

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.

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:

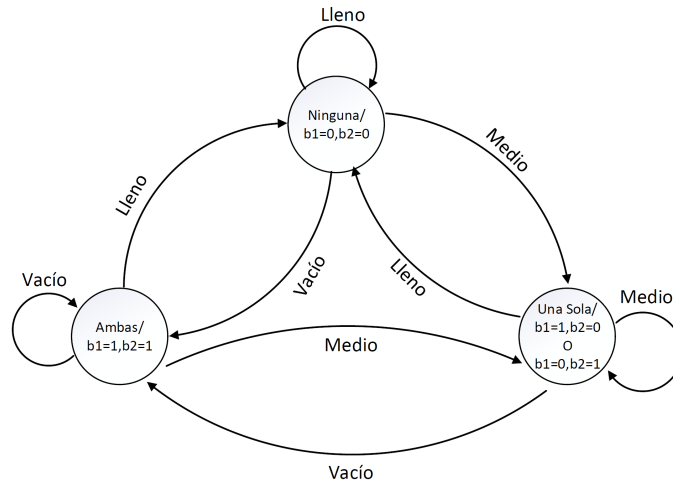


Figura 1: Diagrama de estados y transiciones

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	$y_1 y_0$
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 ($y_1 y_0$)	Próximo Estado($Y_1 Y_0$)			Salida ($b_1 b_2$)
	$SI = 00$	$SI = 01$	$SI = 11$	
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 ($Y_1 Y_0$) en función del estado anterior ($y_1 y_0$) y las entradas (SI).

$y_1 y_0 SI$	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

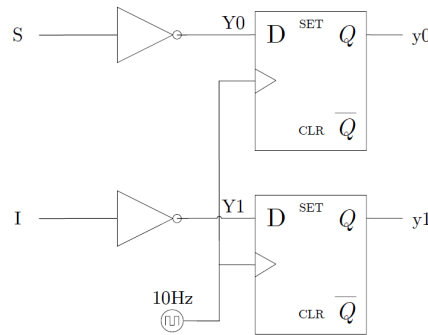


Figura 2: Lógica de transición de estados

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.

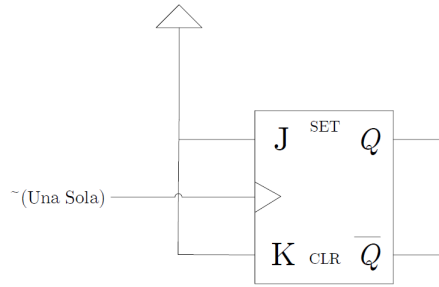


Figura 3: Flip-Flop JK para alternar salidas

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

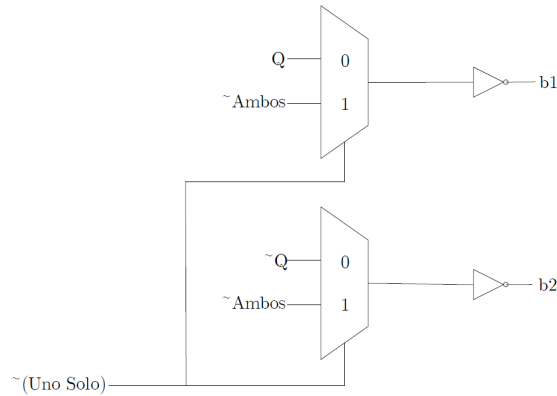


Figura 4: Multiplexación a la salida

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 $Ambas=0$

Simulación Verilog

La Figura 5 muestra el diagrama temporal generado por el código de Verilog presentado en el Anexo.

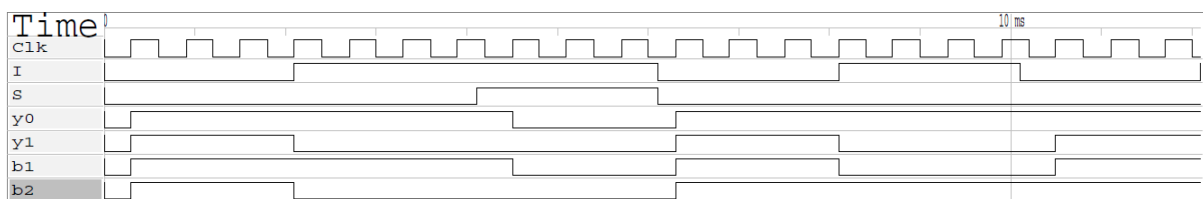


Figura 5: Diagrama temporal de la simulación en Verilog

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