

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Ingeniería en Computación

Laboratorio de Organización y Arquitectura de Computadoras

Práctica 03: Construcción de Máquinas de estados Usando Memorias

Direccionamiento por Trayectoria

Alumnos:

- Murrieta Villegas Alfonso
- Reza Chavarria Sergio Gabriel
- Valdespino Mendieta Joaquin

Profesora: Ayesha Sagrario Román García

Grupo: 7

Fecha de entrega: 1 de octubre de 2021

Práctica 03

Objetivo

Familiarizar al alumno en el conocimiento de construcción de máquinas de estados usando direccionamiento de memorias con el método de direccionamiento por trayectoria.

Introducción

El método por direccionamiento por trayectoria obtiene el estado siguiente y las salidas de cada estado de la carta ASM en la localidad de memoria.

El método de direccionamiento por trayectoria está compuesto por 2 elementos de Hardware, una memoria de tipo ROM y un registro LATCH.

Las ventajas del direccionamiento por trayectoria son:

- Representación y soporte de cualquier carta ASM y cualquier trayectoria en su diseño.
- Puede soportar salidas de tipo condicional
- Se pueden implementar diferentes sistemas en el circuito solo con borrar y reprogramar la memoria

La desventaja que tiene la implementación es el desperdicio de memoria por todas las opciones y configuraciones obtenidas.

La representación de la carta ASM por este método tiene el siguiente encabezado.

Estado Presente + Entradas VS Estado Siguiete + Salidas

Desarrollo

1. Dada la carta ASM de la figura 2, encuentre el contenido de memoria utilizando el direccionamiento por trayectoria. Recuerde que antes de construir la tabla se debe asignar a cada estado de la carta ASM una representación binaria. Así mismo, recuerde que para cada estado es necesario considerar todas las posibles combinaciones de las variables de entrada, aun cuando algunas de ellas no se utilicen en determinado estado.
2. Una vez que haya obtenido el contenido de memoria, implemente el direccionamiento por trayectoria utilizando el software de desarrollo Quartus y escriba el contenido de memoria obtenido.
3. Simule su diseño para probar su funcionamiento. recuerden que en sus simulaciones debe aparecer el contenido de la memoria además del estado presente.

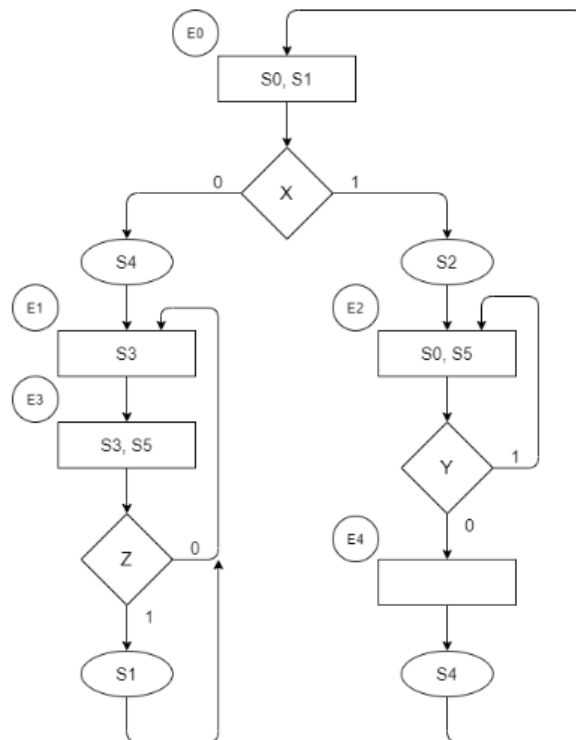


Ilustración 1: Carta ASM a desarrollar

Resultados

Ejercicio 1

A continuación, se realizará la tabla de verdad.

Estado Actual A, B, C	Entradas X, Y, Z	Estado Siguiente $A(D), B(D), C(D)$	Salidas $S_5, S_4, S_3, S_2, S_1, S_0$
000	0**	001	010011
000	1**	010	000111
001	***	011	001000
010	*0*	100	100001
010	*1*	010	100001
011	**0	001	101000
011	**1	001	101010
100	***	000	010000

A partir de la tabla de verdad obtiene la información de los estados siguientes y las salidas correspondientes. Esto servirá para obtener las funciones lógicas. Para obtener la función correspondiente a los estados siguientes manejará la división de la representación binaria de los estados. Las funciones de las salidas igual se manejarán de manera individual. Se tomarán en cuenta, para la realización de las funciones, los casos en donde esté encendido el bit.

Funciones de los estados siguientes (Liga)

Estado Actual			Entradas	Estado Siguiente		
A, B, C			X, Y, Z	$A(D), B(D), C(D)$		
0	0	0	0**	0	0	1
0	0	0	1**	0	1	0
0	0	1	***	0	1	1
0	1	0	*0*	1	0	0
0	1	0	*1*	0	1	0
0	1	1	**0	0	0	1
0	1	1	**1	0	0	1
1	0	0	***	0	0	0

Funciones Lógicas

$$A(D) = \bar{A}\bar{B}\bar{C}\bar{Y}$$

$$B(D) = \bar{A}\bar{B}\bar{C}X + \bar{A}\bar{B}C + \bar{A}B\bar{C}Y$$

$$C(D) = \bar{A}\bar{B}\bar{C}\bar{X} + \bar{A}\bar{B}C + \bar{A}B\bar{C}\bar{Z} + \bar{A}B\bar{C}Z$$

Simplificaciones de funciones lógicas

$$A(D) = \bar{A}\bar{B}\bar{C}\bar{Y}$$

$$B(D) = \bar{A}\bar{B}\bar{C}X + \bar{A}\bar{B}C + \bar{A}B\bar{C}Y = \bar{A}(\bar{B}\bar{C}X + \bar{B}C + B\bar{C}Y)$$

$$B(D) = \bar{A}(\bar{B}(\bar{C}X + C) + B\bar{C}Y) = \bar{A}(\bar{B}(X + C) + B\bar{C}Y)$$

$$C(D) = \bar{A}\bar{B}\bar{C}\bar{X} + \bar{A}\bar{B}C + \bar{A}B\bar{C}\bar{Z} + \bar{A}B\bar{C}Z = \bar{A}(\bar{B}\bar{C}\bar{X} + \bar{B}C + B\bar{C}\bar{Z} + B\bar{C}Z)$$

$$C(D) = \bar{A}(\bar{B}\bar{C}\bar{X} + \bar{B}C) + BC(\bar{Z} + Z) = \bar{A}(\bar{B}\bar{C}\bar{X} + \bar{B}C + BC)$$

$$C(D) = \bar{A}(\bar{B}\bar{C}\bar{X} + C(\bar{B} + B)) = \bar{A}(\bar{B}\bar{C}\bar{X} + C) = \bar{A}(C + \bar{C}\bar{B}\bar{X}) = \bar{A}(C + \bar{B}\bar{X})$$

Funciones de las entradas simplificadas

$$A(D) = \bar{A}B\bar{C}\bar{Y}$$

$$B(D) = \bar{A}(\bar{B}(X + C) + B\bar{C}Y)$$

$$C(D) = \bar{A}(C + \bar{B}\bar{X})$$

Funciones de las salidas

Estado Actual	Entradas	Salidas					
A, B, C	X, Y, Z	$S_5, S_4, S_3, S_2, S_1, S_0$					
000	0**	0	1	0	0	1	1
000	1**	0	0	0	1	1	1
001	***	0	0	1	0	0	0
010	*0*	1	0	0	0	0	1
010	*1*	1	0	0	0	0	1
011	**0	1	0	1	0	0	0
011	**1	1	0	1	0	1	0
100	***	0	1	0	0	0	0

$$S_0 = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C}$$

$$S_1 = \bar{A}\bar{B}\bar{C} + \bar{A}BCZ$$

$$S_2 = \bar{A}\bar{B}\bar{C}X$$

$$S_3 = \bar{A}\bar{B}C + \bar{A}BC$$

$$S_4 = \bar{A}\bar{B}\bar{C}\bar{X} + \bar{A}\bar{B}\bar{C}$$

$$S_5 = \bar{A}\bar{B}\bar{C} + \bar{A}BC$$

Simplificación Funciones lógicas de Salidas

$$S_0 = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} = \bar{A}\bar{C}(\bar{B} + B) = \bar{A}\bar{C} = \overline{(AC)}$$

$$S_1 = \bar{A}\bar{B}\bar{C} + \bar{A}BCZ = \bar{A}(\bar{B}\bar{C} + BCZ)$$

$$S_2 = \bar{A}\bar{B}\bar{C}X$$

$$S_3 = \bar{A}\bar{B}C + \bar{A}BC = \bar{A}C(\bar{B} + B) = \bar{A}C$$

$$S_4 = \bar{A}\bar{B}\bar{C}\bar{X} + \bar{A}\bar{B}\bar{C} = \bar{B}\bar{C}(\bar{A}\bar{X} + A) = \bar{B}\bar{C}(\bar{X} + A)$$

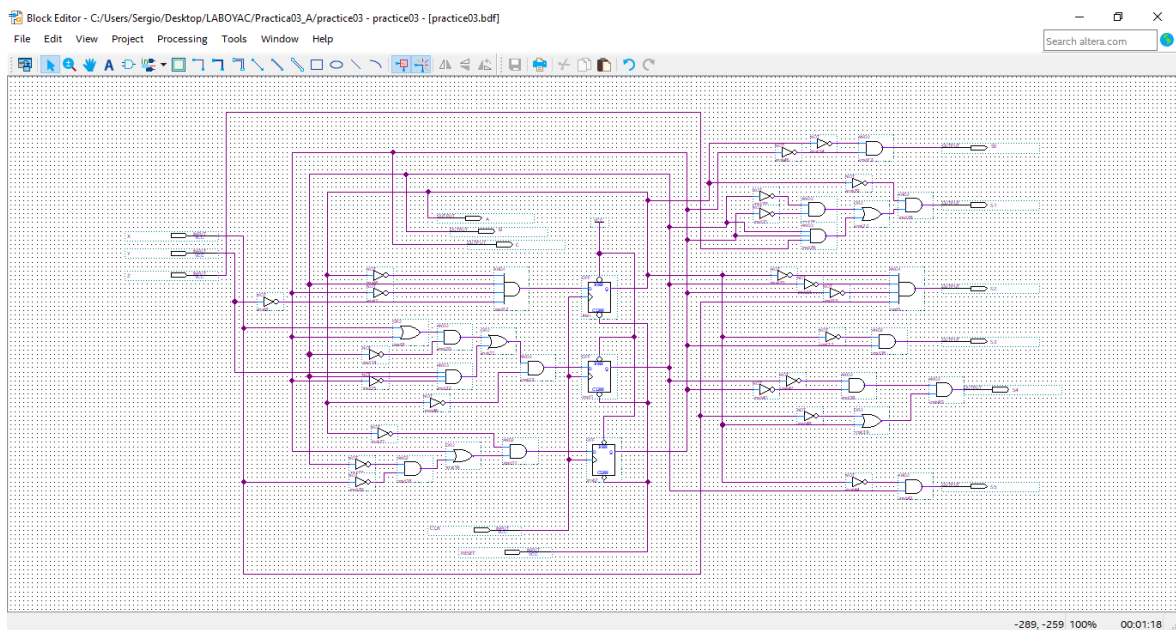
$$S_5 = \bar{A}\bar{B}\bar{C} + \bar{A}BC = \bar{A}B(\bar{C} + C) = \bar{A}B$$

Ejercicio 2

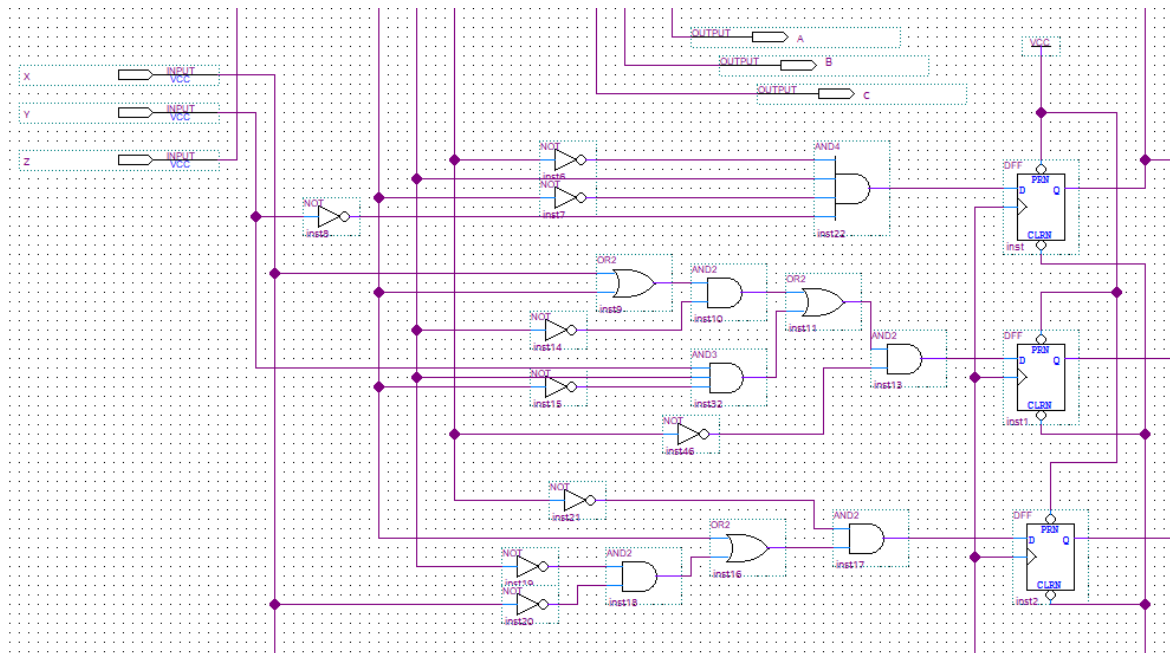
Después de obtener las funciones lógicas, se realizó la simulación en Quartus Prime. Se realizó a partir del uso de Flip Flops de tipo D.

Se conectaron a los flip flops D una fuente VCC, una entrada de reloj y una entrada de reset. En las entradas de cada flip flop se conectaron las funciones lógicas correspondientes de las variables A, B y C (descripción de los números binarios de los estados vistos en la carta ASM). También se conectaron las salidas del sistema a partir de sus funciones lógicas haciendo uso de las salidas A(D), B(D), C(D) de los flip flops y las entradas al sistema.

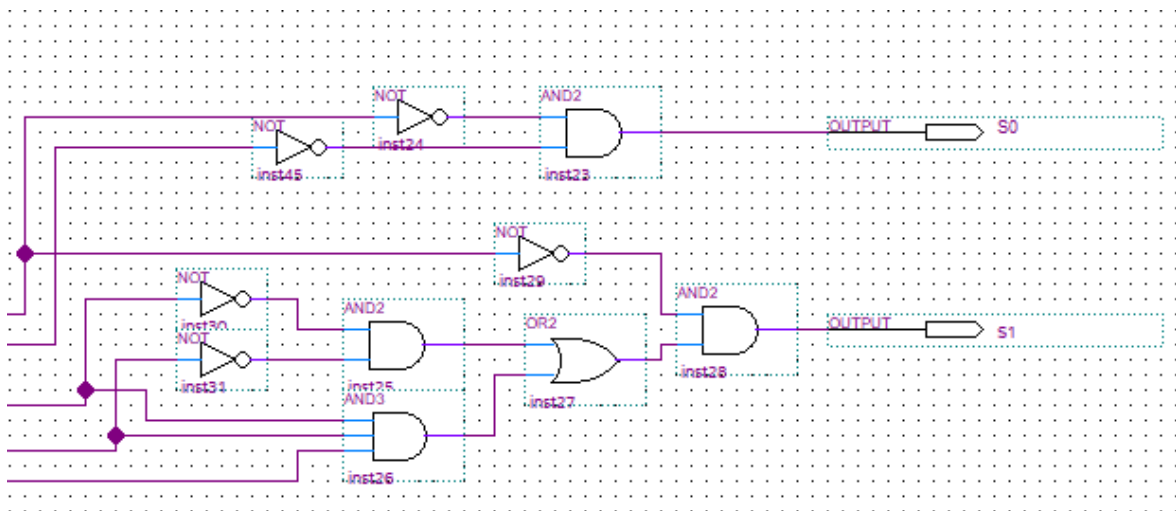
A continuación, se mostrarán las conexiones realizadas y acercamientos a las conexiones lógicas realizadas.

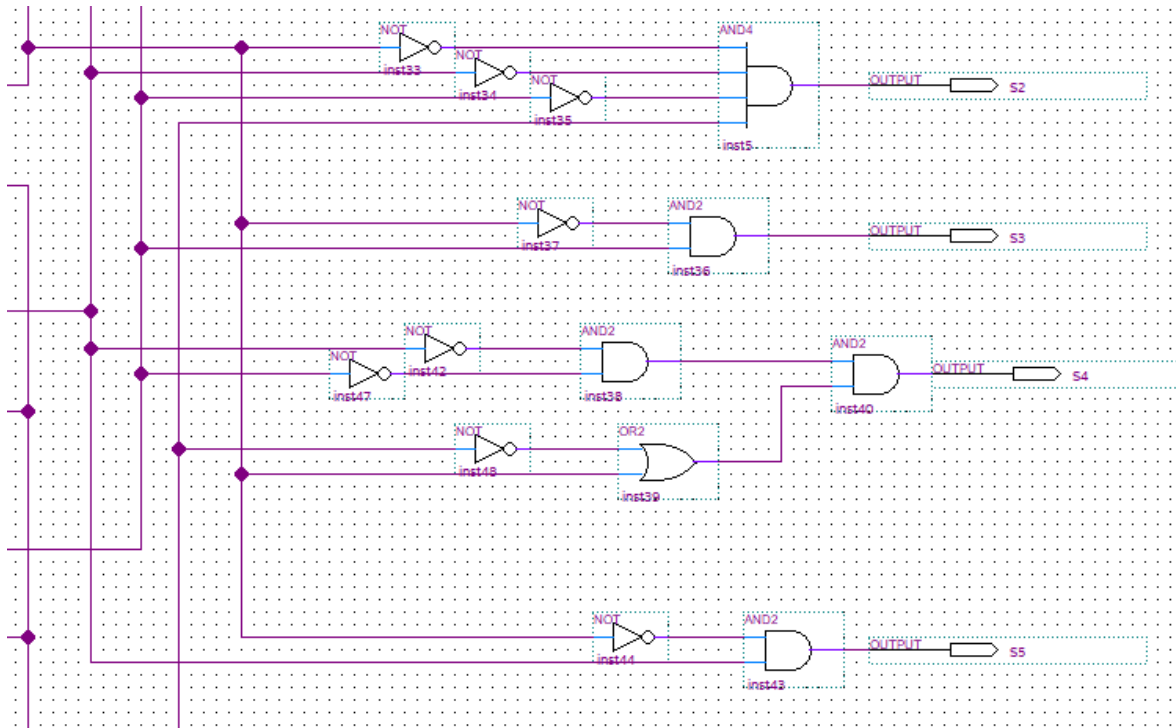


Simulación 1: Sistema completo de la máquina de estados



Simulación 2: Entradas A, B, C para los flip flops D



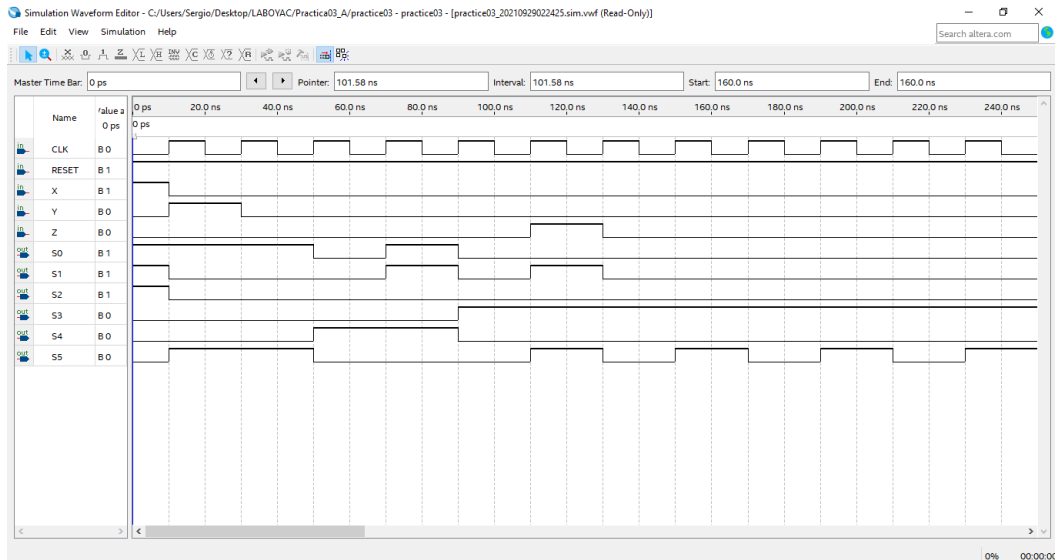


Simulación 3: Funciones de salida de la máquina de estados

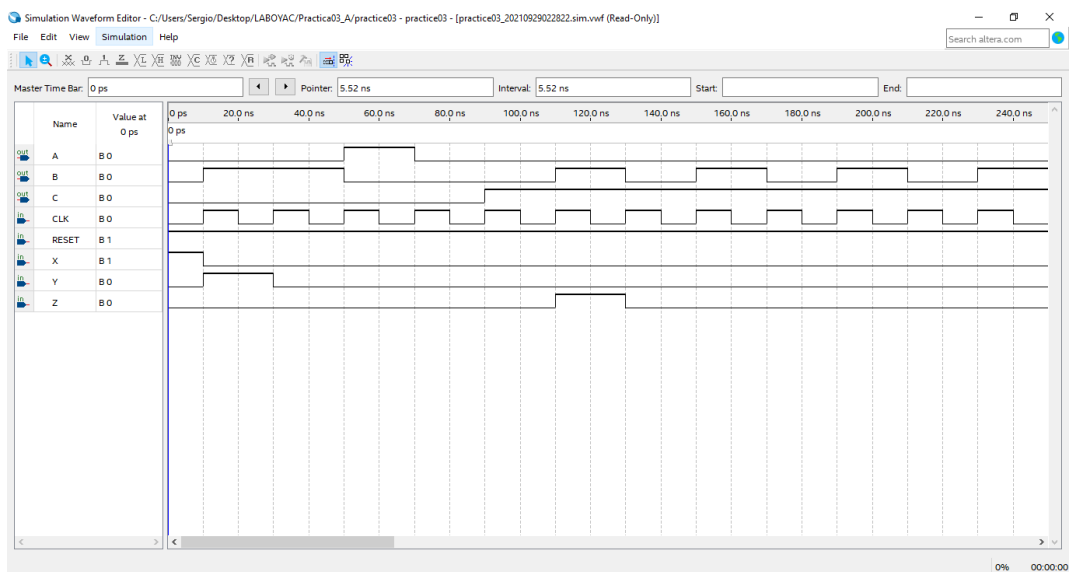
Ejercicio 3

Para la visualización del comportamiento de la máquina de estados se realizó la simulación para observar los estados presentes y las salidas por separado.

Además, se manejó las mismas entradas en ambas simulaciones para que tenga el mismo comportamiento.

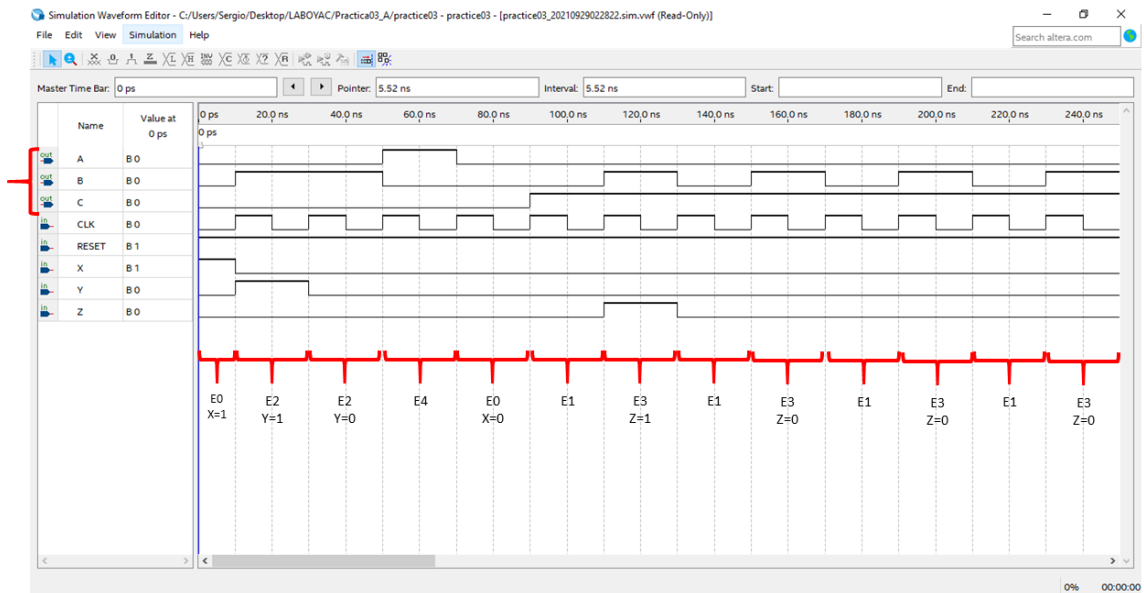


Simulación 4: Visualización de salidas por medio del simulador



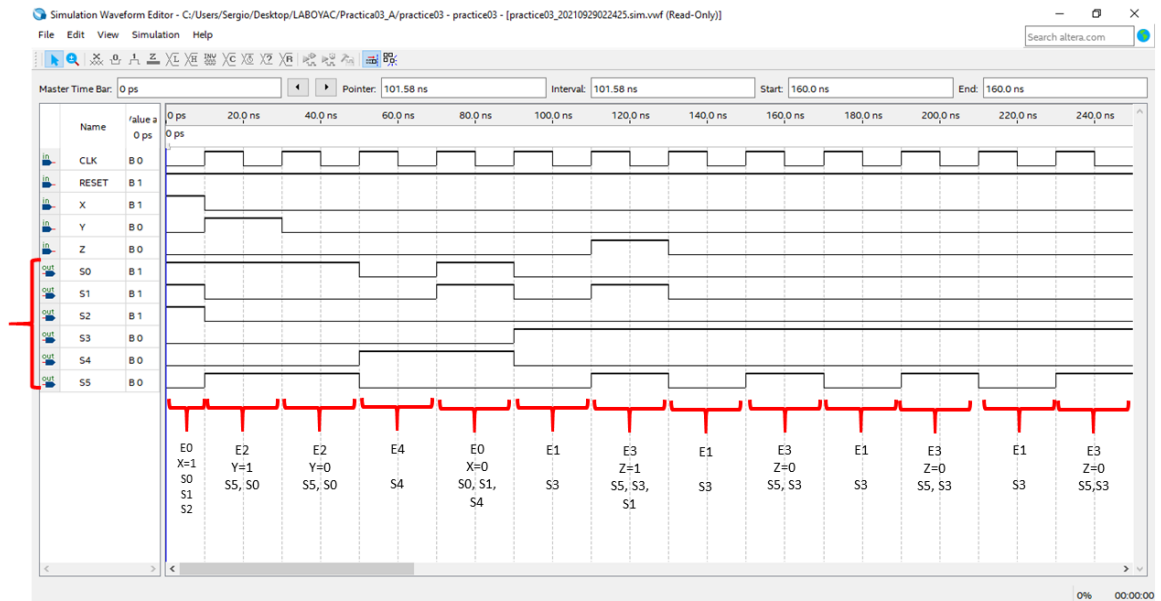
Simulación 5: Visualización de los estados en simulación

A continuación, se marcan las Entradas y los estados presentes correspondientes en cada parte de la simulación.



Simulación 6: Simulación con los estados presentes marcados

A continuación, se marcan las salidas correspondientes en la parte de la simulación.



Simulación 7: Simulación con los estados presentes y las salidas marcados

Conclusiones

Murrieta Villegas Alfonso

Para la presente práctica aplicamos el concepto teórico de las máquinas de estado a través del Método de direccionamiento por trayectoria, de esta forma, es como pudimos considerar la cantidad de memoria necesaria para el uso de las diferentes condiciones además de los estados que la máquina debía contemplar.

Por último, a través de esquemáticos en Quartus es como realizamos diferentes ejercicios empleando Flip Flops tipo D, donde el objetivo implícito es cada vez crear entidades de hardware cada vez más complejas.

Reza Chavarria Sergio Gabriel

Para la realización de la práctica se necesitó de los conceptos para el análisis de una máquina de estados por direccionamiento por trayectoria. En este modo de direccionamiento se tiene en cuenta la cantidad de memoria que se necesita y que es posible el uso de varias condiciones, junto con el uso de estados condicionales.

A partir de la obtención de los datos, por medio de la tabla de verdad, se pudo realizar su armado y comprender el comportamiento en el aspecto de la simulación.

Valdespino Mendieta Joaquin

En la presente practica pudimos aplicar los conocimientos sobre las máquinas de estado, en tanto a su forma de modelado, siendo específicos en las maquinas usando método de direccionamiento por trayectoria, todo a partir de su carta asm, pasando por su tabla de verdad, por último, su esquematización con FFD y posterior ejecución en el entorno de desarrollo, comprendiendo e implementando el comportamiento de dichas maquinas.

Bibliografía

- Savage J., Vázquez G & Chávez N. (2015). Diseño de Microprocesadores, Capítulo III Construcción De Máquinas de Estado usando Memorias. Encontrado

el 27 de septiembre de 2021. Sitio Web: [https://biorobotics.fi-p.unam.mx/wp-content/uploads/Courses/arquitectura de computadoras/material de apoyo/diseño de procesadores.pdf](https://biorobotics.fi-p.unam.mx/wp-content/uploads/Courses/arquitectura_de_computadoras/material_de_apoyo/diseño_de_procesadores.pdf)

- Electrical Engineering (2019). How do D flip-flops (dff) start up in Quartus? Encontrado el 27 de septiembre de 2021. Sitio Web: <https://electronics.stackexchange.com/questions/454677/how-do-d-flip-flops-dff-start-up-in-quartus>