LABORATORIO DE ORGANIZACIÓN
Y
ARQUITECTURA DE COMPUTADORAS

Practica 2

Diseño de máquinas de estado

Integrantes:
Barrón Pérez Marian Andrea
Padilla Castillo Aarón Samir

Grupo 3

Fecha de entrega: 15 Octubre 2020

Objetivo

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

Introducción

Una máquina de estados es un modelo que contiene los elementos necesarios para describir un sistema en términos de entradas, salidas y del tiempo. Existen técnicas para mostrar diagramas existen los diagramas de estados los cuales son parecidos a las cartas ASM sin embargo para circuitos síncronos la técnica de la carta ASM es la mejor notación.

En cartas ASM el estado presente se representa con rectángulo, mientras que las decisiones se representan con un rombo, el cual permite seleccionar el camino que el algoritmo debe de tomar de acuerdo a la variable. Las salidas no condicionales se representan dentro de un rectángulo; Las salidas condicionales sólo se utilizan cuando existen ciertas condiciones de entrada, estas se representan con óvalo.

En esta práctica se diseñará un circuito en el software Quartus, con este circuito se podrá realizar una simulación. Se diseñará y se realizará una simulación de máquina de estados de la carta ASM utilizando el lenguaje de descripción de hardware VHDL. [1]

Desarrollo

Dada la carta ASM de la figura 1, elabore lo que se indica.

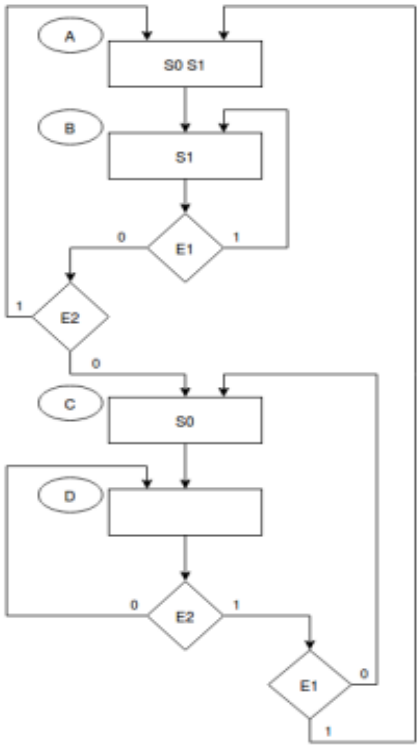


Figura 1: Carta ASM

A. Obtenga el circuito secuencial de la carta ASM utilizando Flip Flops tipo D. Cree un proyecto en Quartus llamado practica2 ff, implemente su diseño en el ambiente de desarrollo Quartus y simularlo para validar su comportamiento.

1. Tabla de verdad

	Estado Presente		Entradas		Estado Siguiente		Salidas	
	Q_1	Q_0	E_1	E_2	D_1	D_0	S_0	S_1
A	0	0	*	*	0	1	1	1
B	0	1	0	0	1	0	0	1
	0	1	0	1	0	0	0	1
	0	1	1	*	0	1	0	1
C	1	0	*	*	1	1	1	0
D	1	1	*	0	1	1	0	0
	1	1	0	1	1	0	0	0
	1	1	1	1	0	0	0	0

Tabla 1: Tabla de verdad con los nombres de estados

2. Mapas de Karnaugh

	D1			
Q1 Q0 / E1 E2	00	01	11	10
00		1	1	1
01			1	1
11				1
10			1	1

$$D1 = Q0\,\overline{E1}\,\overline{E2} + Q1\,\overline{E1} + Q1\,\overline{Q0} + Q1E1\,\overline{E2}$$

Figura 2: Mapa de Karnaugh para D1

	D0			
Q1 Q0 / E1 E2	00	01	11	10
00	1		1	1
01	1			1
11	1	1		1
10	1	1	1	1

$$D0 = \overline{Q0} + \overline{Q1}\,E1 + E1\,\overline{E2} + Q1\,\overline{E1}\,\overline{E2}$$

Figura 3: Mapa de Karnaugh para D0

	S0			
Q1 Q0 / E1 E2	00	01	11	10
00	1			1
01	1			1
11	1			1
10	1			1

$S0 = \overline{Q0}$

Figura 4: Mapa de Karnaugh para S0

	S1			
Q1 Q0 / E1 E2	00	01	11	10
00	1	1		
01	1	1		
11	1	1		
10	1	1		

$S1 = \overline{Q1}$

Figura 5: Mapa de Karnaugh para S1

3. Circuito Secuencial

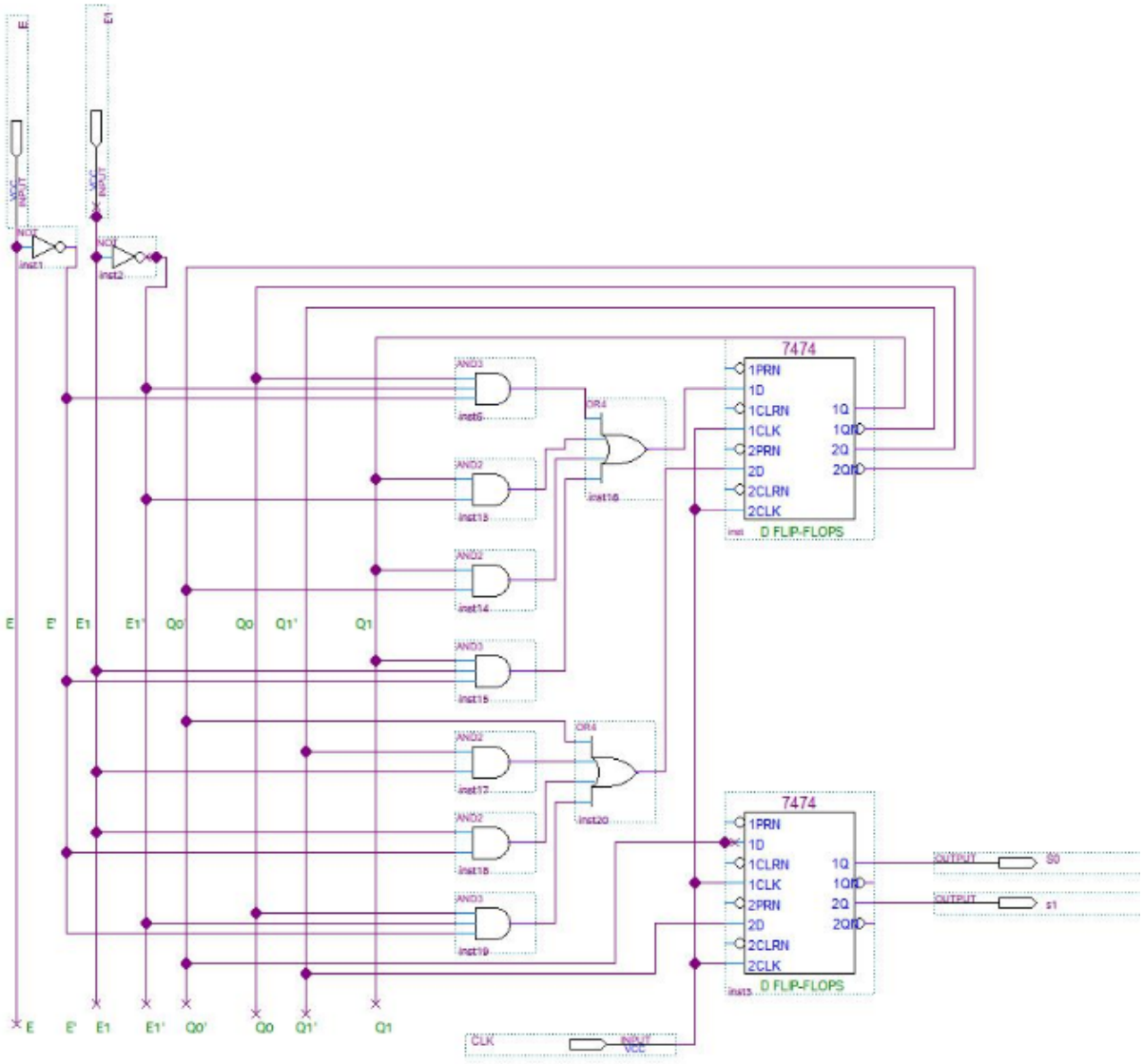


Figura 5. Circuito Secuencial

4. Simulación numero uno

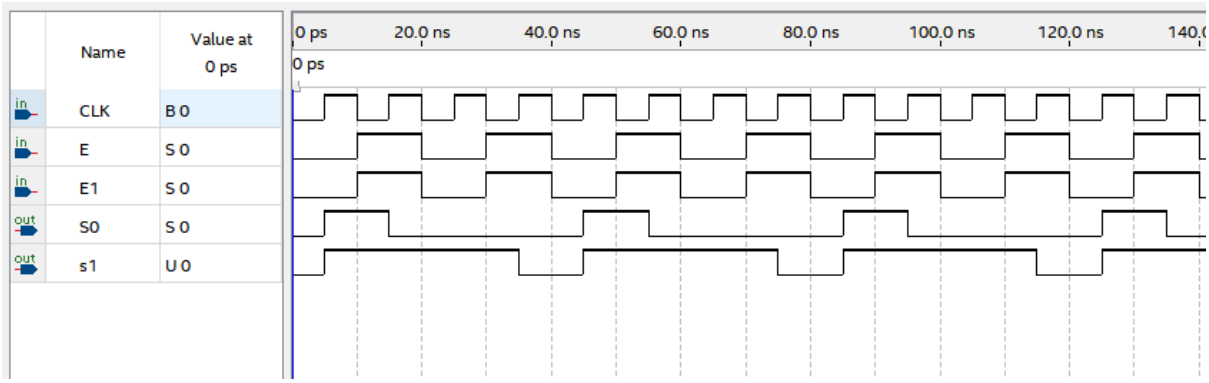


Figura 6. Simulación 1

B. Cree un nuevo proyecto en Quartus llamado practica2 VHDL y diseñe la máquina de estados de la carta ASM utilizando el lenguaje de descripción de hardware VHDL. Simule su diseño para validar que funciona correctamente.

1. Código VHDL

```
library ieee;
use ieee.std_logic_1164.all;
entity p2 is
	port(
		clk: in std_logic;
		reset:in std_logic;
		e: in std_logic_vector(1 downto 0);
		s: out std_logic_vector(1 downto 0)
	);
end p2;

architecture arquitectura of p2 is
	type estado is (A, B, C, D);
	signal estado_presente, estado_siguiente: estado;
		begin
			process(e, estado_presente)
				begin
					s<="00";
					case estado_presente is
				when A =>
					s<="11";
					estado_siguiente <= B;
				when B =>
					s <= "01";
					if (e="00") then estado_siguiente <= C;

					elsif (e="01") then estado_siguiente <= A;
					else
						estado_siguiente <= B;
					end if;
				when C =>
					s<="10";
					estado_siguiente <= D;
				when D =>
					s <= "00";
					if (e="01") then estado_siguiente <= C;

					elsif (e="11") then estado_siguiente <= A;
					else
						estado_siguiente <= D;
					end if;
					end case;
			end process;

			process(clk, reset)
				begin
					if (reset ='1') then
						estado_presente <= A;
					elsif rising_edge(clk) then
						estado_presente <= estado_siguiente;
					end if;
			end process;
end arquitectura;
```

2. Simulación numero dos

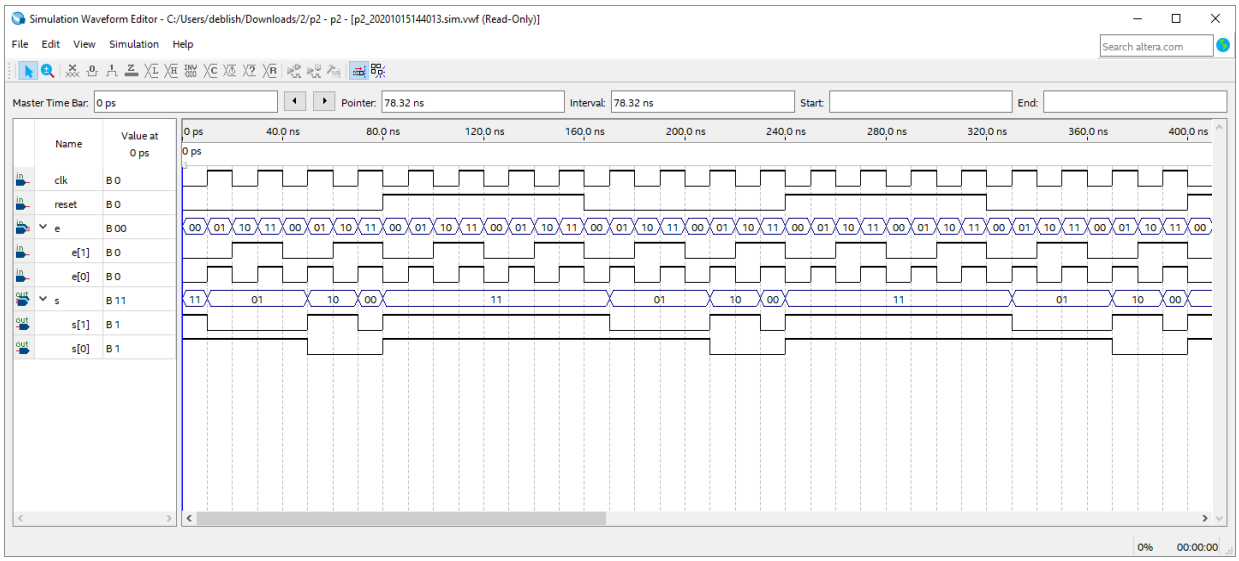


Figura 7. Simulación 2

Conclusion

Con la realización de esta práctica se logró resolver el circuito de manera combinacional a través de diseño de circuitos y con programación en VHDL. Comprobamos que utilizando máquinas de estado se puede simplificar la implementación de un circuito. La programación modular nos ayuda a que los circuitos se pueden simplificar, ya que los bloques se pueden adaptar para cada funcionalidad.la implementación con máquinas de estados es más eficiente y escalable.

Individuales

Padilla Castillo Aarón Samir

En esta practica pudimos aplicar lo visto previamente en clases de teoría, aprendimos como convertir una carta ASM a código VHDL, simular las entradas y salidas del mismo para comparar los resultados teóricos y pudimos hacer mas formalmente el circuito secuencial, como se conectan sus componentes y como interactúan ente ellos.

Barrón Pérez Marian Andrea

Con ayuda de la práctica se pudo entender uno de los tantos algoritmos que se pueden utilizar para representar máquinas de estados, también fue un buen ejercicio para recordar el uso de Quartus y lo aprendido en semestres anteriores sobre VHDL.

Participación en la practica

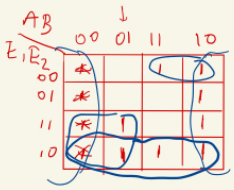
Aportación teórica

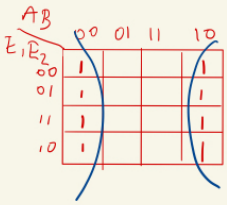
Ambos integrantes ayudamos al desarrollo de todas las partes, excepto la conclusión y pasar a limpio las tablas/figuras, que fue individual. Durante la clase desarrollamos tanto las tablas de Karnaugh como el circuito secuencial.

	Edo presente		Entradas		Estado siguiente		Salidas	
	A	B	E1	E2	A'	B'	S0	S1
3	0	0	*	*	0	1	1	1
4	0	1	0	0	1	0	0	1
5	0	1	0	1	0	0	0	1
6	0	1	1	*	0	1	0	1
7	1	0	*	*	1	1	1	0
8	1	1	*	0	1	1	0	0
9	1	1	0	1	1	0	0	0
10	1	1	1	1	0	0	0	0

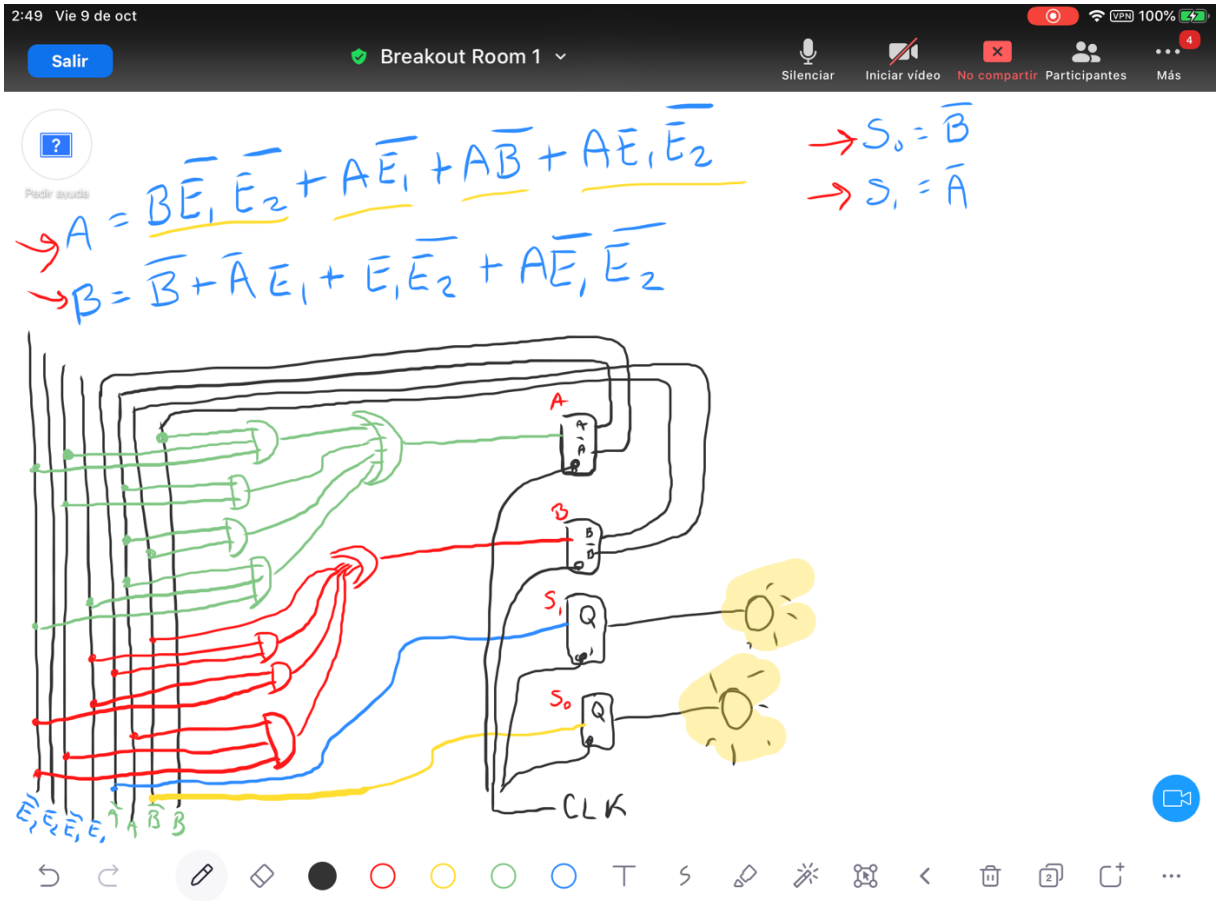
$B = \bar{B} + \bar{A}E_1 + E_1\bar{E_2} + A\bar{E_1}\bar{E_2}$

$S_0 = \bar{B}$





Borrador de tablas en pizarra de Zoom



Circuito Secuencial Borrador en pizarra de Zoom

Aportación experimental

Para la parte experimental nos dividimos las capturas en dos, las capturas del apartado A corresponden al ordenador de Barrón y las del apartado B al de Padilla, sin embargo ambos estuvimos revisando resultados y compartiéndonos el código para corregirlo.

Referencias

[1] Savage, J. (2015). Diseño de microprocesadores. En *Diseño de microprocesadores* (2015.ª ed., pp. 21-23). Facultad de ingenieria.
https://classroom.google.com/c/MTYwNjg5NzQ1Mjgz/m/MTY5OTM1NDA5OTk4/details?hl=es