**Práctica 3:** Construcción de

Máquinas de estados Usando Memorias

Direccionamiento por Trayectoria

Laboratorio de Organización y Arquitectura de Computadoras

**Grupo:** 2

**Alumnos:**

* Guzmán Sánchez José Emmanuel
* Mejia Ortiz Aarón Enrique
* Sáenz Barragán Ricardo

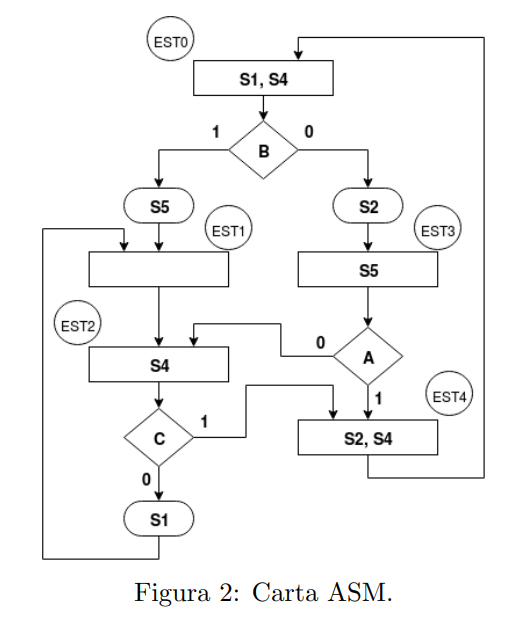
Objetivo

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

**Dispositivo**: MAX10 DE-Lite 10M50DAF484C7G

1. Desarrollo

Para la Implementación del direccionamiento por trayectoria es necesario llenar toda la memoria de acuerdo a la carta ASM que se proporcione. A cada estado se le dio una código y adjunto a cada estado van las posibles combinaciones de entradas que pueden existir, en nuestro caso tenemos 3 entradas y por lo tanto a cada estado le corresponde 8 posibles combinaciones según las entradas.



|  |  |
| --- | --- |
| **Estados** | **Código** |
| EST0 | 000 |
| EST1 | 001 |
| EST2 | 010 |
| EST3 | 011 |
| EST4 | 100 |

Por lo tanto se creo la siguiente tabla con las direcciones de memorias que accederán al contenido de la memoria, la siguiente tabla se tradujo a hexadecimal y con eso se cargo la información en la memoria:

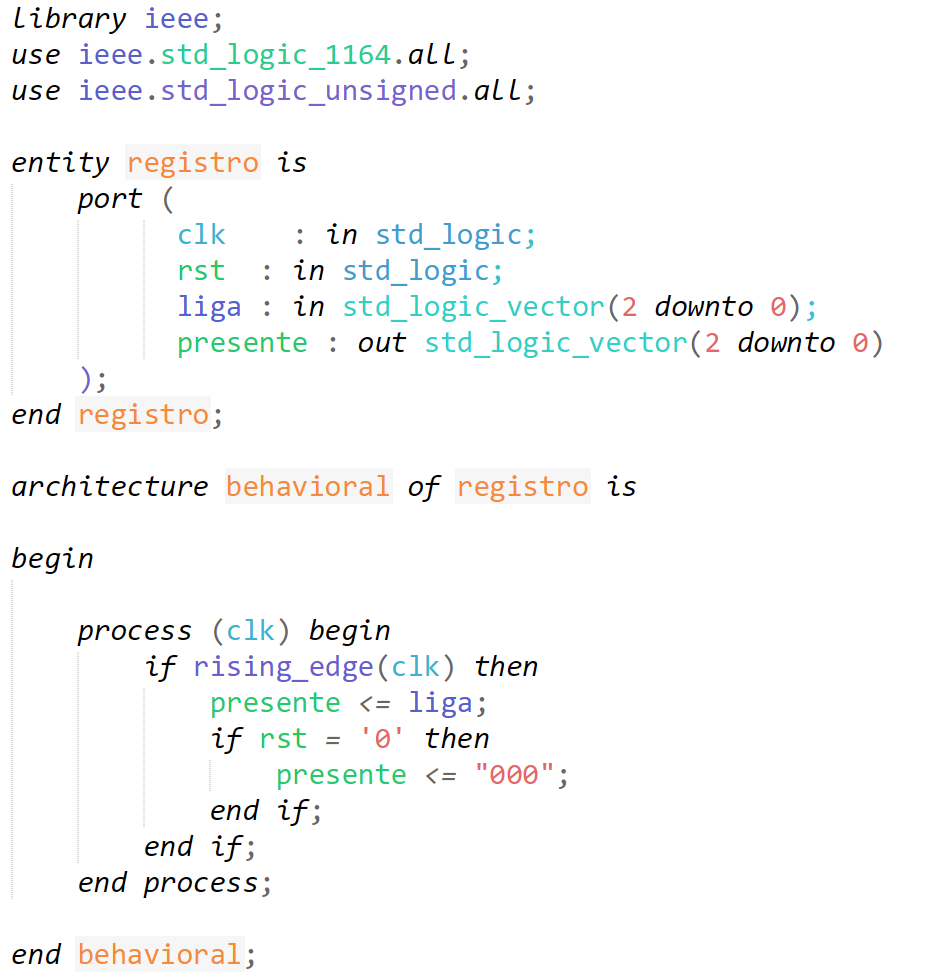
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dirección de Memoria** | | | | | | | **Contenido de la Memoria** | | | | | | | |
| Presente | | | Entrada | | | | Liga | | | Salidas | | | | |
| P3 | P2 | P1 | A | B | C |  | L2 | L1 | L0 | S5 | S4 | S3 | S2 | S1 |
| 0 | 0 | 0 | 0 | 0 | 0 |  | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 |  | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 |  | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 |  | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 |  | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 |  | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 |  | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 |  | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 |  | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |  | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |  | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 |  | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |  | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 |  | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 |  | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 |  | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 |  | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |  | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 |  | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |  | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |  | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |  | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 |  | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |  | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 |  | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |  | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |  | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |  | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 |  | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |  | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 |  | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |  | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |  | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |  | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |  | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |  | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |  | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |  | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Una vez con la tabla, podemos asegurar que los datos correspondientes a las entradas y en los estados correctos sean direccionados al estado siguiente estado correspondiente en la carta ASM. Con la tabla podemos ver las entradas y los estados siguientes con mayor facilidad.

Ahora solo queda definir el comportamiento que tendrá la memoria ROM en VHDL. El contenido de la memoria será escrito por completo según la tabla, se tendrá que llenar sin importar que se usen todos los espacios de memoria para la maquina de estados.



Después de recoger los valores de la memoria, definimos cómo se irá asignando el calor del estado siguiente según la liga recibida y las entradas que se encuentran encendidas. También colocaremos un input que será usado para reiniciar el estado de la maquina al primer estado:



Con los bloques creados anterior mente y algunos que se usaron en prácticas pasadas, podemos implementar nuestro circuito por medio de diagramas de bloques y conexiones. Se usara un divisor de frecuencia para el cambio de estado (Como reloj) y un bloque extra que nos ayudara a visualizar las salidas dependiendo si queremos el estado presente o lo que se encuentre en la memoria en ese momento:

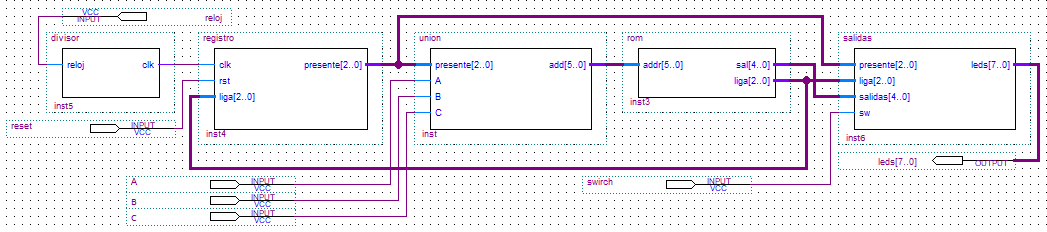
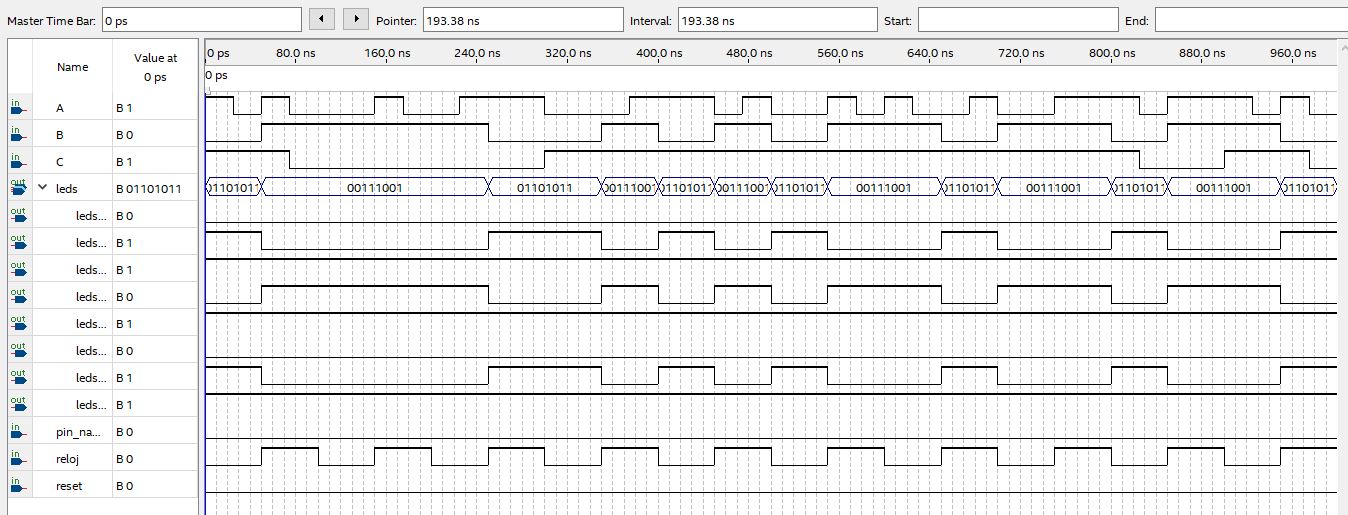


Diagrama de bloques del sistema

1. Simulaciones



1. Ejecución del programa en FPGA
2. Conclusiones

Guzmán Sánchez José Emmanuel

Esta práctica nos permitió conocer y comprender el comportamiento interno de una memoria al momento de direccionar. Si bien, el método que ocupamos es uno de los más sencillos, es el más ineficiente en cuanto a memoria, ya que por cada combinación que se tiene aumenta el tamaño. Conforme aumenta el número de entradas también aumenta el tamaño de la memoria, pero es una buena aproximación para comprender bien el concepto.

Mejia Ortiz Aarón Enrique

Sáenz Barragán Ricardo

Con está práctica se consiguió ver un caso físico para la creación de una maquina de estados de direccionamiento por trayectoria. Al usar los leds como salidas y switches como entradas se consiguió entender mejor el funcionaban de las maquinas de estados, no solo quedarse con la teoría y pruebas de papel, es bueno ver como podemos crear los módulos y conectarlos para crear lo visto en clase.

También se comprendió como este tipo de direccionamiento ocupa mucha memoria y no es lo más optimo pata la implementación de las maquinas, por lo mismo que muchos espacios de memoria no se utilizaban o simplemente el llenado de la memoria es muy tardado y poco eficiente.