

Índice

1. Control de bombas de agua	3
1.1. Maquina de Moore	3
1.2. Maquina de Mealy	6
1.2.1. Diagrama de estados	6
1.2.2. Representación en tabla	6
1.2.3. Lógica combinacional	6
1.2.4. Circuito equivalente	7
1.2.5. Simulación	8
2. Detección de secuencia binaria	8
2.1. Maquina de Moore	8
2.2. Maquina de Mealy	10
2.2.1. Diagrama de estados	10
2.2.2. Representación en tabla	11
2.2.3. Lógica combinacional	11
2.2.4. Circuito equivalente	12
2.2.5. Simulación	12
3. Detector de flancos ascendentes	12
3.1. Maquina de Moore	12
3.2. Maquina de Mealy	15
3.2.1. Diagrama de estados	15
3.2.2. Representación en tabla	15
3.2.3. Lógica combinacional	15
3.2.4. Circuito equivalente	16
3.2.5. Simulación	17
3.3. Control de voltajes externo-interno	17

Introducción

En el siguiente trabajo se pide la implementación de FSM tanto de Moore como de Mealy de diferentes problemas. El diagrama de bloques correspondiente a una FSM de Moore es la siguiente:

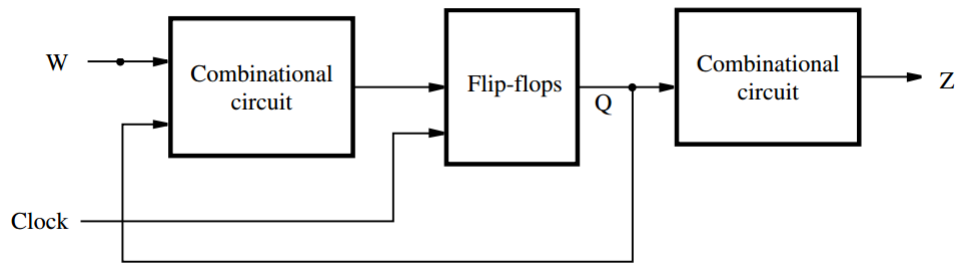


Figura 1: Diagrama de bloques - Maquina de Moore

Como se puede ver en el diagrama, se utilizan dos circuitos combinacionales conformado por compuertas lógicas. El primero de ellos convierte el o los inputs del sistema y el estado actual de la FSM en el nuevo estado de la FSM. Ese nuevo estado se guarda en Flip-Flops, la cual cada Flip-Flop se denomina variable de estado. Las cuales se necesitan los suficientes como para representar en binario la cantidad de estados que se tengan. El segundo circuito combinacional se utiliza para traducir el estado actual al o los outputs de la FSM en ese estado. La maquina de Mealy es muy similar, como se puede apreciar a continuación:

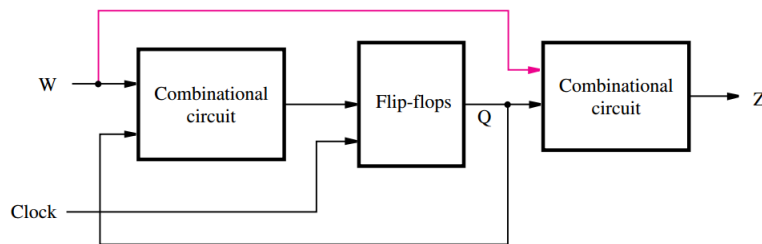


Figura 2: Diagrama de bloques - Maquina de Mealy

La única diferencia a la maquina de Moore, es que en el segundo circuito combinacional, además de el estado actual se tiene como input también a el input del circuito. De esta manera, según el problema, se puede reducir considerablemente el circuito combinacional y hasta reducir la cantidad de estados de la FSM.

Para lograr la maquina de estados deseada mediante la implementación de Mealy o de Moore se analizó el problema en etapas. En primer lugar se plantearon los estados necesarios, las transiciones de los mismos y las salidas correspondientes a cada transición mediante un grafo. A continuación, se volcaron los estados y transiciones obtenidos del grafo en una tabla. Finalmente, se utilizó dicha tabla para obtener la tabla de verdad correspondiente al bloque de entrada de lógica combinacional tanto como para el bloque de salida. Con dichas tablas se procedió a utilizar el método de minimización por mapas de Karnaugh para conseguir la solución menor costo del problema, teniendo en cuenta que en algunos casos es necesario agregar compuertas de manera que no se produzcan glitches.

1. Control de bombas de agua

1.1. Maquina de Moore

Para el control de una bomba de agua, dados los inputs I y S , y los outputs B_0 y B_1 , siendo los inputs los sensores y las outputs las bombas de agua a controlar, la FSM a implementar debe seguir la próxima tabla de verdad:

Input w		Output Z	
I	S	B_0	B_1
1	1	0	0
0	0	1	1
1	0	Alternan entre B_0 y B_1	
0	1	d	d

Cuadro 1: Tabla de verdad

Basándose en la anterior tabla de verdad se llegó a el siguiente diagrama de estados:

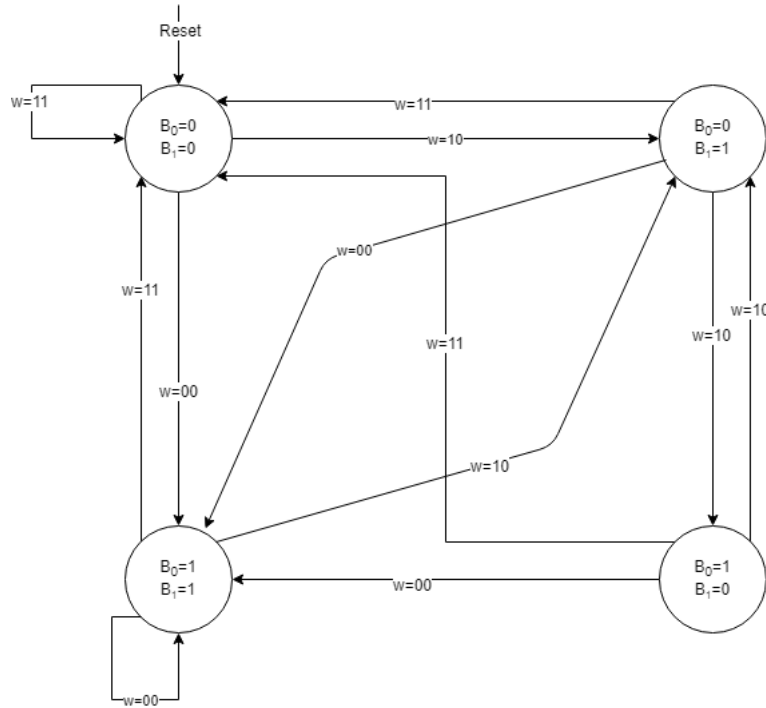


Figura 3: Diagrama de estados - Maquina de Moore

Este diagrama se puede ver reflejado en la siguiente tabla de estados:

Estado Actual		Próximo Estado				Output
Nombre	y_2y_1	$w = 11$	$w = 00$	$w = 10$	$w = 01$	B_0B_1
		Y_2Y_1				
A	00	00	11	01	d	00
B	01	00	11	10	d	01
C	10	00	11	01	d	10
D	11	00	11	01	d	11

Cuadro 2: Tabla de estados - Maquina de Moore

Se puede ver fácilmente que al elegir estas variables de estado, se tiene que:

$$B_1 = y_1$$

$$B_0 = y_2$$

De esta manera no se necesita circuito combinacional tras los Flip-Flops. Para el circuito combinacional de la entrada se resuelven los siguientes mapas de Karnaugh:

cd \ ab	00	01	11	10
00	1	1	1	1
01	d	d	d	d
11	0	0	0	0
10	1	0	1	1

Figura 4: Mapa Karnaugh Y_1

cd \ ab	00	01	11	10
00	1	1	1	1
01	d	d	d	d
11	0	0	0	0
10	0	1	0	0

Figura 5: Mapa Karnaugh Y_2

Lo cual se llega a las siguientes expresiones:

$$\begin{cases} Y_1 = \bar{I} + y_2 \bar{S} + \bar{y}_1 \cdot \bar{S} \\ Y_2 = \bar{I} + \bar{y}_2 \cdot y_1 \cdot \bar{S} \end{cases}$$

Ahora que se tienen todas las etapas se prosigue a diagramar el circuito:

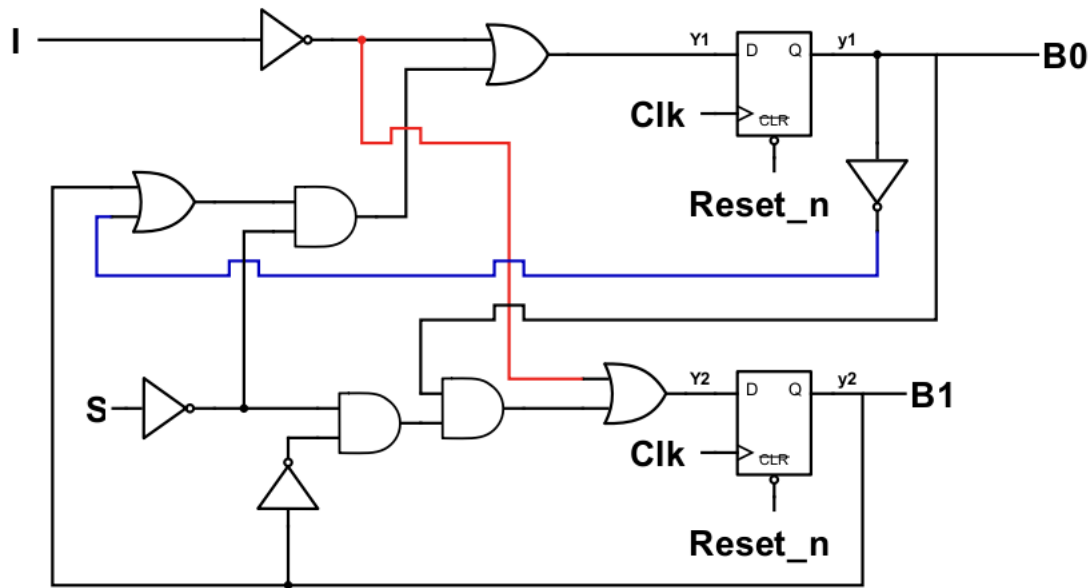


Figura 6: Circuito - Maquina de Moore

Y si se deseara implementarlo con compuertas NAND:

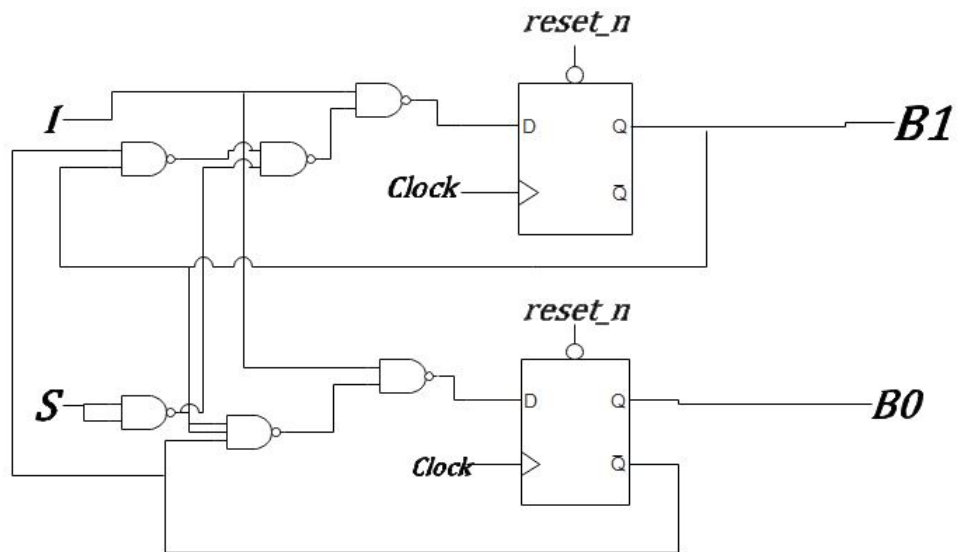


Figura 7: Circuito Compuertas NAND - Maquina de Moore

Para finalizar, se realizo la simulación en Verilog, adjunta en la carpeta, la cual dio el siguiente resultado en GTKwave:

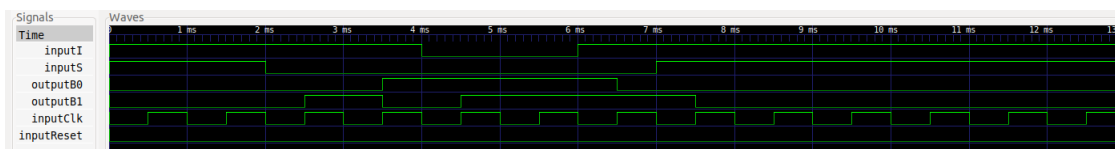


Figura 8: Simulación en GTKwave

1.2. Maquina de Mealy

1.2.1. Diagrama de estados

A continuación se presenta el diagrama de los estados, transiciones y salidas correspondientes a cada transición para la maquina de estados correspondiente al control de los niveles de agua:

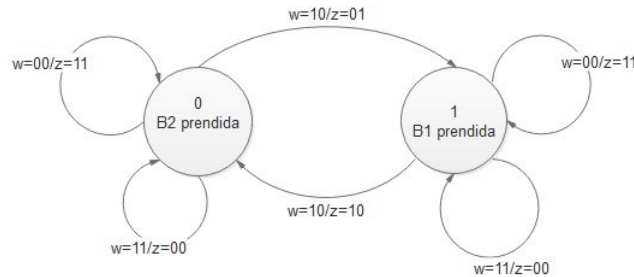


Figura 9: Diagramas de estados planteado

En el diagrama anterior la letra w simboliza el input del circuito mientras que z simboliza el output del mismo. Dichos símbolos fueron elegidos de la siguiente manera:

$$w = IS \text{ y } z = B2B1$$

1.2.2. Representación en tabla

Representando el grafo de la figura 9 en forma de tabla se llega a:

Estado actual	Próximo estado					z(output)				
y	Y					B2B1				
	w=	00	01	10	11	w=	00	01	10	11
0		d	d	1	d		11	dd	01	00
1		d	d	0	d		11	dd	10	00

Cuadro 3: FSM representada en tabla

1.2.3. Lógica combinacional

De la tabla anterior se puede llegar a una tabla de verdad correspondiente a las salidas de los bloques de lógica combinacional del circuito. La tabla de verdad correspondiente es:

I	S	y	Y	B2	B1
0	0	0	d	1	1
0	1	0	d	d	d
1	0	0	1	0	1
1	1	0	d	0	0
0	0	1	d	1	1
0	1	1	d	d	d
1	0	1	0	1	0
1	1	1	d	0	0

Cuadro 4: Tabla de verdad del circuito

Se procedió a simplificar cada uno de los bloques de lógica combinacional mediante el uso de mapas de Karnaugh. Se analizara por un lado la simplificación del bloque de entrada y por otro la del bloque de salida.

Simplificación del bloque de entrada Utilizando la tabla 4 se llega que el mapa de Karnaugh para la salida del primer bloque de lógica combinacional es:

Figura 10: Mapa de Karnaugh para la lógica de entrada

De los grupos elegidos se llega a la siguiente expresión:

$$Y = \bar{y}$$

Simplificación del bloque de salida Como el circuito tiene dos salidas, B_2 y B_1 , se planteo un mapa de Karnaugh para cada una. Dichos mapas son los siguientes:

Figura 11: Mapas de Karnaugh para la lógica de salida

De los grupos elegidos se llega a la siguiente expresión para B_2 :

$$B_2 = \bar{I} + y.\bar{S} = \overline{I.(\bar{y} + S)} = \overline{I.(y.\bar{S})}$$

En cuanto a B_1 se tiene:

$$B_1 = \bar{I} + \bar{S}.\bar{y} = \overline{I.(S + y)} = \overline{I.(\bar{S}.\bar{y})}$$

En ambos casos se busco escribir las salida en función de compuertas nands para que luego sea mas simple plantear el circuito equivalente unicamente en función de estas compuertas.

1.2.4. Circuito equivalente

Utilizando las expresiones simplificadas para el bloque de entrada como para el de salida, así como también el hecho de que solo se necesita un variable de estados para la maquina de estados, se puede llegar al circuito equivalente de la figura 9. Dicho circuito es el siguiente:

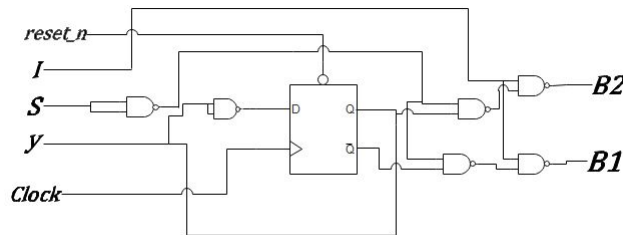


Figura 12:

Cabe aclarar que 'y' no es un input del circuito sino que el estado actual de la FSM. Se posiciono en el circuito de forma tal que la comprensión del mismo sea mas sencilla. Si se desea que la salida del circuito cambie con cada flanco positivo del clock debe colocarse un flip flop adicional a la salida del circuito.

1.2.5. Simulación

Para asegurar que el circuito equivalente funciona de manera deseada se realizó una simulación del mismo en Verilog simulando las mismas conexiones físicas que se presentan en la figura 12. Se elaboró por separado un módulo para cada bloque de lógica combinacional con su banco de prueba correspondiente para asegurar de que cumpla con su tabla de verdad así como también un módulo para toda la FSM. El código de cada módulo se encuentra en el repositorio del proyecto en la carpeta correspondiente al ejercicio dentro de la carpeta 'src'. A continuación se presenta una imagen de la simulación realizada en GtkWave de la implementación en Verilog de la FSM:

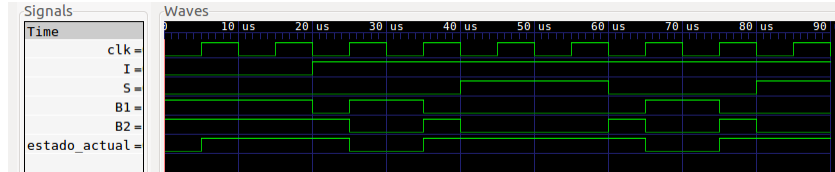


Figura 13: Simulación en GtkWave de la FSM

2. Detección de secuencia binaria

2.1. Maquina de Moore

Como se desea una FSM que sea capaz de detectar la secuencia 1 – 1 – 0 – 1, el input será un solo bit binario (w) y la salida representará si la secuencia se ha detectado (z). Tras detectar la secuencia, el bit queda prendido hasta que se haga un reset de la FSM. El diagrama de estados que corresponde a esta FSM es el siguiente:

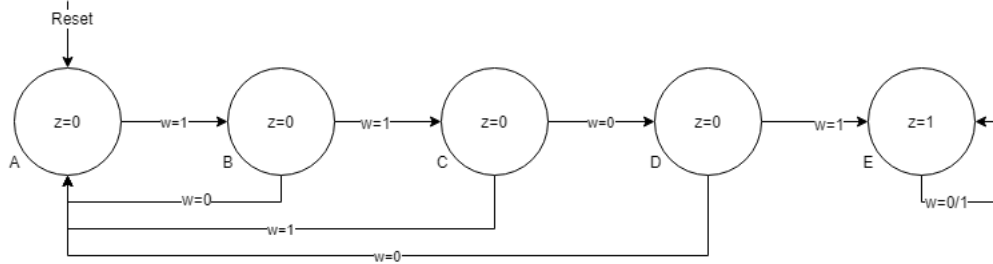


Figura 14: Diagrama de estados - Maquina de Moore

El anterior diagrama puede reflejarse en la siguiente tabla de estados:

Estado Actual		Próximo Estado		Output
Nombre	$y_3y_2y_1$	$w = 0$	$w = 1$	z
		$Y_3Y_2Y_1$		
A	000	000	001	0
B	001	000	010	0
C	010	011	000	0
D	011	000	100	0
E	100	100	100	1

Cuadro 5: Tabla de estados - Maquina de Moore

Es trivial llegar a la siguiente relación entre las variables de estado y el output:

$$z = y_3$$

Al igual que en la primer sección, no se necesitara un circuito combinacional a la salida de los Flip-Flops. Se resuelven los siguientes mapas de Karnaugh para encontrar el circuito combinacional a la entrada:

cd ab	00	01	11	10
	00	01	11	10
00	0	0	0	1
01	0	d	d	d
11	0	d	d	d
10	1	0	0	0

Figura 15: Mapa Karnaugh Y_1

cd ab	00	01	11	10
	00	01	11	10
00	0	0	0	1
01	0	d	d	d
11	0	d	d	d
10	0	1	0	0

Figura 16: Mapa Karnaugh Y_2

cd ab	00	01	11	10
	00	01	11	10
00	0	0	0	0
01	1	d	d	d
11	1	d	d	d
10	0	0	1	0

Figura 17: Mapa Karnaugh Y_3

Lo cual se llega a las siguientes expresiones:

$$\begin{cases} Y_1 = \overline{y_1} \cdot \overline{y_3} \cdot (y_2 \oplus w) \\ Y_2 = y_2 \cdot \overline{y_1} \cdot \overline{w} + \overline{y_2} \cdot y_1 \cdot w + y_1 \cdot y_3 + y_2 y_3 \\ Y_3 = y_3 + y_2 \cdot y_1 \cdot w \end{cases}$$

El circuito correspondiente entonces sera:

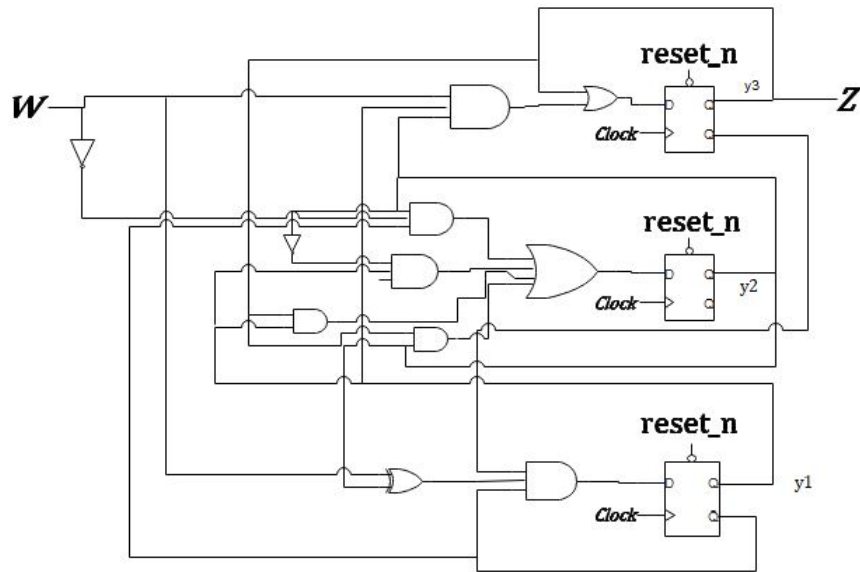


Figura 18: Circuito - Maquina de Moore

Para finalizar, se realizo la simulación en Verilog, adjunta en la carpeta, la cual dio el siguiente resultado en GTKwave:

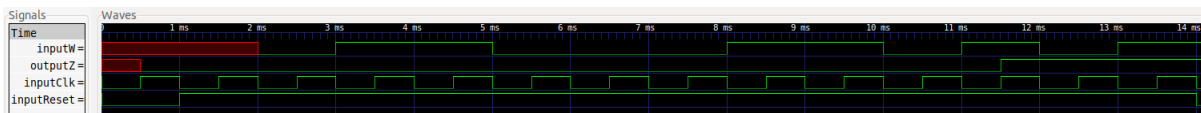


Figura 19: Simulación en GTKwave

2.2. Maquina de Mealy

2.2.1. Diagrama de estados

A continuación se presenta el diagrama de los estados, transiciones y salidas correspondientes a cada transición para la maquina de estados que detecta la secuencia de bits 1-1-0-1:

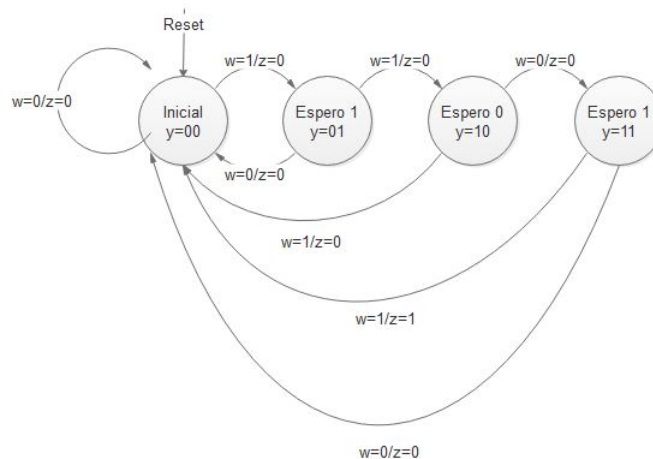


Figura 20: Diagramas de estados planteado

En el diagrama anterior la letra w simboliza el input del circuito mientras que z simboliza el output del mismo e 'y' simboliza el estado actual.

2.2.2. Representación en tabla

Representando el grafo de la figura 20 en forma de tabla se llega a:

Estado actual		Proximo estado		z(Output)	
y_1	y_2	w=0	w=1	w=0	w=1
0	0	00	01	0	0
0	1	00	10	0	0
1	0	11	00	0	0
1	1	00	00	0	1

Cuadro 6: FSM representada en tabla

2.2.3. Lógica combinacional

De la tabla anterior se puede llegar a una tabla de verdad correspondiente a las salidas de los bloques de lógica combinacional del circuito. La tabla de verdad correspondiente es:

y_1	y_2	w	Y_1	Y_2	z
0	0	0	0	0	0
0	1	0	0	0	0
1	0	0	1	1	0
1	1	0	0	0	0
0	0	1	0	1	0
0	1	1	1	0	0
1	0	1	0	0	0
1	1	1	0	0	1

Cuadro 7: Tabla de verdad del circuito

y_1 e y_2 son las variables de estado utilizadas, Y_1 e Y_2 son las salidas del primer bloque de lógica combinacional y simbolizan el próximo estado de la FSM.

Se procedió a simplificar cada uno de los bloques de lógica combinacional mediante el uso de mapas de Karnaugh. Se analizara por un lado la simplificación del bloque de entrada y por otro la del bloque de salida.

Simplificación del bloque de entrada Utilizando la tabla 7 se llega que el mapa de Karnaugh para la salida del primer bloque de lógica combinacional es:

Figura 21: Mapas de Karnaugh para la lógica de entrada

De los grupos elegidos se llega a la siguiente expresión para Y_1 :

$$Y_1 = w \cdot \overline{y_1} \cdot y_2 + \overline{w} \cdot y_1 \cdot \overline{y_2} = \overline{\overline{w \cdot \overline{y_1} \cdot y_2 + \overline{w} \cdot y_1 \cdot \overline{y_2}}}$$

En cuanto a Y_2 se tiene:

$$Y_2 = w \cdot \overline{y_1} \cdot \overline{y_2} + \overline{w} \cdot y_1 \cdot y_2 = \overline{\overline{w \cdot \overline{y_1} \cdot \overline{y_2} + \overline{w} \cdot y_1 \cdot y_2}}$$

Simplificación del bloque de salida Para el bloque de salida el mapa de Karnaugh es el siguiente:

Figura 22: Mapa de Karnaugh para la lógica de salida

De los grupos elegidos se llega a la siguiente expresión para z :

$$z = w \cdot y_1 \cdot y_2$$

2.2.4. Circuito equivalente

Utilizando las expresiones simplificadas para el bloque de entrada como para el de salida, y considerando que se necesitan dos variables de estado. Se puede llegar al circuito equivalente de la figura 20. Dicho circuito es el siguiente:

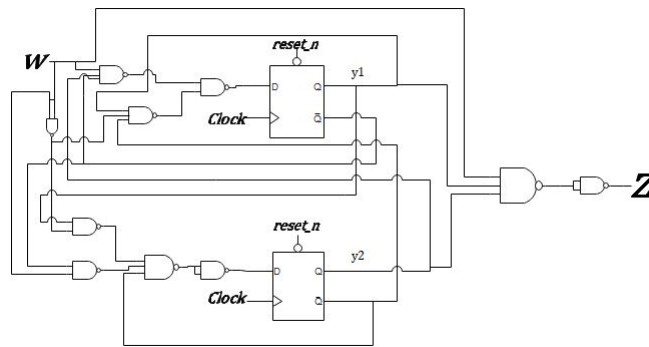


Figura 23: Circuito elaborado con nands

2.2.5. Simulación

A continuación se presenta una imagen de la simulación en GtkWave de la FSM implementada en Verilog utilizando unicamente compuertas nands siguiendo las conexiones presentadas en la figura 23:

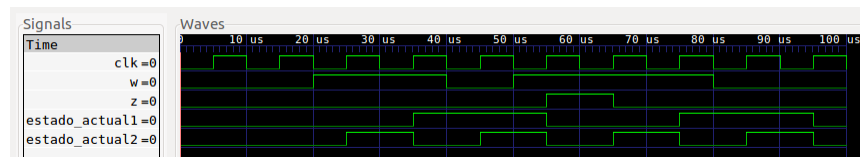


Figura 24: Simulación en GtkWave de la FSM

3. Detector de flancos ascendentes

3.1. Maquina de Moore

El diagrama de estados propuesto puede ser utilizado como un detector de flancos ascendentes. Donde solo se activa cuando el input cambia de 0 a 1, cambiando nuevamente a 0 el clock siguiente, sin importar que la entrada sea 0 o 1. El diagrama de estados es el que se puede encontrar en la consigna, y asociado a este se tiene la siguiente tabla de estados:

Estado Actual		Próximo Estado		Output
Nombre	y_2y_1	$w = 0$	$w = 1$	z
		Y_2Y_1		
A	00	00	01	0
B	01	00	10	1
C	10	00	10	0

Cuadro 8: Tabla de estados - Maquina de Moore

Como en los problemas anteriores, se eligieron las variables de estado de manera que el circuito combinacional se vea innecesario, de manera que el output z es:

$$z = y_1$$

Se resuelven los siguientes mapas de Karnaugh para encontrar el circuito combinacional a la entrada:

		bc			
		00	01	11	10
a	0	0	0	d	0
	1	0	1	d	1

Figura 25: Mapa Karnaugh Y_2

		bc			
		00	01	11	10
a	0	0	0	d	0
	1	1	0	d	0

Figura 26: Mapa Karnaugh Y_1

Lo cual se llega a las siguientes expresiones:

$$\begin{cases} Y_1 = w \cdot \overline{y_1} \cdot \overline{y_2} \\ Y_2 = w \cdot y_1 + w \cdot y_2 = w \cdot (y_1 + y_2) \end{cases}$$

El circuito correspondiente entonces sera:

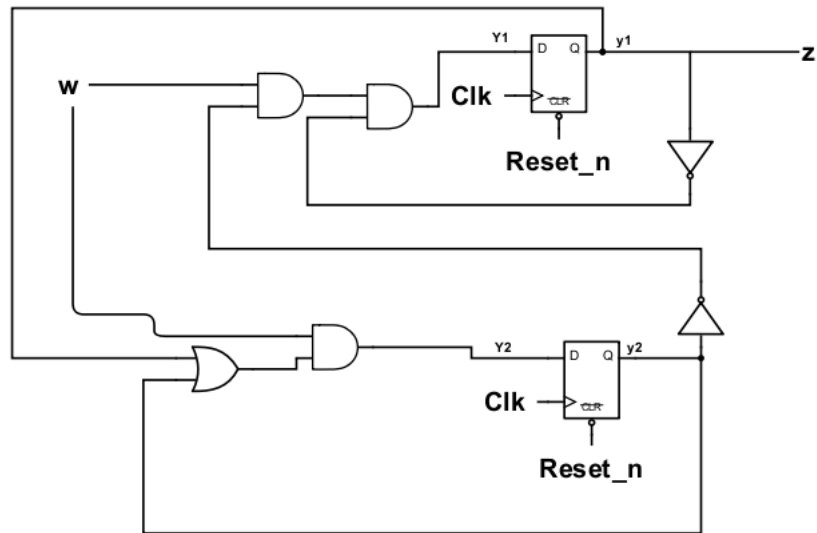


Figura 27: Circuito - Maquina de Moore

Y su equivalente en compuertas NAND:

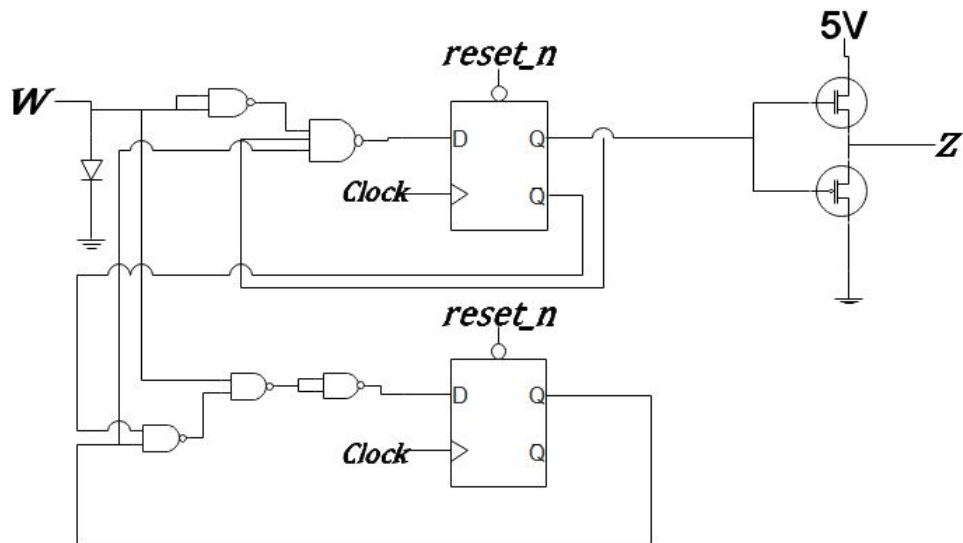


Figura 28: Circuito Compuertas NAND - Maquina de Moore

A continuacion se realizo la simulación en Verilog, adjunta en la carpeta, la cual dio el siguiente resultado en GTKwave:

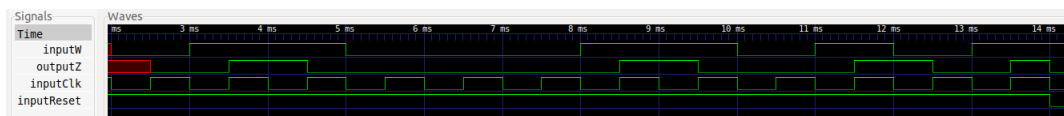


Figura 29: Simulación en GTKwave

Para finalizar, se realizo en placa este circuito, el esquematico es el siguiente:

Figura 30: Esquemático

Y se realizó físicamente:

Figura 31: Placa

3.2. Maquina de Mealy

3.2.1. Diagrama de estados

A continuación se presenta el diagrama de los estados, transiciones y salidas correspondientes a cada transición para la maquina de estados que detecta flancos ascendentes:

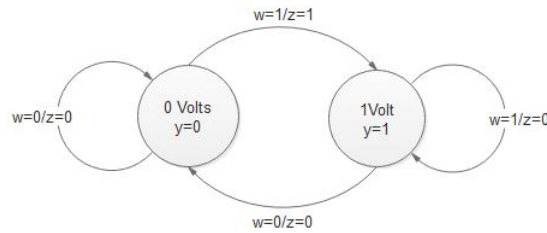


Figura 32: Diagramas de estados planteado

En el diagrama anterior la letra w simboliza el input del circuito mientras que z simboliza el output del mismo e 'y' simboliza el estado actual.

3.2.2. Representación en tabla

Representando el grafo de la figura 32 en forma de tabla se llega a:

Estado actual	Proximo estado		z(Output)	
	w=0	w=1	w=0	w=1
0	0	1	0	1
1	0	1	0	0

Cuadro 9: FSM representada en tabla

3.2.3. Lógica combinacional

De la tabla anterior se puede llegar a una tabla de verdad correspondiente a las salidas de los bloques de lógica combinacional del circuito. La tabla de verdad correspondiente es: El circuito tiene algunos elementos adicionales que podría no tener y seguir funcionando como detector de flancos.

w	y	z	Y
0	0	0	0
0	1	0	0
1	0	1	1
1	1	0	1

Cuadro 10: Tabla de verdad del circuito

Figura 36: Implementación en PCB

Donde 'Y' es el proximo estado de la FSM.

Se procedió a simplificar cada uno de los bloques de lógica combinacional mediante el uso de mapas de Karnaugh. Se analizara por un lado la simplificación del bloque de entrada y por otro la del bloque de salida.

Simplificación del bloque de entrada Utilizando la tabla 10 se llega que el mapa de Karnaugh para la salida del primer bloque de lógica combinacional es:

Figura 33: Mapas de Karnaugh para la lógica de entrada

Con el grupo elegido el bloque de entrada queda sencillo y no se necesita compuerta alguna a la entrada ya que se tiene que:

$$y = w$$

Simplificación del bloque de salida Para el bloque de salida el mapa de Karnaugh es el siguiente:

Figura 34: Mapa de Karnaugh para la lógica de salida

De los grupos elegidos se llega a la siguiente expresión para z:

$$z = w \cdot \bar{y}$$

3.2.4. Circuito equivalente

Utilizando las expresiones simplificadas para el bloque de entrada como para el de salida, y considerando que se necesitan dos variables de estado. Se puede llegar al circuito equivalente de la figura 32. Dicho circuito es el siguiente:

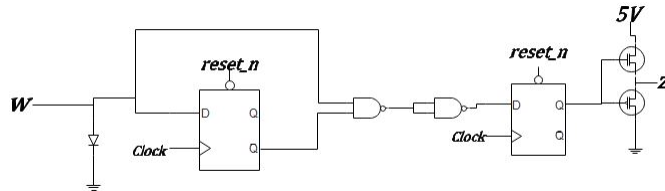


Figura 35: Circuito elaborado con nands

Finalmente, el flip-flop a la salida es para que la salida del circuito sea sincrónica. El circuito tiene algunos elementos adicionales que podría no tener y seguir funcionando como detector de flancos, esto se analiza en la sección 3.3.

A continuación se muestra una imagen de la versión final del circuito elaborada en placa:

3.2.5. Simulación

A continuación se presenta una imagen de la simulación en GtkWave de la FSM implementada en Verilog utilizando unicamente compuertas nands siguiendo las conexiones presentadas en la figura 35:

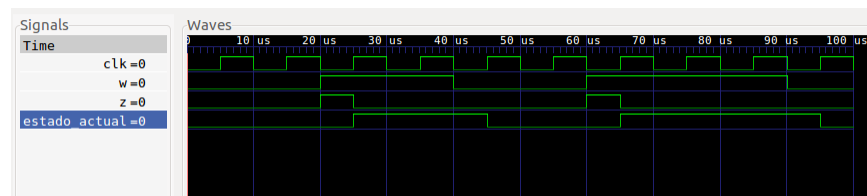


Figura 37: Simulación en GtkWave de la FSM

3.3. Control de voltajes externo-interno

El diodo a la entrada es para llevar la entrada de 5V a 3.3V, en la practica se utilizó un diodo zenner de 3.3V en inversa. Los transistores mosfet a la salida son para que la salida del circuito sea de 5V como se pide, ya que la lógica interna del circuito funciona con 3.3V.