

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Ingeniería en Computación

Laboratorio de Organización y Arquitectura de Computadoras

Práctica 04: Construcción de Máquinas de estados usando Memorias

Direccionamiento Entrada - Estado

Alumnos:

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

Profesora: Ayesha Sagrario Román García

Grupo: 7

Fecha de entrega: 8 de octubre de 2021

Práctica 04: Construcción de Máquinas de estados usando Memorias

Direccionamiento Entrada – Estado

Objetivo

Familiarizar al alumno en el conocimiento de construcción de máquinas de estados usando direccionamiento de memorias con el método de direccionamiento entrada - estado.

Introducción

El direccionamiento Entrada. Estado es una forma para implementar las máquinas de estados, sin embargo, esta tiene como característica principal que está restringida aquellas representaciones ya sea en ASM, donde solamente se valide una sola entrada por cada estado.

Este tipo de direccionamiento cuenta con 4 elementos de hardware, que son: memoria ROM, un registro Latch y dos multiplexores, para el caso base, como se muestra a continuación.

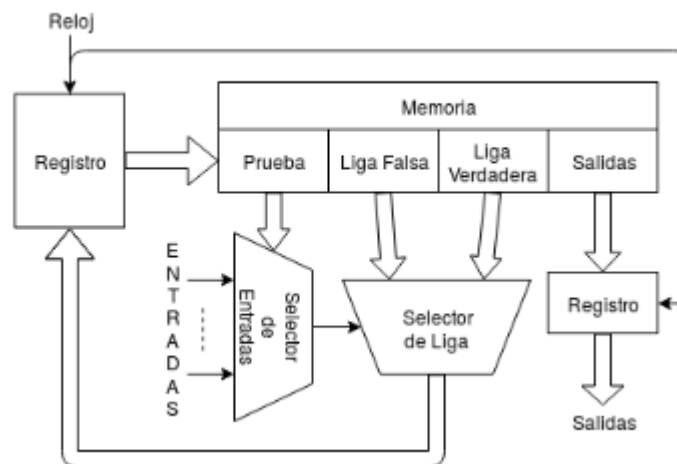


Ilustración 1: Direccionamiento Entrada Estado

La transición entre estados se da por la prueba (valores binarios de las entradas) y las ligas, dependiendo la evaluación de dicha entrada ira a lo que indique la liga Falsa (con 0) o La verdadera (con 1), en caso de que no haya evaluación de la entrada, el valor de la liga verdadera y la falsa son las mismas.

Las ventajas claras de este tipo de direccionamiento son:

- Ahorra mucha memoria respecto al direccionamiento por trayectoria
- Solo hay 4 elementos de Hardware
- Es versátil, para crear diversos sistemas con el mismo circuito basta con borrar y reprogramar memoria, usando

Sin embargo, la clara desventaja es que solo puede evaluar una entrada por estado. Y para el caso base, no permite salidas condicionales, por lo que se tiene que cambiar la estructura, quedando de la siguiente forma

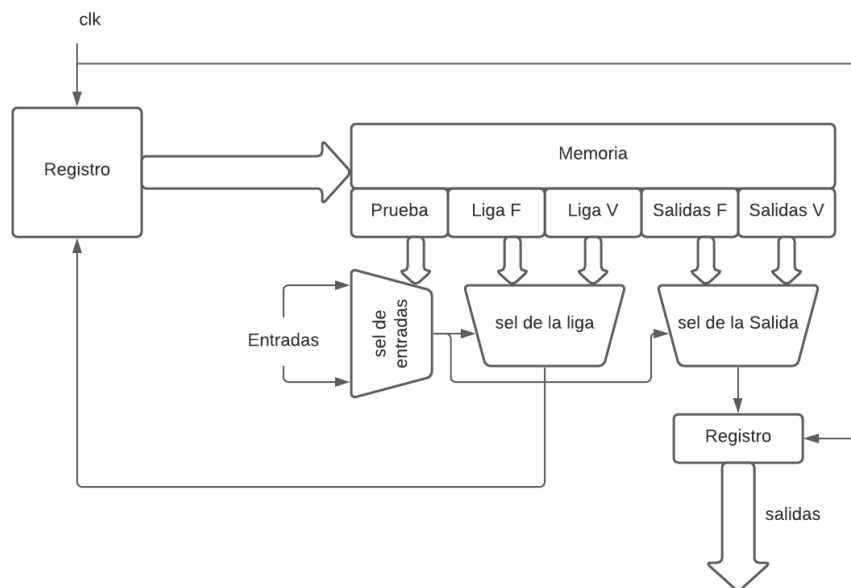


Ilustración 2: Direccionamiento Entrada Estado con salidas condicionales

En este caso, hay dos tipos de salidas, la salida observada depende si hay una salida condicional, es decir, que dependa de la toma de decisión o en otras palabras de la evaluación de la Entrada, en caso podrán variar entre salidas (Salidas F o V) o ser la misma si no se presentan las condicionales.

Desarrollo

1. Dada la carta ASM de la figura 3, encuentre el contenido de memoria utilizando el direccionamiento entrada-estado. Recuerde que antes de construir la tabla se debe asignar a cada estado de la carta ASM una representación binaria. Así mismo, no olvide asignar una representación binaria a las entradas.
2. Una vez que haya obtenido el contenido de memoria, implemente el direccionamiento de entrada-estado utilizando el software de desarrollo Quartus y utilice el contenido de memoria obtenido.
3. Simule su diseño para probar su funcionamiento. Recuerda mostrar el estado presente, el valor de prueba, el estado siguiente y las salidas.

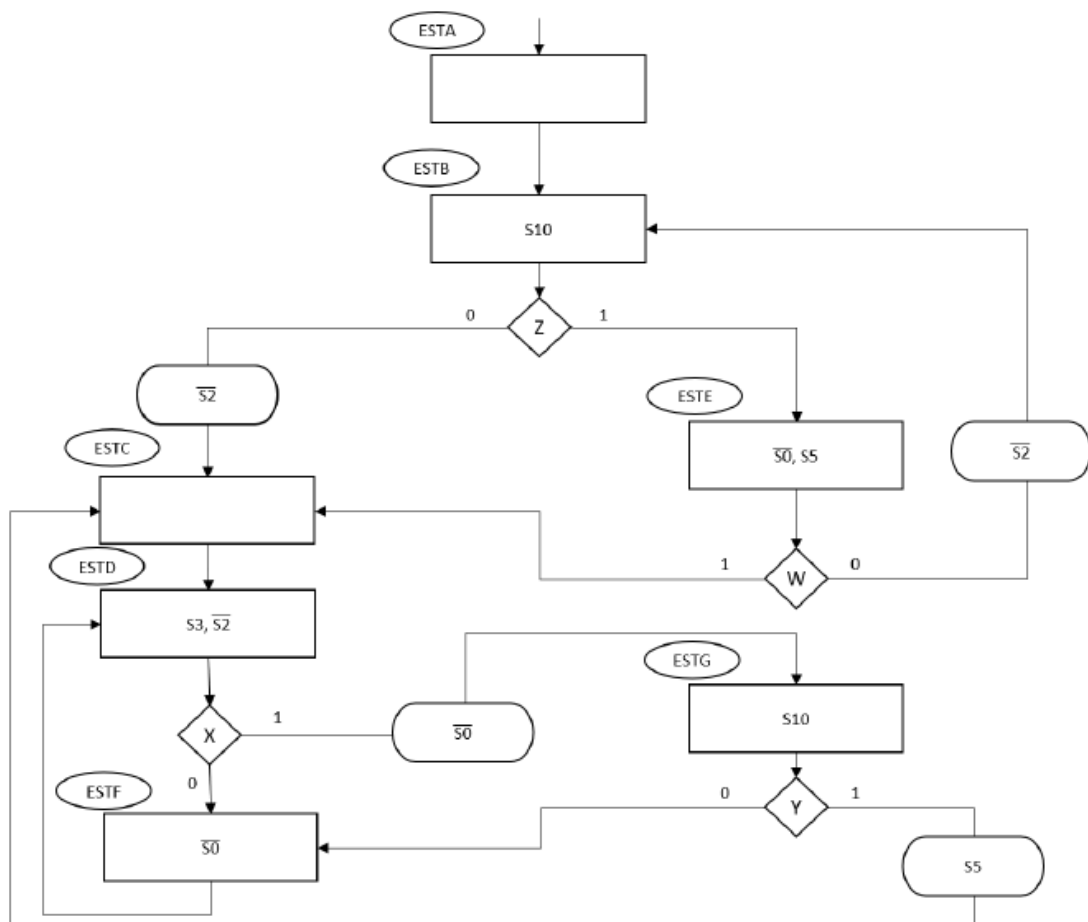


Ilustración 3: Carta ASM

Resultados

Ejercicio 1

Para la realización de la práctica se necesitó de la tabla de verdad. Como en la práctica se está revisando el método de direccionamiento Entrada Estado se necesita del siguiente encabezado.

Estado Presente	Prueba	Liga Verdadera	Liga Falsa	Salida Verdadera	Salida Falsa
P_2, P_1, P_0	K_1, K_0	V_2, V_1, V_0	F_2, F_1, F_0	$S_5, S_3, S_2, S_{10}, S_0$	$Z_5, Z_3, Z_2, Z_{10}, Z_0$

Para la representación de las entradas, la prueba será representado con 2 bits. La cantidad de estados son 7, para su representación es utilizaran 3 bits. La cantidad anterior también será utilizada para las ligas verdaderas y falsas. Para la representación de las salidas de la máquina de estados será de 5 bits.

Pruebas utilizadas

- $W = 00$
- $X = 01$
- $Y = 10$
- $Z = 11$

Tabla de verdad

Estado	Estado Presente P_2, P_1, P_0	Prueba K_1, K_0	Liga Verdadera V_2, V_1, V_0	Liga Falsa F_2, F_1, F_0	Salida Verdadera $S_5, S_3, S_2, S_{10}, S_0$	Salida Falsa $Z_5, Z_3, Z_2, Z_{10}, Z_0$
A	000	**	001	001	00000	00000
B	001	11	100	010	00010	00010
C	010	**	011	011	00000	00000
D	011	01	110	101	01000	01000
E	100	00	010	001	10000	10000
F	101	**	011	011	00000	00000
G	110	10	010	101	10010	00010

Nota: Los elementos que estaban negados en las salidas se representaron con 0. Y las pruebas con * significan que no se requiere alguna entrada para el cambio de estado.

Ejercicio 2

Para la implementación de la máquina de estados se utilizó el software de Quartus Prime, usando el tipo de archivo State Machine File. El archivo *.smf nos permite la realización de máquinas de estados de manera fácil a partir de la herramienta grafica para realizar la implementación de los estados, salidas y transiciones entre estos.

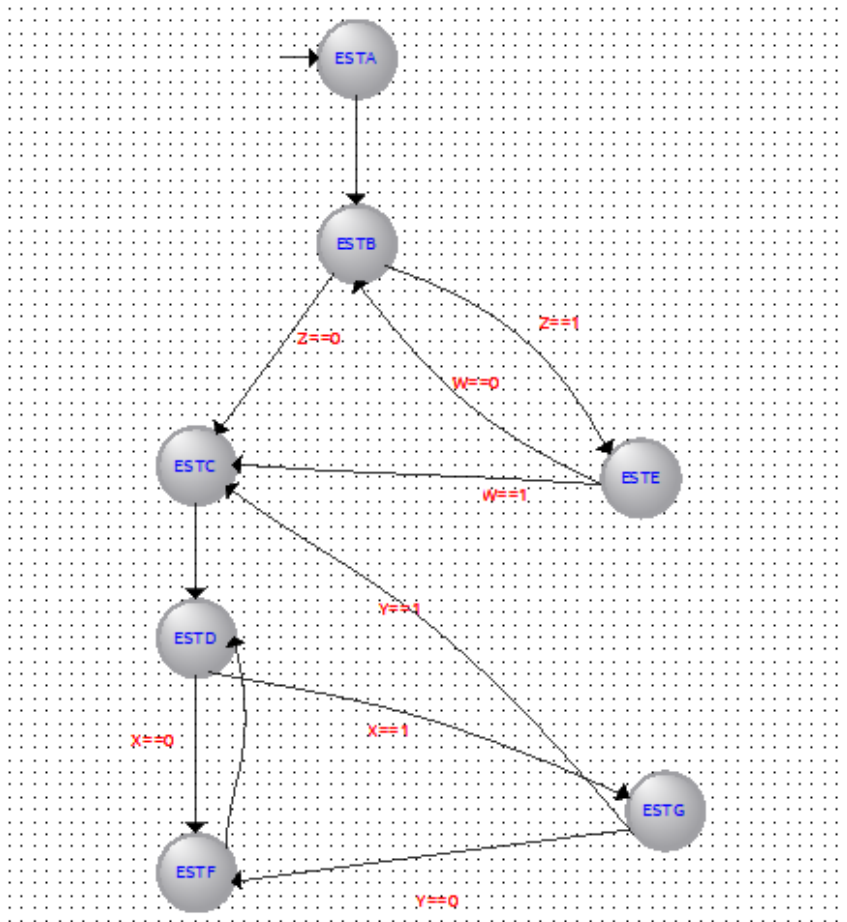


Imagen 1: Implementación de la máquina de estados en SMF

State Machine Editor - C:/Users/Sergio/Desktop/LABOYAC/Practica04/practica04 - practica04.smf

File Edit View Tools Window Help

Search altera.com

Input Table

Input Port

1 reset

Output Table

Output Port

1 SS

Source State Destination State Transition (in Verilog or VHDL 'OTHERS')

1	ESTA	ESTB	
2	ESTG	ESTC	Y==1
3	ESTC	ESTD	
4	ESTB	ESTC	Z==0
5	ESTE	ESTC	W==1
6	ESTB	ESTE	Z==1
7	ESTD	ESTG	X==1
8	ESTE	ESTB	W==0
9	ESTD	ESTF	X==Q
10	ESTF	ESTD	
11	ESTG	ESTF	Y==Q

State Table

General Inputs Outputs States Transitions Actions

100% 00:01:47

Imagen 2: Transiciones implementadas, junto con las condiciones de las entradas

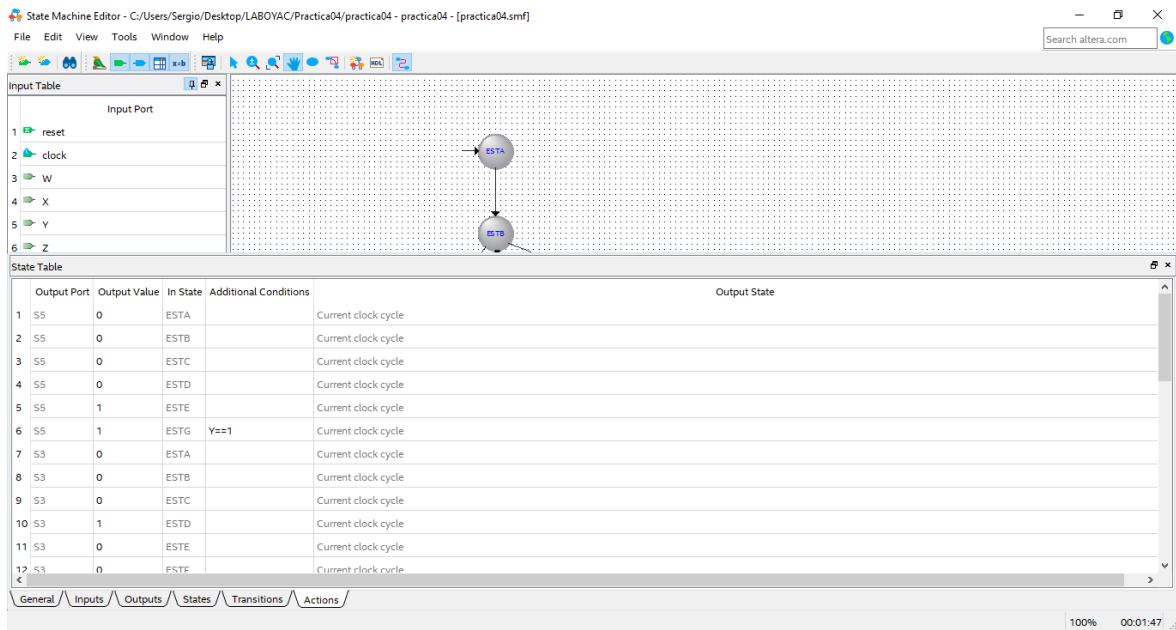


Imagen 3: Salidas de estados y salidas condicionales implementadas

Ejercicio 3

A continuación, se implementó la simulación de la carta ASM por medio de la simulación de Waveform. Se manejó de la misma manera como con las simulaciones anteriores. Incluye el reloj interno a 20 ns, entrada de reinicio, entradas de la carta ASM y Salidas.

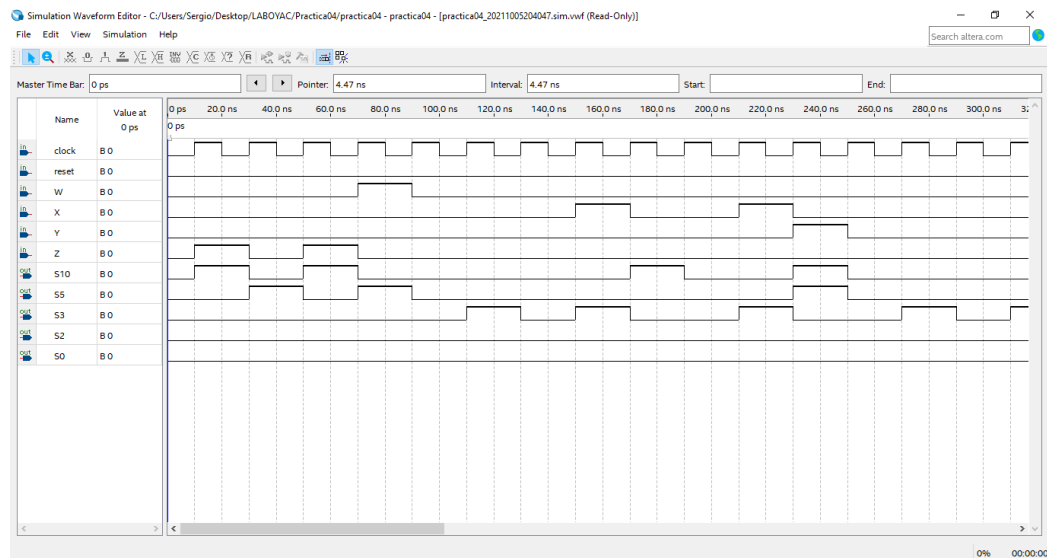


Ilustración 4: Simulación realizada

A continuación, se mostrará la señalización de las entradas, salidas y estados en la simulación obtenida.

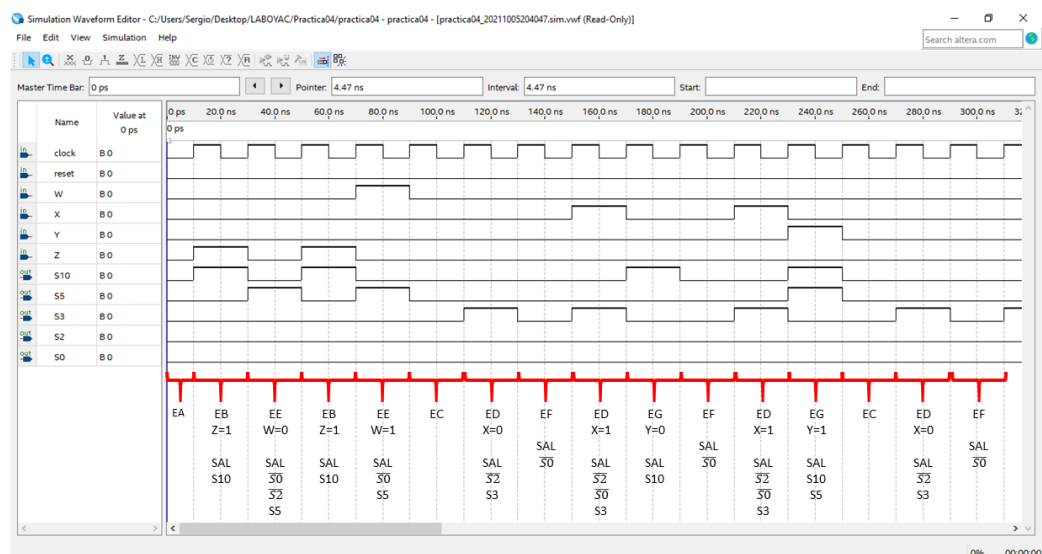


Ilustración 5: Simulación obtenida descrita

Conclusiones

Murrieta Villegas Alfonso

En la presente práctica aprendimos a partir del método de direccionamiento Entrada – Estado a construir máquinas de estados que a diferencias del método de Direccionamiento por Trayectoria nos ayuda a simplificar de manera significativa la cantidad de memoria debido a que para considerar ligas verdaderas y falsas es necesario contemplar las diferentes vías por las que los estados se mueven en nuestra máquina.

Por último y como parte de Quartus, aprendimos a aterrizar nuestras ideas – máquinas de estados a través de diseños gráficos para posteriormente validarlos mediante el waveform.

Reza Chavarria Sergio Gabriel

A partir del direccionamiento de Entrada Estado nos permite dar una simplificación a la memoria en comparación al direccionamiento por trayectoria, ya que se considera en este direccionamiento entre las ligas verdaderas y falsas, esto toma en cuenta las diferentes vías por las que se transita el comportamiento de la máquina de estados.

Con el uso del archivo de la máquina de estados fue un manejo en la implementación de estos sistemas de manera simplificada, así elementos más complejos pueden manejarse de manera gráfica y puede simplificar la estructura.

En la presente practica pudimos aplicar los conocimientos sobre las máquinas de estado, en tanto a su forma de modelado, sus características principales, ventajas y sus desventajas como solo una entrada por estado en este caso, como se ha dicho en las maquinas usando método de direccionamiento Entrada estado, todo a partir de su carta ASM, pasando por su tabla de verdad, por último, su esquematización mediante diagrama de estados y posterior ejecución y simulación en el entorno de desarrollo, comprendiendo e implementando el comportamiento de dichas maquinas.

Bibliografía

- Velasco J. (2015). Máquina de estado con Verilog HDL y Quartus II.
Consultado el 1 de octubre de 2021. Enlace del video:
<https://youtu.be/vHci5NY7Q48>
- 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/discussion_de_procesadores.pdf