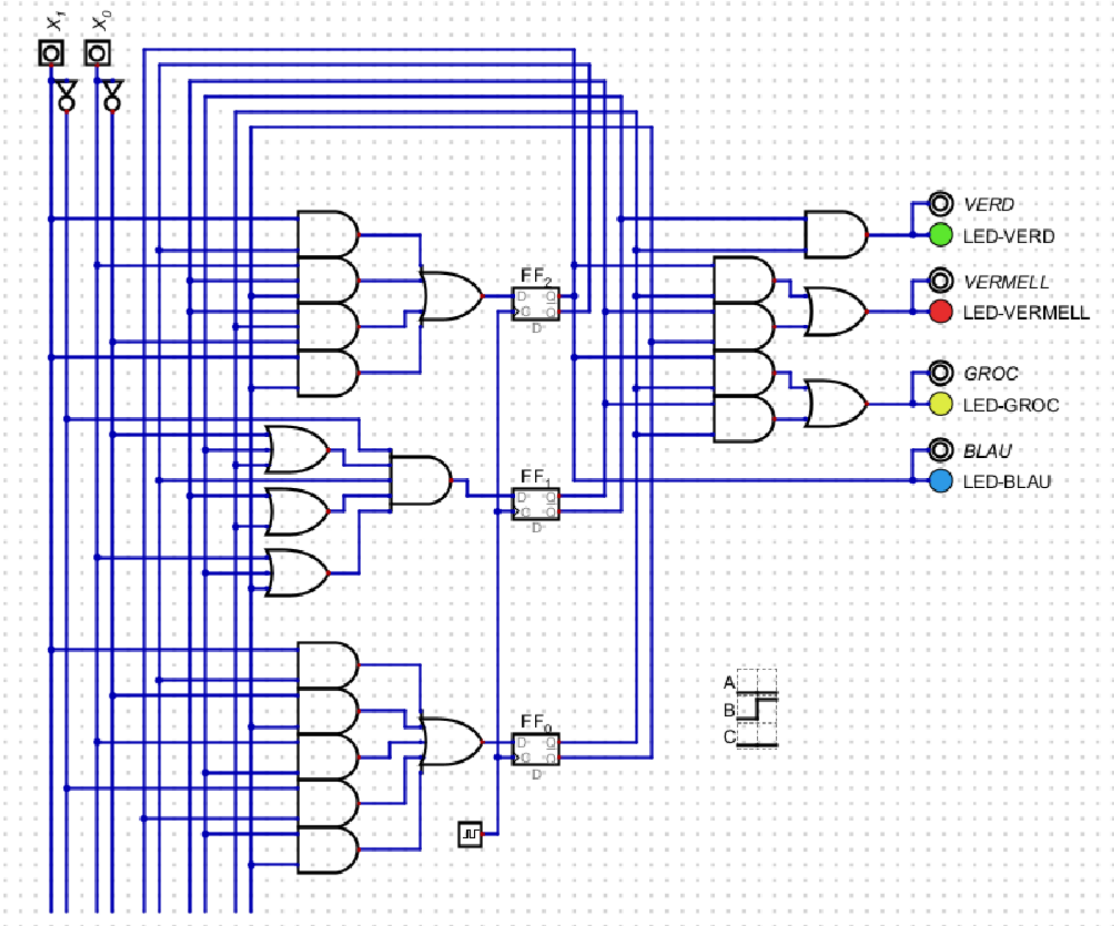
Sistemas digitales: Práctica secuenciales



GORDIOLA FRAU, ANTONI 43482642-S

SAYES SÁNCHEZ, ALEJANDRO 43215170-X

ÍNDICE

INTRODUCCIÓN:

A lo largo de la práctica de sistemas secuenciales, hemos desarrollado lo que representa un circuito controlador de las luces de árboles de navidad en la temporada de 2021/2022. Concretamente, con luces verdes, rojas, amarillas y azules, con las posibilidades de estar encendidas por separado, todas a la vez o todas apagadas, por lo que el sistema controla el manejo de estas.

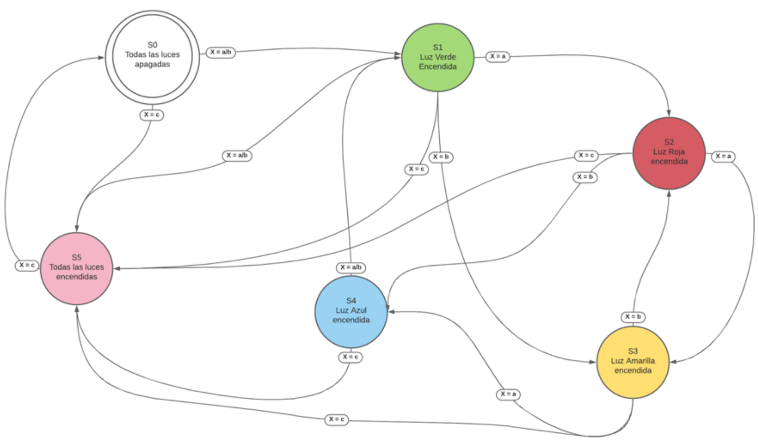
Para desarrollar este circuito hemos implementado lo que se llama una máquina de Moore, que, aunque los motivos y desarrollo se explicarán próximamente, es una máquina que permite el manejo y acciona las secuencias necesarias para el funcionamiento de este.

Como codificaciones de los valores en el circuito, hemos usado las indicadas por la guía de la práctica de forma que se recibe una entrada con los posibles valores A, B y C que representan el interruptor. Las salidas, por tanto, al ser 4 indican cada luz por separado si se encuentran encendidas o apagadas.

En cuanto a los procesos seguidos, han sido la realización del diagrama de estados para identificar los estados necesarios que hay que considerar, la construcción de una tabla de excitación de estados y de salida a partir de las codificaciones del diagrama anterior. Posteriormente obtener las expresiones algebraicas de las salidas y de los valores a tomar por los biestables. Al final de todo montamos el sistema en la herramienta digital Digital para comprobar su funcionamiento.

DIAGRAMAS DE TRANSICIÓN DE ESTADOS:

Como podemos apreciar en la imagen contamos con un estado inicial que se trata del S_0 en donde todas las luces se encuentran apagadas y además es desde donde empieza la secuencia. En el caso de que este estado inicial reciba una "a" o una "b" siempre pasará al estado S_1 en el que la única luz encendida es la verde. Si la entrada sigue siendo "a" se producirá una secuencia de colores de: verde, rojo, amarillo, azul y de nuevo a la verde, creando así un ciclo. En el caso de que nos encontremos en el estado S_1 y la entrada sea continúe siendo "b" durante todo el tiempo se producirá una secuencia de colores de: verde, amarillo, rojo, azul y de nuevo un reinicio de ciclo. En el caso de que en medio de la secuencia cambiemos la entrada de "a" a "b", por ejemplo, si nos encontrábamos en el S_2 (luz roja encendida) en vez de ir hacia el estado S_3 iríamos hacia el estado S_4 ya que al ser diferentes entradas tienen diferentes secuencias. Por último, cabe destacar que nos encontremos en el estado que nos encontremos si la entrada que lee la máquina es "c", se encenderán todas las luces y hasta que no cambie el valor de la entrada, todas las luces seguirán encendidas.



CODIFICACIÓN DE ESTADOS:

Hemos codificado los estados con la posibilidad de tener todas las luces apagadas, solamente la verde, la roja, la amarilla, la azul y todas a la vez como los estados del S_0 hasta el S_5 respectivamente.

Estados	Q_2	Q_1	Q_0
S_0	0	0	0
S_1	0	0	1
S_2	0	1	0
S_3	0	1	1
S_4	1	0	0
S_5	1	0	1

Como podemos ver en la tabla el S_0 es el estado en que todas las luces se encuentran apagadas y por eso las representamos con todo 0's. En el resto de casos hemos seguido el orden normal de codificación binaria aprendida en los vídeos de la asignatura. En la tabla también podemos observar que no llegamos nunca a tener todo 1's y esto se debe a que tenemos seis estados a codificar y para poder llevar a cabo esta codificación necesitamos el uso de 3 variables, dando la posibilidad de codificar hasta 8 estados. Aquí los dos estados restantes no los mostramos, pero en el resto de tablas aparecerás como valores don't care (X).

JUSTIFICACIÓN DEL TIPO DE MÁQUINA:

Hemos escogido el tipo de máquina de Moore. Esto es porque, al ser las salidas si las luces se encuentran apagadas o encendidas, sus condiciones solo dependen del estado en el que se encuentren, definidos por cómo deben estar las luces. De forma que las entradas, como es el modo de secuencia de luces, no intervienen en el efecto directo de las salidas.

TABLA DE TRANSICIÓN DE ESTADOS:

Para definir las transiciones entre estados, indicado todo en la siguiente tabla, hemos partido del diagrama de estados inicial. Mediante este, hemos podido ver directamente cómo eran las evoluciones y transiciones entre estados y las salidas según sus entradas.

Obteniendo por una parte los valores codificados de las transiciones de estados (Q_2^{n+1}, Q_1^{n+1}, Q_0^{n+1}) y las salidas (Y_3, Y_2, Y_1, Y_0) también hemos podido indicar directamente las tablas necesarias para las funciones de entrada de los flip-flops tipo D que requerirá el circuito final. Estas las hemos sabido por la tabla de funcionamiento del flip-flop tipo D y las funciones que devuelve este.

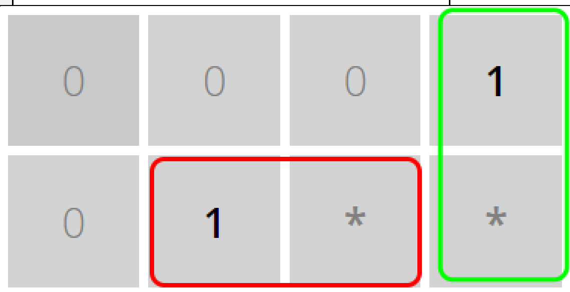
Cabe destacar también, que al no haber estados S_6 y S_7 y entrada 11 definidos, todos los valores que llegan a tomar las evoluciones de estados, salidas y por tanto funciones de los biestables, se convierten en valores don't care, indicadas con 'x'.

Q_2	Q_1	Q_0	X_1	X_0	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	Y_3	Y_2	Y_1	Y_0	D_2	D_1	D_0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
0	0	0	0	1	0	0	1	0	0	0	0	0	0	1
0	0	0	1	0	1	0	1	0	0	0	0	1	0	1
0	0	0	1	1	x	x	x	0	0	0	0	x	x	x
0	0	1	0	0	0	1	0	0	0	0	1	0	1	0
0	0	1	0	1	0	1	1	0	0	0	1	0	1	1
0	0	1	1	0	1	0	1	0	0	0	1	1	0	1
0	0	1	1	1	x	x	x	0	0	0	1	x	x	x
0	1	0	0	0	0	1	1	1	0	0	0	0	1	1
0	1	0	0	1	1	0	0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	1	1	0	0	0	1	0	1
0	1	0	1	1	x	x	x	1	0	0	0	x	x	x
0	1	1	0	0	1	0	0	0	1	0	0	0	1	1
0	1	1	0	1	0	1	0	0	1	0	0	1	0	0
0	1	1	1	0	1	0	1	0	1	0	0	1	0	1
0	1	1	1	1	x	x	x	0	1	0	0	x	x	x
1	0	0	0	0	0	0	1	0	0	1	0	1	0	0
1	0	0	0	1	0	0	1	0	0	1	0	0	1	0
1	0	0	1	0	1	0	1	0	0	1	0	1	0	1
1	0	0	1	1	x	x	x	0	0	1	0	x	x	x
1	0	1	0	0	0	0	1	1	1	1	1	0	0	1
1	0	1	0	1	0	0	1	1	1	1	1	0	0	1
1	0	1	1	0	0	0	0	1	1	1	1	0	0	0
1	0	1	1	1	x	x	x	1	1	1	1	x	x	x
1	1	0	0	0	x	x	x	x	x	x	x	x	x	x
1	1	0	0	1	x	x	x	x	x	x	x	x	x	x
1	1	0	1	0	x	x	x	x	x	x	x	x	x	x
1	1	0	1	1	x	x	x	x	x	x	x	x	x	x
1	1	1	0	0	x	x	x	x	x	x	x	x	x	x
1	1	1	0	1	x	x	x	x	x	x	x	x	x	x
1	1	1	1	0	x	x	x	x	x	x	x	x	x	x
1	1	1	1	1	x	x	x	x	x	x	x	x	x	x

MINIMIZACIÓN DE FUNCIONES:

Minimización de Y_3

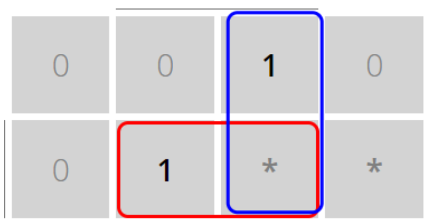
	Q_0	0	1
Q_2	Q_1		
0	**0**	0	0
0	**1**	1	0
1	**1**	x	x
1	**0**	0	1



$Y_3 = Q_1 * Q_0' + Q_2 * Q_0$

Minimización de Y_2

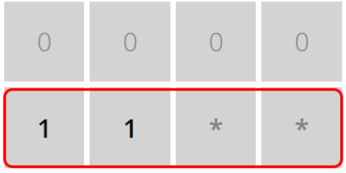
	Q_0	0	1
Q_2	Q_1		
0	**0**	0	0
0	**1**	0	1
1	**1**	x	x
1	**0**	0	1



$Y_2 = Q_1 * Q_0 + Q_2 * Q_0$

Minimización de Y_1

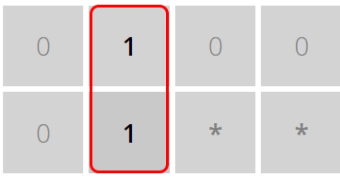
	Q_0	0	1
Q_2	Q_1		
0	**0**	0	0
0	**1**	0	0
1	**1**	x	x
1	**0**	1	1



$Y_1 = Q_2$

Minimización de Y_0

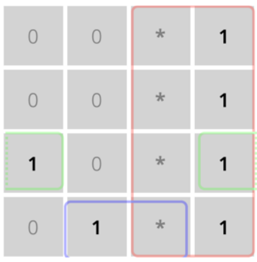
	Q_0	0	1
Q_2	Q_1		
0	0	0	1
0	1	0	0
1	1	x	x
1	0	0	1



$Y_0 = Q_1{}'*Q_0$

Minimización de D_2 con $Q_2 = 0$

D_2		X_1	0	0	1	1
		X_0	0	1	1	0
Q_1	Q_0					
0	0		0	0	X	1
0	1		0	0	X	1
1	1		1	0	X	1
1	0		0	1	X	1

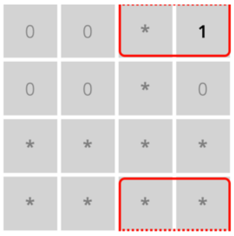


$D_2 = X_1 + X_0*Q_1*Q_0{}' + Q_1*Q_0*X_0{}'$

Minimización de D_2 con $Q_2 = 1$

D_2		X_1	0	0	1	1
		X_0	0	1	1	0
Q_1	Q_0					
0	0		0	0	X	1
0	1		0	0	X	0

1	**1**		X	X	X	X
1	**0**		X	X	X	X

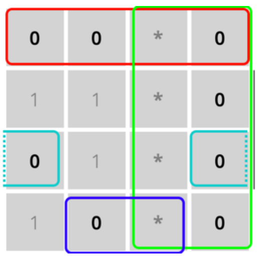


$D_2 = X_1*Q_0$

Expresión algebraica final de $D_2 = X_1 * Q_2' + X_0*Q_1*Q_0' + Q_1*Q_0*X_0' + X_1*Q_0$

Minimización de D_1 con $Q_2 = 0$

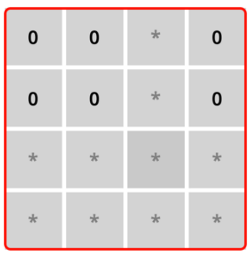
D_1		**X_1**	**0**	**0**	**1**	**1**
		X_0	**0**	**1**	**1**	**0**
Q_1	**Q_0**					
0	**0**		0	0	X	0
0	**1**		1	1	X	0
1	**1**		0	1	X	0
1	**0**		1	0	X	0



$D_1 = X_1' * (X_0'+Q_1'+Q_0) * (Q_1+Q_0) * (X_0+Q_1'+Q_0') + Q_2'$

Minimización de D_1 con $Q_2 = 1$

D_1		**X_1**	**0**	**0**	**1**	**1**
		X_0	**0**	**1**	**1**	**0**
Q_1	**Q_0**					
0	**0**		0	0	X	0
0	**1**		0	0	X	0
1	**1**		X	X	X	X
1	**0**		X	X	X	X

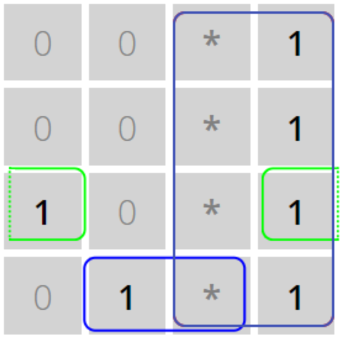


$D_1 = Q_2'$

Expresión algebraica final de $D_1 = X_1' * (X_0'+Q_1'+Q_0) * (Q_1+Q_0) * (X_0+Q_1'+Q_0') * Q_2'$

Minimización de D_0 con $Q_2 = 0$

D_0		X_1	0	0	1	1
		X_0	0	1	1	0
Q_1	Q_0					
0	0		1	1	X	1
0	1		0	1	X	1
1	1		0	0	X	1
1	0		1	0	X	1



$D_0 = X_1 + X_0' * Q_0' + X_0*Q_1'$

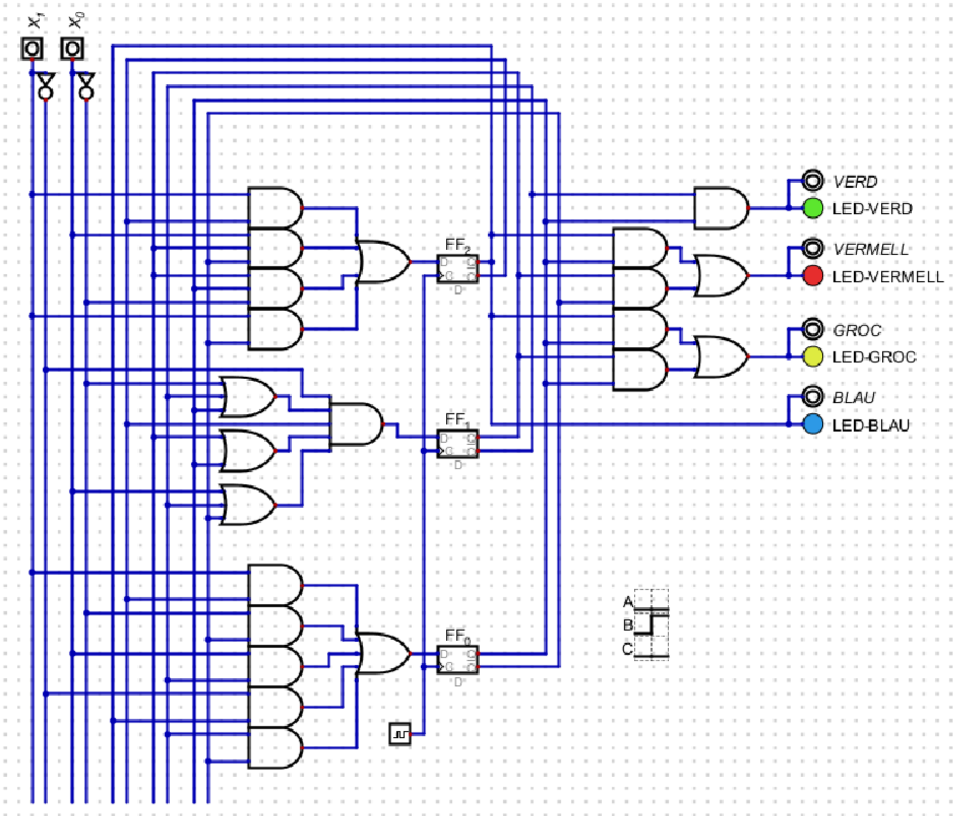
Minimización de D_0 con $Q_2 = 1$

D_2		X_1	0	0	1	1
		X_0	0	1	1	0
Q_1	Q_0					
0	0		1	1	X	1
0	1		1	1	X	0
1	1		X	X	X	X
1	0		X	X	X	X

$D_0 = X_1' + Q_1'*Q_0'$

Expresión algebraica final de $D_0 = X_1 * Q_2'+ X_0' * Q_0' + X_0*Q_1' + X_1'*Q_2 + Q_1'*Q_0'$

CIRCUITO:



x1
x0
FF2
FF1
FF0
VERD
LED-VERD
VERMELL
LED-VERMELL
GROC
LED-GROC
BLAU
LED-BLAU
A
B
C

JUEGO DE PRUEBAS:

Para comprobar los resultados del circuito mediante un juego de pruebas hemos grabado una serie de vídeos comprobando diferentes secuencias y combinaciones valores de entrada. Están principalmente diferenciados entre las entradas del interruptor A, B y C.

Juego de pruebas con la entrada A:

https://uibes-my.sharepoint.com/:v:/g/personal/afg011_id_uib_cat/EaCsAo8SQ1ZNmCj2l8DTwTMBdJyLGrqQ8A9rtC3SQPcnhQ?e=CAvtXS

Juego de pruebas con la entrada B

https://uibes-my.sharepoint.com/:v:/g/personal/afg011_id_uib_cat/EQmUI6KUw5NFtqObMGdCfLkBUW5ZkHy86oOUf1eiIxr1aw?e=mtgfCq

Juego de pruebas con la entrada C:

https://uibes-my.sharepoint.com/:v:/g/personal/afg011_id_uib_cat/ETzPuUemEJ5AtyNj7pUUSmYBuxP1vFOLaUwVZqM2ZtcApA?e=DFCKBY

Juego de pruebas completo y ampliado con diferentes cambios entre entradas:

https://uibes-my.sharepoint.com/:v:/g/personal/afg011_id_uib_cat/EXJK3XYNedRHhhQ54oacD34BJcWOxgZ2B1S4zkpWi7lCAw?e=Hl0H6T

CONCLUSIÓN:

Tras haber realizado enteramente esta segunda práctica de circuitos, pero en este caso secuenciales, hemos podido ver las diferencias notables con los circuitos combinacionales. Completamente, el proceso sistemático de síntesis aprendido para realizar los sistemas secuenciales ha sido de mucha ayuda a la hora de la planificación y desarrollo de la práctica, así como también las bases de la primera práctica de circuitos combinacionales.

Hemos podido ver un nuevo punto de vista, como es el funcionamiento de los sistemas secuenciales en los que se guarda información anterior con flip-flops y elementos nuevos, con un resultado que consideramos más ilustrativo que la práctica anterior, principalmente por la posibilidad de tener más cambios y opciones al poder manejar y probar el resultado final. La visualización también se apoya en la línea temporal de entradas y salidas, útil para ver directamente los cambios y resultados.

Cabe destacar que el uso de sistemas combinacionales usados también ha sido más variado por el principal cambio de simplificación de funciones con 5 variables en lugar de 4 como era anteriormente. Esto en parte también era más complejo y hemos requerido un tiempo mayor para resolverlo. Para finalizar, destacamos la importancia de la lógica secuencial obtenida en las clases para organizar la estructura.