Instituto Tecnologico de Buenos Aires

Ingeniera Electronica

Electronica III

Implementación de circuitos logicos

Autores:
Martín Rodriguez Turco
Tobias Scala
Guido Panaggio
Juan Martin Laguinge

Profesores:
Kevin DeWald
Pablo Wundes
Sebastian Falconaro

5 de noviembre de 2018

Índice general

Capítulo 1

Ejercicio 2

Se desea diseñar una máquina de estados que, al recibir la siguiente secuencia de bits en forma sincrónica 1-1-0-1 encienda una salida y en caso contrario, la mantiene apagada. Se obtienen 5 estados para la misma, en los cuales va a haber un default que va a ser el estado al cual todos los demás estados van a volver en caso de no recibir los deseados además de ser el estado por el cual va a empezar la máquina de estados.
Podemos representar los mismos en el siguiente diagrama de estados:
En donde Z es la salida dada por la máquina de estados al encontrarse en el estado co-

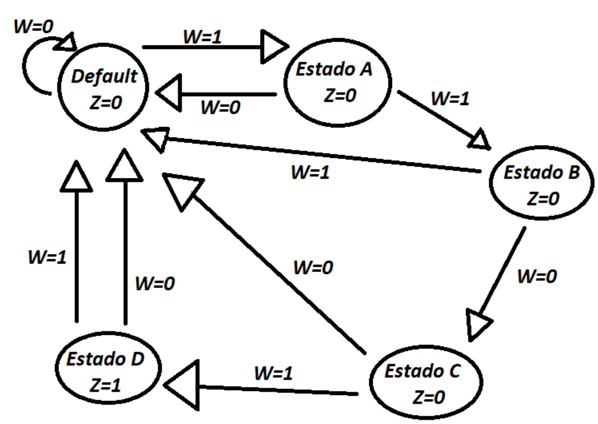

Figura 1.1: Diagrama de estados

rrespondiente y W es la entrada necesaria para que traicione al siguiente estado y la flecha es la encargada de indicar el sentido de la transición.
Este mismo esquema también queda encapsulado en la siguiente tabla de estados:

Cuadro 1.1: Tabla de estados

Estado actual	Estado siguiente **W=0**	**W=1**	**Salida Z**
Default	**Default**	**A**	0
A	**Default**	**B**	0
B	**C**	**Default**	0
C	**Default**	**D**	0
D	**Default**	**Default**	1

Para la implementación de esté falta realizar la asignación de valores de estado, lo cual nos lleva cambiar la tabla anterior por la siguiente:

Cuadro 1.2: Tabla de estados asignados

Estado actual	**Asignacion del Estado actual**	Estado siguiente **W=0**	**W=1**	**Salida Z**
Default	000	000	001	0
A	001	000	010	0
B	010	011	000	0
C	011	000	100	0
D	100	000	000	1

1.1. Máquina de Moore

Para nuestro caso tenemos el siguiente circuito secuencial genérico:

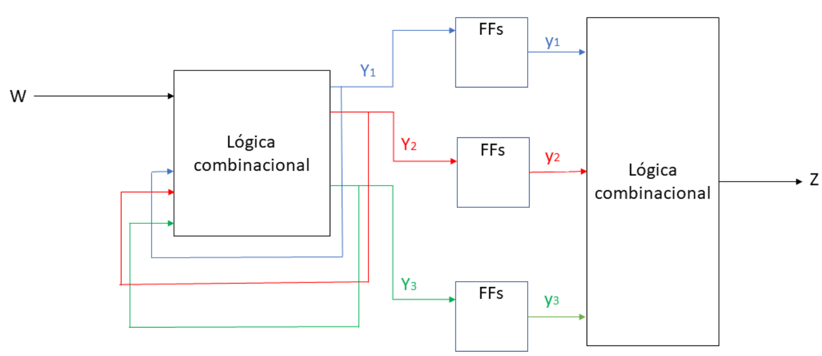

Figura 1.2: Circuito generico

Donde vamos a utilizar Flip-Flops D dado que la entrada D de estos va a corresponder con el estado siguiente Y_i y van a estar seteados por el clock para que esta salida luego cambia la variable y_i a Y_i, dado que y_i son las variables de estado actual.De la tabla 1, obtenemos los siguientes mapas de Karnaugh:

En donde las X representan el don't care y se les decidió dar un valor acorde al cual permiten la simplificación del circuito. Dando como resultado las siguientes ecuaciones:

$$Y_1 = W \cdot \overline{y_3} \cdot \overline{y_2} \cdot \overline{y_1} + \overline{W} \cdot \overline{y_1} \cdot y_2$$
$$Y_2 = W \cdot \overline{y_2} \cdot y_1 + \overline{W} \cdot \overline{y_1} \cdot y_2$$
$$Y_3 = W \cdot y_2 \cdot y_1$$
$$Z = y_3$$

1.2. Máquina de Mealy

Para poder realizar la máquina de estados con el modelo de Mealy es necesario que la salida dependa tanto de los estados como de la entrada de esta, con lo cual van a ser necesarios realizar cambios a la actual máquina de estados.
Quitando el último estado del diagrama 1 y expresando la salida junto con la entrada podemos obtener el siguiente diagrama:

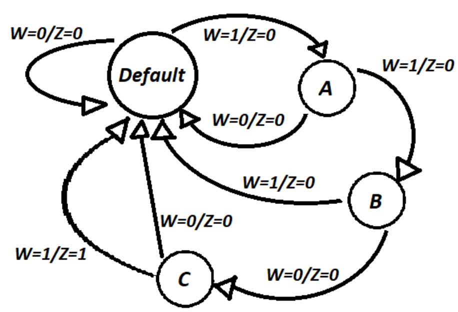

Figura 1.3: Diagrama de estados

El diagrama tiene un estado menos ahorra, esto provoca los siguientes cambio en la tabla de asignación:

La cual nos permite obtener los siguientes mapas de Karnaugh:

Dando como resultado de las simplificaciones las siguientes ecuaciones:

$$Y_1 = W \cdot \overline{y_2} \cdot \overline{y_1} + \overline{W} \cdot \overline{y_1} \cdot y_2$$
$$Y_2 = W \cdot \overline{y_2} \cdot y_1 + \overline{W} \cdot \overline{y_1} \cdot y_2$$
$$Z = W \cdot y_2 \cdot y_1$$

Appendix

Bibliografía

[1] Stephen Brown and Zvonko Vranesic. ”Fundamentals of Digital Logic with Verilog Design”third edition.