

Bitácora del Proyecto Lógica Combinatoria: Calculadora tomógrafo

Curso: Fundamentos de Arquitectura de Computadores

Integrantes: Bryan Stiphen Feng Feng

Fecha de inicio: 11 de marzo de 2025

Repositorio Git: <https://github.com/Feng672/L-gica-Combinatoria-Calculadora-tomografo>

1. Objetivos

- Diseñar un circuito combinatorio que realice una suma de 4 bits
- Mostrar el resultado de la suma en un display de 7 segmentos mediante un decodificador
- Diseñar un actuador que se habilite en dos rangos de valores lógicos no contiguos
- Conectar un arreglo de sensor al circuito combinatorio
- Diseñar un visualizador de LEDs conectado a los sensores

2. Materiales y herramientas

Componentes	Cantidad
DIP switches	1
Compuerta XOR (74LS86)	1
Compuerta AND (74LS08)	1
Decodificador 74LS47	1
Protoboard	1
Arduino (Fuente de 5V)	1
Display 7 segmentos (cátodo común)	1
Vibration motor	1
Transistor TIP120	1
Flip Flop D (74LS175)	
Diodo 1N4007	1
Jumpers	20+
Resistencias 220Ω	5
Resistencias 100Ω	5
Resistencias 2000Ω	4
Resistencia 1000Ω	1
Botón	1

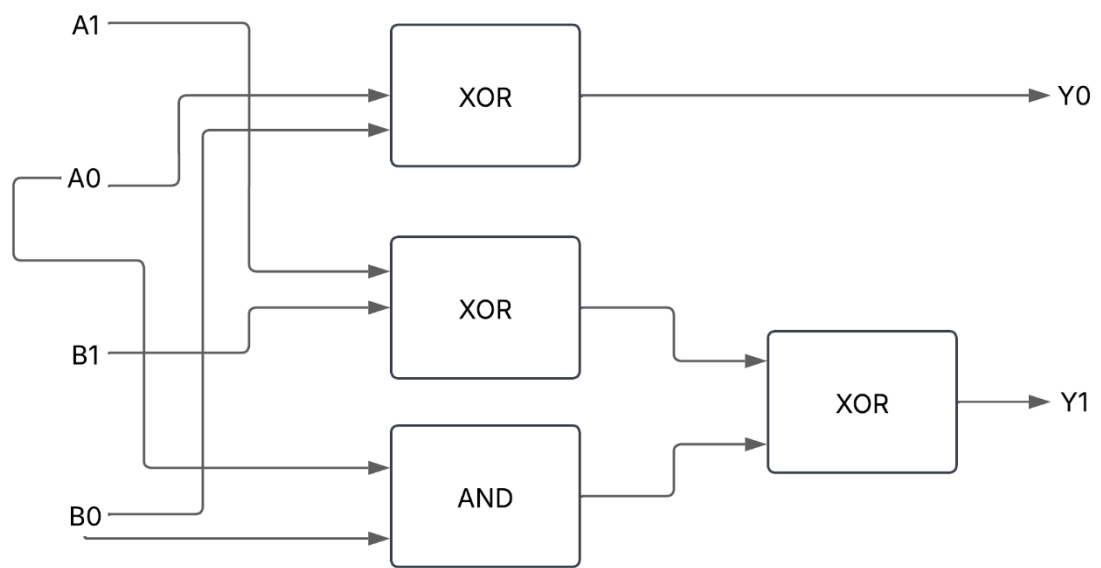
3. Registro diario

Día 1: Diseño lógico

- Tareas:

- Definir el diseño del circuito combinatorio
- Definir las tablas de verdad
- Simplificar ecuaciones
- Resultados

Mediante la herramienta de Lucidchart se diseñó el circuito combinatorio. Esta tiene 4 entradas y da como resultado la suma de $A1A0+B1B0$. El bit menos significativo se obtiene haciendo un XOR de los bits A0 y B0. Para obtener el bit más significativo se tiene que hacer un XOR del resultado del XOR de A1 y B1; y del resultado de un AND de A0 y B0.



Las tablas de verdad quedarían:

XOR			
A0	B0	Y0	
	0	0	0
	0	1	1
	1	0	1
	1	1	0

XOR			
A1	B1	S1	
	0	0	0
	0	1	1
	1	0	1
	1	1	0

AND			
A0	B0	S2	
	0	0	0
	0	1	0
	1	0	0
	1	1	1

XOR			
S1	S2	Y1	
	0	0	0
	0	1	1
	1	0	1
	1	1	0

De esta manera la ecuación para obtener Y1 y Y0 sería

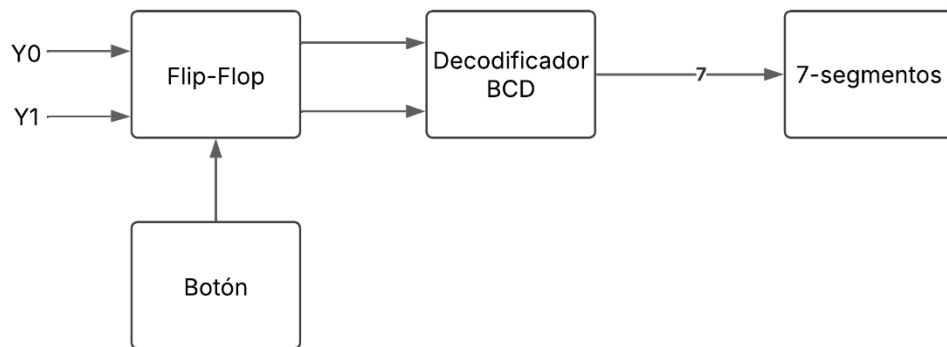
$$A0 \oplus B0 = Y0$$

$$(A1 \oplus B1) \oplus (A0 \cdot B0) = Y1$$

Día 2: Diseño 7 segmentos

- Tareas:
 - Diseñar el módulo BCD
 - Conectar el módulo BCD al visualizador 7 segmentos
- Resultados:

Nuevamente con Lucidchart se diseñó la conexión a los 7 segmentos. Las salidas del circuito combinacional se conectarán con el flip-flop, el cual tendrá un botón controlando su reloj; por lo cual sus dos salidas solo se actualizarán cuando el botón envíe una señal y no cuando las entradas cambien. Luego el decodificador recibirá las dos entradas y las convertirá en 7 bits para representar en un 7 segmento el número en decimal, del 0 al 3.

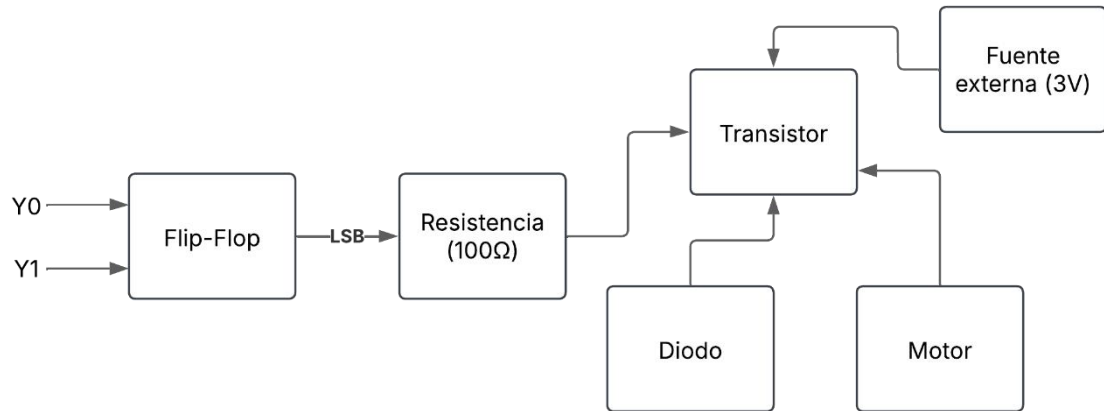


Día 3: Diseño del motor

- Tareas:
 - Elegir una lógica para que el motor se encienda
 - Diseñar el desacople
 - Diseñar el módulo del motor
- Resultados:

Se creó el diagrama con la herramienta de Lucidchart. En el diseño se reutilizará el flip-flop del módulo del BCD. En donde esta vez se utilizará la salida del bit menos significativo y con una resistencia en serie se conectará a la base del transistor. Se conectará el diodo y el motor en paralelo, en donde estarán conectados con el colector del transistor y la tierra del circuito en general. También se conectará una fuente externa al colector del transistor. Este diseño permite que dependiendo de la señal que reciba la base del transistor, la corriente del colector pasará al emisor de forma que activa el motor. Este diseño hará que el motor se active cuando el número mostrado en el 7 segmento sea un 0 o un 2.

Observación importante, en teoría el transistor es un NPN entonces debería activar el motor cuando la base reciba corriente, sin embargo, el transistor presenta un funcionamiento inverso y solamente activa el motor cuando no recibe corriente en la base.



Día 4: Diseño de las entradas

- Tareas:
 - Diseño de las entradas del circuito combinatorio
 - Elección de pull down o pull up para los switches y botón
- Resultados:

Utilizando Lucidchart se crearon los dos diagramas. El dip switch a utilizar tiene 7 bits, por lo que solo se utilizará el 1, 3, 5 y 7. Tanto para el dip switch como para el botón se va a hacer un pull down para evitar señales flotantes y solo entregue 1 o 0.

