

Instituto Tecnológico de Buenos Aires

22.13 ELECTRÓNICA III

Trabajo práctico N°3

Grupo 3

MECHOULAM, Alan	58438
LAMBERTUCCI, Guido Enrique	58009
MARTORELL, Ariel	56209
LONDERO BONAPARTE, Tomás Guillermo	58150

Profesores

DEWALD, Kevin
WUNDES, Pablo Enrique
AGUIRRE, Miguel Pablo

Presentado: 14/11/19

Índice

1. Ejercicio 1	2
1.1. Introducción	2
1.2. Análisis del sistema	2
1.3. Implementación y mediciones	4
2. Ejercicio 2	8
3. Ejercicio 2	8
3.1. Introducción	8
3.2. Implementación	8
4. Ejercicio 3	12
4.1. Introducción	12
4.2. Simulación	14
4.3. Implementación	14
4.3.1. Level Shifting	14
4.3.2. Diseño Final	14
4.3.3. Componentes	15
4.4. Mediciones	15
4.5. Conclusiones	16

1. Ejercicio 1

1.1. Introducción

En este ejercicio se implementa un sistema de control para un tanque de agua, el cual cuenta con dos sensores, siendo estos I y S, los cuales indican si el tanque está lleno, por la mitad o vacío. Las condiciones de diseño son las siguientes:

- Cuando está vacío ($I = 0, S = 0$) se prenden las dos bombas B_0 y B_1 .
- Cuando se encuentra lleno ($I = 1, S = 1$) se apagan las bombas.
- Cuando está por la mitad ($I = 1, S = 0$) se activa una sola bomba, pero estas se alternan entre sí al establecer cual trabaja.

1.2. Análisis del sistema

Las limitaciones previamente mencionadas se corresponden con la siguiente tabla de verdad:

I	S	B_1	B_2
0	0	1	1
0	1	x	x
1	0	Alternado	
1	1	0	0

Tabla 1: Tabla de verdad del sistema.

A partir de lo expuesto previamente, se diseña la siguiente FSM.

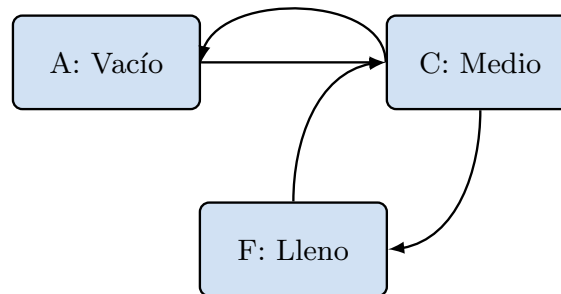


Figura 1: Finite State Machine.

Con lo presentado en la Figura (1), se confecciona una tabla de transiciones.

Estado actual	Próximo estado				Salida	
	I-S	I-S	I-S	I-S	Ambos	Toggle
	0-0	0-1	1-0	1-1		
A	X	X	B	X	1	0
B	A	X	X	C	0	1

Tabla 2: Tabla de transiciones del sistema.

A partir de la Tabla (2) y la Figura (1) se puede llegar a la siguiente tabla, donde y_1 e y_2 representan la salida de los flip-flops, mientras que Y_1 e Y_2 la entrada de los mismos.

Estado actual	Codificación	Próximo estado				Salida	
	$y_2 - y_1$	$Y_2 - Y_1$	$Y_2 - Y_1$	$Y_2 - Y_1$	$Y_2 - Y_1$	Ambos	Toggle
		I-S	I-S	I-S	I-S		
		0-0	0-1	1-0	1-1		
A	00	X	X	01	X	1	0
B	01	00	X	X	11	0	1
C	10	X	X	01	X	0	0
D	11	X	X	X	X	X	X

Se destaca que la variable “Ambos” hace referencia a estado en el cual se deben prender ambas bombas, mientras que la variable “Toggle” a cuando debe prenderse una sola e intercambiar.

Luego, se prosigue a resolver los mapas de Karnaugh para cada variable:

y2y1	IS			
	00	01	11	10
00	X	X	X	1
01	0	X	1	X
11	X	X	X	X
10	X	X	X	1

(a) Tabla de Karnaugh para Y_1 .

y2y1	IS			
	00	01	11	10
00	X	X	X	0
01	0	X	1	X
11	X	X	X	X
10	X	X	X	0

(b) Tabla de Karnaugh para Y_2 .

y2	y1	
	0	1
0	1	0
1	0	X

(c) Tabla de Karnaugh para “Ambos”.

y2	y1	
	0	1
0	0	1
1	0	X

(d) Tabla de Karnaugh para “Toggle”.

Figura 2: Tablas de Karnaugh para cada variable analizada.

A partir de la Figura (2) se derivan las siguientes expresiones:

$$\begin{aligned} Y_1 &= I \\ Y_2 &= S \end{aligned} \quad (1)$$

$$\begin{aligned} Ambos &= \overline{y_2 + y_1} \\ Toggle &= y_1 \end{aligned} \quad (2)$$

Luego, se procede a obtener los circuitos para la FSM.

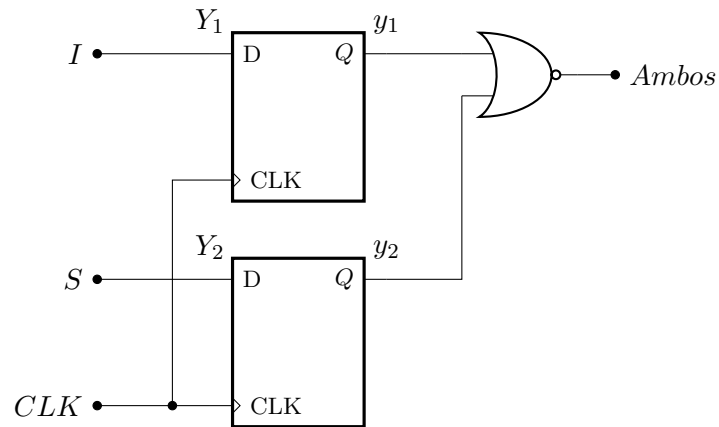


Figura 3: Circuito FSM.

Agregando el siguiente circuito lógico, se implementa la función de Toggle junto a la lógica de salida.

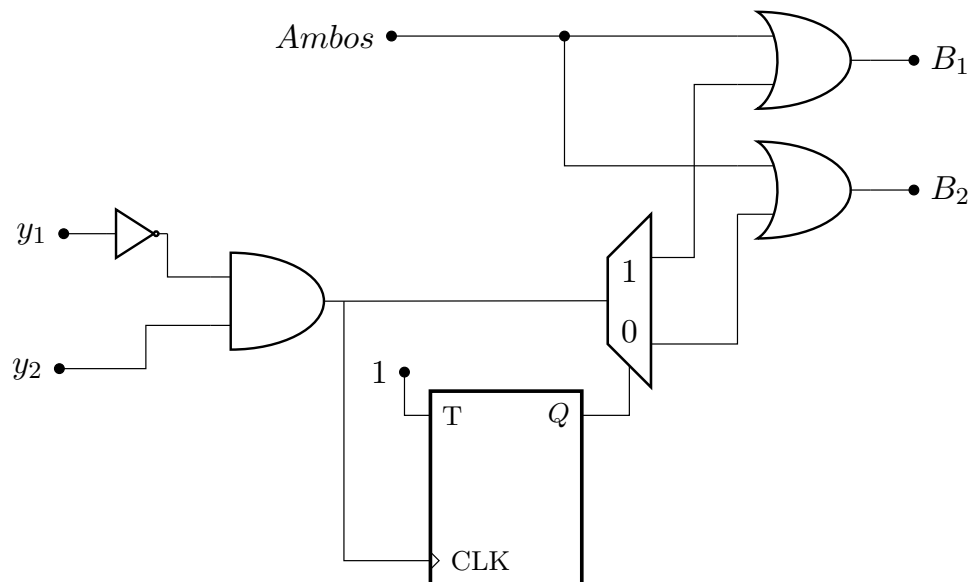


Figura 4: Circuito FSM con Toggle.

1.3. Implementación y mediciones

Una vez establecidos los circuitos y a partir de ellos, se procedió a implementarlos en PCB:

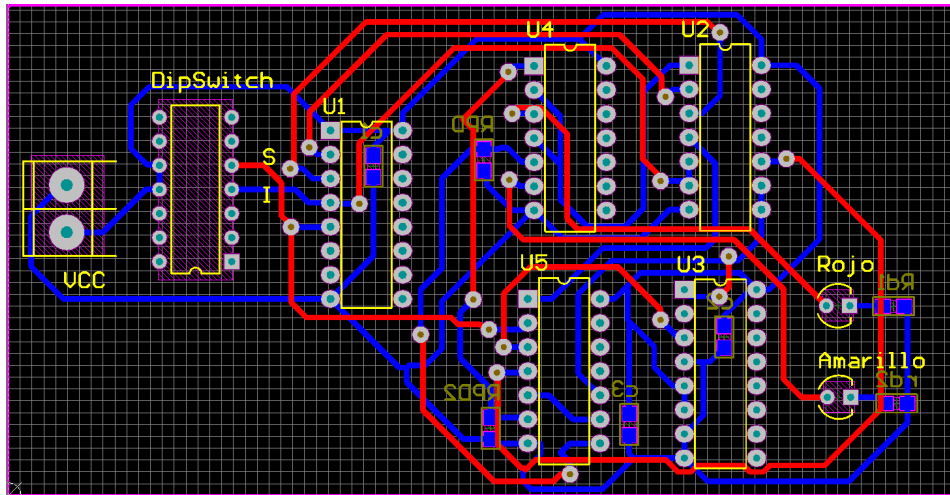


Figura 5: PCB en Altium de los circuitos.

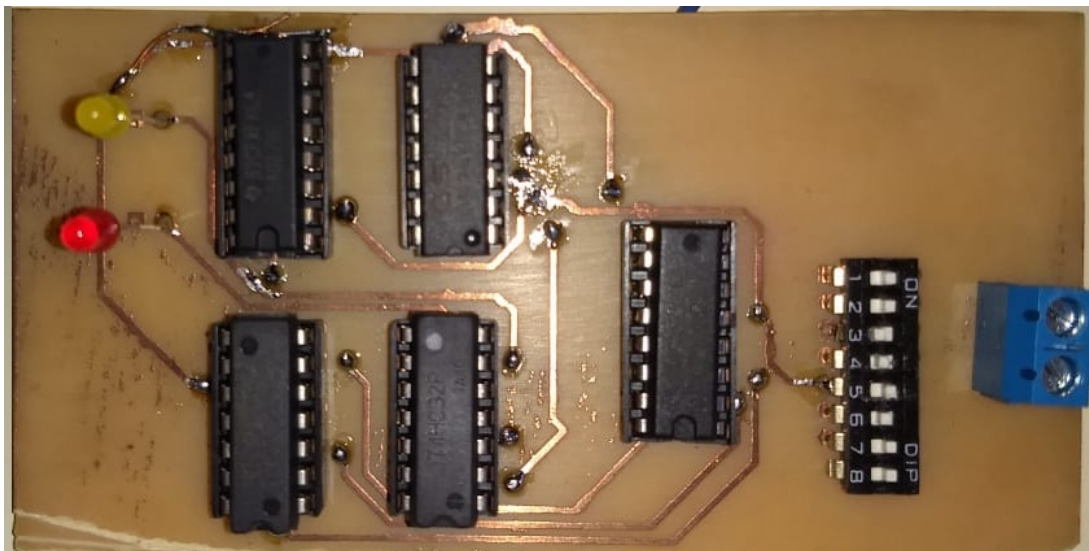


Figura 6: Placa implementada.

Finalmente, se procedió a medir los niveles de tensión para las transiciones posibles. A continuación se presentan los resultados.

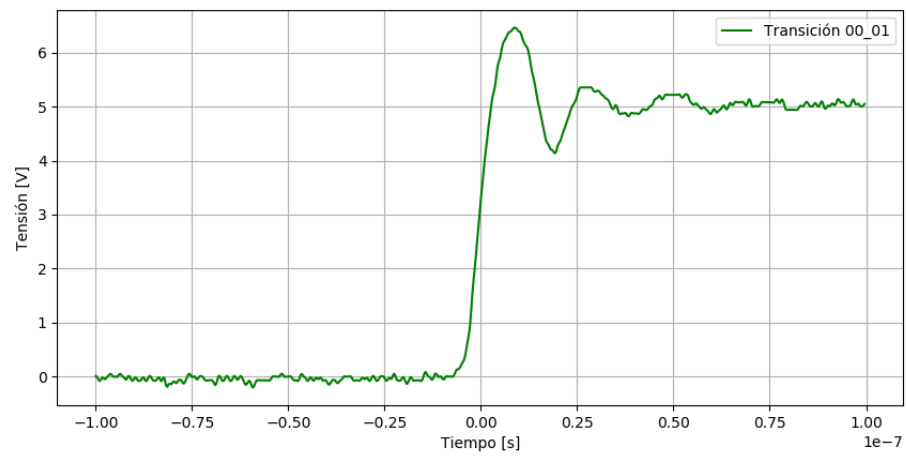


Figura 7: Transición 00-01.

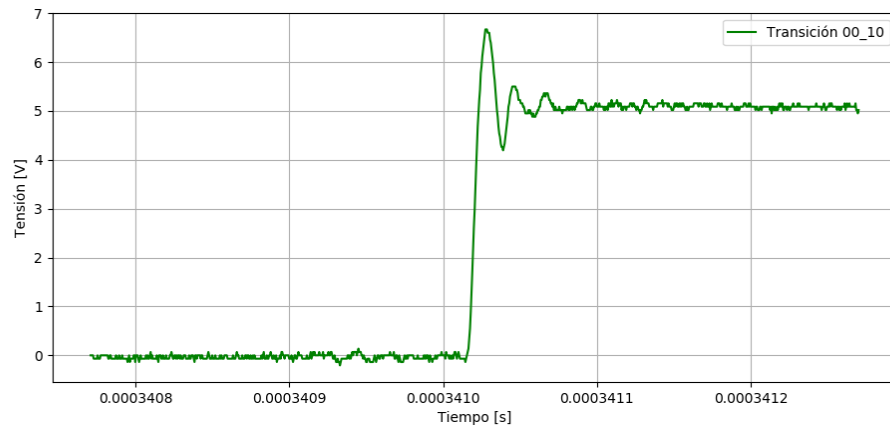


Figura 8: Transición 00-10.

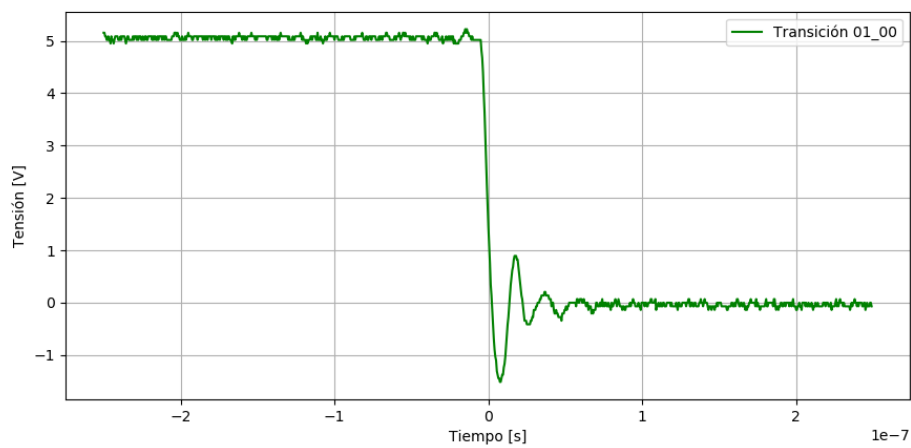


Figura 9: Transición 01-00.

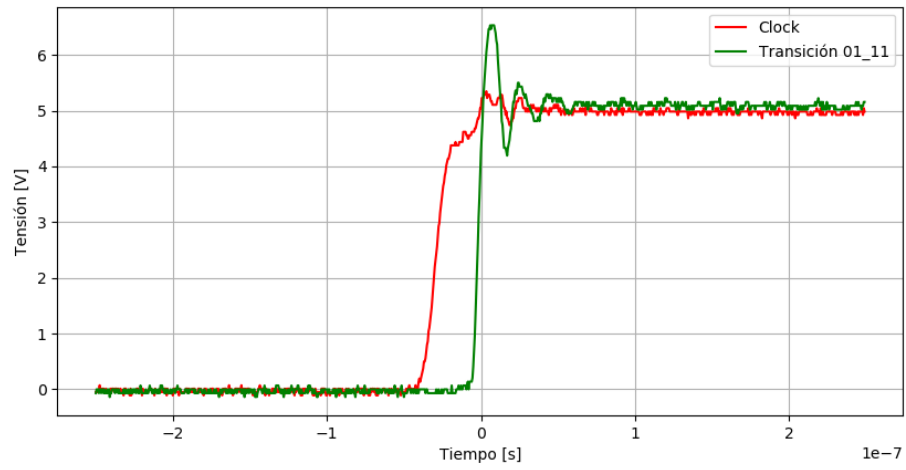


Figura 10: Transición 01-11.

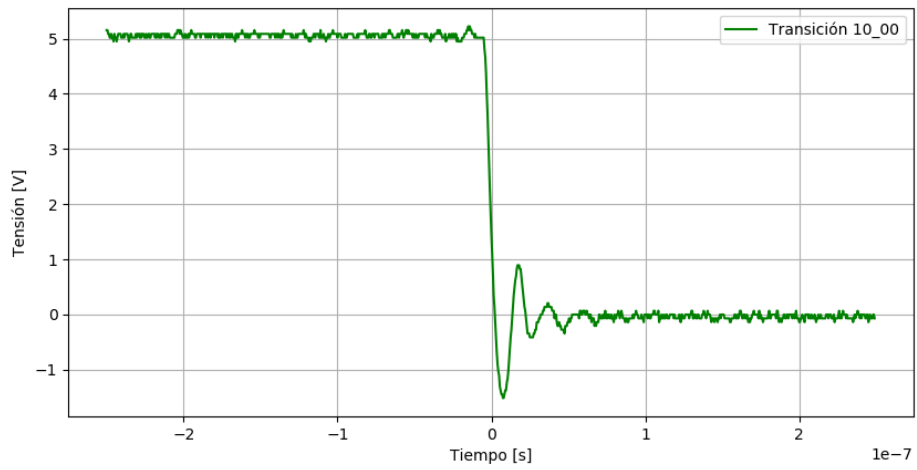


Figura 11: Transición 10-00.

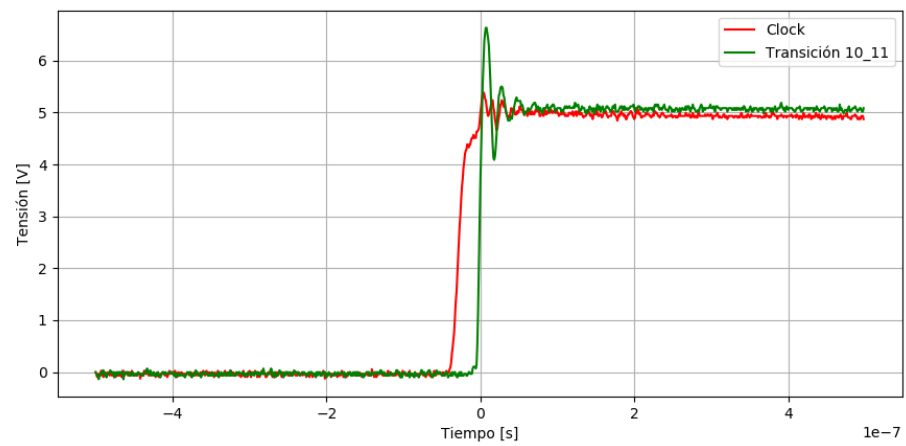


Figura 12: Transición 10-11.

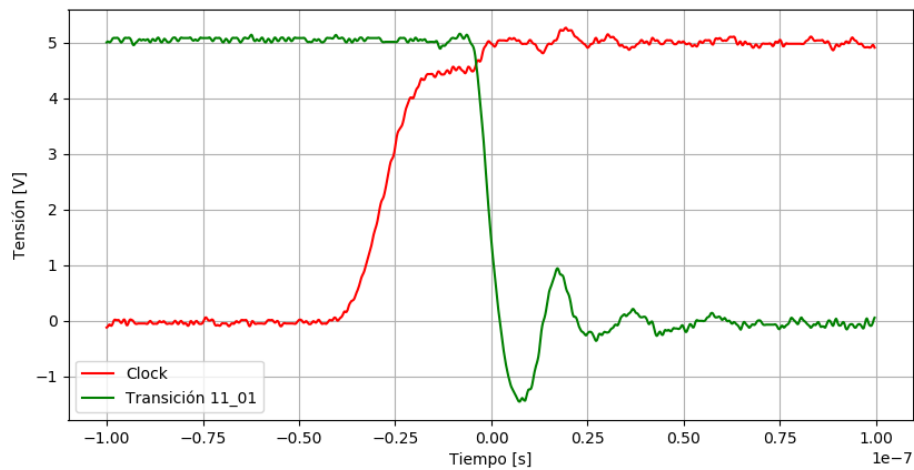


Figura 13: Transición 11-01.

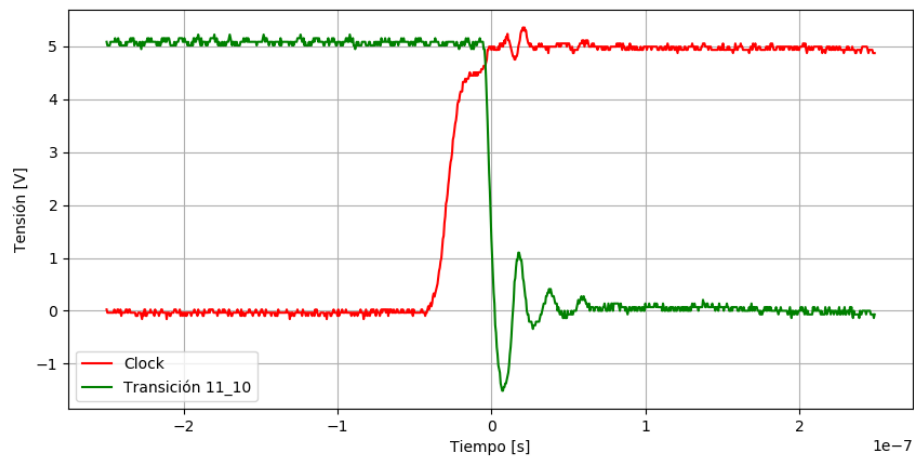


Figura 14: Transición 11-10.

2. Ejercicio 2

3. Ejercicio 2

3.1. 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.

3.2. 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 realizo el siguiente diagrama de estados, utilizando una máquina de Mealy:

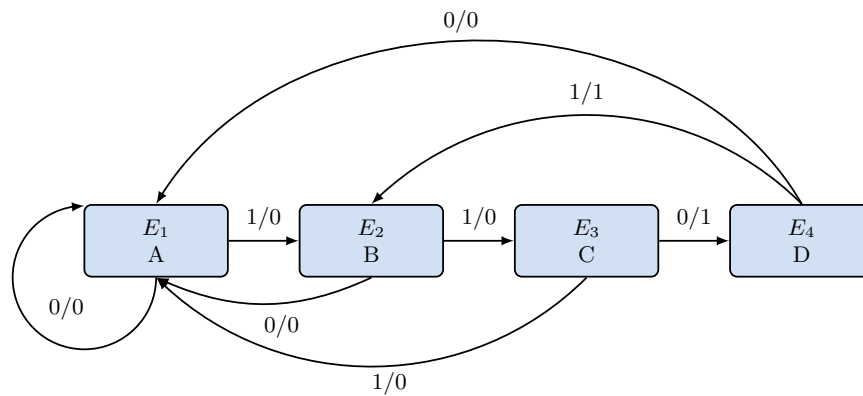


Figura 15: Diagrama de la máquina de Mealy utilizada para detectar la secuencia de bits.

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 3: Tabla de estados de la máquina de Mealy.

A partir de lo expuesto en la Tabla (3), 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 16: Mapas de Karnaugh implentados.

Con los mapas presentados en la Figura (16) 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:

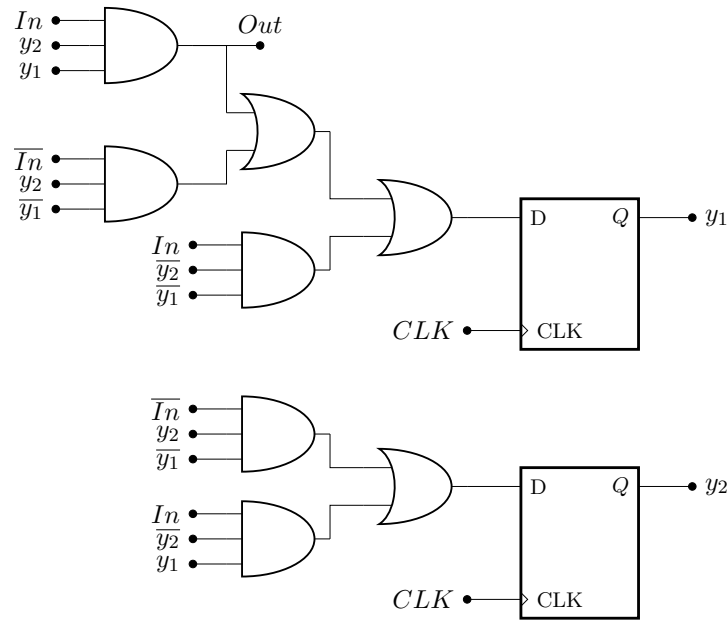
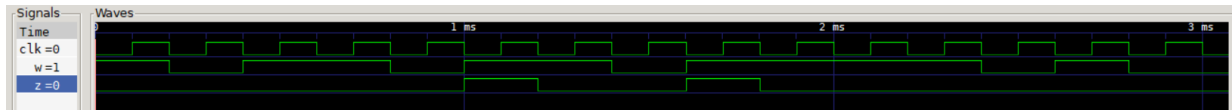


Figura 17: Circuito implementado.

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

Figura 18: 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.

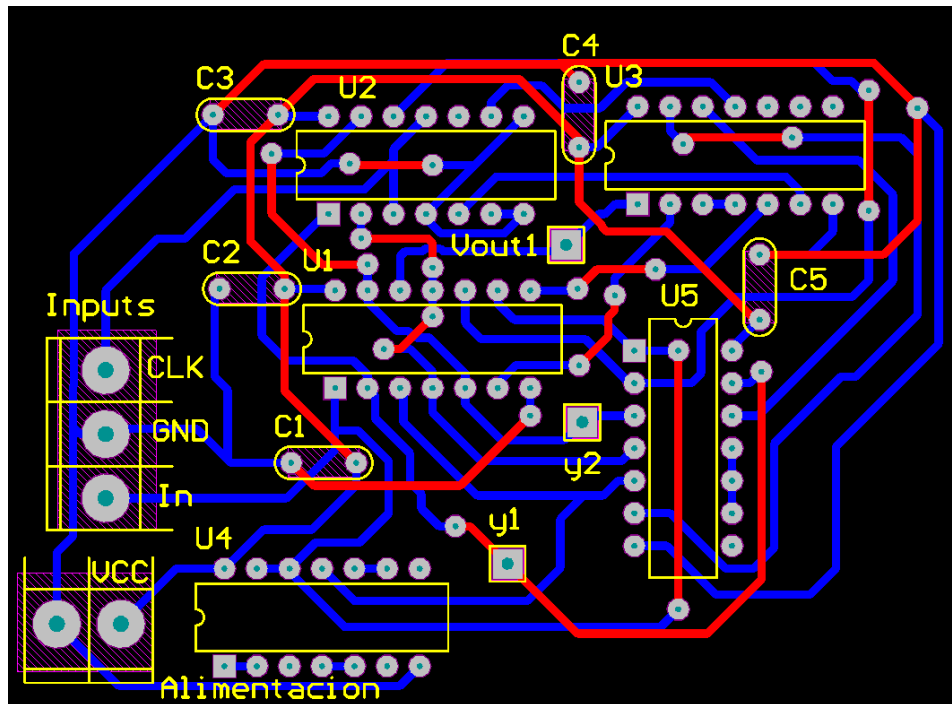


Figura 19: Implementación en PCB del circuito implementado.

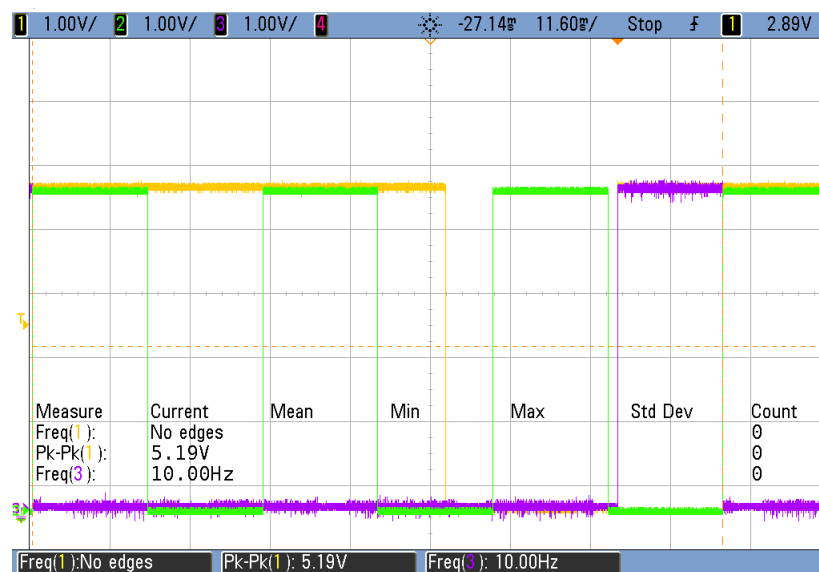


Figura 20: Detección de la secuencia deseada vista en el osciloscopio.

En la Figura (20) se presenta una medición de la detección de la secuencia deseada. En ella, se presentan en verde la señal de clock, en amarillo la señal de entrada y en violeta la salida. Además, es posible presenciar el funcionamiento de la maquina de estados. En ella, se presenta un desfase, en el que pareciera que la salida se logra en un estado bajo del clock. Dicha suposición es errónea, ya que esto se debe a que no se logró una sincronización entre la señal de entrada y el generador de clock. Dichas señales, a su vez, dado que la señal de salida posee una frecuencia distinta, se dificulta la captura de esta en el osciloscopio a un mismo tiempo, por lo tanto, se produce dicho desplazamiento entre las 3 señales.

4. Ejercicio 3

4.1. Introducción

En esta sección se buscó implementar la siguiente máquina de estados finitos:

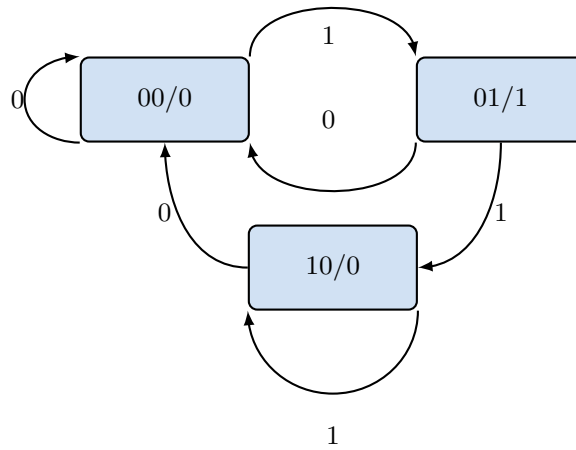


Figura 21: Máquina de estados finitos a implementar.

Para esto, se conformó la tabla de estados considerando al estado 00 como el inicial, resultando:

Estado actual y_2y_1	Próximo estado		Salida
	Y_2Y_1 $\omega = 0$	Y_2Y_1 $\omega = 1$	
00	00	01	0
01	00	10	1
10	00	10	0
11	X	X	X

Tabla 4: Tabla de estados para la máquina de estados finita a implementar.

Como fue necesario implementar tres estados, se requirió utilizar dos flip-flops. Luego, se hallaron las fórmulas lógicas para los estados siguientes utilizando mapas de Karnaugh.

a \ bc	00	01	11	10
0	0	0	X	0
1	0	1	X	1

$$Y_2 = wy_1 + wy_2$$

(a) Solución para Y_2 .

a \ bc	00	01	11	10
0	0	0	X	0
1	1	0	X	0

$$Y_1 = w(\overline{y_2} \cdot \overline{y_1})$$

(b) Solución para Y_1 .

Figura 22: Mapas de Karnaugh para los próximos estados de la maquina de estados finitos.

Utilizando el teorema de De Morgan y simplificando se obtienen dos posibles implementaciones análogas:

$$\begin{cases} Y_1 = w(\overline{y_2} \cdot \overline{y_1}) \\ Y_2 = w(\overline{y_2} \cdot \overline{y_1}) \end{cases}$$

(a) Implementación con NAND.

$$\begin{cases} Y_1 = w(\overline{y_2 + y_1}) \\ Y_2 = w(y_2 + y_1) \end{cases}$$

(b) Implementación con AND y NOR.

Si este circuito fuese trabajado directamente sobre el silicio, se elegiría la implementación con compuertas NAND,

ya que esta forma es la más simple de realizar. Sin embargo, como se realizó sobre un PCB, se decidió utilizar la implementación con AND y NOR, debido a que de esta manera se utilizan solamente dos integrados para el circuito lógico de entrada y salida, a diferencia de la implementación con NAND, el cual requiere de tres integrados, utilizando un total de nueve NAND's. Finalmente, a partir de las ecuaciones obtenidas se esquematizó la implementación teórica.

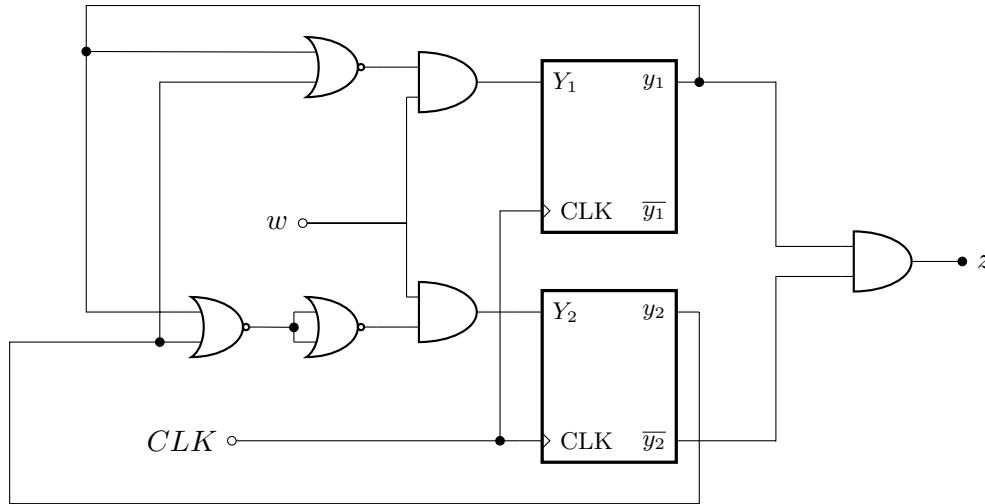


Figura 24: Implementación teórica de la lógica de entrada, estados y lógica de salida.

4.2. Simulación

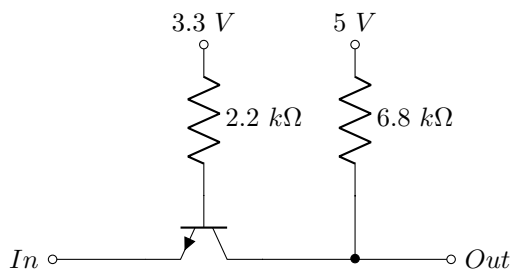
Se simuló la implementación obtenida en la sección anterior utilizando *Verilog*. Además, se construyó un test bench con todas las combinaciones posibles de entradas. Esta simulación obtuvo resultados exitosos. Se encuentra anexada esta simulación junto al test bench y un ejecutable junto a este informe.

4.3. Implementación

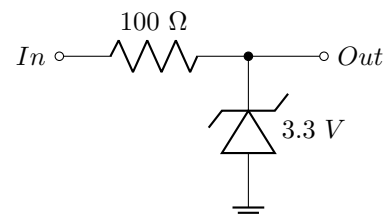
Para esta etapa se tuvo un cuidado especial dado que era un requisito en la implementación que la lógica interna del circuito funcione con 3.3 V, mientras que las entradas y salidas debían operar con 5 V.

4.3.1. Level Shifting

Para la conversión de 3.3 V a 5 V de la salida se decidió utilizar un transistor bipolar NPN como indica la Figura (25a). Luego, para las entradas, las cuales deben pasar de 5 V a 3.3 V, se utilizó un diodo zener de 3.3 V con una resistencia limitadora de corriente, la cual fue calculada conociendo la corriente de codo del diodo y la corriente de entrada de las compuertas de tecnología CMOS empleadas. Esta implementación se puede observar en la Figura (25b).



(a) Transistor NPN en base común utilizado como step-up level-shifter.



(b) Regulador de tensión de 3.3 V con zener y resistencia utilizados como step-down level-shifter.

4.3.2. Diseño Final

Finalmente se presenta a continuación el diseño final de la máquina de estados finitos implementada en un PCB de 50mm x 50mm.

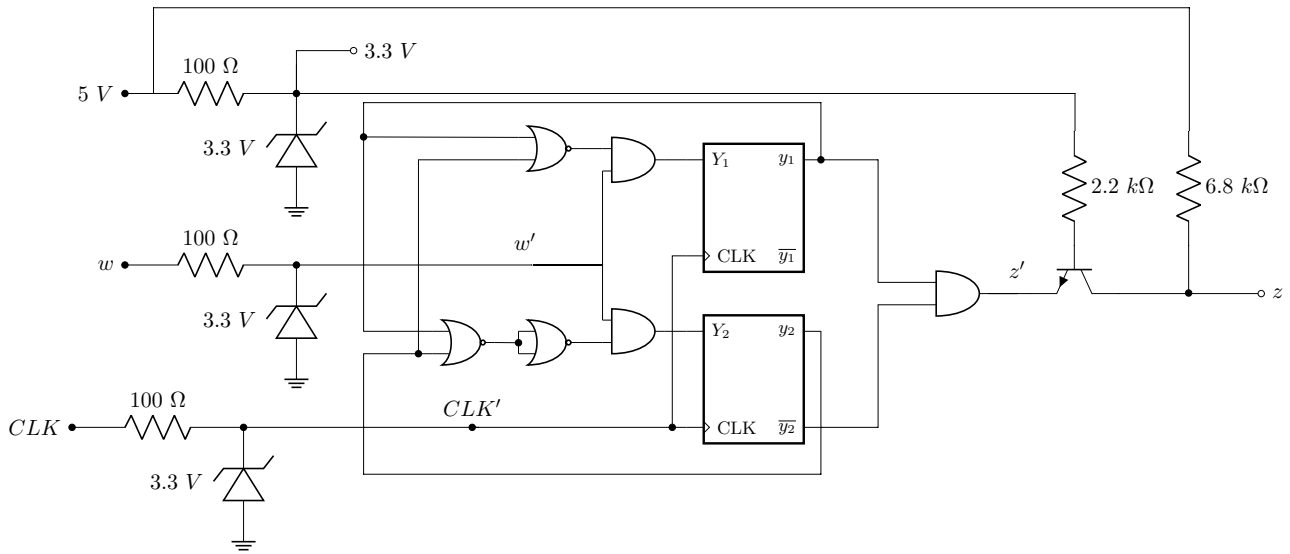


Figura 26: Implementación de la máquina de estados finitos junto a la conversión de niveles de tensión.

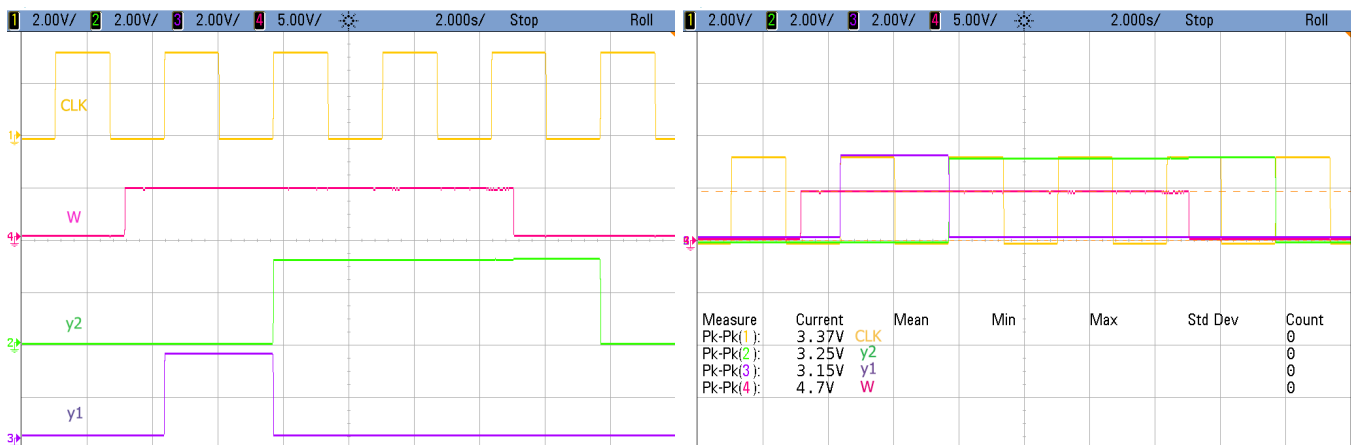
4.3.3. Componentes

A continuación se detallan los componentes utilizados en la implementación:

- Dual Flip-flop D: [CD4013](#)
- Quad 2-input AND: [74HC08](#)
- Quad 2-input NOR: [74HC02](#)
- BJT NPN: [BC548](#)

4.4. Mediciones

Se realizaron mediciones de tanto correcta transición entre estados como de niveles de tensión en el circuito.



(a) Medición de transiciones de estados: ..00-01-10-...-10-00-..

(b) Medición de los niveles de tensión.

Figura 27: Mediciones del circuito implementado.

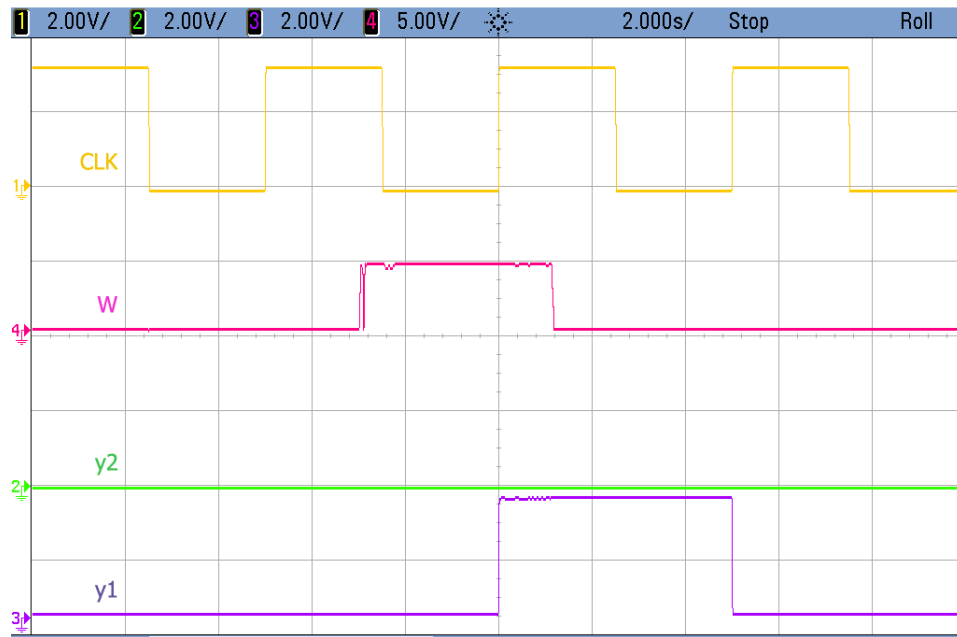
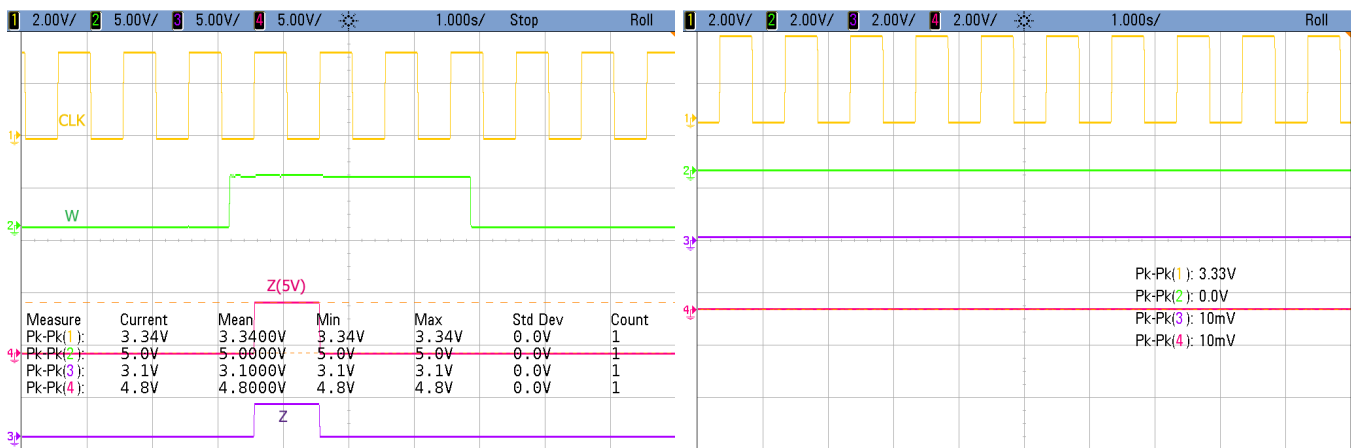


Figura 28: Medición de la transición de estados: ..00-01-00-..



(a) Medición de la transición de la salida del circuito.

(b) Medición de los niveles de tensión en estado bajo..

Figura 29: Mediciones del circuito implementado.

Se puede observar que el level shifting de niveles de tensión funcionan correctamente. Los niveles de tensión son $V_{OL} = 10 \text{ mV}$, $V_{OH} = 4.8 \text{ V}$, $V_{3.3H} \geq 3.1 \text{ V}$ y $V_{3.3L} \approx 0 \text{ V}$. Se destaca que $V_{3.3}$ representa el nivel de tensión de la lógica interna del circuito, mientras que V_{OL} y V_{OH} son los niveles de tensión altos y bajos conseguidos a la salida Z .

Luego, se observa en las Figuras (27a) y (28) que la transición de estados funciona correctamente habiendo probado todas las combinaciones posibles. Por otro lado, en la Figura (29a) se observa que la salida posee los valores adecuados, siendo cero para todos los estados excepto en el 01, para el cual la salida es igual a un uno lógico. Finalmente, se destaca que el consumo de corriente del circuito es de 15 mA en todos los estados excepto en el estado 01 posee un consumo de 32 mA .

4.5. Conclusiones

Se implementó la FSM propuesta habiendo cumplido los requisitos de niveles de tensión de lógica interna. Si bien la transición de 5 V a 3.3 V funcionó correctamente, el consumo de corriente es muy elevado, observándose un máximo de 32 mA . Si se hubiera deseado un consumo menor, se debería utilizar un step-down level-shifters, valiéndose transistores como se realizó para el step-up level-shifter. Este logró mantener la salida en una tensión muy baja para el estado lógico bajo y en una tensión aceptable para el estado de tensión lógico alto, con un error de 200 mV por debajo del deseado. Sin embargo, estos niveles de tensión se encuentran totalmente dentro de estándares de márgenes de ruido

tanto para la tecnología TTL como para CMOS. Luego, la tensión alta de la lógica interna de 3.1 V se encuentra también dentro de los márgenes de ruido para ambas tecnologías.

Otras posibles implementaciones para el step-down parten del uso de un divisor resistivo, lo cual disminuye el consumo de corriente, ya que dentro de los valores de resistencias que se pueden utilizar, se hayan aquellas de 10 k Ω o más, ya que estos valores no son lo suficientemente altos como para generar un divisor resistivo con la impedancia de entrada de la compuerta CMOS ni para provocar una corriente comparable con el consumo de estas compuertas. Otra implementación se centra en el uso de un comparador. Este último es más barato que un transistor, pero se descartó por su gran tamaño y por poseer implementaciones más simples con mismo resultado.