos en la salidas y en las entradas, notamos que en las salidas son: (0,0), (0,1) y (1,0) en las entradas tenemos las mismas combinaciones excepto (1,1). Como conclusión de este análisis podemos decir que la entrada (1,1) no sé usa, ya que ninguna salida hace que lleguemos a este estado, por ende nuestros estados son: (0,0), (0,1) y (1,0). Una vez tengamos los estados definidos procedemos a hacer las flechas basandonos en las entradas x_1 y x_0.

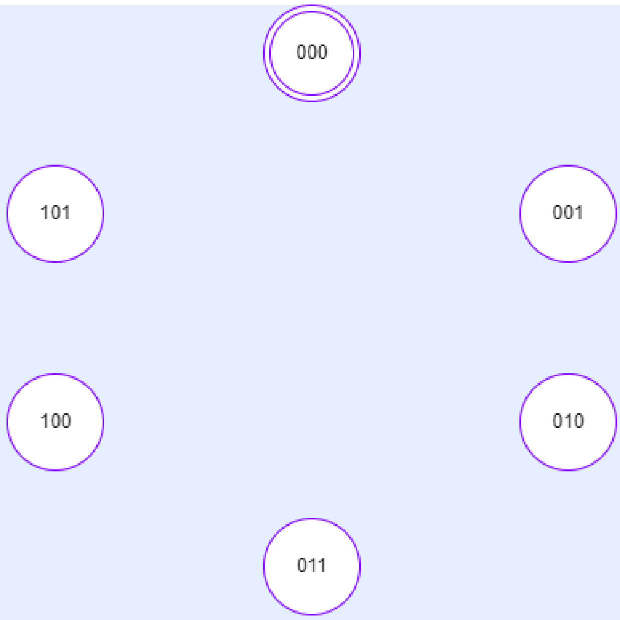


Nota que algunas flechas tiene dos estados, por ejemplo: 01/11 o 11/10. Cuando tenemos estos casos podemos simplificarlo ya que considerando la forma que tienen (x_1,x_0), no nos importa el valor de x_1 y no nos importa x_0 respectivamente en cada caso, asi que podemos escribir una x indicando no importa el estado de alguna entrada, simplificando quedaría así:

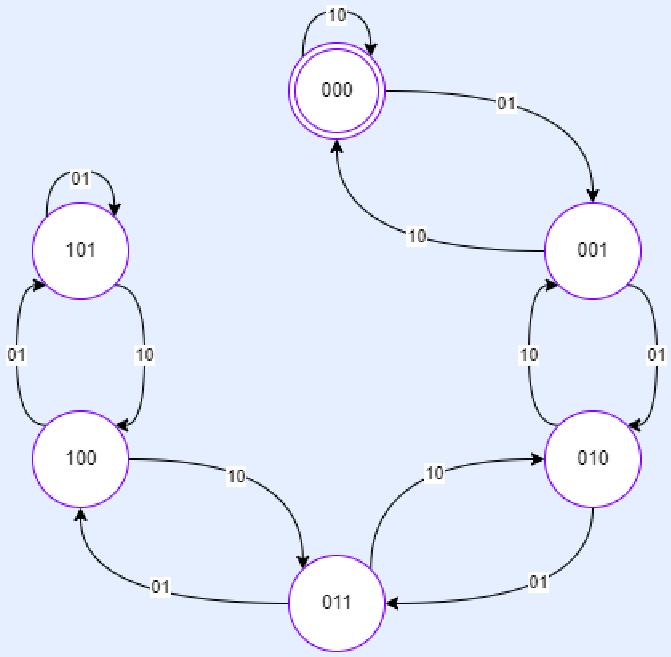


Aplicando la teoría a un ejemplo más real:

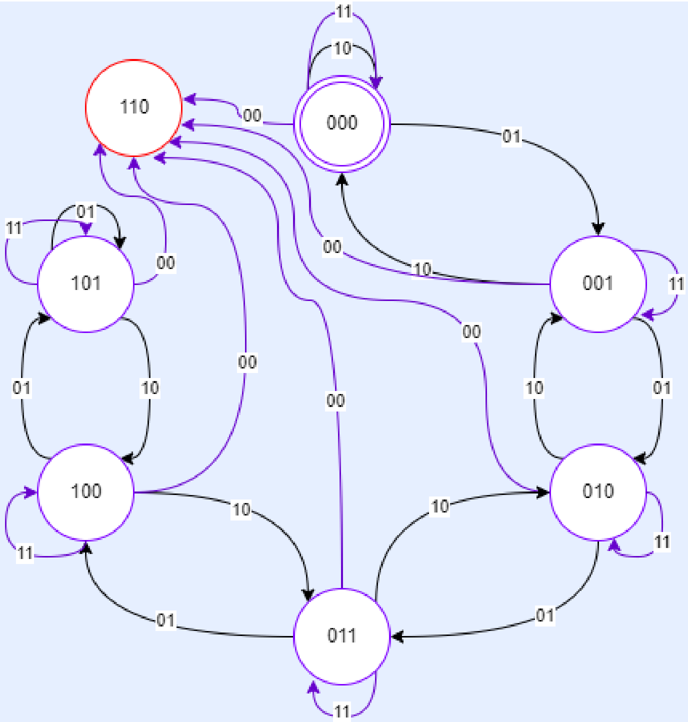
Ejercicio 0.3 Tenemos un contador desde 0 al 5, con entradas x_1 y x_0. Hacer el grafos teniendo en cuenta que x_1=+1 y x_0=-1 con estado inicial en 0,0.
Solución:
Desde el 0 al 5 tenemos 6 números, por lo tanto haremos que cada número en binario sea un estado teniendo un total de 6, así que necesitaremos 2^3=8 estados(como solo necesitamos 6 dos estados no serán usados):



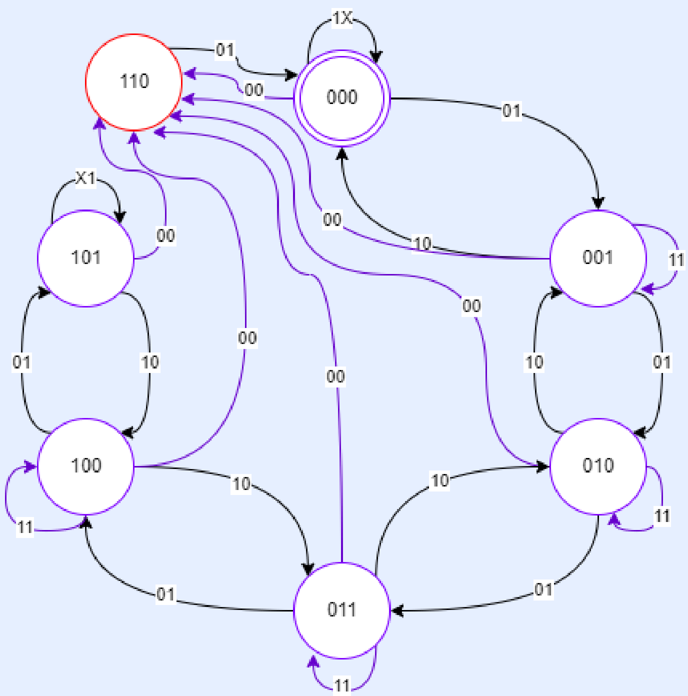
El problema nos dice que si la combinación de las entradas es (0,1) significa que sumamos uno y si es (1,0) restamos uno. Sabiendo esto vamos a unir los estados con flechas:



Listo, tenemos las sumas y las restas, nota que cuadno llegue al estado 000 y restamos se queda en el mismo estado; lo mismo que si sumamos al último estado (101) se quedará en el mismo estado, pero podemos mejorar este modelo. Definiremos lo siguiente: que cuando las entradas sean (1,1) nos mantengamos en el mismo estado, ademas añadiremos un estado nulo que lo definiremos como (110^a):



Ya hemos implementado el estado OFF, el estado (1,1) nos mantenemos en el mismo estado, con el estado (0,0) apagamos el circuito. Se puede mejorar los estados donde hay dos entradas que nos llevan al mismo estado y aumentamos un flecha para salir del estado OFF:



Resumiendo, sabiendo las entradas (x_1,x_0):

- (0,1) incremento
- (1,0) decremento

- (1,1) ni uno pulsado
- (0,0) apagamos el circuito

La tabla de verdad sería la siguiente:

Estado	q_2	q_1	q_0	x_1	x_0	q_2^+	q_1^+	q_0^+
Estado 0	0	0	0	0	0	1	1	0
	0	0	0	0	1	0	0	1
	0	0	0	1	0	0	0	0
	0	0	0	1	1	0	0	0
Estado 1	0	0	1	0	0	1	1	0
	0	0	1	0	1	0	1	0
	0	0	1	1	0	0	0	0
	0	0	1	1	1	0	0	1
Estado 2	0	1	0	0	0	1	1	0
	0	1	0	0	1	0	1	1
	0	1	0	1	0	0	0	1
	0	1	0	1	1	0	1	0
Estado 3	0	1	1	0	0	1	1	0
	0	1	1	0	1	1	0	0
	0	1	1	1	0	0	1	0
	0	1	1	1	1	0	1	1
Estado 4	1	0	0	0	0	1	1	0
	1	0	0	0	1	1	0	1
	1	0	0	1	0	0	1	1
	1	0	0	1	1	1	0	0
Estado 5	1	0	1	0	0	1	1	1
	1	0	1	0	1	1	0	1
	1	0	1	1	0	1	0	0
	1	0	1	1	1	1	0	1
Estado OFF	1	1	0	0	0	x	x	x
	1	1	0	0	1	0	0	0
	1	1	0	1	0	x	x	x
	1	1	0	1	1	x	x	x

Ya solo nos queda simplificar las salidas q^+:

Mapa de Karnaugh para q_2^+

q_1q_0 \ x_1x_0	00	01	11	10
00	1	0	0	0
01	1	0	0	0
11	1	1	0	0
10	1	0	0	0

$q_2 = 0$

q_1q_0 \ x_1x_0	00	01	11	10
00	1	1	1	0
01	1	1	1	1
11	x	x	x	x
10	x	0	x	x

$q_2 = 1$

Mapa de Karnaugh para q_1^+

q_1q_0 \ x_1x_0	00	01	11	10
00	1	0	0	0
01	1	1	0	0
11	1	0	1	1
10	1	1	1	0

$q_2 = 0$

q_1q_0 \ x_1x_0	00	01	11	10
00	1	0	0	1
01	1	0	0	0
11	x	x	x	x
10	x	0	x	x

$q_2 = 1$

Mapa de Karnaugh para q_0^+

q_1q_0 \ x_1x_0	00	01	11	10
00	0	1	0	0
01	0	0	1	0
11	0	0	1	0
10	0	1	0	1

$q_2 = 0$

q_1q_0 \ x_1x_0	00	01	11	10
00	0	1	0	1
01	1	1	1	0
11	x	x	x	x
10	x	0	x	x

$q_2 = 1$

Agrupamos según nos convenga e implementamos los circuitos con las funciones lógicas obtenidas. ∎

[a]Aunque este es el binario del número 7, definiremos que este estado sea neutro/apagado/off.