

Laboratorio #06

Este laboratorio será trabajado de forma individual y se entregará de forma digital de acuerdo a la fecha de entrega en Canvas. Haga los ejercicios **SIN** usar una calculadora (a menos que se le indique lo contrario). Deberá identificar su entrega con su nombre, carné y sección.

Ejercicio 01

Implemente la siguiente Máquina de Estados Finitos

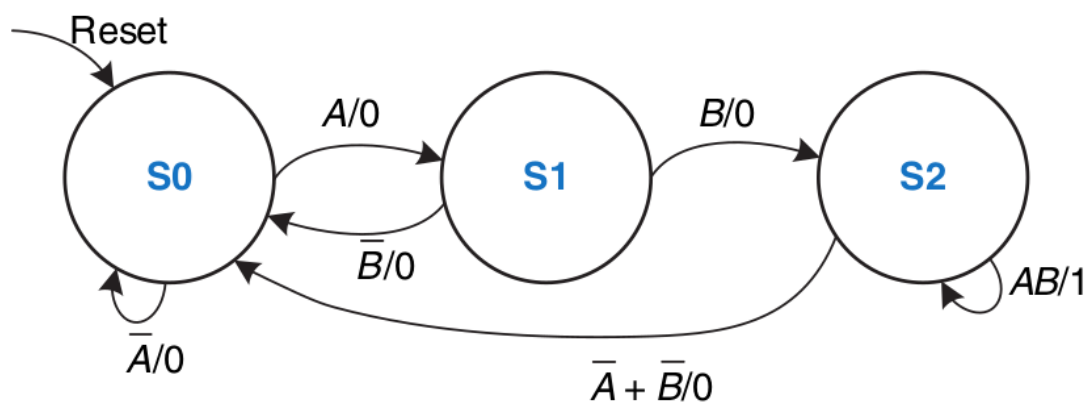


Figure 1: Diagrama Ejercicio 01

Su solución debe incluir:

1. Caja Negra
2. Tabla de transiciones de estado sin codificar
3. Tabla de transiciones de estado codificada
4. Screenshots de Logic Friday resolviendo las tablas
5. Ecuaciones booleanas
6. Implementación completa en CircuitVerse.

Ejercicio 02

Re-dibuje el diagrama del Ejercicio 01 como una FSM de Moore. **Únicamente haga el diagrama.**

Ejercicio 03

Diseñe una FSM de Moore que represente un contador Gray de 3 bits módulo 8. Un contador cambia de valor en el flanco de reloj. La escala de Gray tiene la característica que sólo cambia un bit a la vez. Un contador módulo N cuenta de 0 hasta $N-1$ y luego vuelve a empezar. El conteo debe ser de la siguiente forma:

Número	Código Gray
0	000
1	001
2	011
3	010
4	110
5	111
6	101
7	100

Adicionalmente su contador debe tener una entrada de 1 bit (UP/\overline{DOWN}) que indique la dirección de conteo (hacia arriba o hacia abajo). Si la entrada = 1 entonces el contador aumenta. Si es 0 entonces el contador disminuye.

Su solución debe incluir:

1. Diagrama de transiciones de estado (implementado en una herramienta como <https://app.diagrams.net/>)
2. Caja Negra
3. Tabla de transiciones de estado sin codificar
4. Tabla de transiciones de estado codificada
5. Screenshots de **Logic Friday** resolviendo las tablas
6. Ecuaciones booleanas
7. Implementación completa en CircuitVerse.

Ejercicio 04

Lea la sección 4.5.4 (toda la sección) de su libro (página 205) y explique en sus propias palabras qué es **non-blocking assignment**, cuál es la diferencia entre **non-blocking** y **blocking** assignment y en qué situaciones debe utilizarse cada uno. Incluya ejemplos (no es necesario hacer un testbench).

Ejercicio 05

Implemente un Flip Flop tipo D de 4 bits con un *reset* asíncrono y un *set* síncrono en Verilog. Diseñe un *testbench* para probar su código.

Ejercicio 06

Implemente los ejercicios 01 y 03 en Verilog. Recuerde que Verilog sólo es una descripción en código de su circuito. Diseñe un testbench para probar su máquina completa. Incluya **screenshots** de su diagrama de timing en su entrega en Canvas. Recuerde que debe subir su código a su repositorio en GitHub y agregar el link del repo en su entrega en Canvas.