

Ejercicio 1

Se nos pidió diseñar e implementar una maquina de estados capaz de controlar la activación de dos bombas B1 y B2 (simulando su encendido con un LED) que deben mantener el nivel de agua de un depósito que dispone de dos sensores S e I, colados en la parte superior e inferior del depósito, respectivamente. La operatoria de la máquina sigue las siguientes reglas:

- Si el agua ha superado un sensor, su valor de salida será: 1.
- Si el deposito estuviera lleno ($I=S=1$) no se activaría ninguna bomba.
- Si el deposito estuviera vacío ($I=S=0$) se activarían ambas bombas.
- Si el deposito estuviera lleno por la mitad ($I=1, S=0$) se activaría la ultima bomba en no activarse.

Se nos solicitó tener en cuenta las siguientes consideraciones para la implementación de la maquina de estados:

- Implementar la solución utilizando tanto una maquina de Mealy como de Moore.
- Muestre claramente el diagrama de estados y transiciones.
- Respaldar el diseño con una simulación en Verilog.

Se debe mencionar que en ambas implementaciones(Moore y Mealy) se utilizó un Flip-Flop JK en modo toggle para implementar el alternado de las bombas. La lógica secuencial del Flip-Flop en cuestión es función solo de los estados de la máquina, y no del clock directamente (como si lo son los Flip-Flops que almacenan el estado actual de la maquina), por lo tanto se consideró apropiada tal implementación aún no cumpliendo con la definición estricta de máquina de estados de Moore y Mealy, que dice que la salida es una lógica combinacional de los estados (y las entradas en el caso de Mealy). Puede considerarse este Flip-Flop JK en modo toggle que controla el alternado de las salidas como una maquina de estados englobada por otra.

Implementación Maquina de Moore

Diagrama de Estados y Transiciones

Se comenzó por la implementación de la máquina de estados de Moore, para ello, se comenzó por realizar un diagrama de estados y transiciones que describa la maquina de estados. Para lo cual se listará primero a las entradas, estados y salidas posibles, y se hará una breve descripción para una mejor comprensión del diagrama.

Entradas:

Vacío: Configuración de entrada $S=0, I=0$, que indica que el depósito se encuentra vacío.

Medio: Configuración de entrada $S=0, I=1$, que indica que el depósito esta lleno por la mitad.

Lleno: Configuración de entrada $S=1, I=1$, que indica que el depósito se encuentra lleno.

Estados:

Ninguna: Estado que indica que ninguna bomba esta encendida.

Una Sola: Estado que indica que solo una bomba se encuentra encendida.

Ambas: Estado que indica que ambas bombas se encuentran encendidas.

Salidas:

$b_1 = 0$ y $b_2 = 0$: Esta configuración de salida no enciende ninguna bomba.

$b_1 = 1, b_2 = 0$ o $b_1 = 0, b_2 = 1$: Solo se enciende una de las bombas que controla el circuito.

$b_1 = 1$ y $b_2 = 1$: Esta configuración de salida enciende las dos bombas.

A continuación se presenta el diagrama de estados y transiciones, a partir del cual se diseñó la maquina de estados:

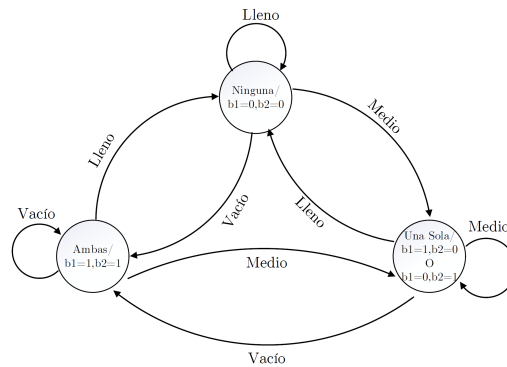


Figura 1: Diagrama de estados y transiciones

Al contar con 3 estados diferentes, mi maquina de estados necesitará como mínimo dos Flip-Flop's para almacenar el estado actual. Los tres estados posibles se codifican a traves de dos variables y_1 e y_0 , de esta forma, tendremos las siguientes configuraciones:

Estado	y_1y_0
Ninguna	00
Una Sola	01
Ambas	11

Habiendo definido los estados del diseño, se procedió a completar la correspondiente tabla de asignación de estados:

Estado Actual (y_1y_0)	Próximo Estado(Y_1Y_0)			Salida (b_1b_2)
	$SI = 00$	$SI = 01$	$SI = 11$	
00	11	01	00	00
01	11	01	00	01 o 10 (Alternado)
11	11	01	00	11

Cuadro 1: Tabla de asignación de estados

A partir del Cuadro 1 se confeccionó el Cuadro 2, que implementa la tabla de verdad que determina el próximo estado (Y_1Y_0) en función del estado anterior (y_1y_0) y las entradas (SI).

y_1y_0SI	Y_1	Y_0
0000	1	1
0001	0	1
0010	x	x
0011	0	0
0100	1	1
0101	0	1
0110	x	x
0111	0	0
1000	x	x
1001	x	x
1010	x	x
1011	x	x
1100	1	1
1101	0	1
1110	x	x
1111	0	0

Cuadro 2: Tabla de verdad cambio de estado

Se determinó que es estado $SI = 10$ no es una combinación posible, ya que indicaría que el agua ha superado el sensor superior pero no el inferior, lo cual es incompatible con el modelo del deposito, e indicaría un error en los sensores. Como el manejo de errores en los sensores excede los requisitos de la consigna es que se determinó que no son combinaciones posibles y las salidas correspondientes a estas configuraciones se determinaron como 'don't care'

La simplificación mediante Mapas de Karnaugh arrojó los siguientes resultados:

y_1y_0 \ SI	00	01	11	10
00	1	0	0	X
01	1	0	0	X
11	1	0	0	X
10	X	X	X	X

$$Y_1 = \bar{I}$$

y_1y_0 \ SI	00	01	11	10
00	1	1	0	X
01	1	1	0	X
11	1	1	0	X
10	X	X	X	X

$$Y_0 = \bar{S}$$

Se observa que el 'próximo estado' no depende del estado actual, sino solamente de las entradas.

Implementación

La lógica que determina la transición de un estado a otro está dada por las expresiones de Y_1 e Y_0 , desarrolladas en el apartado anterior. Se determinó adecuada la utilización de Flip-Flop's D para almacenar las variables y_1 e y_0 que determinan el estado de la máquina. La implementación del circuito se muestra en la Figura 2

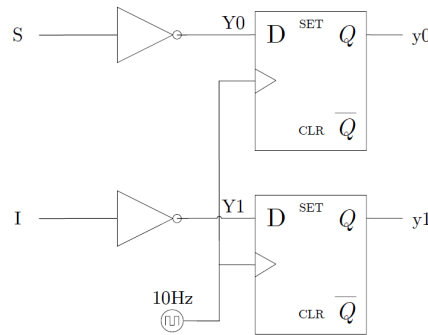


Figura 2: Lógica de transición de estados

Para implementar la lógica combinacional de las salidas en función de los estados actuales, se decidió utilizar un multiplexor, de forma que por un lado se implemente la lógica que controle las salidas cuando éstas son iguales (estados *Ninguna* y *Ambas*) y por otro la lógica cuando las salidas deben ser complementarias (estado *Una Sola*).

Para decodificar el estado de la máquina implementada, y poder controlar el mutiplexor mencionado y la lógica combinacional de las salidas, se utilizó un decodificador 2-a-4 74HC139. Este integrado implementa un decodificador de salidas active LOW, de esta forma, puedo decodificar el estado actual de la máquina para el posterior uso en la lógica de salida.

Cuando la máquina se encuentra en el estado *Una Sola* ($Una\ Sola=1$), las salidas no solo son complementarias (una bomba está encendida y la otra apagada), sino que además cada vez que 'entro' y 'salgo' del estado en cuestión, las salidas se deben intercalar, como indica la consigna. Para lograr esto, se utilizó un Flip-Flop JK en modo Toggle ($J=1, K=1$). El Clock del Flip-Flop en cuestión es alimentado por la señal (*Una Sola*), como indica la Figura 3.

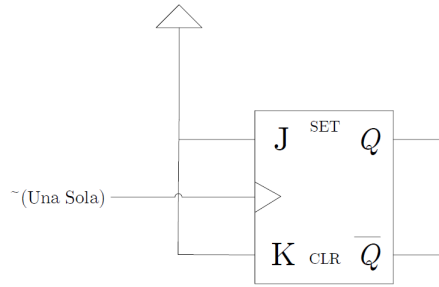


Figura 3: Flip-Flop JK para alternar salidas

Las salidas de este Flip-Flop alimentaron las entradas '0' de dos multiplexores como indica la Figura 4. De esta forma cuando la máquina se encuentre en el estado *Una Sola*, las salidas b_1 y b_2 serán complementarias

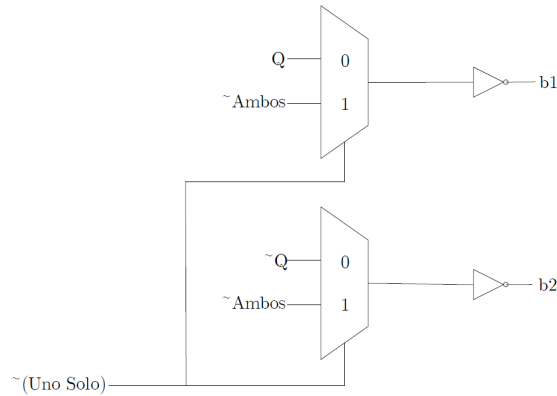


Figura 4: Multiplexación a la salida

Por otro lado, cuando la máquina de estados se encuentre en el estado *Ambos*, se seleccionará la entrada '1' del multiplexor, y obtendré un 1 a la salida de los multiplexores cuando $Ambos=0$ y un 0 cuando $Ambos=1$. Como las salidas b_1 y b_2 son las salidas de los multiplexores negadas, entonces obtendremos $b_1 b_2 = 11$ cuando $Ambas=1$ y $b_1 b_2 = 00$ cuando $Ambas=0$

El esquemático completo del circuito implementado en Altium Designer, a partir del cual se diseñó el circuito impreso se muestra en la Figura 5.

Mediciones y Simulación

En esta sección se contrastan las mediciones realizadas sobre el dispositivo implementado con los resultados arrojados por los códigos de Verilog que se encuentran disponibles en el repositorio. El diagrama de tiempo de las variables del modulo simulado se observan en la Figura 6.

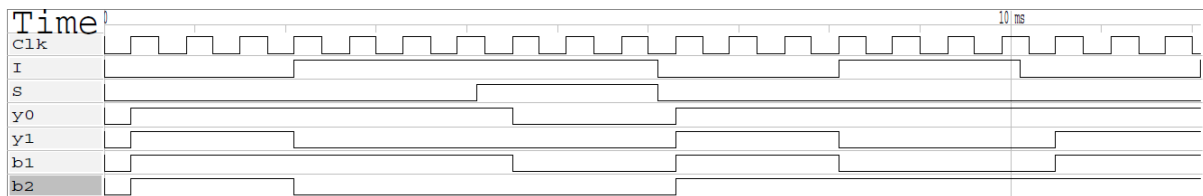


Figura 6: Diagrama de tiempo simulado

El comportamiento de la simulación es el esperado, cumpliendo con los requerimientos de la consigna.

A continuación, en la figura 7 se muestra una medición realizada sobre la placa implementada. En el mismo se muestran las entradas de los sensores I y S en los canales 3 y 4, y las salidas b_1 y b_2 en los canales 1 y 2. Como se mencionó anteriormente el caso $S=1, I=0$ se consideró como 'don't care', la medición muestra que para esta configuración de entrada ambas bombas están en estado 'apagadas'.

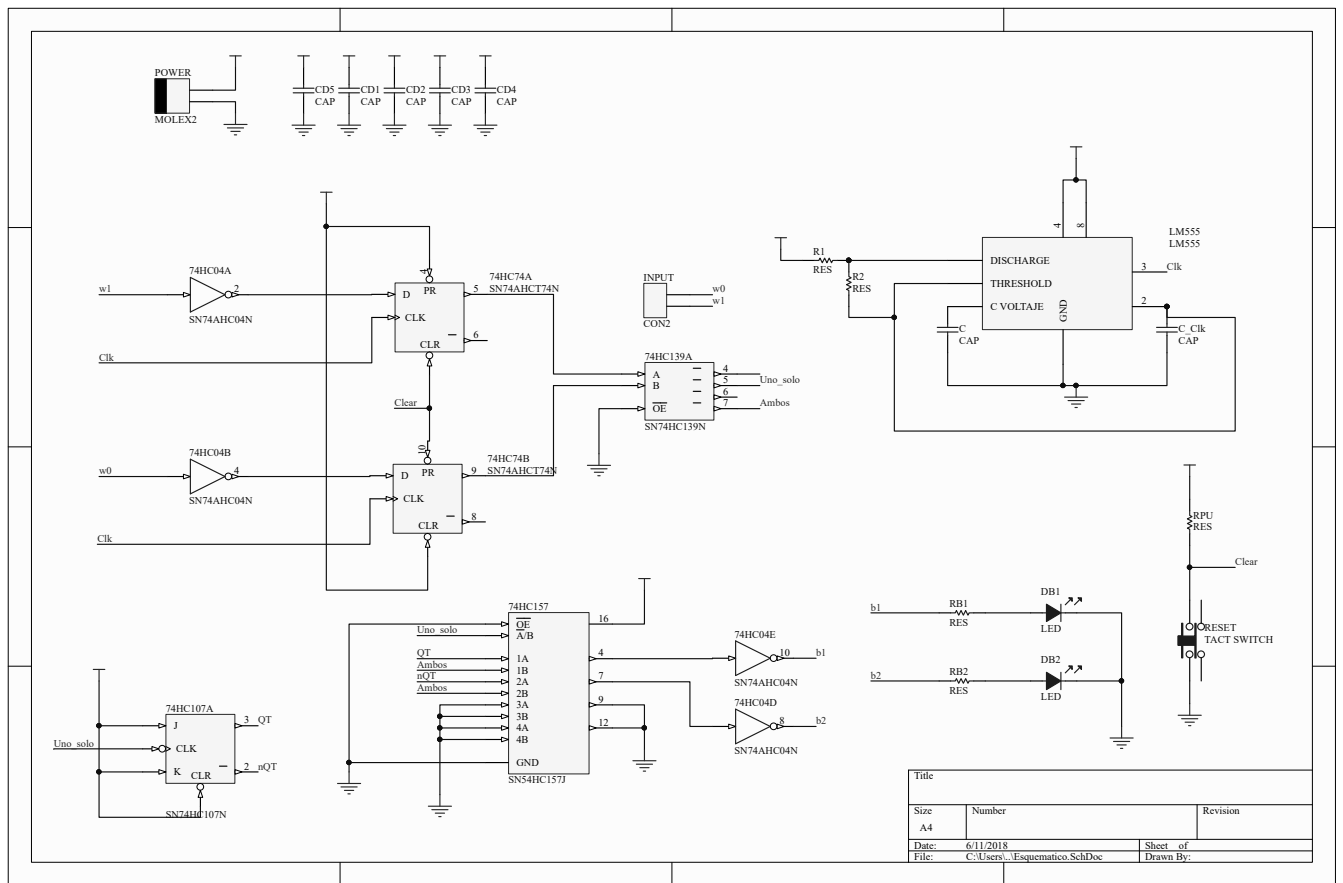


Figura 5: Esquemático del circuito implementado. Máquina de Moore

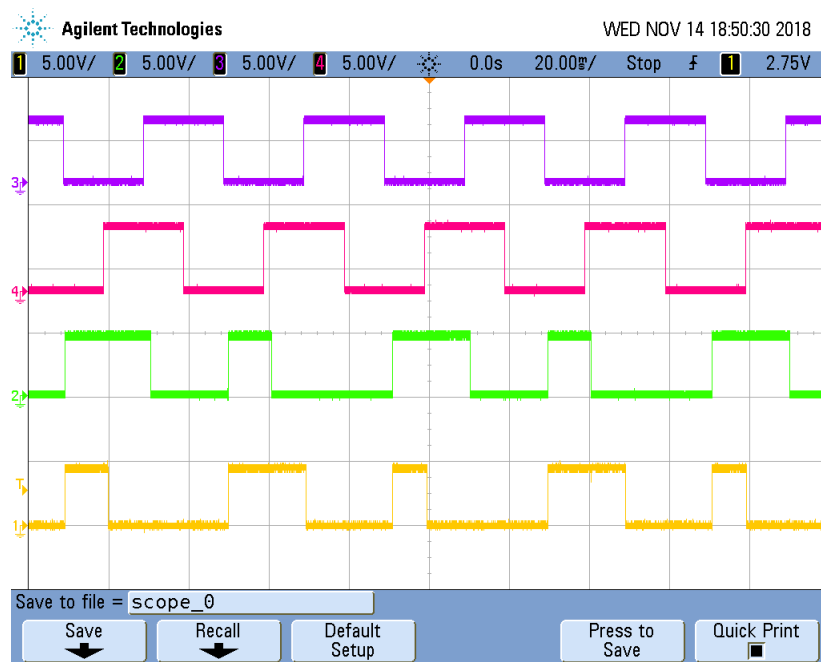


Figura 7: Medición de entradas y salidas

Implementación Máquina de Mealy

Diagrama de Estados y Transiciones

Al igual que en el proceso de diseño de la máquina de estados de Moore, el primer paso fue determinar los estados de la máquina de estados. Las entradas y salidas son las mismas que para la máquina de Moore.

Estados:

Juntas: Estado que indica que ambas bombas están en el mismo estado, así sea encendidas o apagadas.

Complementarias: Estado que indica que los estados de las bombas son complementarios, es decir, una encendida y la otra apagada.

y	Estado
0	Complementarias
1	Juntas

Al solamente necesitar de dos estados, puedo representarlos utilizando solamente una variable: y . El diagrama de estados de la máquina de Mealy se muestra en la Figura 8.

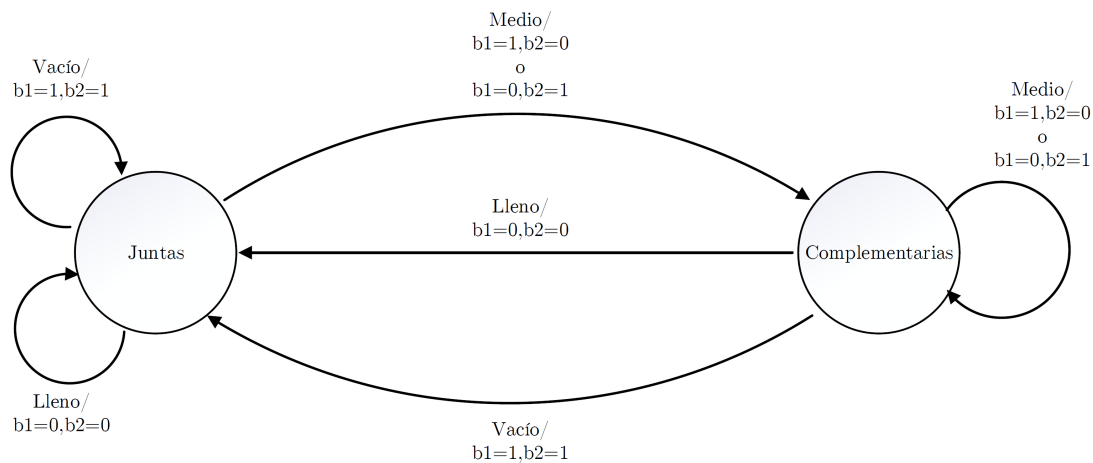


Figura 8: Diagrama de estados y transiciones

Con el diagrama de estados definido, se completo la tabla de asignación de estados, representada en el Cuadro 3.

Estado Actual (y)	Próximo Estado(Y)			Salida(b_1b_2)		
	$SI = 00$	$SI = 01$	$SI = 11$	$SI = 00$	$SI = 01$	$SI = 11$
0	1	0	0	11	10 o 01	00
1	1	0	0	11	10 o 01	00

Cuadro 3: Tabla de asignación de estados

Al igual que se procedió en el caso de la máquina de Moore, a partir del Cuadro 8 se confeccionó el Cuadro 4, que implementa la tabla de verdad que determina el próximo estado (Y) y las salidas(b_1b_2) en función del estado anterior (y) y las entradas (SI).

ySI	Y	b_1b_2
000	1	11
001	0	xx
010	x	xx
011	1	00
100	1	11
101	0	xx
110	x	xx
111	1	00

Cuadro 4: Tabla de verdad cambio de estado

Cabe destacar que en la tabla de verdad del Cuadro 4, las salidas se definen como 'don't care'(x) para las configuraciones de entradas $ySI = x01$ y $ySI = x10$, esto es porque al igual que en la máquina de Moore, se utilizará un multiplexor y un Flip-Flop JK en modo Toggle ($J=1$, $K=1$) para seleccionar que las salidas sean iguales o complementarias.

Se simplificaron las funciones mediante mapas de Karnaugh, arrojando los siguientes resultados:

$y \backslash SI$	00	01	11	10
0	1	X	0	X
1	1	X	0	X

$$b_1 = \bar{S}, b_2 = \bar{S}$$

$y \backslash SI$	00	01	11	10
0	1	0	1	X
1	1	0	1	X

$$Y = S + \bar{I} = \bar{S}I$$

Implementación

Se utilizó un Flip-Flop D para representar los dos estados de la maquina de Mealy, siguiendo los resultados arrojados por las tablas de verdad y los mapas de Karnaugh del apartado anterior. La Figura 9 muestra el diagrama implementado.

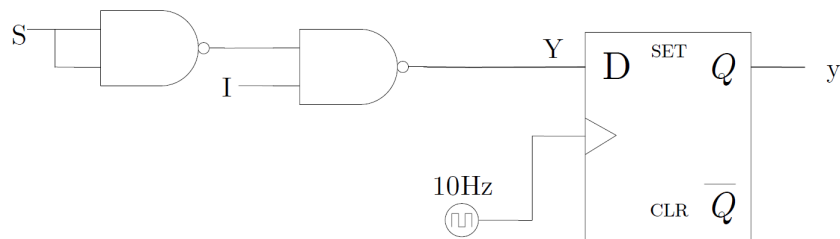


Figura 9: Lógica de transición de estados

La Figura 10 muestra la lógica implementada a la salida de la máquina de Mealy:

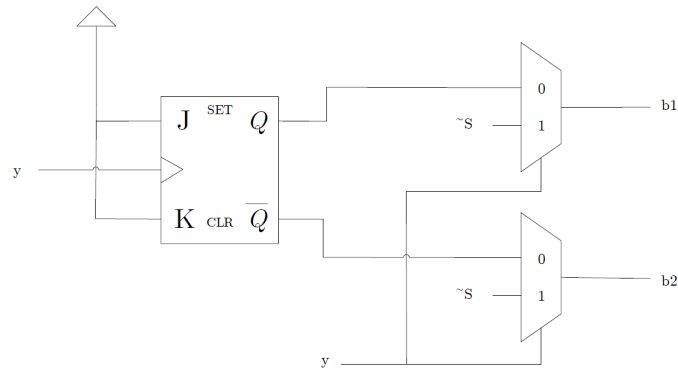


Figura 10: Lógica a la salida

Al igual que en la máquina de Moore, el Flip-Flop JK togglea sus salidas cada vez que la variable de estados y cambia de 0 a 1, es decir, cada vez que las bombas dejan de trabajar de forma complementaria. Las dos salidas del Flip-Flop alimentan la entrada '0' del multiplexior. Las entradas '1' de los mismos están alimentadas por la señal \bar{S} . Las líneas de selección de los multiplexores están controladas por la variable de estado y . De esta forma cuando $y = 0$ (bombas complementarias), a la salida de los multiplexores obtengo las salidas complementarias del Flip-Flop JK. Por otro lado cuando $y = 1$ (ambas bombas trabajan juntas) obtengo a la salida de los multiplexores la señal \bar{S} , entonces cuando el depósito este lleno ($S = 1$) obtendré en ambas salidas $\bar{1} = 0$, es decir estarán apagadas. Cuando el deposito esté vacío la máquina esta en el estado $y = 1$ y las entradas indicaran $SI = 00$, entonces tendré en las salidas $\bar{0} = 1$, es decir ambas bombas estarán encendidas.

El esquemático completo del circuito implementado en Altium Designer, a partir del cual se diseñó el circuito impreso se muestra en la Figura 11.

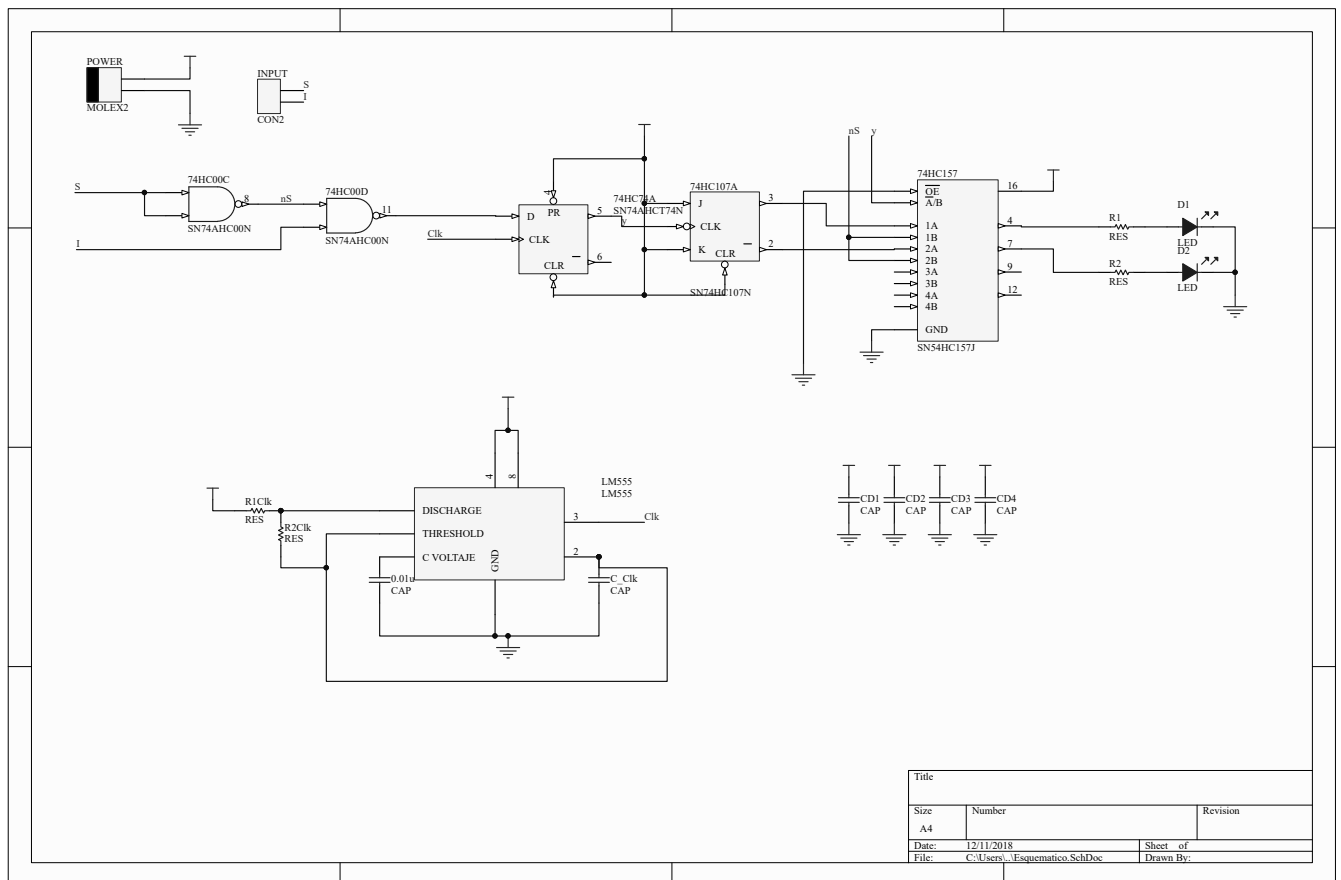


Figura 11: Esquemático del circuito implementado. Máquina de MEaly

Mediciones y Simulación

En esta sección se contrastan las mediciones realizadas sobre el dispositivo implementado con los resultados arrojados por los códigos de Verilog que se encuentran disponibles en el repositorio. El diagrama de tiempo de las variables del modulo simulado se observan en la Figura 12.

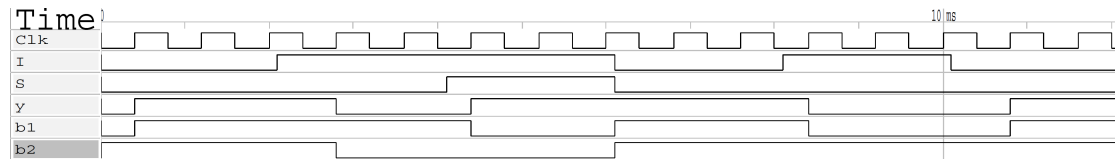


Figura 12: Diagrama de tiempo simulado

El comportamiento de la simulación es el esperado, cumpliendo con los requerimientos de la consigna.

A continuación, en la figura 7 se muestra una medición realizada sobre la placa implementada. En el mismo se muestran las entradas de los sensores S e I en los canales 3 y 4 respectivamente, y las salidas b_1 y b_2 en los canales 1 y 2. Como se mencionó anteriormente el caso $S=1, I=0$ se consideró como 'don't care' y la medición al igual que en la máquina de Moore, la máquina de Mealy muestra como salida $b_1b_2 = 00$ para el caso mencionado.

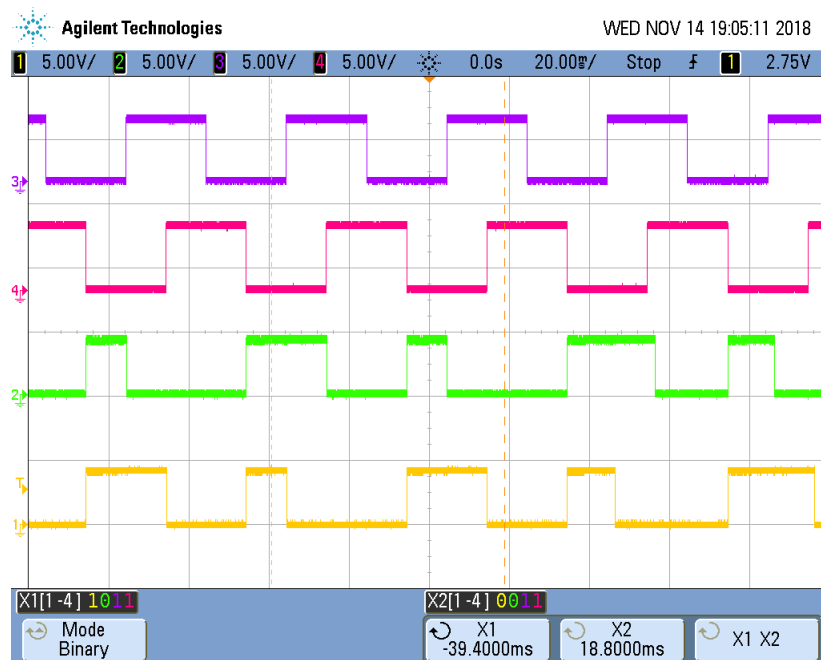


Figura 13: Medición de entradas y salidas