Objetivo

Familiarizar al alumno en el conocimiento de los algoritmos de las máquinas de estados utilizando Quartus y el lenguaje VHDL.

Introducción

Se denomina máquina de estados a un modelo de comportamiento de un sistema con entradas y salidas en donde las salidas dependen no solo de las señales de entradas actuales, sino también de las anteriores. Las máquinas de estado pueden ser:

- Síncronas: Necesitan de la intervención de un pulso de reloj. Si la entrada participa también en la salida se denomina Máquina de estado de Mealy, y si no participa se denomina de Moore.
- Asíncronas: No necesitan de la intervención de un pulso de reloj. Estos circuitos evolucionan cuando cambian las entradas.

Una máquina de estado finita o FSM representa un sistema como un conjunto de estados, transiciones entre estos estados, que dependen de las entradas, conjuntamente con las salidas y las entradas asociadas. De modo tal que una máquina de estado es una representación, de un circuito secuencial particular.

El diseño de una FSM involucra lo siguiente:

- Definición de estados.
- Definición de transición entre estados (dependientes de las entradas de la máquina).
- Optimización/minimización.

Una de las características de VHDL es que es posible describir circuitos digitales a diferentes niveles de abstracción. Esto quiere decir que podemos conectar componentes para crear el circuito que representa la máquina de estados, o que podemos simplemente describir su funcionamiento usando el lenguaje. La principal ventaja de usar una descripción funcional es que será la herramienta de síntesis que utilicemos la que a partir de esta descripción generará el circuito digital resultante, ahorrando muchísimo tiempo al diseñador. A la hora de diseñar una FSM en VHDL, se puede utilizando dos procesos o uno solo.

Si utilizamos dos procesos, las salidas se asignarán desde el proceso combinacional, por tanto, tendrán oscilaciones y el camino crítico de la FSM se unirá al camino crítico del módulo al que esté conectada la salida de la máquina. Si necesitamos evitar esto, tendremos que registrar las salidas de la FSM.

Si usamos una estructura con un solo proceso, las salidas estarán registradas, es decir se almacenarán en registros, esto ocupará más lógica, pero las señales serán estables y los caminos críticos estarán separados. Por contra, la salida llevará un ciclo de reloj de retraso frente a una máquina descrita en dos procesos.

Desarrollo

Como primer punto, realizamos el circuito secuencial de la carta ASM que se muestra a continuación:

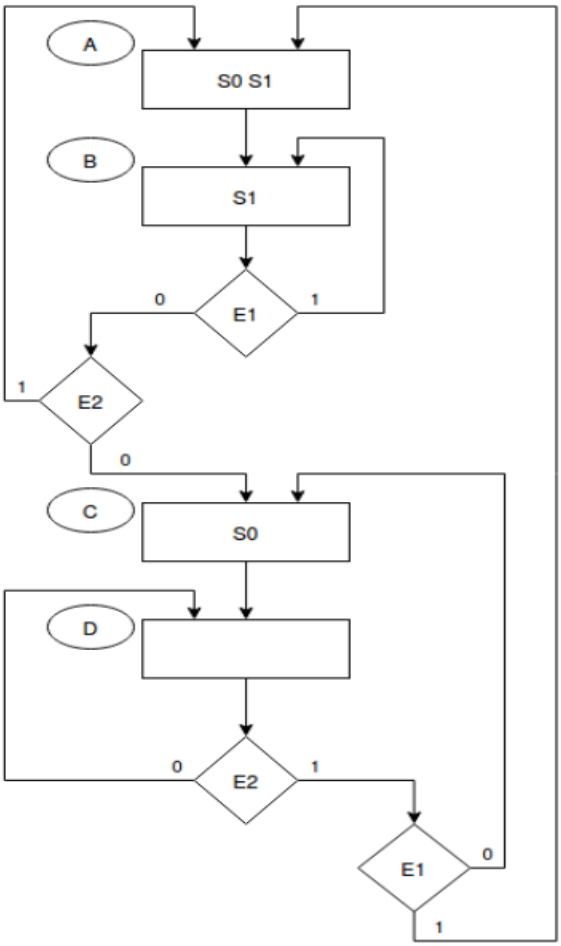

Figura 1: Carta ASM.

Analizando primeramente la carta ASM, realizamos la tabla de verdad correspondiente al comportamiento que esta tiene. Asignando primeramente los valores binarios para cada uno de los estados y así identificarlos de manera correcta

A=00
B=01
C=10
D=11

Con lo cual obtuvimos la siguiente tabla de verdad

Edo presente		Entradas		Estado Siguiente		Salidas	
Q1	Q0	E1	E2	D1	D0	S0	S1
0	0	*	*	0	1	1	1
0	1	0	0	1	0	0	1
0	1	0	1	0	0	0	1
0	1	1	*	0	1	0	1
1	0	*	*	1	1	1	0
1	1	*	0	1	1	0	0
1	1	0	1	1	0	0	0
1	1	1	1	0	0	0	0

Figura 1.1 Tabla de verdad de la carta ASM.

Con lo cual posteriormente realizamos los mapas de Karnaugh para los estados siguientes y las salidas.

- Para D1:

$$D1=Q_0E_1'E_2'+Q_1Q_0'+Q_1E_1'+Q_1E_2'$$

E1E2\Q1Q0	0	1	11	10
0	0	1	1	1
1	0	0	1	1
11	0	0	0	1
10	0	0	1	1

Figura 1.2 Mapa de Karnaugh para D1.

- Para D0:

$$D0=Q_1'E_1+Q_1E_2'+Q_0'$$

E1E2\Q1Q0	0	1	11	10
0	1	0	1	1
1	1	0	0	1
11	1	1	0	1
10	1	1	1	1

Figura 1.3 Mapa de Karnaugh para D0.

- Para S1:

$$S1 = Q_1'$$

E1E2\Q1Q0	0	1	11	10
0	1	1	0	0
1	1	1	0	0
11	1	1	0	0
10	1	1	0	0

Figura 1.3 Mapa de Karnaugh para S1.

- Para S0:

$$S2 = Q_0'$$

E1E2\Q1Q0	0	1	11	10
0	1	0	0	1
1	1	0	0	1
11	1	0	0	1
10	1	0	0	1

Figura 1.4 Mapa de Karnaugh para S0.

Una vez obtenido los mapas de Karnaugh y sus respectivas formulas, pudimos realizar el esquema del circuito secuencial correspondiente a la carta ASM inicial.

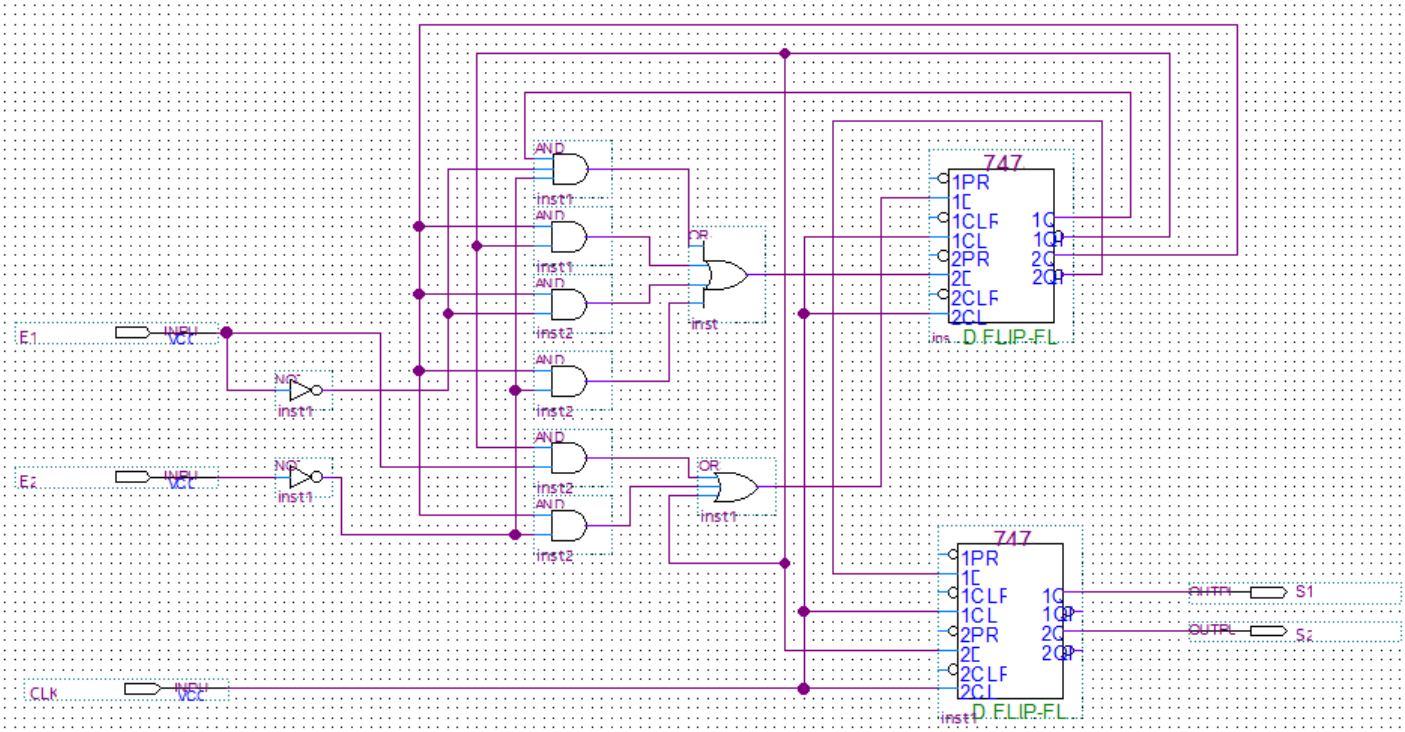

Figura 1.5 Esquema del circuito secuencial

Siguiendo las fórmulas obtenidas anteriormente podemos ya conectar compuertas lógicas simples y Flip Flops tipo D, para poder tener el circuito secuencial de manera correcta, como se muestra en la imagen, permitiéndonos así poder simularlo.

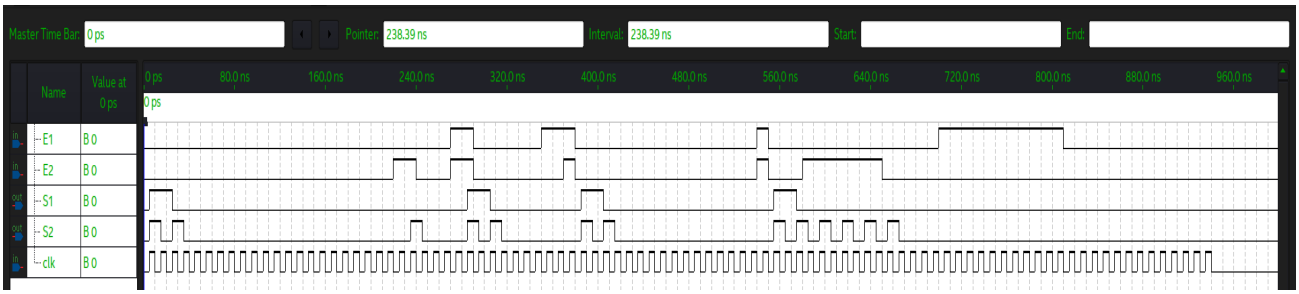

Figura 1.6 Simulación 01 del Esquemático

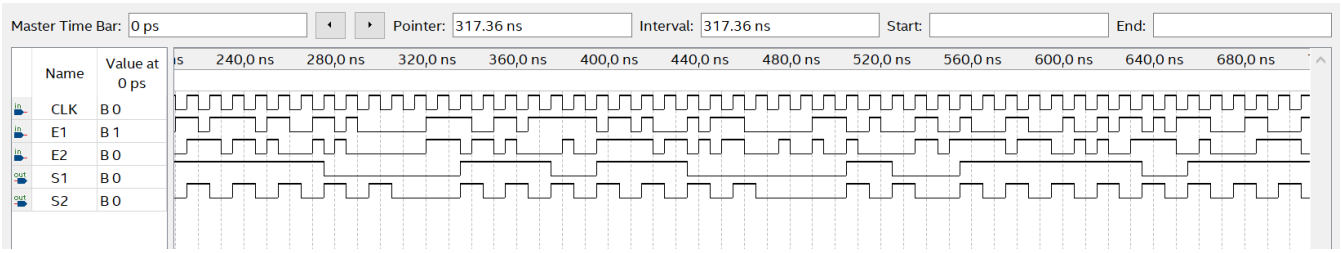

Figura 1.7 Simulación 02 del Esquemático

Como segundo punto a realizar en la práctica, se realizó la máquina de estados programada en VHDL que represente a la carta ASM vista con anterioridad. Dando el código que se muestra a continuación:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY carta_asm IS PORT (
   clk, reset : IN std_logic;
   e : IN std_logic_vector (1 DOWNTO 0);
   s : OUT std_logic_vector (1 DOWNTO 0)
);
END carta_asm;
ARCHITECTURE Behavioral OF carta_asm IS
   TYPE estados IS (A, B, C, D);
   SIGNAL estado_presente, estado_siguiente : estados;
   -- Maquina de estados
BEGIN
   PROCESS (estado_presente) BEGIN
       CASE estado_presente IS
           WHEN A =>
               s <= "11";
               estado_siguiente <= B;
           WHEN B =>
               s <= "10";
               IF e(0) = '1' THEN
                   estado_siguiente <= B;
               ELSIF e(0) = '0' and e(1) = '1' THEN
                   estado_siguiente <= A;
               ELSIF e(0) = '0' and e(1) = '0' THEN
                   estado_siguiente <= C;
               ELSE
                   estado_siguiente <= estado_presente;
               END IF;
           WHEN C =>
               s <= "01";
               estado_siguiente <= D;
           WHEN D =>
               IF e(1) = '0' THEN
                   estado_siguiente <= D;
               ELSIF e(0) = '0' and e(1) = '1' THEN
                   estado_siguiente <= C;
               ELSIF e(0) = '1' and e(1) = '1' THEN
                   estado_siguiente <= A;
               ELSE
                   estado_siguiente <= estado_presente;
               END IF;
       END CASE;
```

Comprobando su funcionamiento a través de una simulación

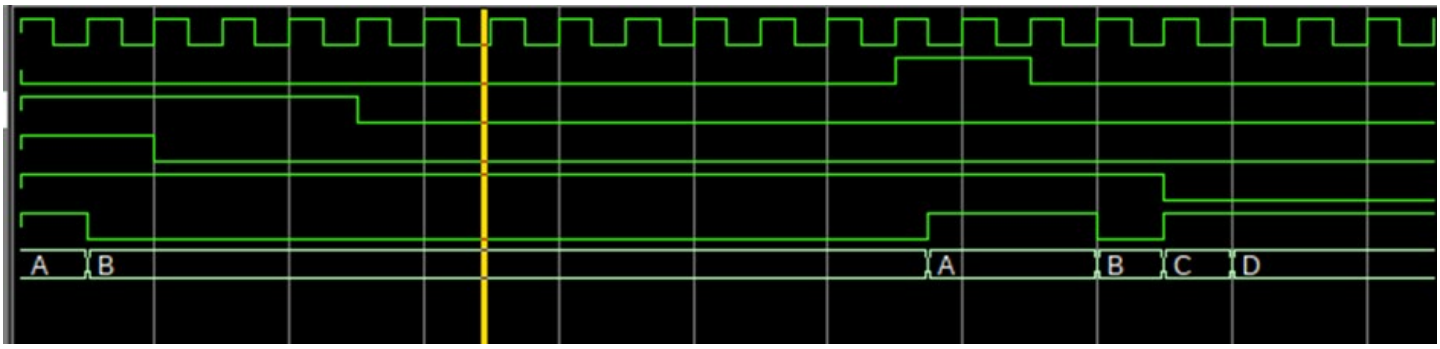

Figura 1.6 Simulación de la Maquina de estado en VHDL

Como logramos observar en la simulación se observa las letras de los estados en los que se encontraría nuestra máquina de estado finito dependiendo de las secuencias de entrada que haya tenido a lo largo de su ciclo.

Fuentes de Consulta

- Casanovas, M. (2014) Máquinas de Estado Finitas. Consultado el 12 de octubre del 2020 de: https://www.profesores.frc.utn.edu.ar/electronica/tecnicasdigitalesi/pub/file/AportesDelCudar/Maquinas%20de%20Estado%20MC%20V5.pdf
- Anónimo (S.F) Máquinas de estado en VHDL. Consultado el 12 de octubre del 2020 de: https://vhdl.es/maquinas-de-estado-en-vhdl/
- García, A (S.F) Máquinas de estado finito en VHDL. Consultado el 12 de octubre del 2020 de: https://www.digilogic.es/maquinas-de-estado-finito-fsm-vhdl/
-