

INSTITUTO TECNOLÓGICO DE BUENOS AIRES

22.13 ELECTRONICA III

TRABAJO PRÁCTICO 3

TP3: Máquinas de Estado

Grupo 4:

Lisandro ALVAREZ 57.771

Milton DELGADO 56.451

Paulo NAVARRO 57.775

Matias FOGG 56.252

Profesores:

Kevin DEWALD

Pablo WUNDES

Realizado: 10/10/2018

Presentado: 17/10/2018

Corrección:

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:

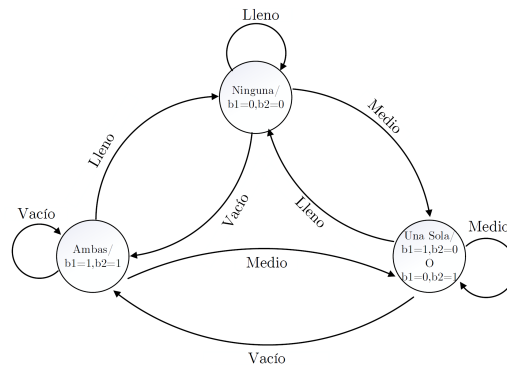


Figura 1: Diagrama de estados y transiciones

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

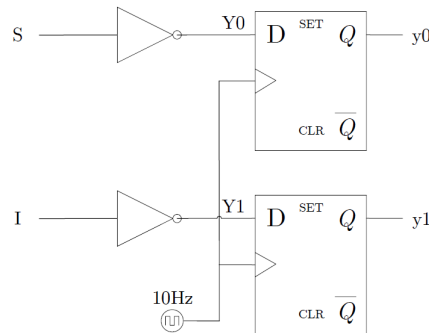


Figura 2: Lógica de transición de estados

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.

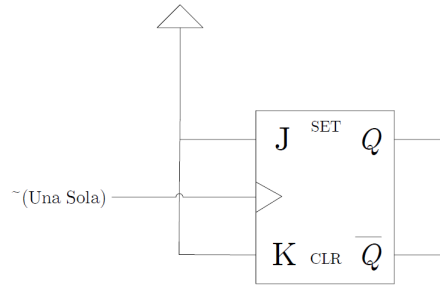


Figura 3: Flip-Flop JK para alternar salidas

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

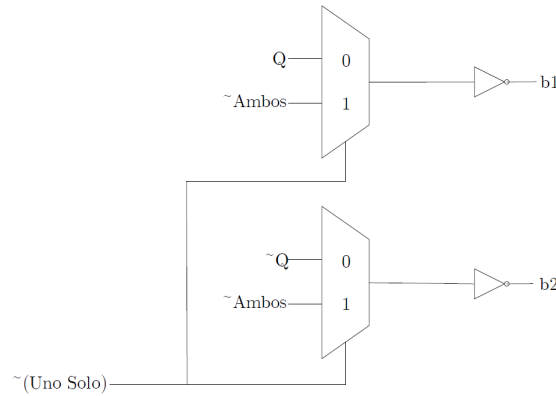


Figura 4: Multiplexación a la salida

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 $b_1b_2 = 11$ cuando $Ambas=1$ y $b_1b_2 = 00$ 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.

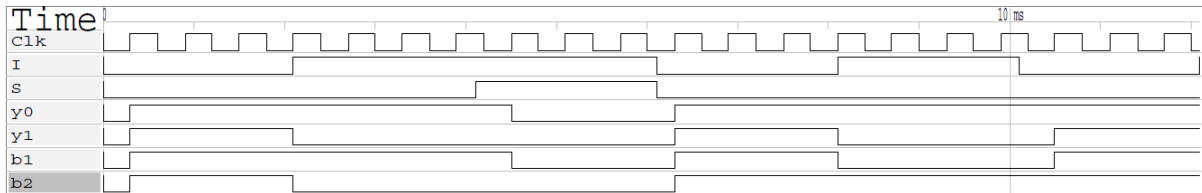


Figura 6: Diagrama de tiempo simulado

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'.

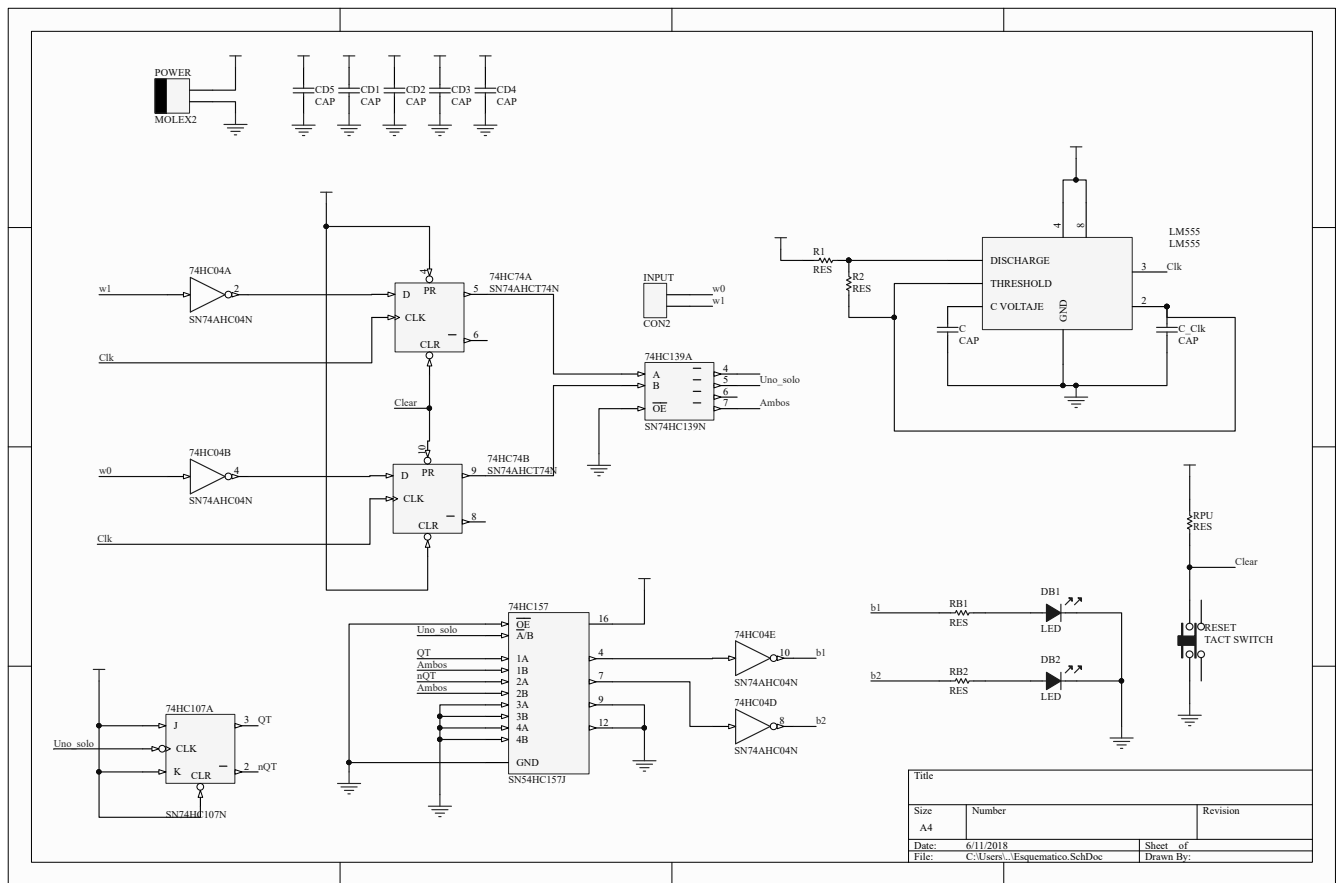


Figura 5: Esquemático del circuito implementado. Máquina de Moore

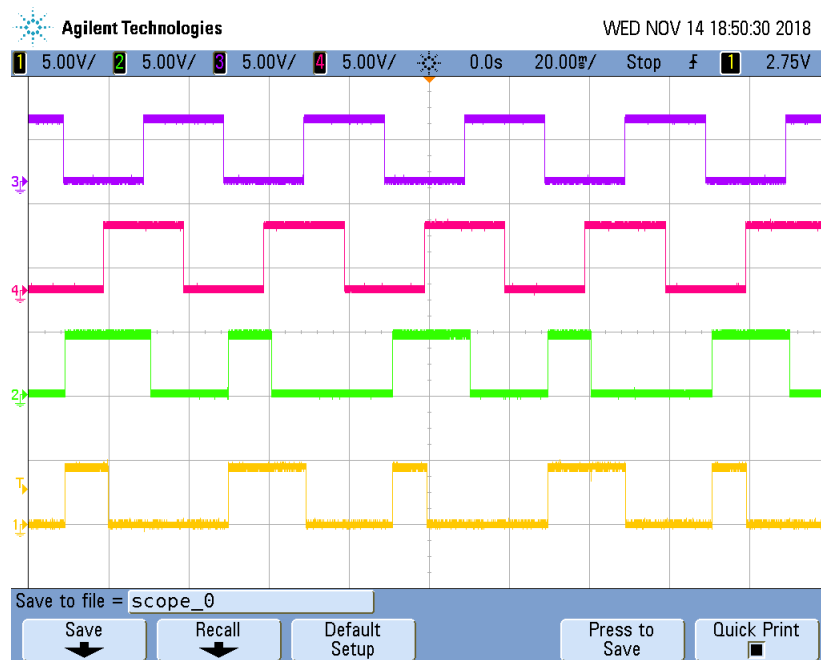


Figura 7: Medición de entradas y salidas

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.

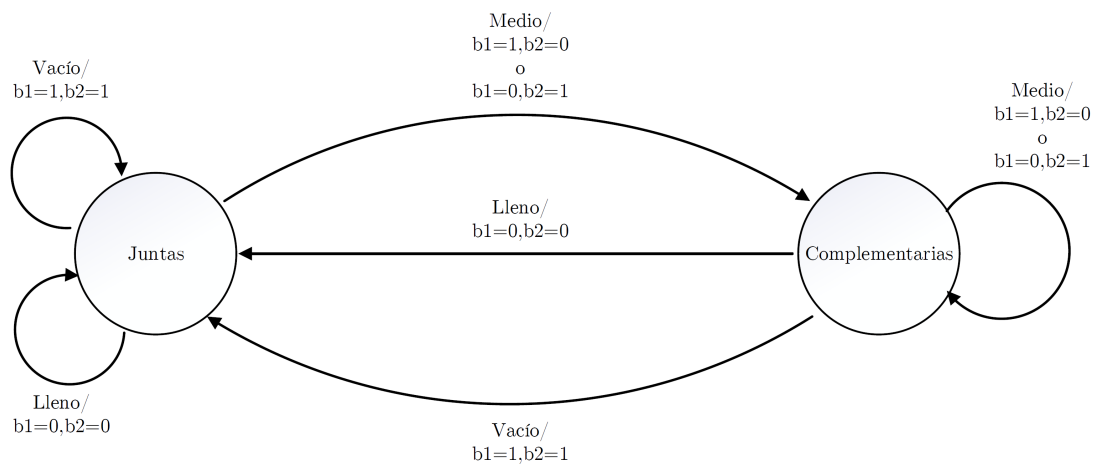


Figura 8: Diagrama de estados y transiciones

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.

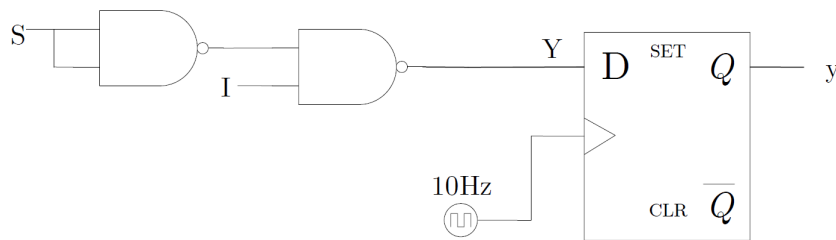


Figura 9: Lógica de transición de estados

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

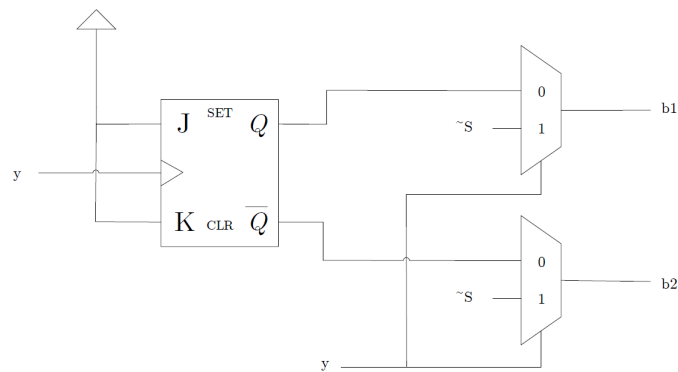


Figura 10: Lógica a la salida

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 multiplexor. 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 esté lleno ($S = 1$) obtendré en ambas salidas $\bar{I} = 0$, es decir estarán apagadas. Cuando el depósito esté vacío la máquina está 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.

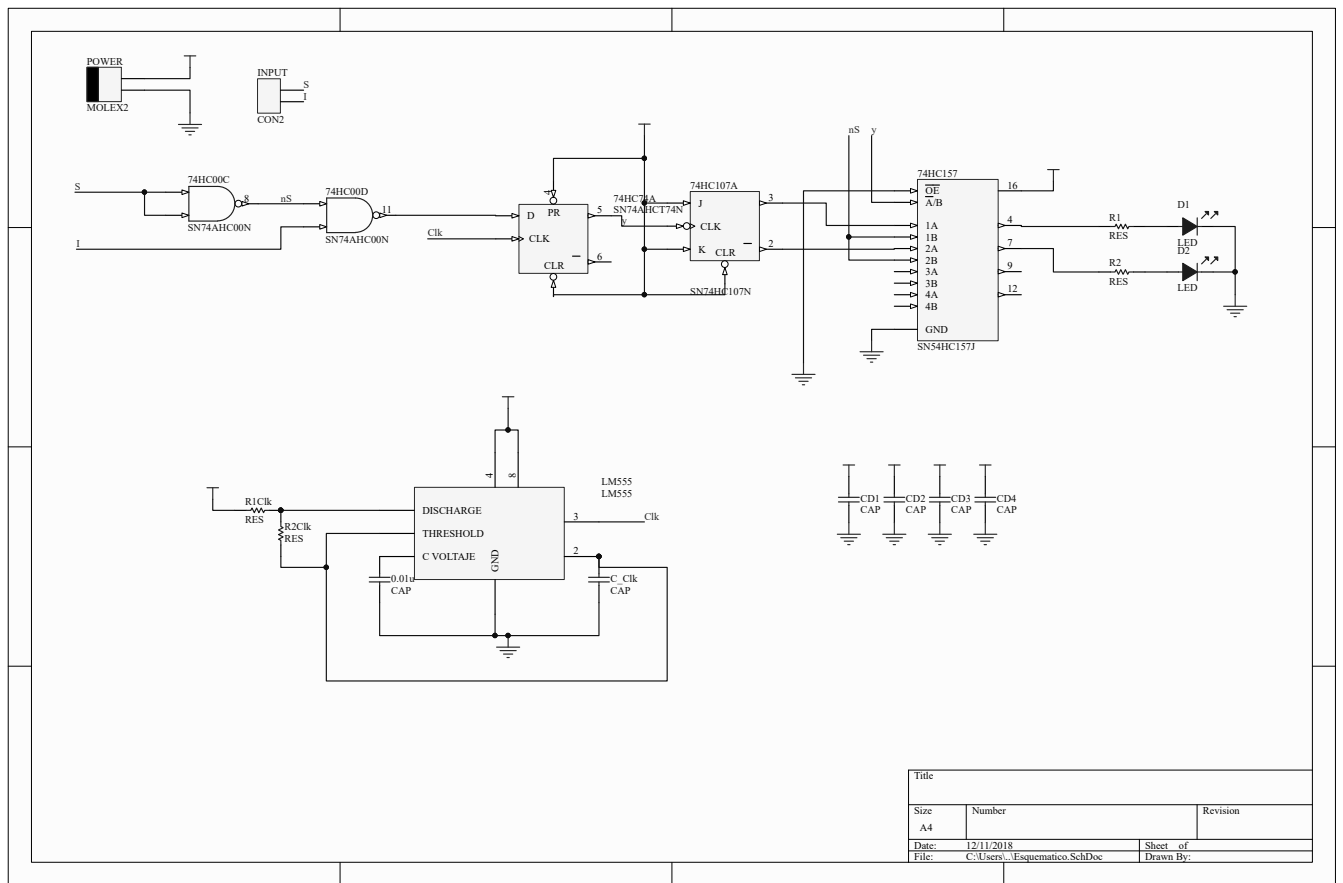


Figura 11: Esquemático del circuito implementado. Máquina de MEaly

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.

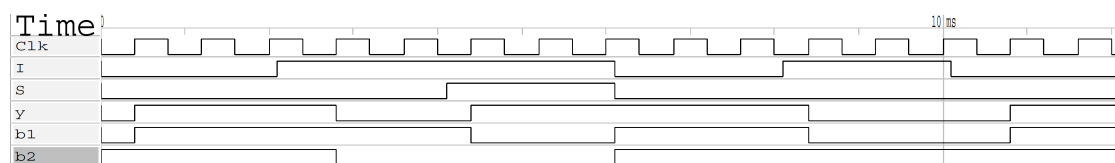


Figura 12: Diagrama de tiempo simulado

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.

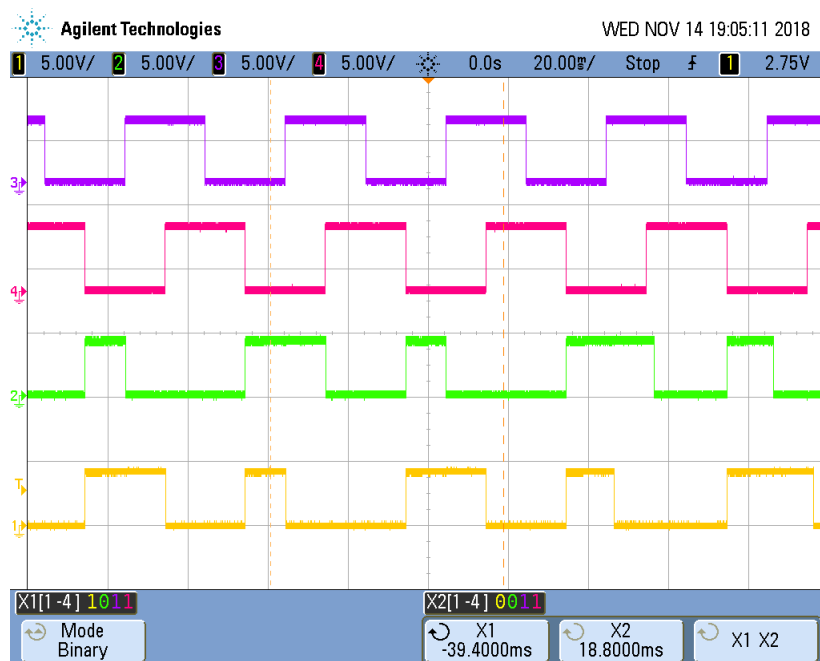


Figura 13: Medición de entradas y salidas

Ejercicio 2

Moore

Se nos pidió realizar una máquina de Moore y una de Mealy que puedan detectar la secuencia 1-1-0-1 y avise en la salida al detectarla. Para esto tuvimos en cuenta que cuando termina la secuencia y es detectada se toma el último 1 de la secuencia como el primero de la siguiente secuencia en caso de que ocurran 2 secuencias seguidas, la cual sería 1-1-0-1-1-0-1. Primero diseñamos un diagrama de estados basándonos en la máquina de Moore, donde tendremos 5 estados. El estado A es el caso base donde se recibió un 0 fuera de la secuencia pedida, el estado B es el caso donde se recibió el primer 1, el estado C es el caso donde se recibe el segundo 1, el estado D donde se recibe la secuencia 1-1-0 y el estado E es cuando se recibe la secuencia completa. El diagrama queda de la siguiente manera:

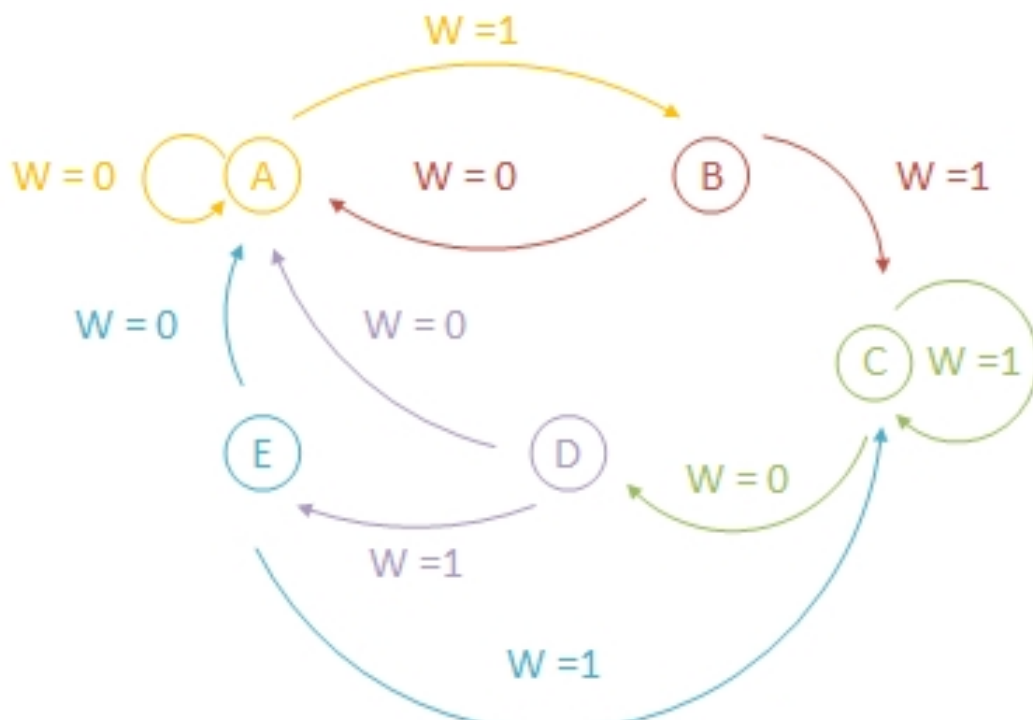


Figura 14: Diagrama de Estados - Máquina de Moore

Se le otorgó a cada estado un valor representativo en binario, siendo el estado A un 0 y el estado E un 6 (el cual al estar en binario se representara como 1-1-0), no utilizamos los valores comunes del 0 al 4 ya que preferimos utilizar el código de Grey para que los flip flops tuvieran menos transiciones al pasar de un estado al otro. Como el estado E es un número que necesita 3 bits de memoria, se utilizarán 3 flip flop para almacenar el número del estado actual, cada flip flop se representará en este ejercicio como Q_n , siendo entonces cada uno representación de un bit del estado en el cual se encuentra el circuito. El flip flop Q_0 representa el bit menos significativo, y el Q_2 el bit más significativo. Cada estado podrá pasar a otro según la entrada que reciba, como se mostró en el diagrama anterior, ahora pasamos a representar el esquema en una tabla con las variaciones de los estados:

	W = 0	W = 1	Z
A ₀₀₀	A	B	0
B ₀₀₁	A	C	0
C ₀₁₁	D	A	0
D ₀₁₀	A	E	0
E ₁₁₀	A	C	1

Figura 15: Transiciones con Estados - Máquina de Moore

Si ahora representamos a cada estado con sus respectivos valores Q_n para ver las transiciones la tabla quedará con valores 1 y 0 que representarán una salida High o Low respectivamente. Así podremos analizar cada flip flop por separado y llegar a un circuito combinacional que los alimente, para esto tenemos que discriminar entre los estados actuales Q_{n_t} y los estados siguientes $Q_{n_{t+1}}$. La tabla dicha es la siguiente:

Estado Actual			Estado						Salida
			W = 0			W = 1			
Q_{2_t}	Q_{1_t}	Q_{0_t}	$Q_{2_{t+1}}$	$Q_{1_{t+1}}$	$Q_{0_{t+1}}$	$Q_{2_{t+1}}$	$Q_{1_{t+1}}$	$Q_{0_{t+1}}$	Z
0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	1	1	0
0	1	1	0	1	0	0	1	1	0
0	1	0	0	0	0	1	1	0	0
1	1	0	0	0	0	0	1	1	1
1	1	1	X	X	X	X	X	X	X
1	0	1	X	X	X	X	X	X	X
1	0	0	X	X	X	X	X	X	X

Figura 16: Transiciones con Flip Flop - Máquina de Moore

Para analizar esta tabla debemos tener en cuenta que para la máquina de Moore los estados son dependientes de las entradas y de ellos mismos, por lo que cada estado Q_{n_t} dependerá tanto de la entrada W como de los estados Q_{2_t}, Q_{1_t} y Q_{0_t} . Mientras que la salida Z depende solo de los estados Q_{2_t}, Q_{1_t} y Q_{0_t} . Por lo que pasaremos a analizar cada columna $Q_{n_{t+1}}$ dependiendo de cada combinación Q_{n_t} y la entrada W , y luego analizaremos la columna Z para cada combinación de los Q_{n_t} . Para analizar las columnas las resolvimos con mapas de Karnaugh para simplificar más rápido los minitérminos quedando los estados y la salida de las siguientes maneras:

		$Q_1 Q_0$			
		00	01	11	10
$W Q_2$	00	0	0	0	0
	01	X	X	X	0
	11	X	X	X	1
	10	1	1	1	0

Figura 17: $Q_{0_{t+1}}$ Máq. de Moore

		Q_1Q_0			
		00	01	11	10
WQ_2	00	0	0	1	0
	01	X	X	X	0
	11	X	X	X	1
	10	0	1	1	1

Figura 18: Q_{1t+1} Máq. de Moore

		Q_1Q_0			
		00	01	11	10
WQ_2	00	0	0	0	0
	01	X	X	X	0
	11	X	X	X	0
	10	0	0	0	1

Figura 19: Q_{2t+1} Máq. de Moore

		Q_1Q_0			
		00	01	11	10
Q_2	0	0	0	0	0
	1	X	X	X	1

Figura 20: Z : Salida - Máq. de Moore

Tomando las agrupaciones marcadas en cada mapa podemos formar las ecuaciones que representarán a Q_{0t+1} , Q_{1t+1} , Q_{2t+1} y Z respectivamente, quedando las ecuaciones de la siguiente manera:

$$\begin{aligned}
 Q_{0t+1} &= W * \overline{Q_1} + W * (Q_2 + Q_0) \\
 Q_{1t+1} &= W * (Q_2 + Q_0) + Q_1 * (W + Q_0) \\
 Q_{2t+1} &= W * \overline{Q_2} * Q_1 * \overline{Q_0} \\
 Z &= Q_2
 \end{aligned}$$

Una vez que ya tenemos las ecuaciones procedemos a armar el circuito:

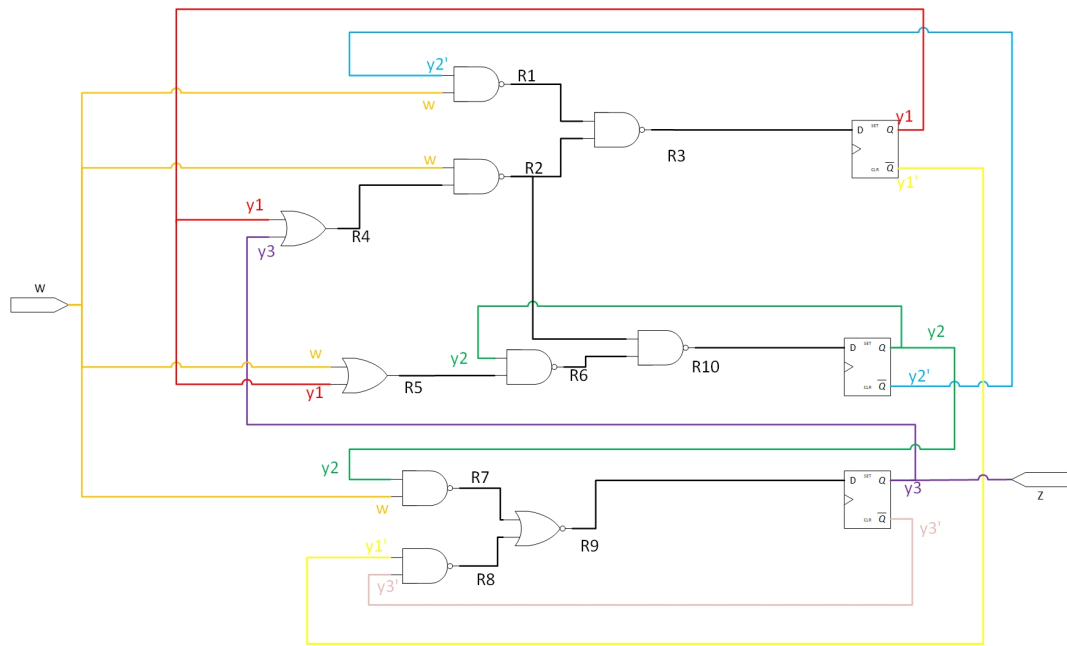


Figura 21: Circuito lógico de Moore

Con este circuito y una simulación en Verilog que se encuentra en el repositorio pudimos ver si se cumplía lo pedido y se observó que el circuito encuentra bien la secuencia pedida.

Mealy

También se nos pidió armar una máquina pero de Mealy en lugar de Moore. Para esto rediseñamos el diagrama de la máquina de estados utilizando la notación de Mealy con flechas que indiquen la entrada y la salida entre transición de estados. Utilizando este método tendremos menos estados pero la salida esta vez será dependiente tanto de los estados como de las entradas.

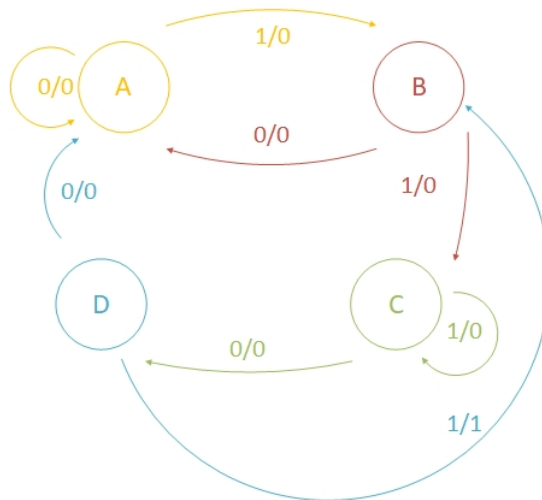


Figura 22: Diagrama de Estados - Máquina de Mealy

Partiendo de este nuevo diagrama, se tabula un nuevo cuadro con la transiciones de estados y la salida, para luego asignarle un valor numérico a cada estado. Como esta vez son 4 estados solo se necesitara representar desde el 0 al 3, los cuales en binario serán 0-0 y 1-1, solo se necesitaran 2 flip flops para esta máquina de estados. La siguiente tabla muestra los valores que $Q_{1_{t+1}}$, $Q_{0_{t+1}}$ y Z tendrán ahora.

Estado Actual	Estado Siguiente		Salida: Z	
	W = 0	W = 1	W = 0	W = 1
A ₀₀₀	A	B	0	0
B ₀₀₁	A	C	0	0
C ₀₁₀	D	C	0	0
D ₀₁₁	A	B	0	1

Figura 23: Transiciones - Máquina de Mealy

En este caso hicimos un análisis previo y decidimos ordenar los estados por código de Grey para q de un estado a otro solo haya un cambio de estado, solo un bit cambiaría entre cada transición excepto del estado C al A, lo que significaría que un solo flip flop cambiaría entre estado. Siendo entonces el estado A la combinación 0-0, B la combinación 0-1, C la combinación 1-1 y por último D la combinación 1-0. Utilizamos este método ya que nos ahorra un integrado a comparación del anterior. La siguiente tabla muestra las transiciones con el código de Grey:

Estado Actual		Estado Siguiente				Salida: Z	
		W = 0		W = 1			
Q_{1_t}	Q_{0_t}	$Q_{1_{t+1}}$	$Q_{0_{t+1}}$	$Q_{1_{t+1}}$	$Q_{0_{t+1}}$	W = 0	W = 1
0	0	0	0	0	1	0	0
0	1	0	0	1	1	0	0
1	1	1	0	1	1	0	0
1	0	0	0	0	1	0	1

Figura 24: Transiciones con Flip Flop - Máquina de Mealy

Teniendo en cuenta que la única diferencia en esta parte con respecto a Moore es que la salida Z depende de la entrada y los estados, pasamos a analizar las columnas de la tabla anterior. Las resolvimos con mapas de Karnaugh para simplificar más rápido los minitérminos quedando los estados y la salida de las siguientes maneras:

		Q_1Q_0			
		00	01	11	10
W	0	0	0	0	0
	1	1	1	1	1

Figura 25: Q_{0t+1} Máq. de Mealy

		Q_1Q_0			
		00	01	11	10
W	0	0	0	1	0
	1	0	1	1	0

Figura 26: Q_{1t+1} Máq. de Mealy

		Q_1Q_0			
		00	01	11	10
W	0	0	0	0	0
	1	0	0	0	1

Figura 27: Z: Salida - Máq. de Mealy

Tomando las agrupaciones marcadas en cada mapa podemos formar las ecuaciones que representarán a $Q_{0_{t+1}}$, $Q_{1_{t+1}}$, $Q_{2_{t+1}}$ y Z respectivamente, quedando las ecuaciones de la siguiente manera:

$$\begin{aligned} Q_{0t+1} &= W \\ Q_{1t+1} &= Q_0 * (Q_1 + w) \\ Z &= w * Q_1 * \overline{Q_0} \end{aligned}$$

De estas ecuaciones podemos formar el siguiente circuito:

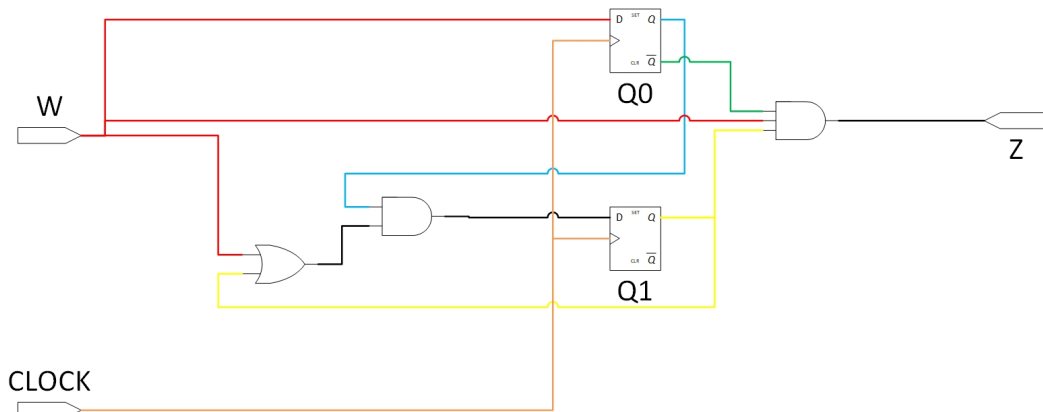


Figura 28: Circuito lógico de Mealy

Con esto podemos concluir que el circuito de Mealy es mejor que el de Moore ya que utiliza menos integrados y realiza la misma función. Observando ambos circuitos pudimos ver que realizaban lo pedido de encontrar la secuencia.

Ejercicio 3

Moore

Se nos dio un diagrama de estado y se nos pidió implementar una máquina de estado de Moore que la resolviera. Para esto hicimos un análisis con tablas que muestre las transiciones de estados, y luego vimos como serían las transiciones con Flip Flop D (). Como ya hicimos anteriormente utilizamos mapas de Karnaugh para resolver el problema.

$Q_1 Q_0$	00	01	11	10
0	0	0	X	0
1	1	0	X	0

Figura 32: Q_0 : Máq. de Moore

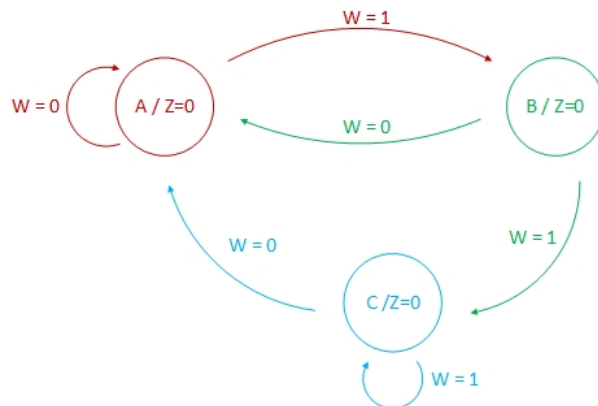


Figura 29: Diagrama Moore

ESTADO ACTUAL	ESTADO SIGUIENTE		SALIDA
	W=0	W=1	
A	A	B	0
B	A	C	1
C	A	C	0
D	X	X	X

Figura 30: Transiciones de Estados - Máquina de Moore

ESTADO ACTUAL		ESTADO SIGUIENTE				SALIDA
		W = 0		W = 1		
Q_{1_t}	Q_{0_t}	$Q_{1_{t+1}}$	$Q_{0_{t+1}}$	$Q_{1_{t+1}}$	Q_{0_t}	Z
0	0	0	0	0	1	0
0	1	0	0	1	0	1
1	0	0	0	1	0	0
1	1	X	X	X	X	X

Figura 31: Transiciones con Flip Flop - Máquina de Moore

		Q_1Q_0			
		00	01	11	10
W	0	0	0	X	0
	1	0	1	X	1

Figura 33: Q_1 Máq. de Moore

		Q_0	
		0	1
Q_1	0	0	1
	1	0	X

Figura 34: Z : Máq. de Moore

Al reducir los minitérminos obtuvimos las siguientes expresiones que representan el circuito lógico que se usara para resolver la máquina de estado:

$$\begin{aligned}
 Q_{0_{t+1}} &= W * \overline{Q_1} * \overline{Q_0} \\
 Q_{1_{t+1}} &= W * (Q_1 + Q_0) \\
 Z &= Q_0
 \end{aligned}$$

Entonces el circuito quedará de la siguiente manera:

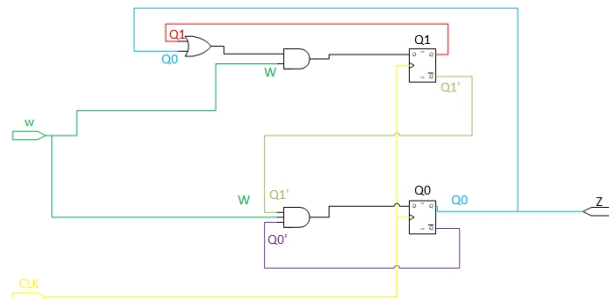


Figura 35: Circuito lógico - Moore

Mealy

Analizando las tablas de transiciones de estados pudimos notar que la función de la máquina es prender la salida cuando se recibe la primer señal en HIGH y luego se apaga en el segundo CLOCK. Para implementar ahora la máquina de Mealy tuvimos en cuenta esto y diseñamos el siguiente diagrama.



Figura 36: Diagrama de Mealy

De este diagrama representamos en una tabla la transiciones de estados directamente con un Flip Flop, ya que al haber solo 2 estados solo se necesita un Flip Flop para representar ambos estados, el estado 0 y estado 1. La tabla es la siguiente:

ESTADO ACTUAL	ESTADO SIGUIENTE		SALIDA	
	Q		Z	
Q	W=0	W=1	W=0	W=1
0	0	1	0	1
1	0	1	0	0

Figura 37: Transiciones con Flip Flop - Máquina de Moore

Sin un análisis muy complejo, el flip flop devuelve una salida en HIGH solo cuando la entrada es HIGH, e igualmente cuando devuelven una señal LOW, por lo que la salida será directamente la entrada. Pero la salida depende tanto del estado como de la entrada, característica de este tipo de máquina de estado, por lo que la salida será:

$$Z = W * \overline{Q_0}$$

El circuito lógico queda muy simple a comparación del logrado con Moore, demostrado en la siguiente figura:

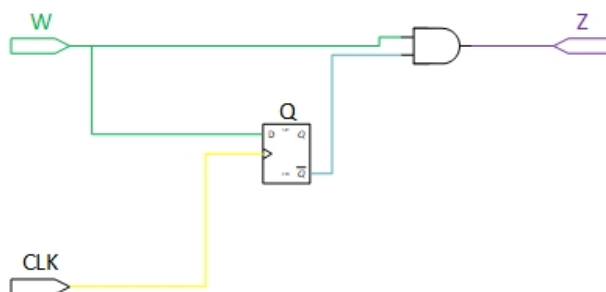


Figura 38: Circuito lógico Mealy

Implementación

Para crear ambas máquinas se pidió que las entradas y salidas del circuito deberán ser lógica de 5V, mientras que toda la lógica interna trabajará con 3,3V. Para esto hicimos un circuito que pueda bajar de 5V a 3,3V y viceversa. Este circuito puede verse abajo, en donde LI es la tensión con la que trabajamos (3,3V o 5V dependiendo el caso), HV es la tensión de alimentación (de nuevo 3,3V o 5V pero sin ser igual a LI) y HO la salida que va a tener la tensión deseada. Cuando LI sea interpretada como estado alto, HO será 5V si LI es 3,3V y HV 5V, en caso de ser LI 5V y HV 3,3V, HO será 3,3V. Si LI es interpretada como estado bajo, HO será 0V.

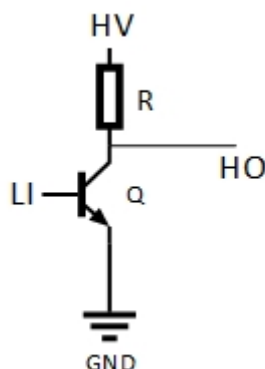


Figura 39: Conversor

Observamos que las tecnologías TTL y CMOS aceptaban bien los valores de 3,3V como señal HIGH para estos circuitos, por lo que no tuvimos restricción con respecto a las tecnologías. En la siguiente imagen se puede ver como la salida cambia cuando el clock es HIGH y la entrada también, pero al segundo CLOCK se apaga aunque la entrada siga en HIGH, demostrando que solo se prende con el primer 1 lógico recibido.

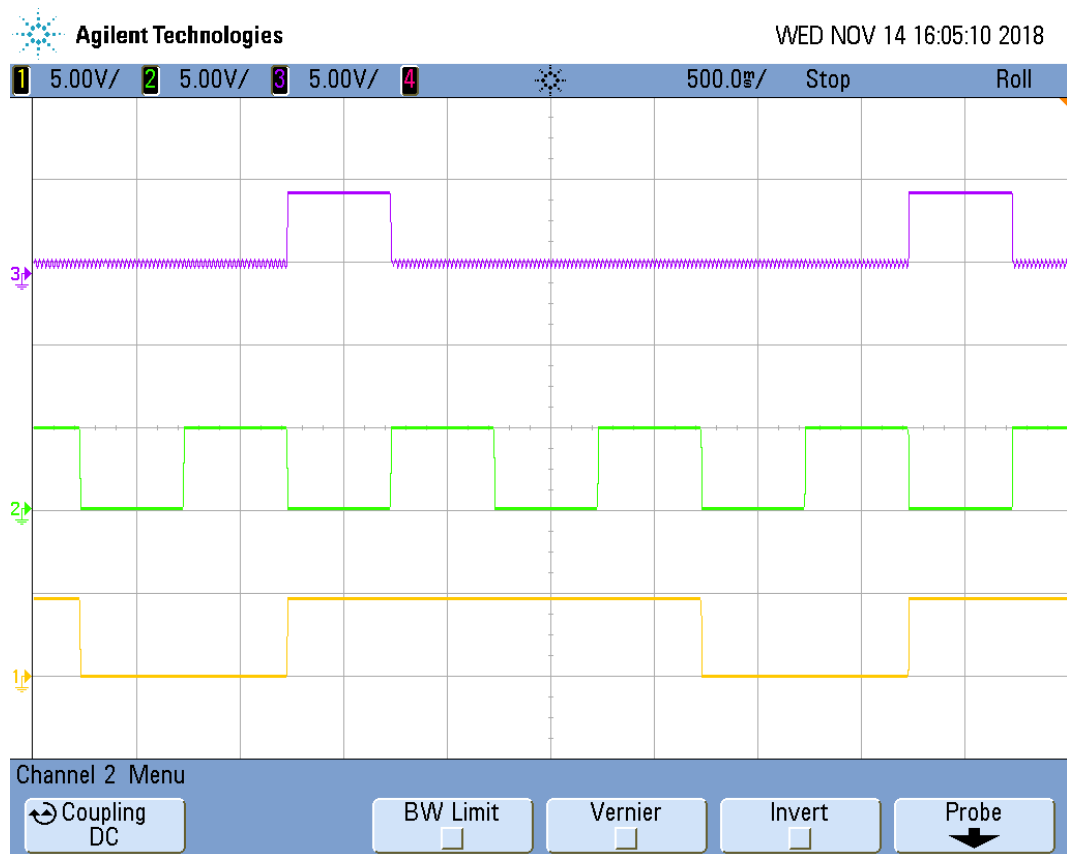


Figura 40: Medición del Ejercicio 3

Conclusión

Pudimos observar el buen funcionamiento de ambas máquinas cumpliendo la misma función, aunque el circuito resuelto con Mealy necesito solo 2 componentes (que serían 2 integrados) por lo que resulta más simple y económico, también tiene menos delay la salida ya que solo consta de una compuerta.