

0.1. Introducción

Se quiso implementar la siguiente máquina de estados finitos:

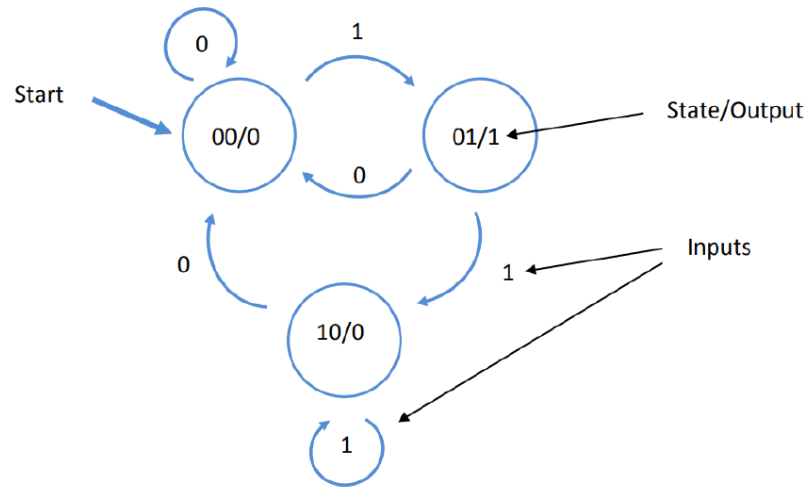


Figura 1: Máquina de estados finitos a implementar.

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.

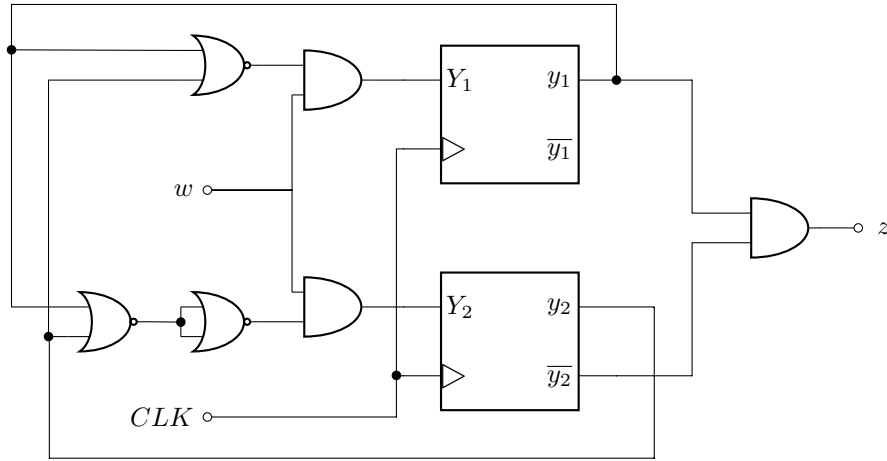


Figura 4: Implementación teórica de la lógica de entrada, estados y lógica de salida de la máquina de estados finitos a implementar.

0.2. Simulación

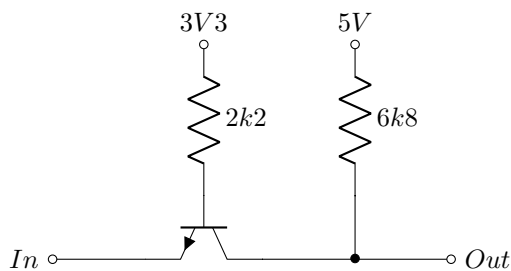
Se simuló la implementación obtenida en la sección anterior utilizando *Verilog*, un lenguaje descriptivo de hardware y *GTKWave*, un visualizador de señales.

0.3. Implementación

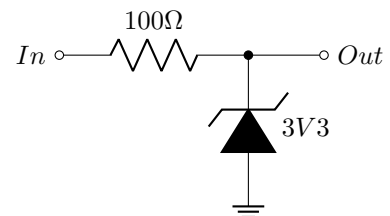
Para esta etapa se debió tener un cuidado especial dado que era un requisito en la implementación que la lógica interna del circuito funcione con 3.3V mientras que las entradas y salidas debían operar con 5V.

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.

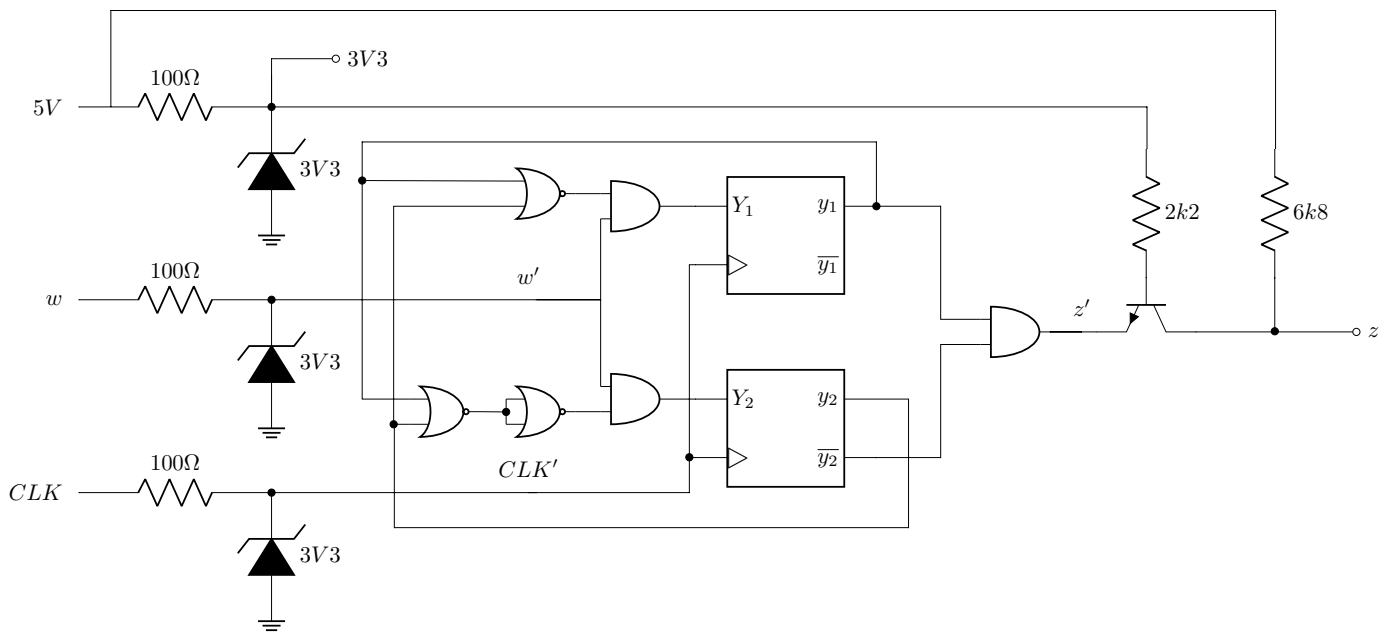


Figura 6: Implementación de la máquina de estados finitos junto a la conversión de niveles de tensión.

0.3.3. Componentes

A continuación se detallan los componentes utilizados en la implementación:

- Dual Flip-flop D: [CD4013](#)
- Quad 2-input AND: [74HC02](#)
- Quad 2-input NOR: [74HC02](#)
- BJT NPN: FALTA
- Diodo Zener 3V3: FALTA

0.4. Mediciones

0.5. Conclusiones