

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:

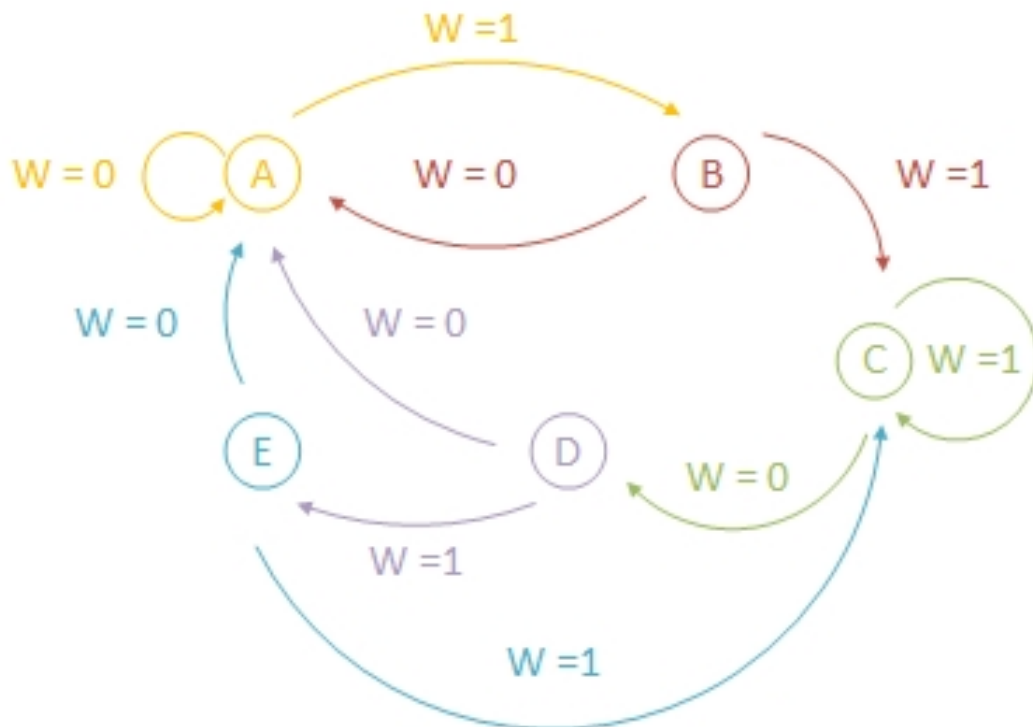


Figura 1: Diagrama de Estados - Máquina de Moore

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
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 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 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_{2t}	Q_{1t}	Q_{0t}	Q_{2t+1}	Q_{1t+1}	Q_{0t+1}	Q_{2t+1}	Q_{1t+1}	Q_{0t+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 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_1Q_0			
		00	01	11	10
WQ_2	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

		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 5: $Q_{1_{t+1}}$ 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 6: 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 7: 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:

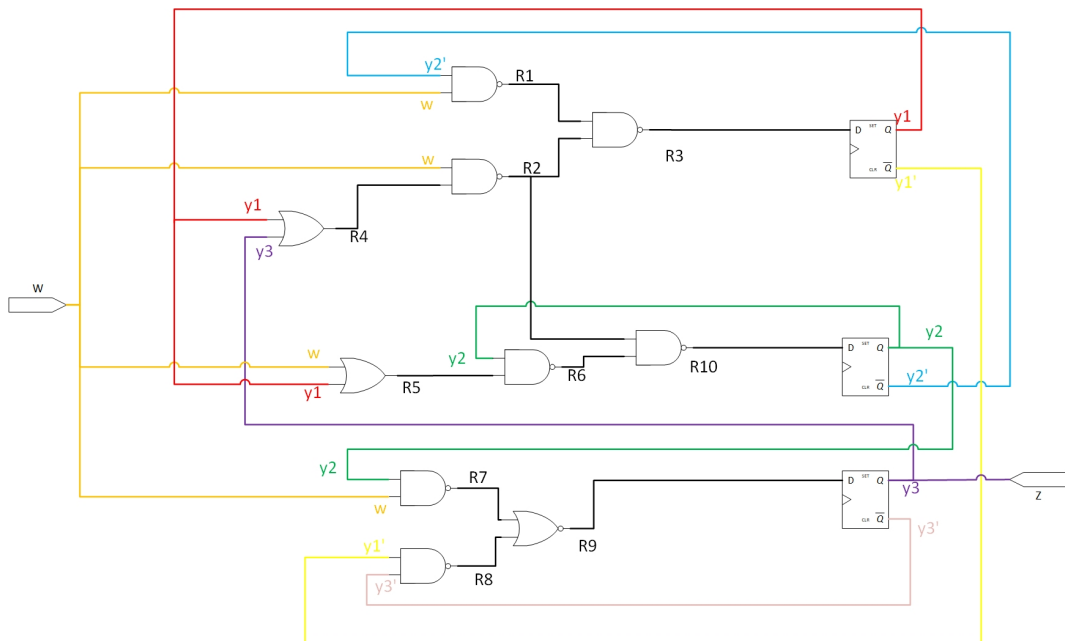


Figura 8: Circuito lógico de Moore

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.

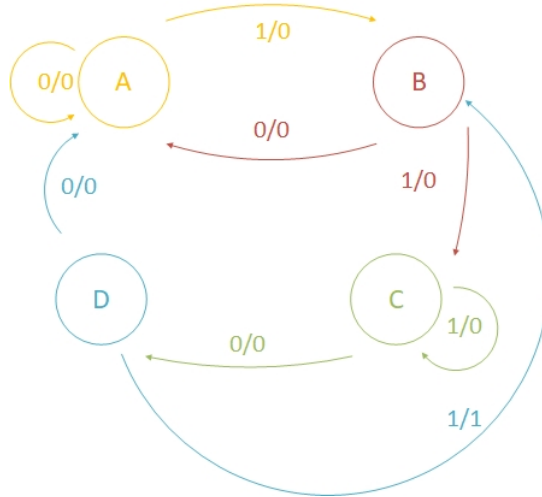


Figura 9: Diagrama de Estados - Máquina de Mealy

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_{1t+1} , Q_{0t+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 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 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_{1t}	Q_{0t}	Q_{1t+1}	Q_{0t+1}	Q_{1t+1}	Q_{0t+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:

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

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

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

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

		Q_1Q_0			
		00	01	11	10
W	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 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:

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

De estas ecuaciones podemos formar el siguiente circuito:

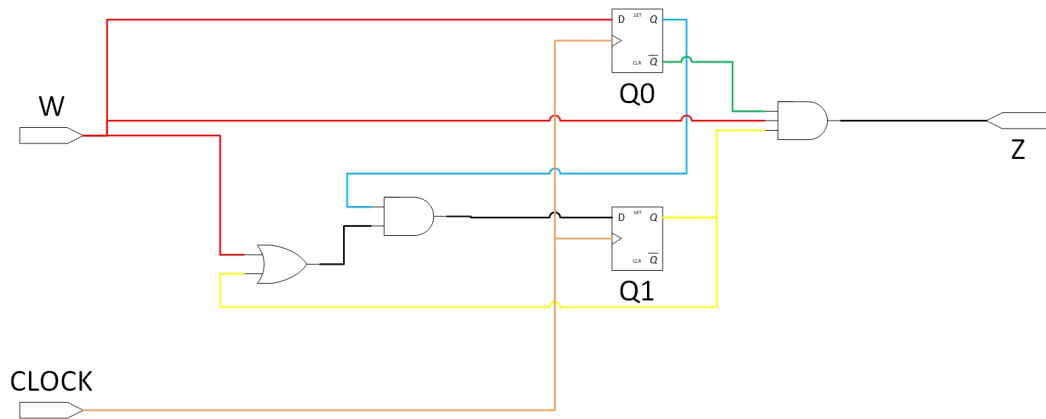


Figura 15: Circuito lógico de Mealy

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.