

1. Ejercicio 2

1.1. Introducción

En esta sección se procede a realizar una máquina de estados capaz de detectar una secuencia de bits dada, siendo esta 1-1-0-1.

1.2. Implementación

Para poder realizar este detector se consideró que el último bit de la secuencia puede ser el primero de una nueva, es decir, dada una cadena de bits de la forma 1-1-0-1-1-0-1, la máquina de estados detecta dos secuencias correctas de bits. También se consideró que, si un carácter es incorrecto, la el sistema vuelve a su estado base inicial, comenzando el proceso de nuevo. Es así que se realizó el siguiente diagrama de estados, utilizando una máquina de Mealy:

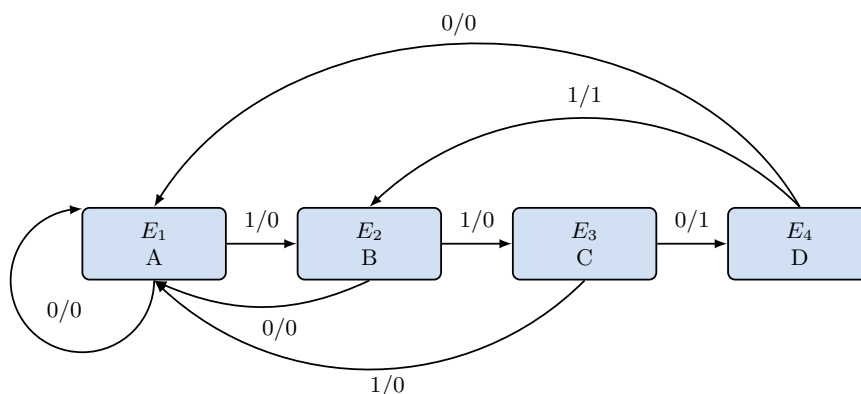


Figura 1: Diagrama de la máquina de Mealy utilizada para detectar la secuencia de bits.

Se le otorgó a los estados la siguiente numeración: $A = 00$, $B = 01$, $C = 10$, $D = 11$. Es así que se obtiene la siguiente tabla:

Estado actual	Próximo estado		Salida	
	w=0	w=1	w=0	w=1
y_2y_1	Y_2Y_1	Y_2Y_1	Z	Z
A	A(00)	B(01)	0	0
B	A(00)	C(10)	0	0
C	D(11)	A(00)	0	0
D	A(00)	B(01)	0	1

Tabla 1: Tabla de estados de la máquina de Mealy.

A partir de lo expuesto en la Tabla (1), se realizaron los respectivos mapas de Karnaugh para las salidas Y_1 e Y_2 . Es así que se pudo observar que, tanto para los maxterminos como para los minterminos, el número de operaciones lógicas a realizar son las mismas, por lo tanto se optó por utilizar los minterminos, obteniendo así los siguientes mapas:

		y_2y_1			
		00	01	11	10
w	0	0	0	0	1
	1	0	1	0	0

(a) Mapa del primer Flip-Flop.

		y_2y_1			
		00	01	11	10
w	0	0	0	0	1
	1	1	0	1	0

(b) Mapa del segundo Flip-Flop.

		y_2y_1			
		00	01	11	10
w	0	0	0	0	0
	1	0	0	1	0

(c) Mapa de la salida deseada.

Figura 2: Mapas de Karnaugh implementados.

Con los mapas presentados en la Figura (2) se obtienen las siguientes funciones:

$$Y_1 = (w\bar{y}_2\bar{y}_1) + (wy_1y_2) + (\bar{w}y_2\bar{y}_1)$$

$$Y_2 = (w\bar{y}_2y_1) + (\bar{w}y_2\bar{y}_1)$$

$$Z = wy_2y_1$$

Se emplean 2 flip-flops tipo D para representar las 4 combinaciones posibles, obteniendo así todos los estados posibles, considerando a su vez que los futuros pueden representar tanto la entrada y la salida de los flip-flops como el estado actual. Es así que se presenta el siguiente circuito:

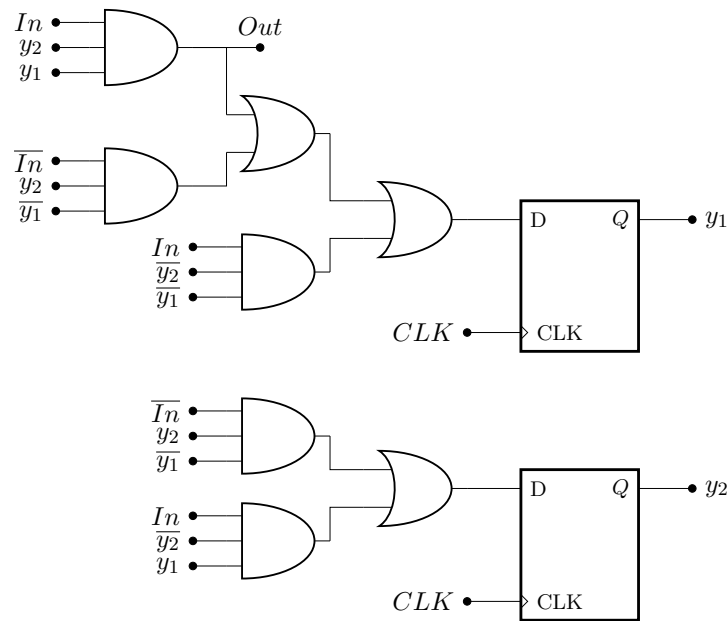
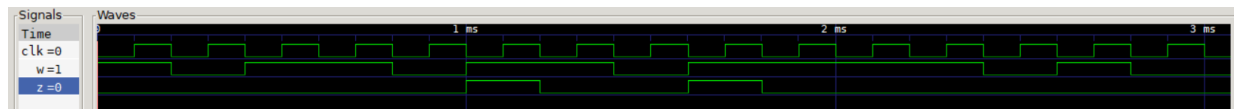


Figura 3: Circuito implementado.

Consecuentemente, se procedió a realizar la simulación en Verilog de dicho circuito. De esta forma se obtuvieron los siguientes resultados:

Figura 4: Simulaciones en GTKwave, donde w simboliza la entrada, z la salida y clk el clock del circuito

Como era de esperarse, dicho circuito emula satisfactoriamente la máquina de estados planteada, por lo que se puede detectar correctamente la secuencia de bits deseados cuando esta se presenta. Posteriormente a las simulaciones, se procedió a diseñar dicho circuito en PCB, con el cual se obtuvieron los mismos resultados, por lo que se corroboró el correcto funcionamiento de la máquina de estados implementada, como se puede presenciar en las siguientes figuras.

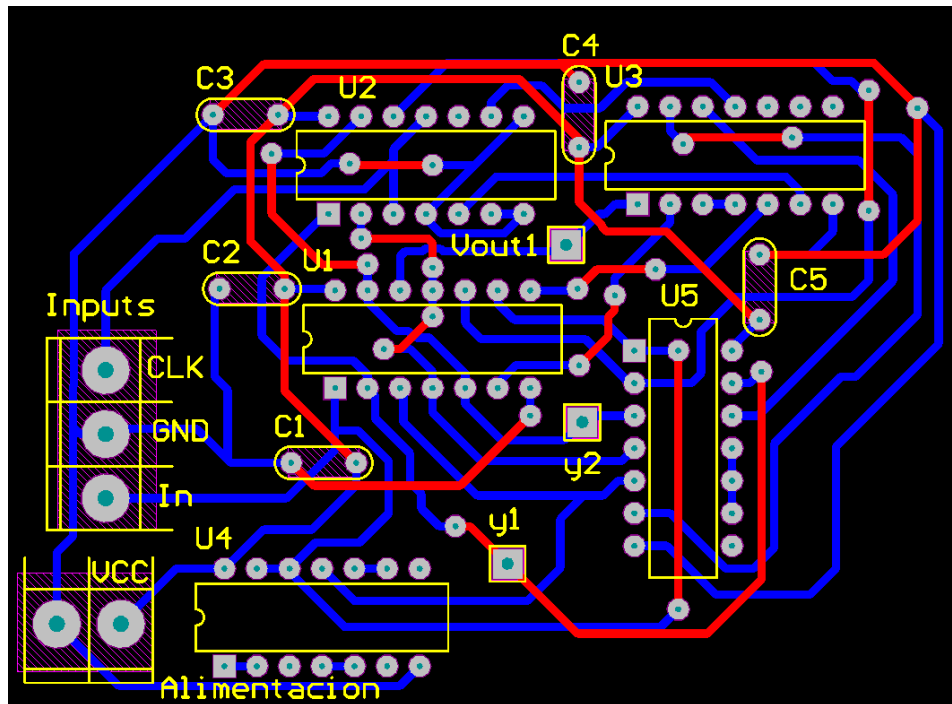


Figura 5: Implementación en PCB del circuito implementado.

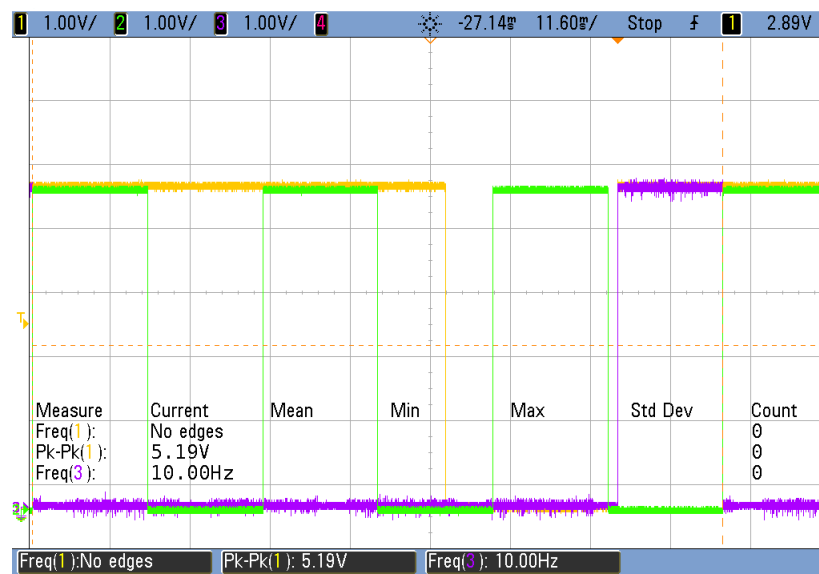


Figura 6: Detección de la secuencia deseada vista en el osciloscopio.