Instituto Tecnológico de Buenos Aires

22.13 Electronica III

Trabajo Práctico 2

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 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:

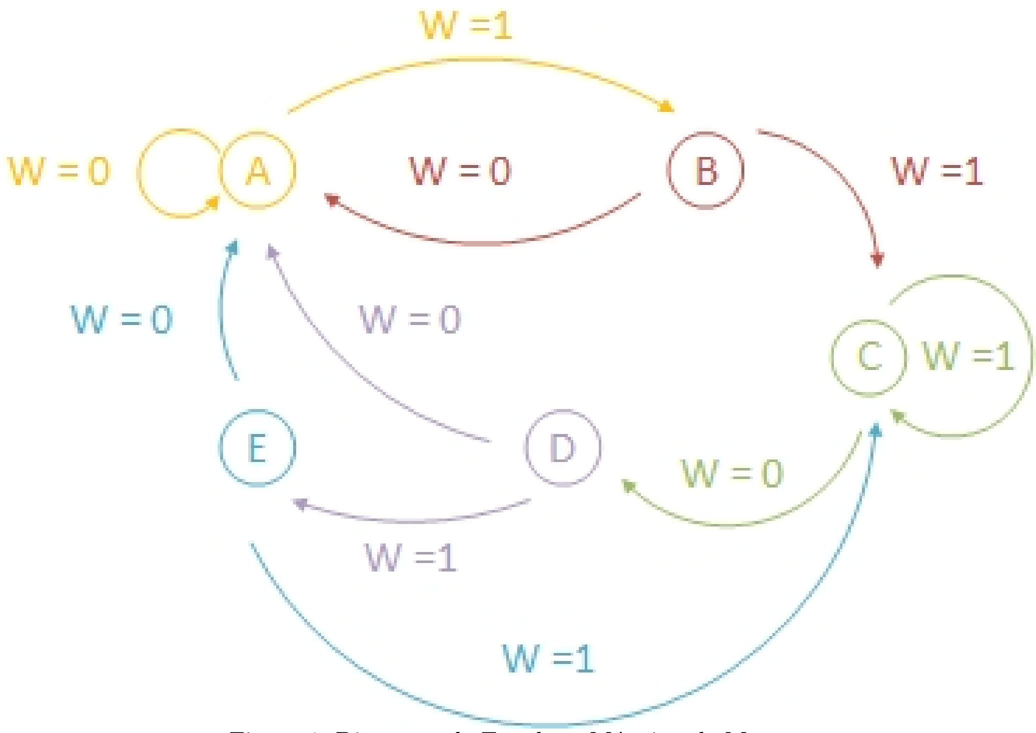

Figura 1: Diagrama de Estados - Máquina de Moore

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**
$\mathbf{A}_{000}$	**A**	**B**	**0**
$\mathbf{B}_{001}$	**A**	**C**	**0**
$\mathbf{C}_{011}$	**D**	**A**	**0**
$\mathbf{D}_{010}$	**A**	**E**	**0**
$\mathbf{E}_{110}$	**A**	**C**	**1**

Figura 2: 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 Qn_t y los estados siguientes Qn_{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 3: 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 Qn_t y la entrada W, y luego analizaremos la columna Z para cada combinación de los Qn_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:

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

Figura 4: $Q_{0_{t+1}}$ Máq. de Moore

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

Figura 5: $Q_{1_{t+1}}$ Máq. de Moore

$WQ_2 \backslash Q_1Q_0$	00	01	11	10
00	0	0	0	0
01	X	X	X	0
11	X	X	X	0
10	0	0	0	1

Figura 6: $Q_{2_{t+1}}$ Máq. de Moore

$Q2 \backslash Q_1Q_0$	00	01	11	10
0	0	0	0	0
1	X	X	X	1

Figura 7: Z: Salida - Máq. de Moore

Tomando las agrupaciones marcadas en cada mapa podemos formas 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:

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

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

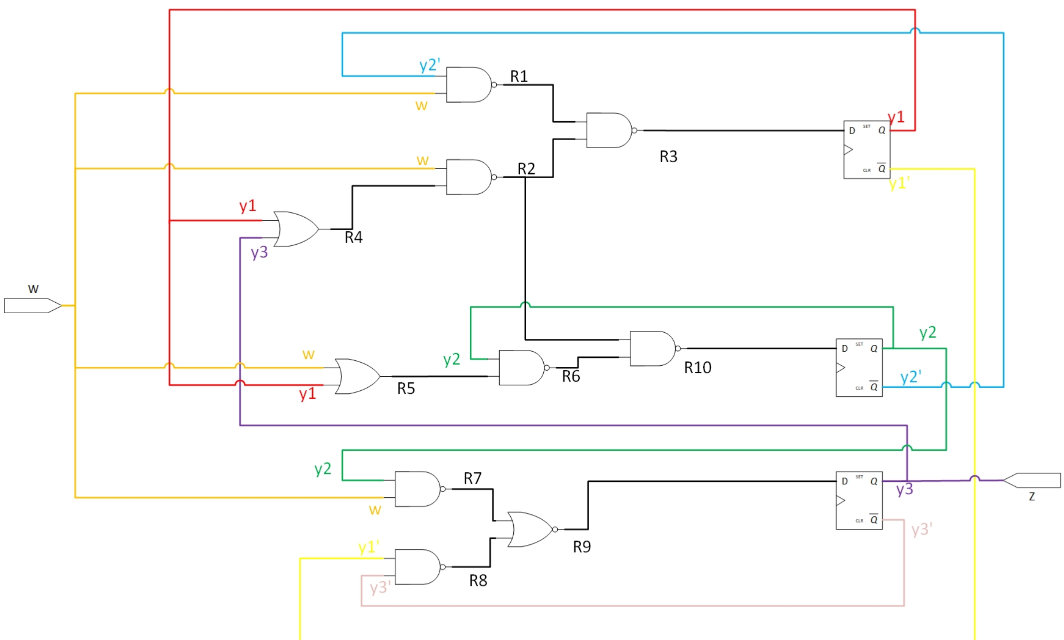

Figura 8: Circuito lógico de Moore

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.

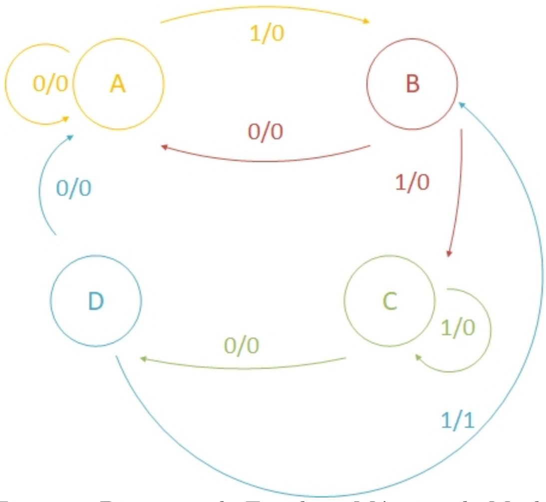

Figura 9: Diagrama de Estados - Máquina de Mealy

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**
$\mathbf{A}_{000}$	**A**	**B**	**0**	**0**
$\mathbf{B}_{001}$	**A**	**C**	**0**	**0**
$\mathbf{C}_{010}$	**D**	**C**	**0**	**0**
$\mathbf{D}_{011}$	**A**	**B**	**0**	**1**

Figura 10: 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 ahorraba 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 11: 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:

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

Figura 12: $Q_{0_{t+1}}$ Máq. de Mealy

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

Figura 13: $Q_{1_{t+1}}$ Máq. de Mealy

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

Figura 14: Z: Salida - Máq. de Mealy

Tomando las agrupaciones marcadas en cada mapa podemos formas 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:

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

De estas ecuaciones podemos formar el siguiente circuito:

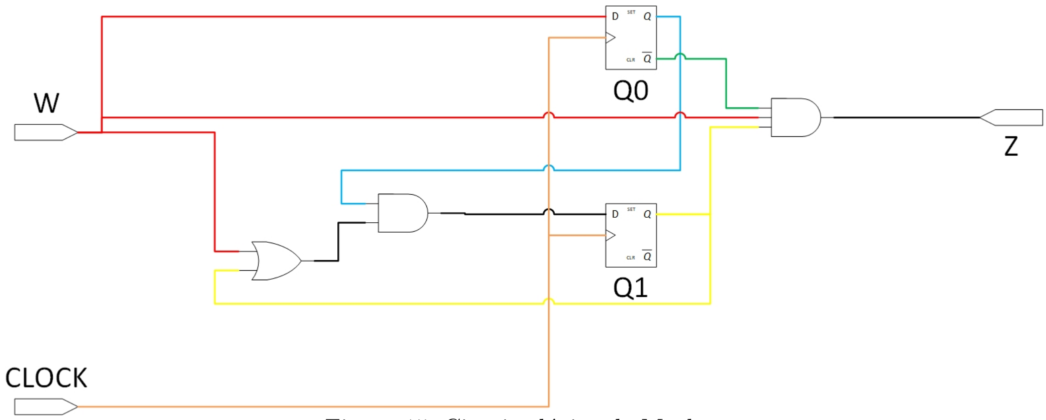

Figura 15: Circuito lógico de Mealy

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.

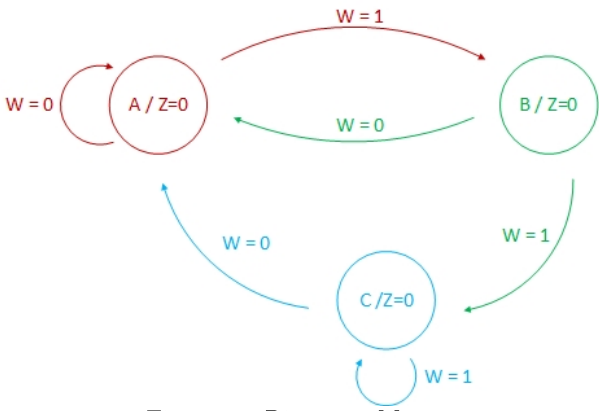

Figura 16: Diagrama Moore

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

Figura 17: Transiciones de Estados - Máquina de Moore

ESTADO ACTUAL		ESTADO SIGUIENTE				SALIDA
		W = 0		W = 1		
$Q1_t$	$Q0_t$	$Q1_{t+1}$	$Q0_{t+1}$	$Q1_{t+1}$	$Q0_t$	**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 18: Transiciones con Flip Flop - Máquina de Moore

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

Figura 19: Q_0: Máq. de Moore

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

Figura 20: Q_1 Máq. de Moore

Q_1 \ Q_0	0	1
0	0	1
1	0	X

Figura 21: 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:

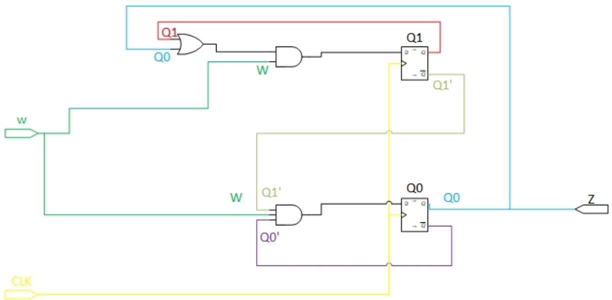

Figura 22: Circuito lógico - Moore

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.

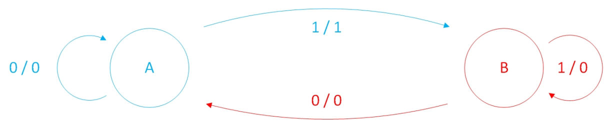

Figura 23: Diagrama de Mealy

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 24: 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:

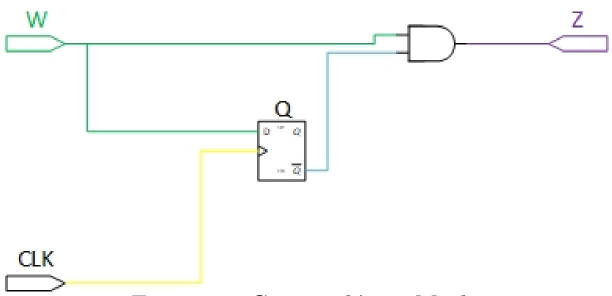

Figura 25: Circuito lógico Mealy

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 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.

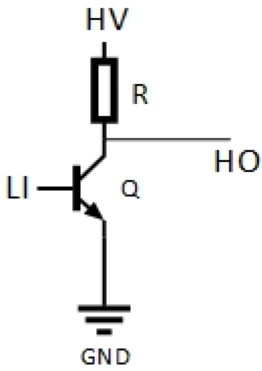

Figura 26: Conversor

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.

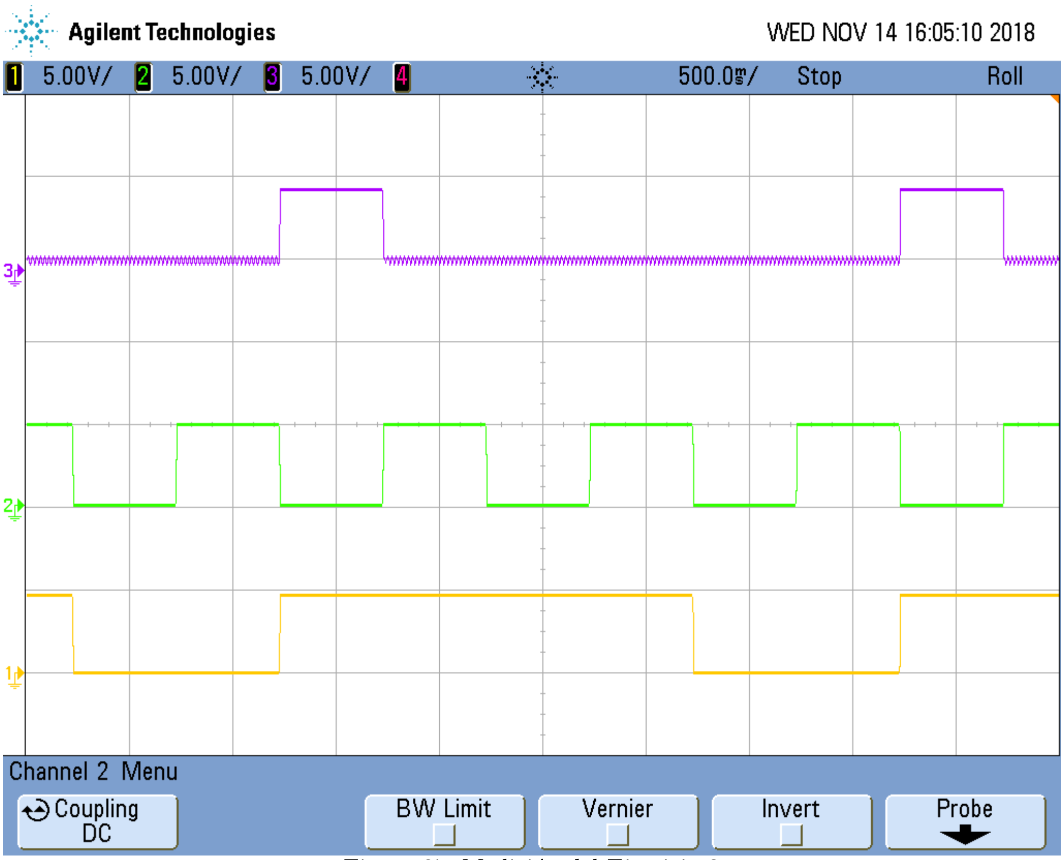

Figura 27: Medición del Ejercicio 3

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.