



**Universidad
Internacional
de Valencia**

10-2021 GRADO EN INGENIERÍA INFORMÁTICA

05GIIN
FUNDAMENTOS DE COMPUTADORES

ACTIVIDAD EVALUATIVA 3

07/01/2022

Juan Diego Tomás Nolan

Miguel Ángel Gagliardo

Salceda Fernández-Barredo del Amo

Verónica Romero Márquez



Universidad
Internacional
de Valencia

ÍNDICE

1. Introducción	2
2. Objetivos	3
3. Desarrollo	4
4. Conclusiones	16



1. Introducción

Un circuito lógico secuencial es un circuito combinacional cuya salida depende de los valores actuales y pasados de las señales de entrada. Se trata de circuitos en los que aparecen lazos de feedback (salidas del circuito pueden actuar como valores de entrada).

Los componentes de un circuito combinacional secuencial son:

- Señales de entrada y salida (señales binarias).
- Señal de reloj (señal binaria con forma periódica)
- Lógica combinacional (determina la salida y el próximo estado)
- Almacenamiento (mantiene información sobre el estado actual)



2. Objetivos

Diseñar un sistema combinacional secuencial sincrónico capaz de controlar un semáforo inteligente en un cruce de peatones.

La primera fase de la secuencia será con el semáforo de peatones en rojo y el de circulación de coches en verde. Mediante un pulsador se cambiará la luz verde del semáforo de coches a color ámbar, la cual se mantendrá encendida durante 5 segundos. Acto seguido la luz ámbar se apagará y se encenderá la roja. El semáforo de peatones se pondrá verde y se activará un contador descendente de 12 segundos que se podrá visualizar. Cuando acabe el temporizador todo volverá a la fase inicial, la luz del peatón volverá a ponerse roja y la del semáforo de coches de color verde. Este último estado permanecerá así hasta que se vuelva a pulsar el botón, momento en que la secuencia volverá a activarse.

En caso de volverse a presionar el botón para el cruce de los peatones mientras se ejecuta la secuencia anterior, éste no tiene ningún efecto.

3. Desarrollo

- CONTADOR DESCENDENTE DE 12

Para el diseño del contador descendente de 12 a 0, se han utilizado 4 flip-flops y 4 biestables tipo D.

Tabla de la verdad con las salidas Q de cada biestable (del 0 al 3) (negro) que son las entradas a los flip-flops, los cambios de estado t+1 (rojo) en orden descendente y las salidas (verde) de los flip-flops para cada dígito, que constituyen la entrada a los biestables D de cada bit del dígito de 4 bits que se necesita para codificar el número 12 en código binario.

ESTADO ACTUAL				ESTADO FUTURO				ENTRADAS BIESTABLES			
Q3	Q2	Q1	Q0	Q3	Q2	Q1	Q0	D3	D2	D1	D0
0	0	0	0	1	1	0	0	1	1	0	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0	0	0	1	0
0	1	0	0	0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	0	0	1	0	0
0	1	1	0	0	1	0	1	0	1	0	1
0	1	1	1	0	1	1	0	0	1	1	0
1	0	0	0	0	1	1	1	0	1	1	1
1	0	0	1	1	0	0	0	1	0	0	0
1	0	1	0	1	0	0	1	1	0	0	1
1	0	1	1	1	0	1	0	1	0	1	0
1	1	0	0	1	0	1	1	1	0	1	1

- Tabla de verdad del biestable D:

Operación	Entradas				Salidas	
	Preset	Clear	Reloj (CLK)	D	Q	\bar{Q}
Preset	0	1	X	X	1	0
Clear	1	0	X	X	0	1
Prohibido	0	0	X	X	1	1
Set	1	1	↑	1	1	0
Reset	1	1	↑	0	0	1
Hold	1	1	0	X	Q _o	\bar{Q}_o

Tabla de excitación:

Q	\bar{Q}	D
0	0	0
0	1	1
1	0	0
1	1	1

- Ecuaciones para el contador descendente de 12:

$$D_3 = \bar{Q}_3 \bar{Q}_2 \bar{Q}_1 \bar{Q}_0 + Q_3 \bar{Q}_2 \bar{Q}_1 Q_0 + Q_3 \bar{Q}_2 Q_1 \bar{Q}_0 + Q_3 \bar{Q}_2 Q_1 Q_0 + Q_3 Q_2 \bar{Q}_1 \bar{Q}_0$$

$$D_2 = \bar{Q}_3 \bar{Q}_2 \bar{Q}_1 \bar{Q}_0 + \bar{Q}_3 Q_2 \bar{Q}_1 Q_0 + \bar{Q}_3 Q_2 Q_1 \bar{Q}_0 + \bar{Q}_3 Q_2 Q_1 Q_0 + Q_3 \bar{Q}_2 \bar{Q}_1 \bar{Q}_0$$

$$D_1 = \bar{Q}_3 \bar{Q}_2 Q_1 Q_0 + \bar{Q}_3 Q_2 \bar{Q}_1 \bar{Q}_0 + \bar{Q}_3 Q_2 Q_1 Q_0 + Q_3 \bar{Q}_2 Q_1 Q_0 + Q_3 Q_2 \bar{Q}_1 \bar{Q}_0 + Q_3 \bar{Q}_2 \bar{Q}_1 \bar{Q}_0$$

$$D_0 = \bar{Q}_3 \bar{Q}_2 Q_1 \bar{Q}_0 + \bar{Q}_3 Q_2 \bar{Q}_1 \bar{Q}_0 + \bar{Q}_3 Q_2 Q_1 \bar{Q}_0 + Q_3 \bar{Q}_2 \bar{Q}_1 \bar{Q}_0 + Q_3 \bar{Q}_2 Q_1 \bar{Q}_0 + Q_3 Q_2 \bar{Q}_1 \bar{Q}_0$$

- Mapas de karnaugh:

D3

		Q1 Q3			
Q0 Q2		00	01	11	10
	00	1	0	1	0
	01	0	1	0	0
	11	0	0	0	0
	10	0	1	1	0

$\bar{Q}_3 \bar{Q}_2 \bar{Q}_1 \bar{Q}_0 + Q_3 Q_1 \bar{Q}_2 + Q_3 Q_2 \bar{Q}_1 \bar{Q}_0 + Q_0 \bar{Q}_2 Q_3$

D2

		Q1 Q3			
Q0 Q2		00	01	11	10
	00	1	1	0	0
	01	0	0	0	1
	11	1	0	0	1
	10	0	0	0	0

$(\bar{Q}_2 \bar{Q}_1 \bar{Q}_0) + (\bar{Q}_3 Q_2 Q_0) + (\bar{Q}_3 Q_2 Q_1)$

D1

		Q1 Q3			
Q0 Q2		00	01	11	10
	00	0	1	0	0
	01	1	1	0	0
	11	0	0	0	1
	10	0	0	1	1

$(Q_2 \bar{Q}_1 \bar{Q}_0) + (Q_3 \bar{Q}_1 \bar{Q}_0) + (\bar{Q}_2 Q_1 Q_0) + (\bar{Q}_3 Q_1 Q_0)$

D0

		Q3 Q1			
Q2 Q0		00	01	11	10
		0	1	1	1
01		1	1	0	1
11		0	0	0	0
10		0	0	0	0

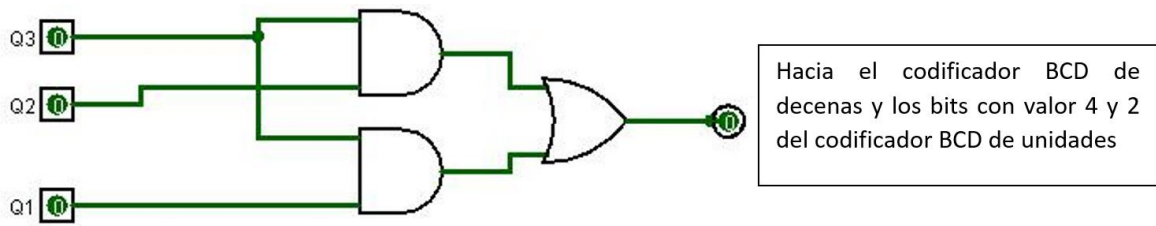
$(\bar{Q}_2 Q_1 \bar{Q}_0) + (\bar{Q}_3 Q_2 \bar{Q}_0) + (Q_3 \bar{Q}_1 \bar{Q}_0)$

Para solucionar el problema de mostrar en el display los números del 12 hacia el cero, se dispusieron dos displays de BCD a 7-segmentos. Uno para las unidades y otro para las decenas.

En el display de las decenas, solamente aparecerá el 1 cuando sea necesario, por lo que, combinando dos tablas para mostrar la secuencia de las unidades y de las decenas de forma conjunta, queda de la siguiente manera:

NÚMERO BINARIO COMPLETO	BINARIO PARA LAS DECENAS	BINARIO PARA LAS UNIDADES	NÚMERO DECIMAL COMPLETO
0000	0000	0000	0
0001	0000	0001	1
0010	0000	0010	2
0011	0000	0011	3
0100	0000	0100	4
0101	0000	0101	5
0110	0000	0110	6
0111	0000	0111	7
1000	0000	1000	8
1001	0000	1001	9
1010	0001	0000	10
1011	0001	0001	11
1100	0001	0010	12

En las celdas marcadas en gris, puede observarse que, si el número es mayor a 9, siempre va a tener los bits Q3yQ2 con valor 1 ó los que tienen Q3 y Q1 en 1 (marcados en amarillo). Con esta información, generamos dos puertas AND y una puerta OR que queda como sigue, y que irá al contador de las decenas cuando la salida sea 1, generando el número 0001.



Así, cuando en el contador de las unidades tenga que marcar 10, en el de las decenas tendrá que codificar un binario 0001 y en el de las unidades, un binario 0000, y hasta 12 según muestra la tabla.

Para que ambos displays muestren los números 10, 11 y 12, es necesario modificar sólo en estos números el contador de las unidades, sumando el número 0110 (6 decimal) sin tener en cuenta los acarreos de las sumas. Esta suma tiene que coincidir en tiempo cuando el display de las decenas muestre 1, por lo que el resultado de este sumador irá a la misma salida que la de la puerta OR descrita anteriormente, cuando ésta sea 1.

Así si sumamos al número 10 (1010), el número 6 (0110):

$$\begin{array}{r} \boxed{111} \text{ no se tienen en cuenta los acarreos} \\ 1010 = (10)_{10} \\ + 0110 = (6)_{10} \\ \hline \boxed{0000} \rightarrow \text{n}^\circ \text{ para las decenas} \end{array}$$

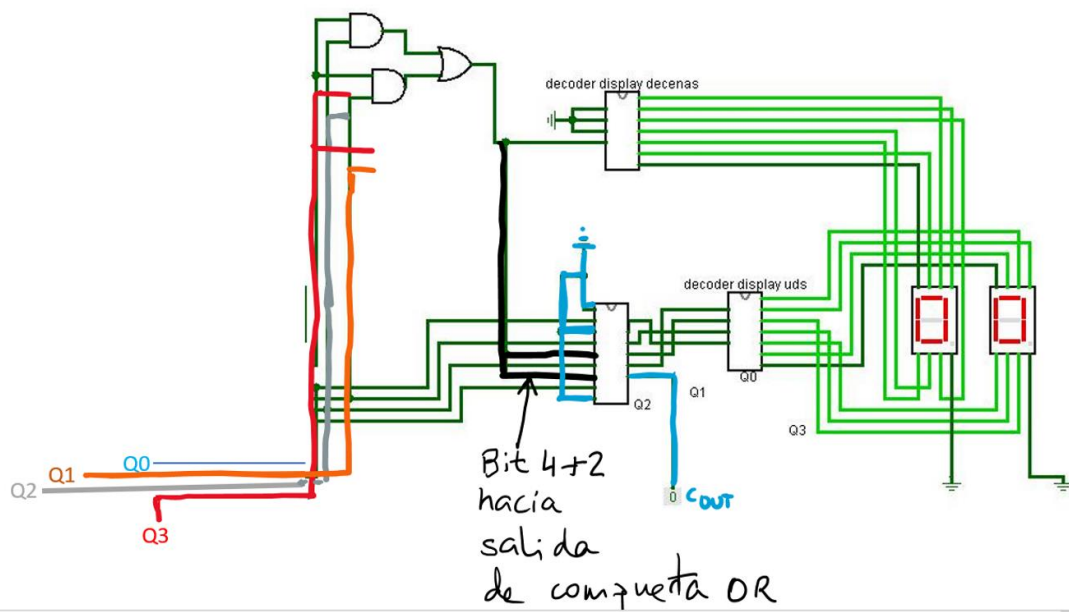
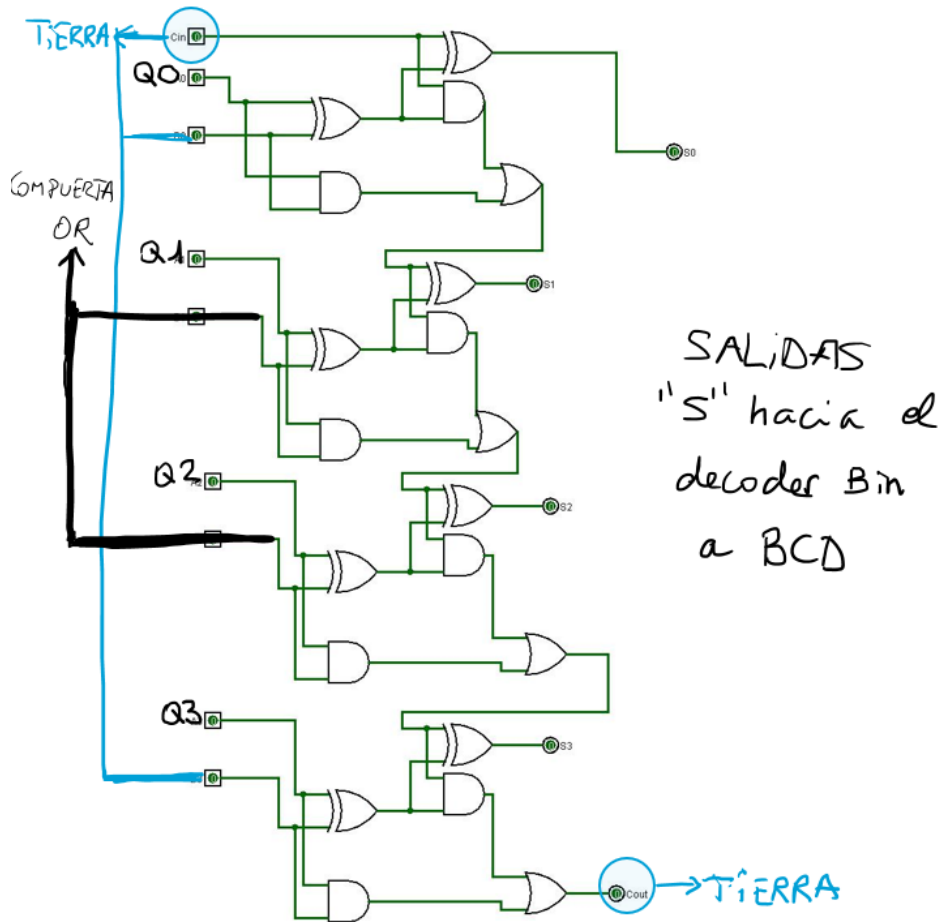
Hacemos lo mismo para el (11)₁₀ y el (12)₁₀:

$$\begin{array}{r} \boxed{111} \rightarrow \text{Descontamos el acarreo} \\ 1011 = (11)_{10} \\ + 0110 = (6)_{10} \\ \hline \boxed{0001} \rightarrow \text{n}^\circ \text{ para las decenas} \end{array}$$

$$\begin{array}{r} \boxed{111} \rightarrow \text{Descontamos el acarreo} \\ 1100 = (12)_{10} \\ + 0110 = (6)_{10} \\ \hline \boxed{0010} \rightarrow \text{n}^\circ \text{ para las decenas} \end{array}$$

Montamos un sumador que añada de forma constante el n° 0110 cuando la salida de la compuerta OR sea 1. Se pasan a tierra las salidas de los bits con valor 0 y 8, carry in y carry out (porque no tenemos en cuenta el acarreo de salida), y se unen los bits con valor 2 y 4, y dicha línea se une a la compuerta OR como se describe a continuación, para que solamente sume cuando la salida de OR sea 1.

Sumador 4 bits:



En el decodificador de binario a BCD que usamos para las decenas, solamente hemos dejado la entrada que sale de la compuerta OR más la del sumador y que entra en el LSB. Así sólo codifica 1 o 0, mientras que el resto de los bits más significativos quedan en 0.

- TEMPORIZADOR DESCENDIENTE DE 5 SEGUNDOS:

ESTADO ACTUAL			ESTADO FUTURO			ENTRADAS BIESTABLES		
Q2	Q1	Q0	Q2	Q1	Q0	D2	D1	D0
0	0	0	1	0	0	1	0	0
0	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	1
0	1	1	0	1	0	0	1	0
1	0	0	0	1	1	0	1	1
1	0	1	1	0	0	1	0	0
1	1	0	1	0	1	1	0	1
1	1	1	1	1	0	1	1	0

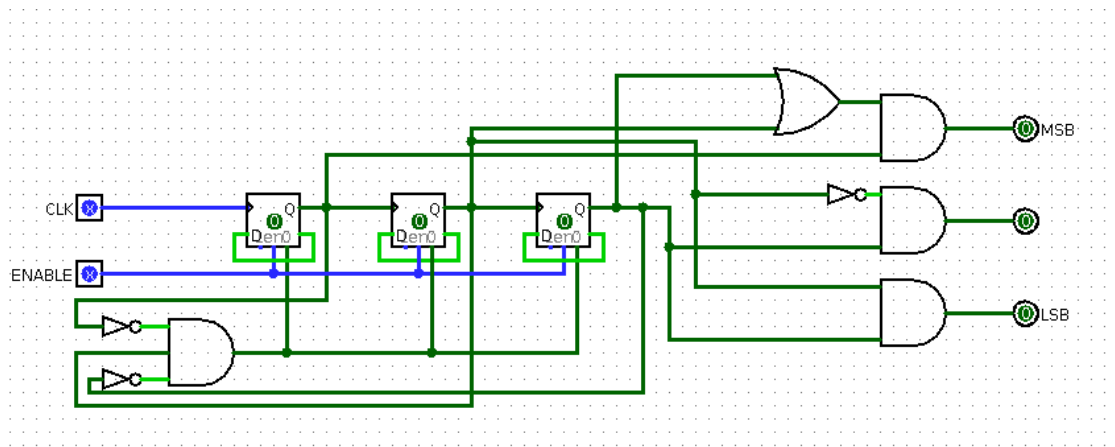
Como el contador va de 7 a 0 (dado que son 3 bits) por la siguiente tabla:

Numero Decimal	A
7	111
6	110
5	101
4	100
3	011
2	010
1	001
0	000

Se necesita que el contador vaya de 5 a 0. Ahora bien, se puede aprovechar el reset (clear) del Flip Flop tipo D para resetear "a 0" (o sea, 000). Como no podemos hacer que el contador empiece de 5 (siempre va a iniciar desde 7), si podemos hacer que vaya a 0.

Con lo cual nos queda mover los outputs 2 bits para el deco a BCD (o sea, en el 111 pasara a ser 5, el 110 el 4 y así).

Dado que estamos aprovechando que cuando el circuito llegue a 010 se ejecutara un trigger que resetea a través de la compuerta CLEAR con un AND + 2 NOT, como se ve en el circuito:



Lo que nos queda es un circuito bastante sencillo que solo es activado por una compuerta ENABLE (en 1) + un CLK.

Para "correr" los 2 bits (de 7 a 5, de 6 a 4 y así) se utilizó la siguiente tabla + diagramas de Karnaugh:

Decimal Original	Entradas			Nuevo Decimal		Salidas	
	C	B	A		O ₀	O ₁	O ₂
7	1	1	1	5	1	0	1
6	1	1	0	4	1	0	0
5	1	0	1	3	0	1	1
4	1	0	0	2	0	1	0
3	0	1	1	1	0	0	1
2	0	1	0	0	0	0	0
1	0	0	1	X	X	X	X
0	0	0	0	X	X	X	X

Mapas de Karnaugh

(MSB) O₀ = BC

O₁ = B'C

(LSB) O₂ = AB+AC = A(B+C)

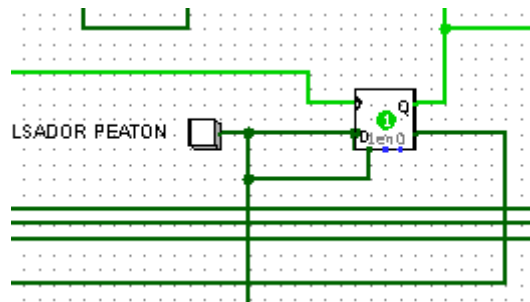
CB					
A		00	01	11	10
		0	1	0	1
0		0	0	1	0
1		0	0	1	0

CB					
A		00	01	11	10
		0	1	0	1
0		0	0	0	1
1		0	0	0	1

CB					
A		00	01	11	10
		0	1	0	1
0		0	0	0	0
1		0	1	1	1

- PULSADOR DEL SEMAFORO:

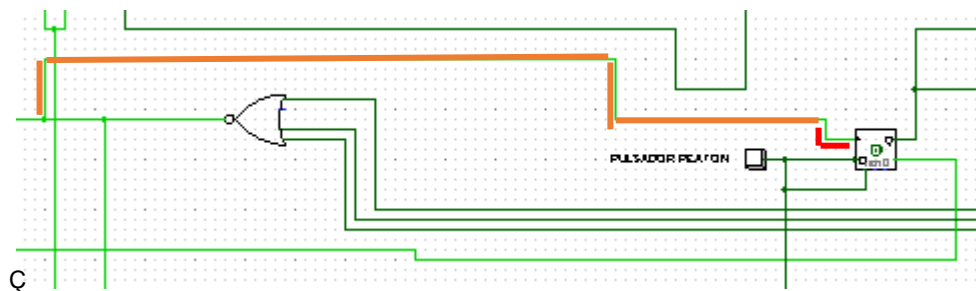
Para diseñar el pulsador del semáforo hemos optado por un biestable ya que al poner un interruptor lógico el “1” o “0” se quedaba perpetuamente marcado, es decir, es como si el peatón no dejara de pulsar el botón de peatones y este estuviera siempre pulsado. Para lograr que solo se pulse el botón una vez y vuelva a su estado lógico de “0”, hemos utilizado un biestable en cuya entrada “D” y “SET” hemos conectado un pulsador.



En cuanto pulsamos el pulsador este introduce un “1” lógico y se queda almacenado en el biestable ya que la entrada “D” y la entrada “SET” hacen que haya un “1” lógico al pulsarlo.

Este “1” hará que en la salida “Q” del biestable haya un “1” y este accionara el “enable” del contador que va de 5 a 0.

Las salidas “Q” del primer contador (5 a 0) van todas a una compuerta lógica NOR para que cuando el primer contador llegue a 0 (que son todas las salidas del primer contador a “0”) obtengamos en la suma lógica “0” que este negado (compuerta NOR) nos saque un “1” lógico que hará que el reloj(CLK) del biestable del pulsador de peatón actualice el estado del pulsador que ahora será “0”



- ACTIVADOR DEL PRIMER CONTADOR Y SEGUNDO CONTADOR:

Para hacer que cuando se pulse el botón de peatones el primer contador funcione y el segundo se quede en espera hemos utilizado biestables para almacenar estados, de esta manera podemos activar el segundo contador utilizando el enable de los biestables del segundo contador.

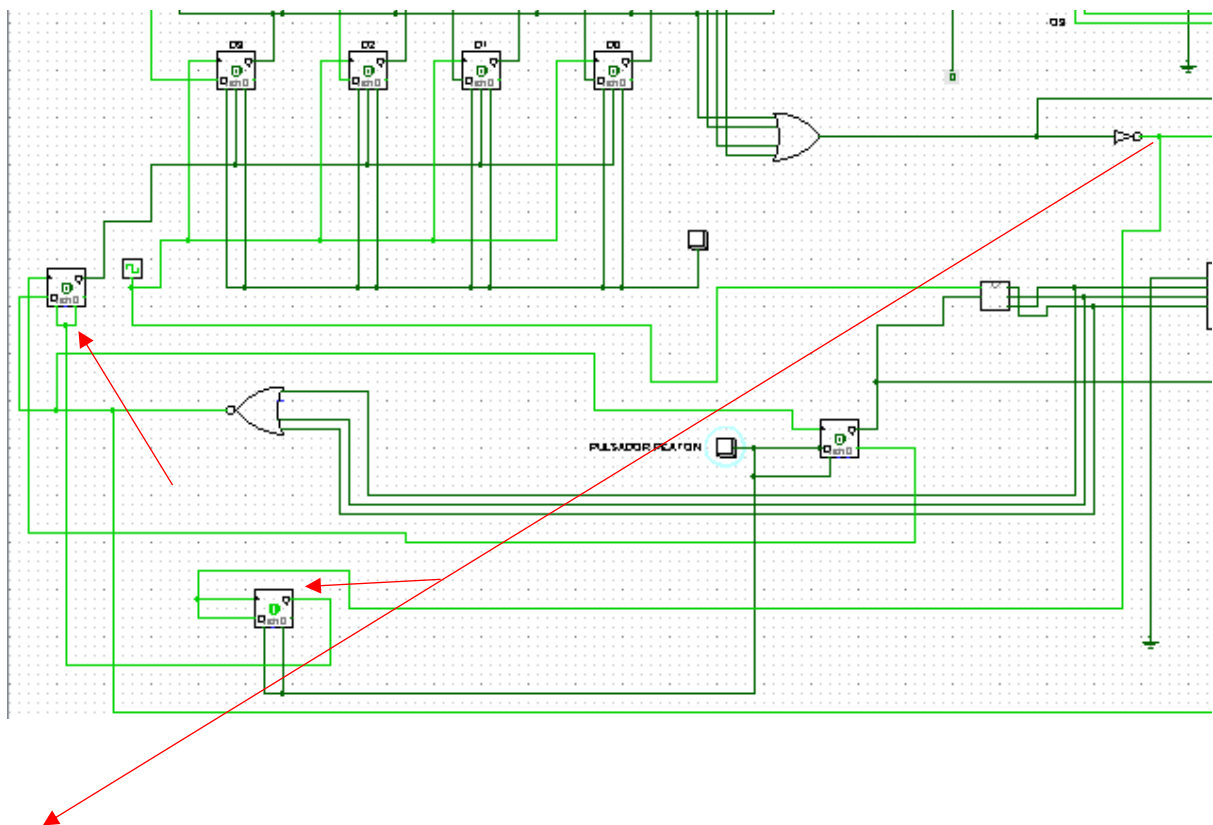
Cuando pulsamos el botón de peatones como hemos explicado anteriormente del biestable que hemos utilizado para almacenar la información del pulsador, cuando en la salida Q del biestable del pulsador hay un “1” lógico este activa el enable del primer contador y hemos utilizado la \bar{Q} para mandar un pulso de reloj

a al biestable que hace que el segundo circuito se active actualizando así la información que hay en la entrada del biestable (que va al segundo contador) que será “1” o “0” dependiendo de si la cuenta del primer biestable ha llegado a “0”.

Del botón del pulsador de peatones hemos sacado una línea a otro biestable que es el que controla que cuando el contador 2 llegue a 0 reseté la entrada del biestable del segundo contador poniendo este a “0” y así se logra parar para que no esté constantemente haciendo loops.

Este biestable adicional que hemos conectado al botón tendrá un “1” lógico en su salida que va conectado directamente al reset del biestable que controla el inicio y parada del contador 2.

Cuando se pulsa el botón de peatones este biestable que controla la activación del segundo contador hace reset cambiando su estado a “0”.

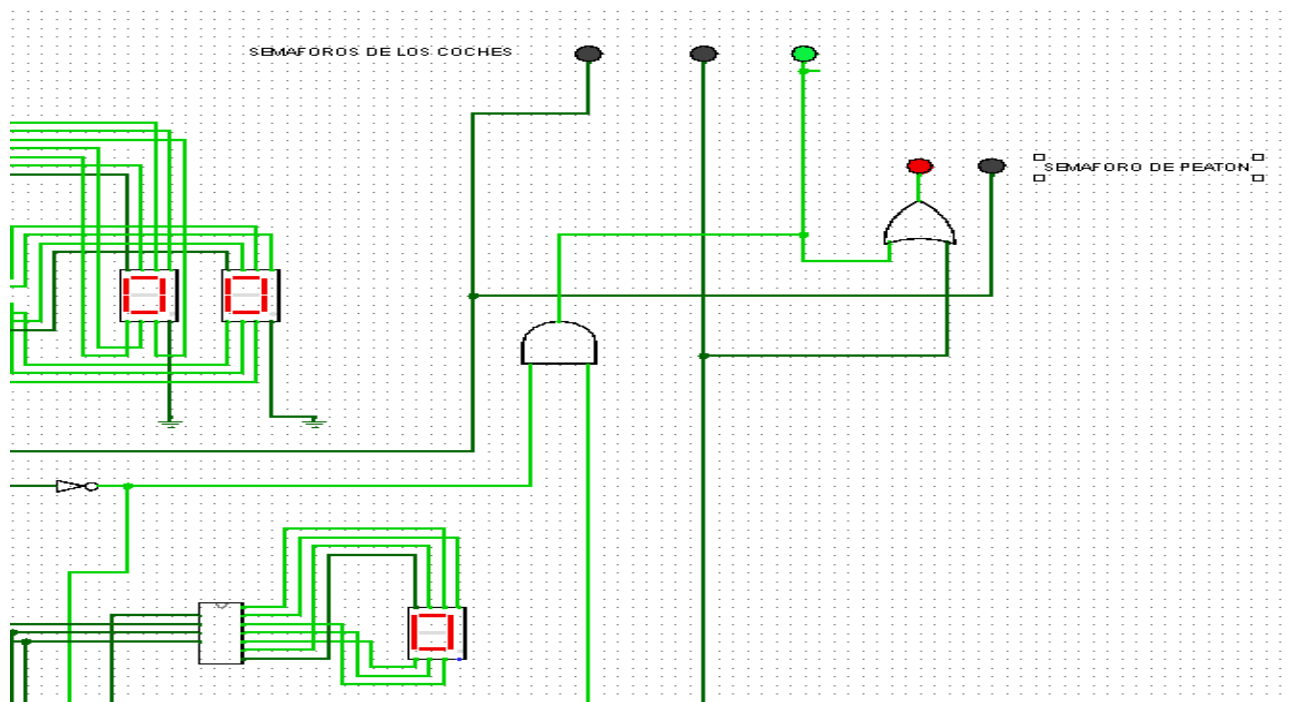


Si el contador 2 está a “0” (es decir no se ha iniciado aun) cogemos la salida del sumador OR de todas las salidas del contador 2 que sumaran “0” y lo invertimos con un inversor (se podría hacer también con una NOR) este va directamente a la entrada del biestable que controla el funcionamiento del segundo biestable mandando un reset constante (para que no se inicialice) hasta que el estado del biestable del pulsador cambie a “1”.

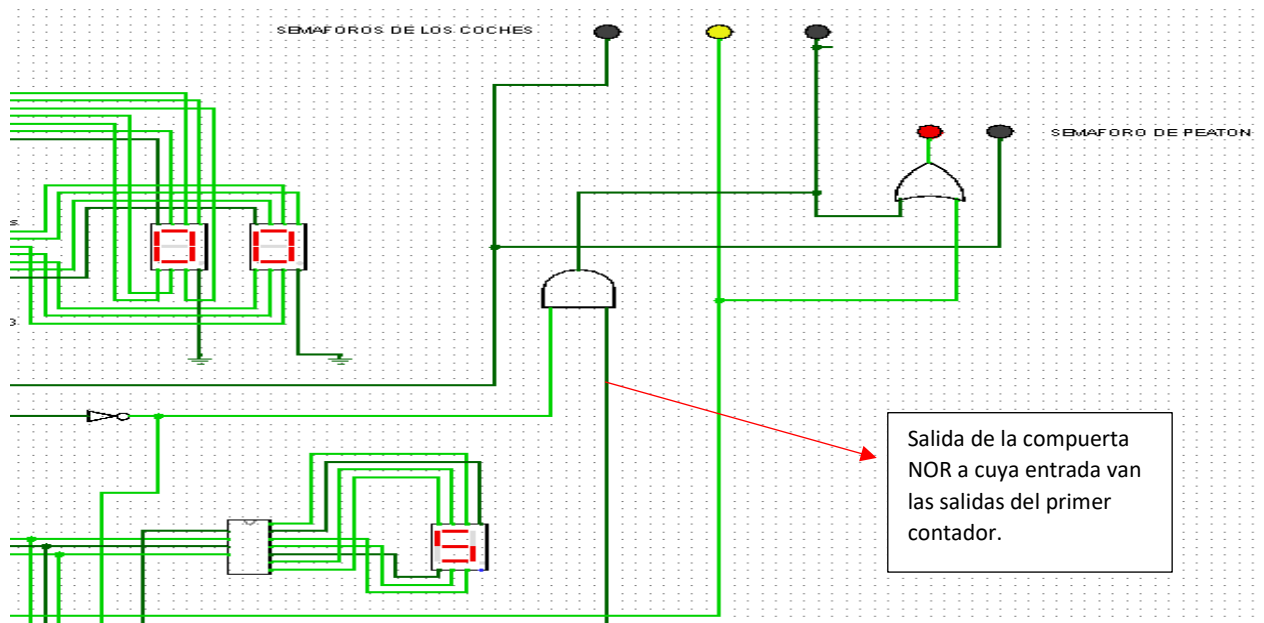
- SEMÁFOROS:

La parte de los semáforos la hemos hecho empleando un poco la lógica como los biestables que controlan los contadores, no ha hecho falta hacer tabla de la verdad de las salidas de los biestables, simplemente aplicando un poco de lógica hemos logrado hacer que los semáforos tengan un estado u otro dependiendo del contador que en ese momento toca.

El estado inicial es "SEMAFORO VERDE PARA LOS COCHES", "ROJO PARA PEATON", este estado lo sacamos del estado inicial que son todos los contadores a 0 sin inicializar ya que el peatón todavía no ha pulsado el botón.

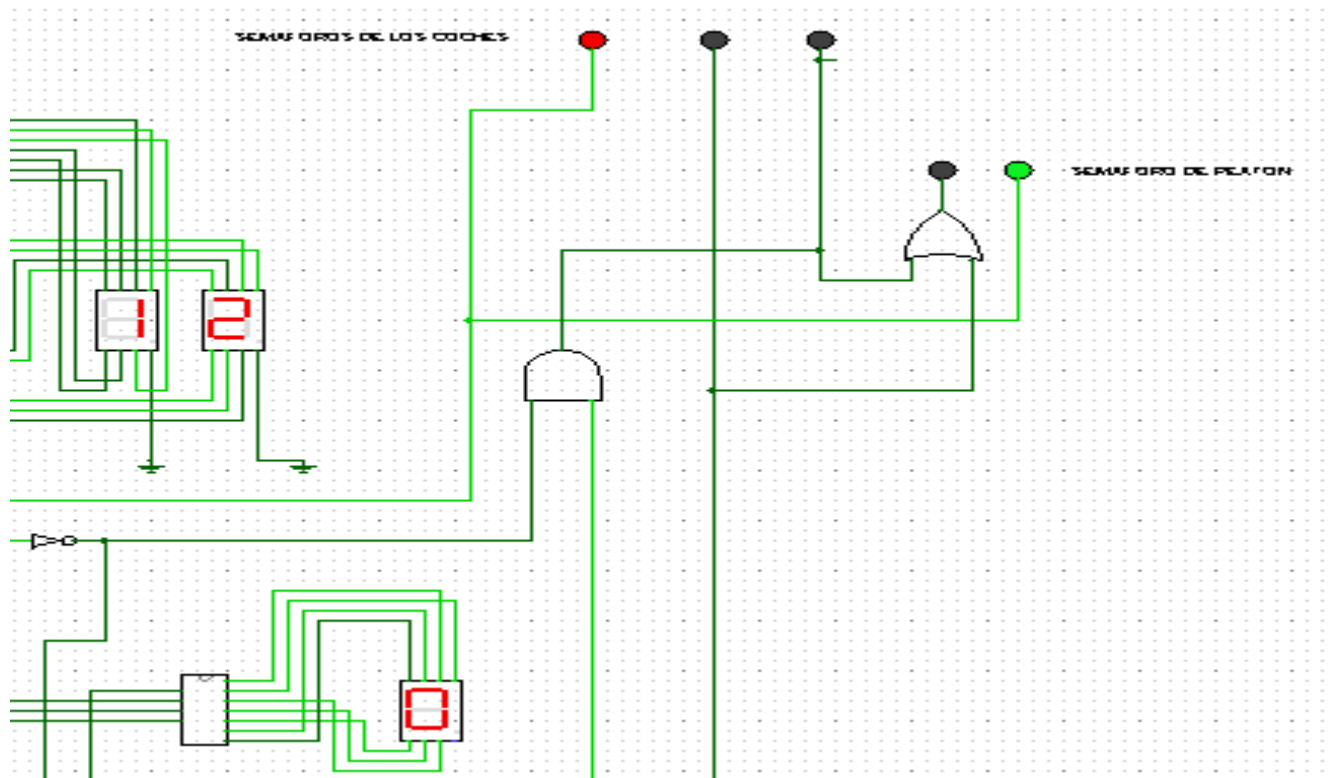


En cuanto se pulsa el botón de peatones el semáforo cambia su estado a "AMBAR COCHES" y "ROJO PEATON"



Si el circuito está inicializado y el primer contador empieza a funcionar la salida de la compuerta NOR cuyas entradas son las salidas del primer biestable, al no haber un "1" (que es cuando las salidas del primer contador están a "0") manda un 0 a la compuerta AND que es la que enciende el semáforo de coches Rojo. La salida que va al led "AMBAR" lo sacamos directamente del biestable del pulsador de peatones que será "1" mientras está el contador funcionando y cuyo estado pasa a "0" cuando el contador 2 empieza a funcionar y hace cambiar el estado del biestable del pulsador a "0".

Por último tenemos el estado del contador 2, que sería cuando el peatón puede cruzar y es "ROJO PARA COCHES" y "VERDE PARA PEATON".



Esto lo conseguimos sacando directamente de la puerta OR en cuyas entradas tenemos las salidas del segundo contador (cuando estas no están a 0 siempre sacara un "1" porque siempre hay algún bit en la salida activo) un cable que va directo al led de "ROJO" y al led de peatón "VERDE"



4. Conclusiones

Hemos tenido dificultades en conseguir que se activarán las decenas. Para que ambos displays muestren los números 10, 11 y 12 ha sido necesario introducir un sumador para estas cifras. Añadiendo un 0110 (6 decimal) en la salida del display de las unidades y sin tener en cuenta los acarreos de las sumas.

Nos ha resultado especialmente dificultoso a la hora de conectar los contadores entre sí porque cuando activábamos el contador, los dos contadores hacían la cuenta decreciente al mismo tiempo. Hemos pensado colocar un botón con un biestable porque si dejamos una entrada tipo 1-0, en cuanto pongamos en la entrada del pulsador de peatones un uno, éste actuará como si estuviera constantemente pulsado. Para ello hemos conectado un pulsador que solo marca un uno en el momento en el que el peatón pulsa el botón, y éste uno que se obtiene es almacenado en el registro del biestable.