

# Maquina de estados Finitos FSM

David Ricardo Martínez Hernández Código: 261931

**Resumen**—Se realizo un circuito que simulara en una FPGA una maquina expendedora de productos, teniendo como teoría el funcionamiento de una maquina de estados, diseñando en Xilinx un programa que cumpla con las condiciones previamente establecidas en el diagrama de estados, tomado en cuenta el cambio de estado del sistema y las posibles condiciones que se puedan presentar, este programa debe entregar un producto y su respectivo cambio, teniendo en cuenta que se manejan diferentes valores de monedas (Asumiendo que las monedas son el dinero que se ingresa a la maquina representado por pulsadores). La salida de estos pulsadores esta determinada por los displays de la FPGA, en ellos se muestran la cantidad de dinero acumulado, el tipo de producto que se quiera obtener y el cambio, además al entregar un producto se debe prender un display que informe sobre la entrega del mismo.

**Palabras clave**—Diagrama de Estados, Estado, Estado Presente, Estado Siguiente, Flanco de Bajada, Flanco de Subida, Flip-Flop, Maquina de Estado, Señal Sincrónica.

## I. OBJETIVOS

- Comprender el funcionamiento, la importancia y la implementación de una maquina de estados finitos (FMS) para controlar un preceso determinado.
- Diseñar una maquina de estados capaz de controlar una maquina expendedora de productos, capaz de recibir un número determinado de monedas con diferentes opciones de productos dando el cambio de dinero si es necesario.
- Comprender y determinar la diferencia entre una **maquina de Mealy** y una **maquina de Moore** y sus diferentes aplicaciones.

## II. INTRODUCCIÓN

Un circuito combinacional está formado por una etapa de lógica combinacional y una etapa de memoria (flip-Flops). Los latch y flip-flops son bloques de construcción básicos de los circuitos secuenciales, pero por si solos son circuitos secuenciales de retroalimentación que se pueden analizar formalmente.

Una maquina de estado es un nombre genérico dado a estos circuitos secuenciales, "*temporizado*" hace referencia al hecho de que sus elementos de almacenamiento (flip-flops) emplean una entrada de reloj y "*sincrónica*" significa que todos los flip-flops utilizan una misma señal de reloj. Una maquina de estado cambia única y exclusivamente cuando es detecta un flanco de disparo o "*pulso*" en la señal de reloj.

No todos los circuitos secuenciales tienen variables de entrada y de salida, sin embargo todos tienen variables de excitación y variables de estado. La memoria de estado es un conjunto de  $n$  flip-flops que almacenan el estado actual de la maquina y tiene  $2^n$  estados distintos. El estado siguiente de la maquina de estado fig. 1 esta determinado por la "*lógica de estado siguiente, F*" como una función de la entrada y estado actuales.

La "*lógica de salida G*" determina la salida como una función de la entrada y el estado actual.<sup>1</sup>

Un circuito secuencial cuya salida depende tanto del estado como de la entrada fig. 1 se conoce como una **Maquina de Mealy**.

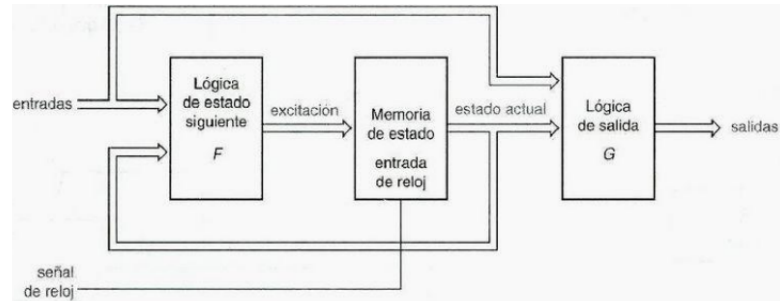


Fig. 1: Maquina de Mealy [3]

La **Maquina de Moore** la salida depende únicamente del estado actual fig. 2.

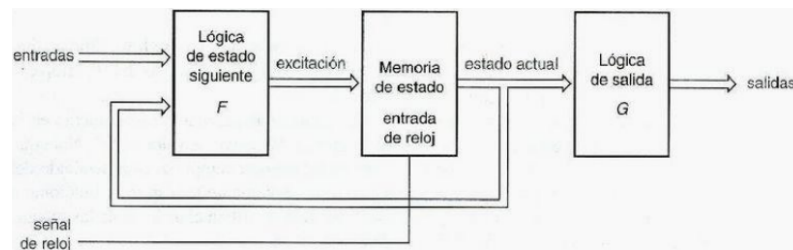


Fig. 2: Maquina de Moore [3]

Existe un procedimiento de diseño general para los circuitos secuenciales a los contadores sincrónicos:

- 1) Diagrama de estados.
- 2) Tabla del estado siguiente.
- 3) Tabla de transiciones de los Flip-Flops.
- 4) Mapas de Karnaugh.
- 5) Expresiones Lógicas para las entradas de los Flip-Flops.
- 6) Implementación del contador.

### A. Pasos para Maquina de estados

1) **Diagrama de Estados:** Presenta la información de los pasos anteriores y siguientes cuando se aplica una señal de reloj fig. 3.

<sup>1</sup>Texto tomado de [3], Págs 550 y 551, [4], Págs 573 y 574

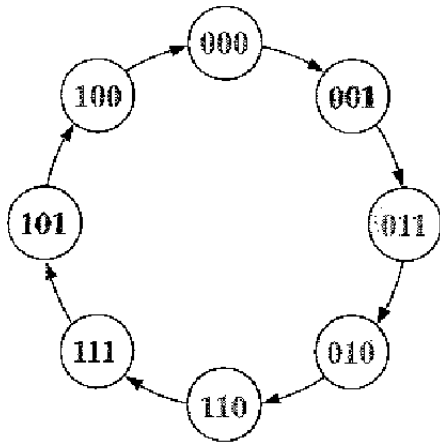


Fig. 3: Diagrama de Estados para un contador en código Gray, [4] pág 574

2) *Tabla del estado siguiente*: Una tabla de estado siguiente enumera cada estado del contador (estado actual) y cada estado siguiente, "el estado siguiente es el estado al que pasa el contador cuando se aplica un impulso de reloj. En la TAB. I se encuentra la tabla que define estos estados.

$Q_2$	Estado Actual $Q_1$	$Q_0$	$Q_2$	Estado Siguiente $Q_1$	$Q_0$
0	0	0	0	0	1
0	0	1	0	0	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

TABLA I: Tabla del estado siguiente para el código Gray

3) *Tabla de transiciones de los Flip-Flops*: Se enumeran todas las posibles transiciones de salida, mostrando como evoluciona la salida  $Q$  del flip-flop al pasar de los asados actuales a los estados siguiente.

4) *Mapas de Karnaugh*: Los mapas de Karnaugh se utiliza para determinar la lógica requerida para las entradas de cada flip-flop del contador. A partir de los estados de las entradas de la tabla de transiciones se introduce un 1, un 0 o una X en cada celda de la tabla correspondiente al estado actual, dependiendo de la transición de salida de cada flip-flop.

5) *Expresiones Lógicas para las entradas de los Flip-Flops*: A partir de los mapas de Karnaugh se obtienen las expresiones lógicas para las entradas de cada flip-flop.

6) *Implementación del contador*: Consiste en implementar la lógica combinacional a partir de las expresiones para las entradas de cada flip-flop, conectados entre si para crear el contador.

### III. MATERIALES Y MÉTODOS

- Computador.
- FPGA.
- Software Xilinx, Inc.

### IV. ANÁLISIS Y RESULTADOS

#### A. Maquinas de estados finitos 1 Producto y 2 Monedas

Se diseñó un programa en VHDL que cumpliera el siguiente Diagrama de estados fig. 4.

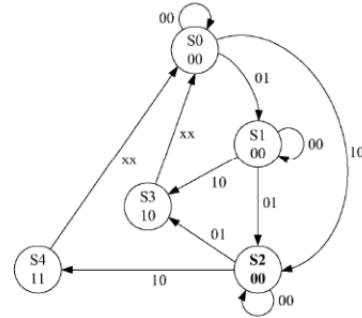


Fig. 4: Diagrama de Estados Maquina Expendidora

Una parte del código que se utilizó para realizar el laboratorio fue

```

case (estado)
3'b000:
begin
if (u==1'b0 && d==1'b0)
begin
nestado=0;
producto=4'b0000;
monedas=2'b00;
cambio=1'b0;
end
else if (u==1'b1 && d==1'b0)
begin
nestado=1;
monedas=monedas+2'b01;
producto=4'b0000;
cambio=1'b0;
end
else if (d==1'b1 && u==1'b0)
begin
nestado=2;
monedas=monedas+2'b10;
producto=4'b0000;
cambio=1'b0;
end
end
end

```

Esta parte del código es el estado  $S_0$ , en este estado se puede ir a 3 posibles estados de acuerdo a lo que se le ingrese a la maquina en un principio, puede ser una moneda de \$1 (u) y \$2 (d).

No se pudo cumplir la práctica por un problema en la visualización de los displays, no hacía bien la multiplexación e los mismos, aunque si se

#### B. Maquinas de estados finitos 2 Productos y 3 Monedas

Se diseñó un programa en VHDL que cumpliera el siguiente Diagrama de estados fig. 5.

Una parte del código que se utilizó para realizar el laboratorio fue

```
s10:begin
if (u&p)nst=s13;
else if (u&!p)nst=s14;
else if (d&!p)nst=s9;
else if (d&p)nst=s8;
else if (c&!p)nst=s19;
else if (c&p)nst=s18;
end
```

Esta parte del código es sobre el estado  $S_{10}$ , en este estado es particular se puede elegir entre dos productos, es decir se puede tomar la decisión de ir por el producto mas caro que es de \$12 o se puede devolver al producto de \$9 devolviendo el cambio correspondiente, al igual que en el estado  $S_{11}$ .

Al momento de implementar el código en el FPGA se obtuvieron los datos esperados, de acuerdo al diagrama de estados.

## REFERENCIAS

- [1] Dorf Svoboda. "Circuitos Eléctricos". Alfaomega, 2006.
- [2] C. J. Savant. "Diseños Electrónicos: Circuitos de Sistema". Prentice-Hall, 2006.
- [3] John F. Wakerly. "Diseño Digital: Principios y Prácticas". Prentice-Hall, 2001.
- [4] Thomas L. Floyd. "Fundamentos de Sistemas Digitales". Prentice-Hall, 2000.
- [5] Sitio Web: <http://www.virtual.unal.edu.co/cursos/ingenieria/2000477/lecciones/070101.htm>

## V. CONCLUSIONES

- Se comprendió el funcionamiento de una maquina de estados finitos, comprobándolo con los resultados obtenidos en el laboratorio sobre esta práctica.
- Se implemento la maquina expendedora utilizando la teoría sobre maquina de estado finitos, realizando algunos de los pasos propuestos en este informe.

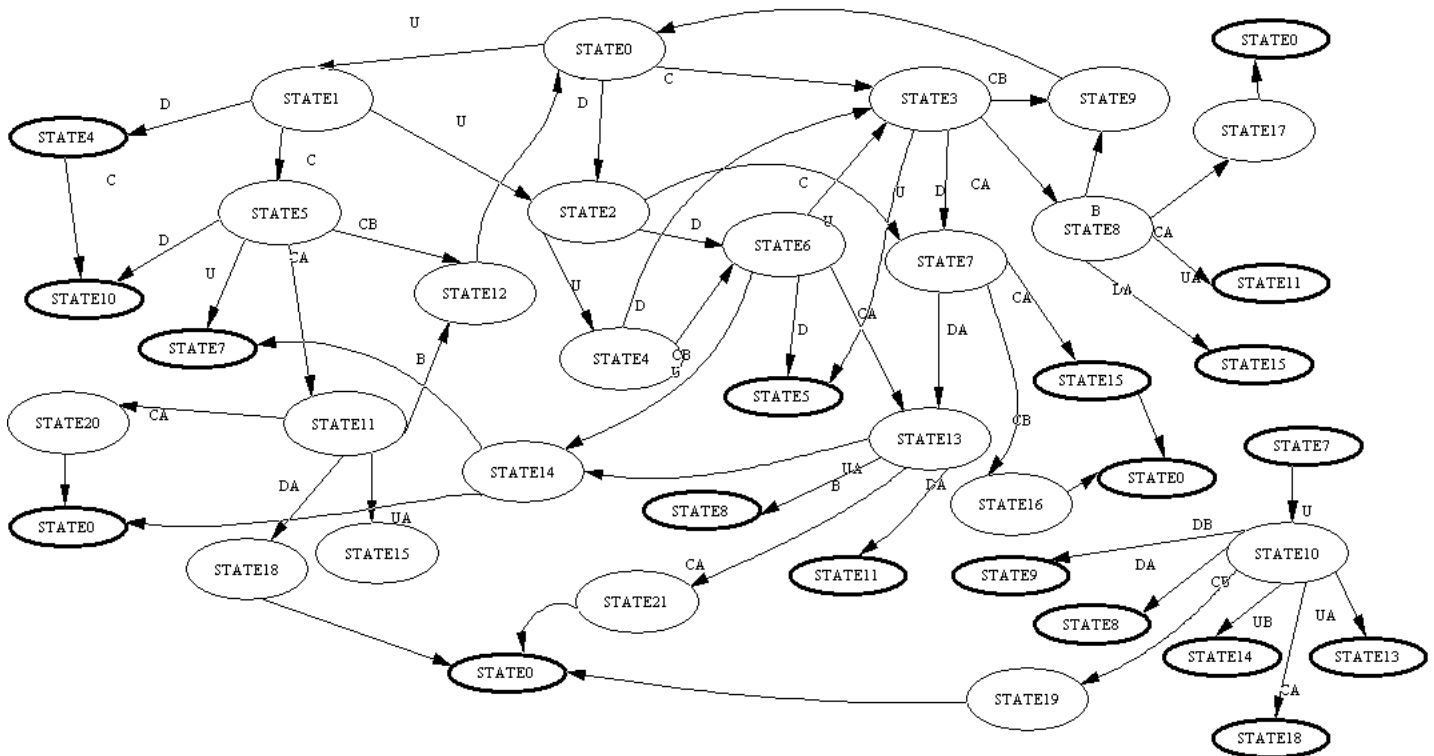


Fig. 5: Diagrama de Estados maquina de dos Productos