

INSTITUTO TECNOLÓGICO DE BUENOS AIRES

# Trabajo Práctico N°3

## Electrónica III - 2019

Grupo 1:

Farall, Facundo David  
Gaytan, Joaquín Oscar  
Kammann, Lucas Agustín  
Maselli, Carlos Javier

Profesores:

Dewald, Kevin  
Wundes, Pablo  
Aguirre, Miguel

28 de octubre de 2019

# Índice

<b>Ejercicio 1: Control de bombas de agua</b>	<b>3</b>
<b>Ejercicio 2: Reconocimiento de secuencia de bits</b>	<b>4</b>
Diseño de Máquina de Estados . . . . .	4
Simulaciones en Verilog . . . . .	6
Diseño en PCB . . . . .	7
Resultados . . . . .	7
Conclusiones . . . . .	7
<b>Ejercicio 3: Máquina de Moore</b>	<b>8</b>

## Ejercicio 1: Control de bombas de agua

## Ejercicio 2: Reconocimiento de secuencia de bits

Se desea diseñar una máquina de estados implementada con máquina de Mealy, la cual sea capaz de analizar una secuencia de bits y detectar si se produjo un patrón seguido por 1-1-0-1, ante lo cual deberá notificar tal suceso activando su salida para ello. En la Fig. 1 se ilustra un esquema general de ello.

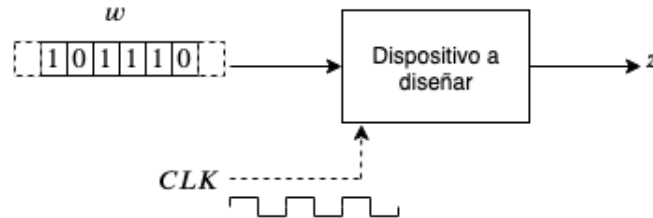


Figura 1: Esquema general del dispositivo a diseñar

### Diseño de Máquina de Estados

En primer lugar, dadas las especificaciones de la máquina de estado, se desea diseñar tal dispositivo el cual consta de una única entrada y una única salida. Para lo cual se emplea un esquema genérico de máquina de estados, en donde la salida será asíncronica pues se busca utilizar el diseño de Mealy para tal lógica. Este esquema general descrito puede visualizarse en la Fig. 2.

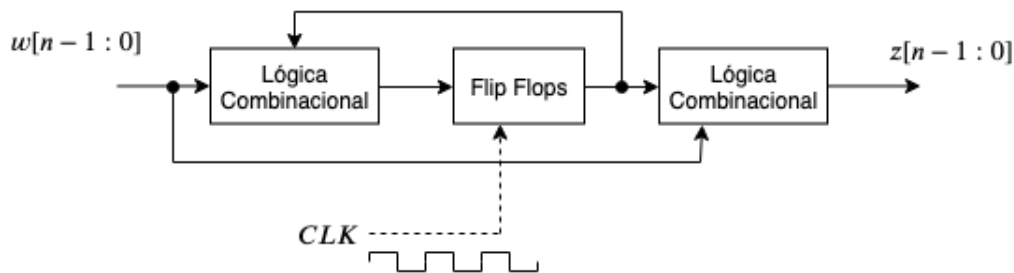


Figura 2: Esquema general de la máquina de Mealy

En la Fig. 3 se puede observar el diagrama de estados propuesto. Es importante mencionar que el estado de Reset es definido como tal para reconocer cuál es el estado inicial de la máquina, y deberá ser tenido en cuenta durante la asignación de estados en caso de proveer la posibilidad de reiniciar la máquina, pues los flip flops deberán ser llevados a dicho estado, según sea asignado.

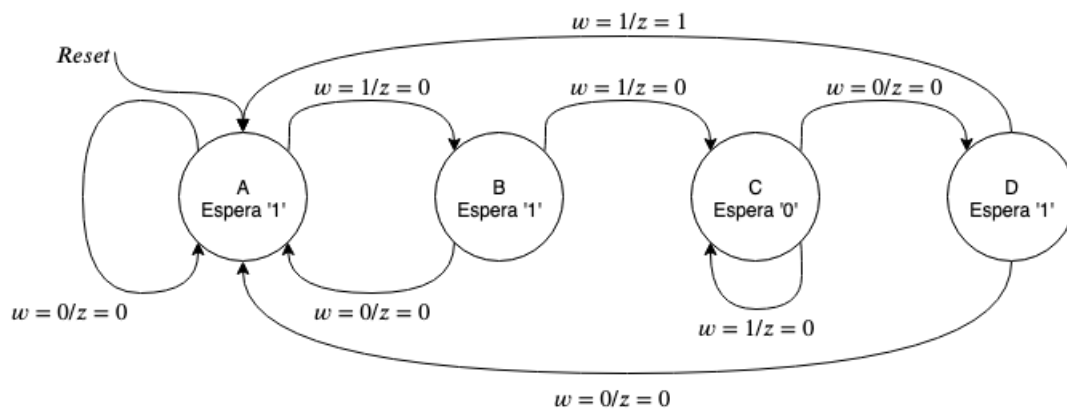


Figura 3: Diagrama de estados

En la Tabla 1 se puede observar la tabla de estados o transiciones de la máquina de estados, habiendo ya asignado correspondientemente a cada estado una configuración de bits. Es de interés mencionar que tal asignación es el resultado de comparar las diferentes alternativas y encontrar que, dada la distribución propuesta, la lógica externa es la mínima necesaria.

Estado $y_2y_1$	Próximo		Salida	
	$w = 0$	$w = 1$	$w = 0$	$w = 1$
$A = 11$	$A = 11$	$B = 00$	$z = 0$	$z = 0$
$B = 00$	$A = 11$	$C = 01$	$z = 0$	$z = 0$
$C = 01$	$D = 10$	$C = 01$	$z = 0$	$z = 0$
$D = 10$	$A = 11$	$A = 11$	$z = 0$	$z = 1$

Tabla 1: Tabla de estados o transiciones

		$y_2y_1$			
		00	01	11	10
$w$	0	0	0	0	0
	1	0	0	0	1

Figura 4: Karnaugh para la variable de salida

$$z = w \cdot y_2 \cdot \overline{y_1} \quad (1)$$

		$y_2y_1$			
		00	01	11	10
$w$	0	1	1	1	1
	1	0	0	0	1

Figura 5: Karnaugh para la variable de estado  $y_2$

$$y_2 = \overline{w} + y_2 \cdot \overline{y_1} \quad (2)$$

		$y_2y_1$			
		00	01	11	10
$w$	0	1	0	1	1
	1	1	1	0	1

Figura 6: Karnaugh para la variable de estado  $y_1$

$$y_1 = \overline{y_1} + \overline{w} \cdot y_2 + w \cdot \overline{y_2} \quad (3)$$

## Simulaciones en Verilog

En la Fig. 7 se puede observar el circuito lógico completo correspondiente a la máquina de estados diseñada en el apartado anterior. Se desea comprobar su correcto funcionamiento a nivel lógico mediante una simulación en Verilog, para lo cual se emplean dos metodologías de diseño.

Por un lado, puede utilizarse un diseño en Verilog que determine si la máquina está bien diagramada, utilizando para ello el bloque procedural `case`. Por otro lado, para determinar si la implementación lógica de la máquina puede funcionar, debe emplearse un diseño a nivel compuertas de los módulos en Verilog, para esto último se divide el problema inicialmente en tres bloques, los flip flops, la lógica combinacional que produce el próximo estado y la lógica de salida. Finalmente, un cuarto bloque o módulo describe la máquina interconectando los módulos mencionados para producir el comportamiento esperado.

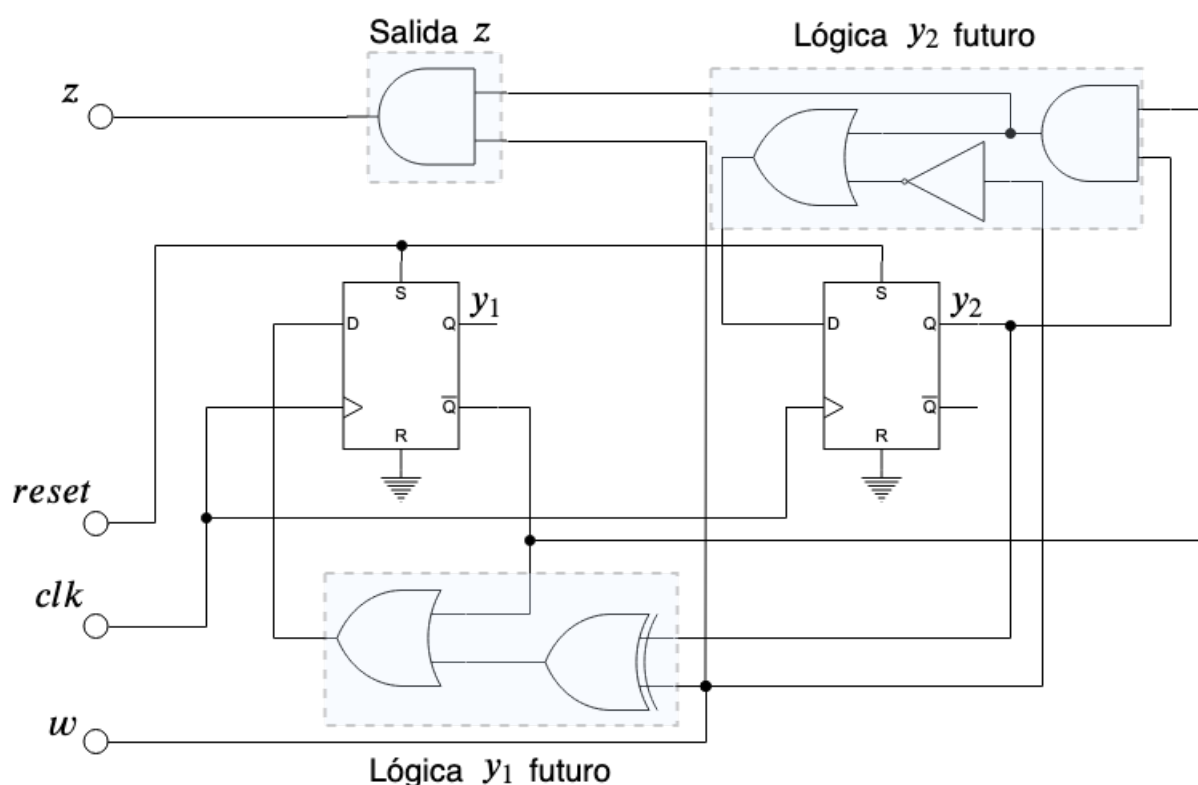


Figura 7: Circuito lógico completo de la máquina de estados

En la Fig. 8 se muestra a modo de referencia la simulación realizada y visualizada con GTKWave. Para determinar los casos de pruebas de la máquina, se partió del diagrama de estados y se consideraron diferentes secuencias. En primer lugar la secuencia 0-1-0 para determinar si permanece correctamente en el primer estado, pasa al segundo y vuelve al detectar el error. En segundo lugar, la secuencia 1-1-1-0-0, para determinar si llega correctamente al tercer estado, permanece y luego transiciona reiniciando la máquina pero con la salida en estado bajo. Finalmente, la secuencia correcta para analizar si la salida responde como es esperado.

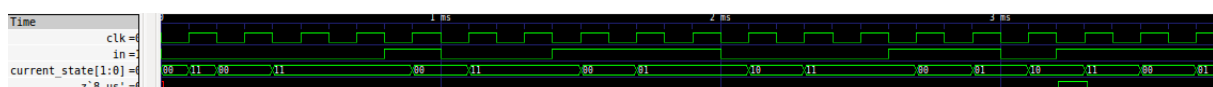


Figura 8: Simulación de Verilog visualizada con GTKWave

**Diseño en PCB**

**Resultados**

**Conclusiones**

### Ejercicio 3: Máquina de Moore