

0.1. Introducción

Se quiso implementar la siguiente máquina de estados finitos:

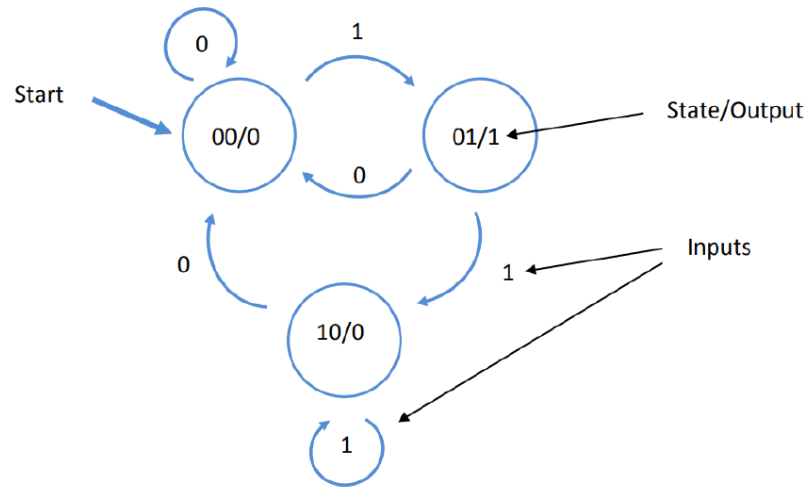


Figura 1: Máquina de estados finitos a implementar.

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

<i>PresentState</i> y_2y_1	<i>NextState</i>		<i>Output(z)</i>
	$w = 0$ Y_2Y_1	$w = 1$ Y_2Y_1	
00	00	01	0
01	00	10	1
10	00	10	0
11	xx	xx	x

Tabla 1: 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.

w	y_2y_1			
	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 .

w	y_2y_1			
	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 2: Mapas de Karnaugh para los próximos estados de la maquina de estados finitos.

Utilizando el teorema de DeMorgan 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 NAND ya que sería la más simple de realizar. Sin embargo, como se contruirá un PCB, se decidió utilizar la implementación con AND y NOR ya que se deberían de utilizar solamente dos integrados para el circuito lógico de entrada y salida, a diferencia de la implementación con NAND, que requeriría de tres integrados (utilizando un total de nueve NAND's). Finalmente, a partir de las ecuaciones obtenidas se esquematizó la implementación teórica.

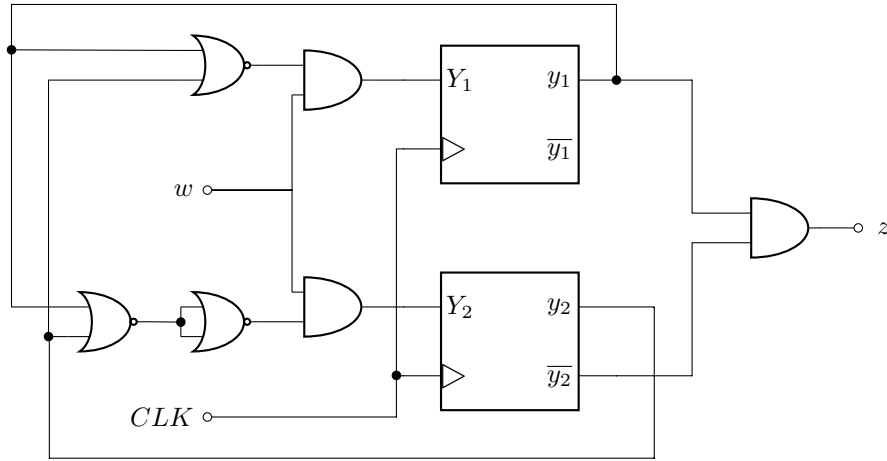


Figura 4: Implementación teórica de la lógica de entrada, estados y lógica de salida de la máquina de estados finitos a implementar.

0.2. Simulación

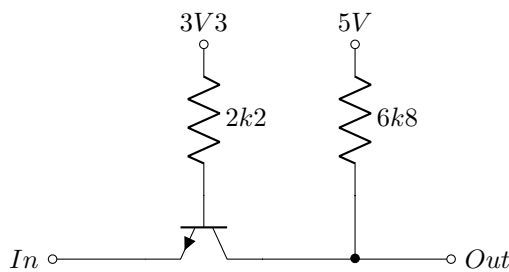
Se simuló la implementación obtenida en la sección anterior utilizando *Verilog*, un lenguaje descriptivo de hardware. Además, se construyó un test bench con todas las combinaciones posibles de entradas. Esta simulación obtuvo resultados exitosos. Se encuentra anexada esta simulación junto al test bench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

0.3. Implementación

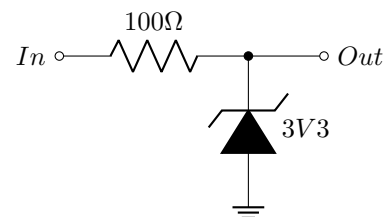
Para esta etapa se debió tener un cuidado especial dado que era un requisito en la implementación que la lógica interna del circuito funcione con 3.3V mientras que las entradas y salidas debían operar con 5V.

0.3.1. Level Shifting

Para la conversión de 3V3 a 5V de la salida se decidió utilizar un transistor bipolar NPN como indica la Figura (5a). Luego, para la entradas, las cuales debían pasar de 5V a 3V3 se utilizó un diodo zener de 3V3 con una resistencia limitadora de corriente calculada conociendo la corriente de codo del diodo y la corriente de entrada de las compuertas de tecnología CMOS utilizadas. Esta implementación se puede observar en la Figura(5b).



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



(b) Regulador de tensión de 3V3 con diodo zener y resistencia utilizado como step-down level-shifter.

0.3.2. Diseño Final

Finalmente se presenta a continuación el diseño final de la máquina de estados finitos implementada en un PCB de 50mmx50mm.

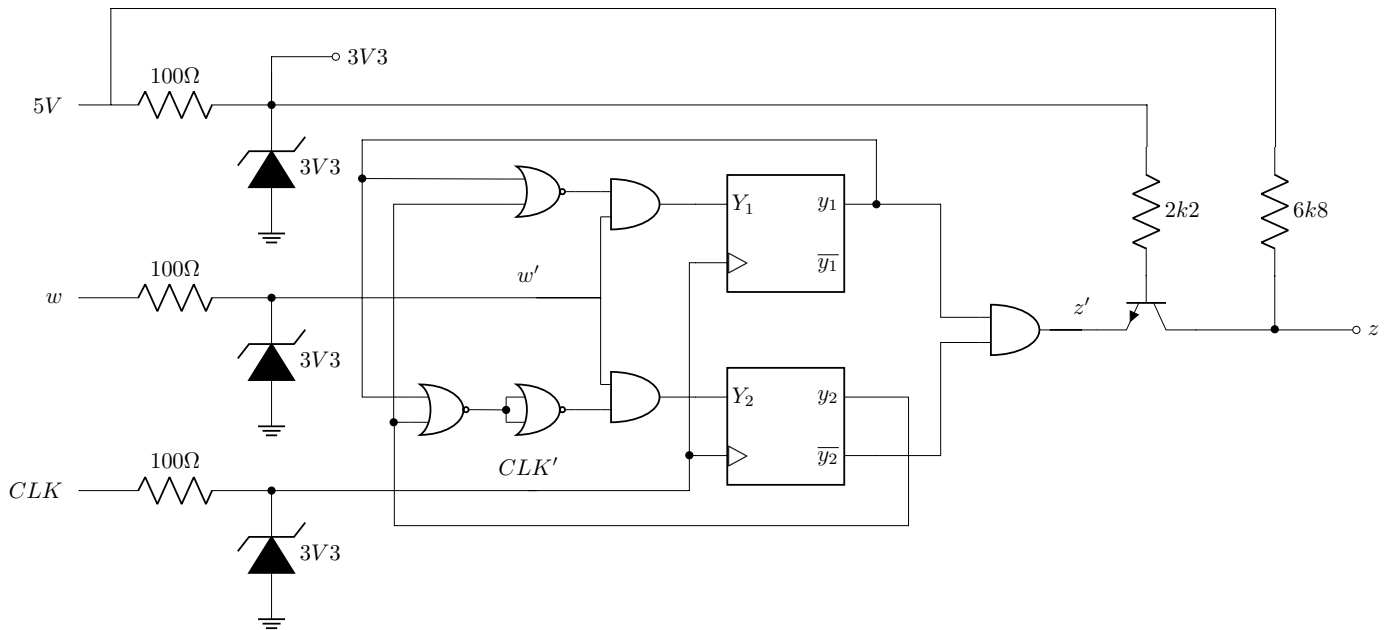


Figura 6: Implementación de la máquina de estados finitos junto a la conversión de niveles de tensión.

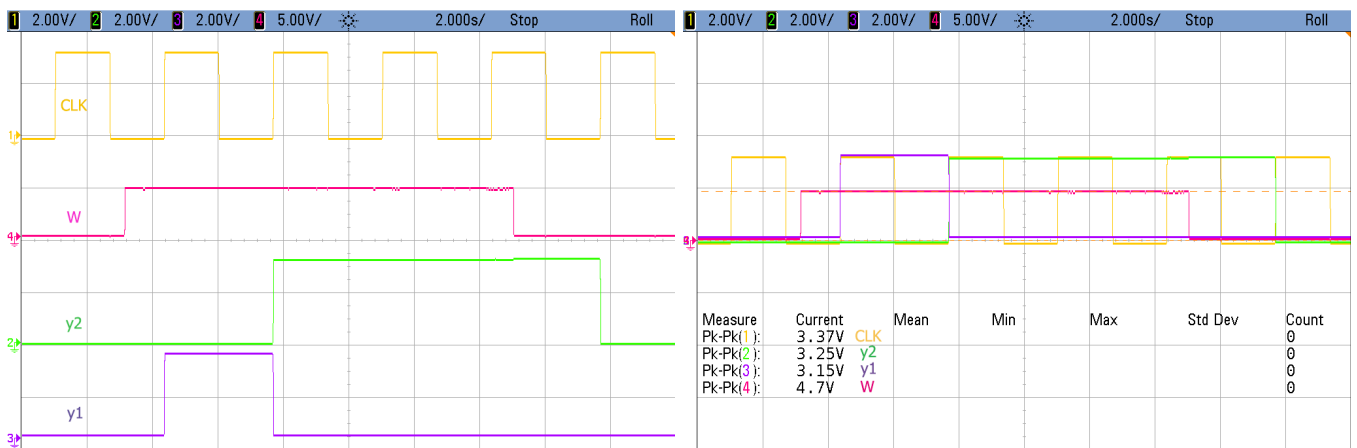
0.3.3. Componentes

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](#)

0.4. 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 7: Mediciones del circuito implementado.

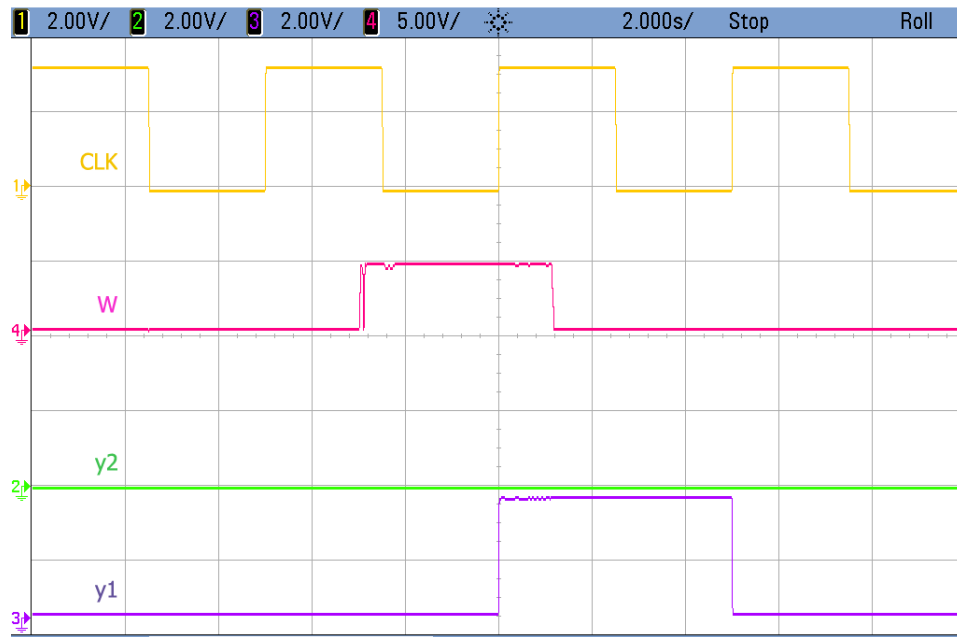
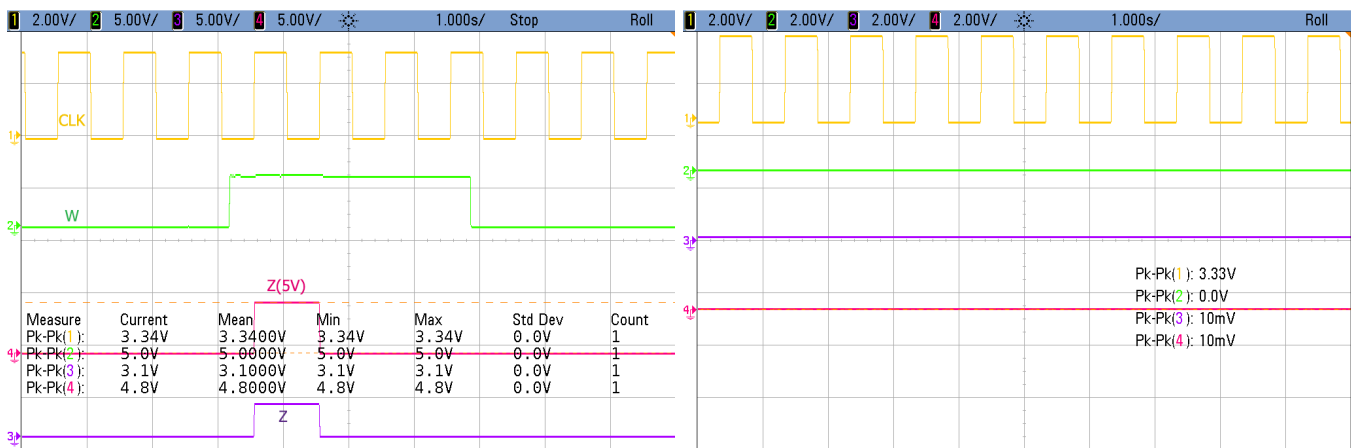


Figura 8: Medición de la transición de estados: ..00-01-00-..



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

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

Figura 9: Mediciones del circuito implementado.

Se puede observar que el level shifting de niveles de tensión funcionan correctamente. Los niveles de tensión son:

- $V_{OL} = 10mV$
- $V_{OH} = 4.8V$
- $V_{3V3H} = \geq 3.1V$
- $V_{3V3L} = \approx 0$

siendo V_{3V3} el nivel de tensión de la lógica interna del circuito y V_{OL} y V_{OH} los niveles de tensión altos y bajos conseguidos a la salida Z.

Luego, se observa en las Figuras (7a) y (8) que la transición de estados funciona correctamente habiendo probado todas las combinaciones posibles, y en la Figura (9a) se observa que la salida posee los valores correctos, siendo cero para todos los estados excepto en el estado 01 en el cual la salida es igual a un uno lógico. El consumo de corriente del circuito es de $15mA$ en todos los estados excepto en el estado 01 con un consumo de $32mA$.

0.5. Conclusiones

Se implementó la FSM propuesta habiendo cumplido los requisitos de niveles de tensión de lógica interna. Si bien la transición de $5V$ a $3V3$ funcionó correctamente, el consumo de corriente es muy elevado siendo este como máximo $32mA$. Si se hubiera querido un consumo menor, se debería de haber utilizado step-down level-shifters utilizando 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 $200mV$ 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 $3.1V$ se encuentra también dentro de los márgenes de ruido para ambas tecnologías.

Otras posibles implementaciones para el step-down hubieran sido el uso de un divisor resistivo, lo cual hubiera disminuido mucho el consumo de corriente ya que los valores de resistencias utilizadas posibles hubieran sido cómodamente de unos $10k\Omega$ o más, ya que estos valores no eran lo suficiente altos ni como para generar un divisor resistivo con la impedancia de entrada de la compuerta CMOS ni para generar una corriente comparable con el consumo de estas compuertas. Otra implementación hubiera sido el uso de un comparador ya que este era más barato que un transistor, pero se descartó por su gran tamaño y por poseer implementaciones más simples con mismo resultado.

Finalmente, se concluye que un posible uso de esta máquina de estados implementada es la de obtener un solo pulso a la salida dado el accionar de un botón, teniendo que dejar de oprimir este botón y volver a apretarlo para poder obtener siempre un solo pulso de longitud de un pulso de clock a la salida. Varias modificaciones pueden ser realizadas a esta máquina de estados para obtener otros posibles usos. Por ejemplo, si se deseara implementar la técnica de debouncing para evitar rebotes mecánicos al momento de accionar una entrada se podría redirigir la salida de $w = 0$ en el estado 01 para que esta continúe hacia el próximo estado también y que este último estado mantenga la salida como $z = 1$. Así efectivamente por un pulso de clock se ignoran posibles rebotes mecánicos que puedan llegar a poner a la entrada en un estado bajo indeseado. Si un pulso de clock fuese muy poco tiempo debido a su frecuencia se podría repetir el estado 01 hasta obtener el margen de tiempo deseado.