Bitácora del Proyecto: Lógica Combinatoria, Calculadora Tomógrafo

José Eduardo Campos Salazar c.2023135620
Jimmy Feng Feng c.2023060347

Fundamentos de Arquitectura de Computadores
Instituto Tecnológico de Costa Rica
10 de abril de 2025

1. Registro de Actividades

1.1. Fecha: 18/03/2025

Actividades:

1. Se definieron las especificaciones del sistema.
2. Se diseño los módulos necesarios, asi como sus entradas y salidas respectivas del caso.
3. Se creó la tabla de verdad para el encoder de 4 bits a 2 bits.

Desarrollo y Resultados:

Figura 1. Diagrama de bloques del circuito planteado.

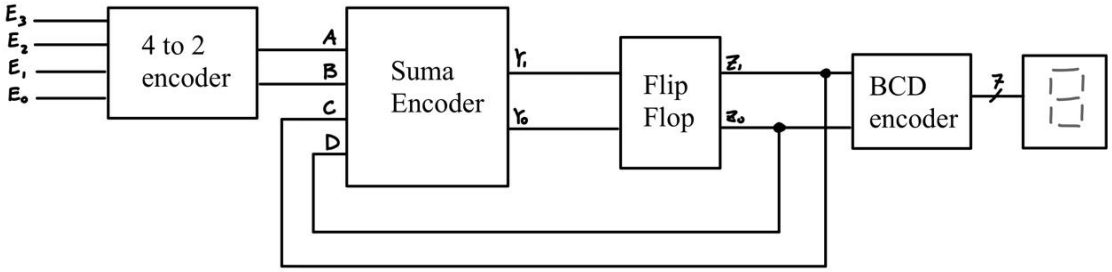


Tabla 1. Tabla de verdad de Encoder 4:2

Entrada	Y1	Y0
0001	0	0
0010	1	0
0100	0	1
1000	1	1

1.2. Fecha: 20/03/2025

Actividades:

1. Se obtuvo la ecuación canónica de las salidas del encoder de 4 bits a 2 bits.
2. Se implementó el diseño del circuito basado en la ecuación obtenida.

Desarrollo y Resultados:
Con la Tabla 1, se generan los siguientes k-maps:

Figura 2. K-map para Y0.

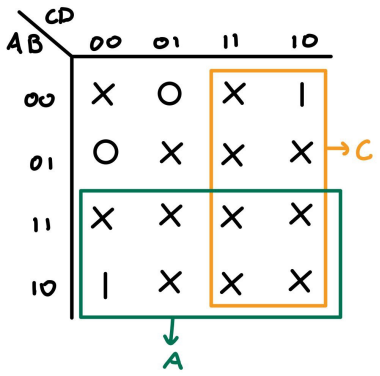
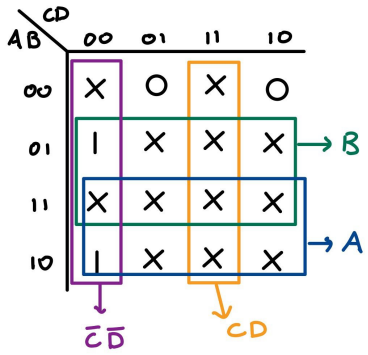


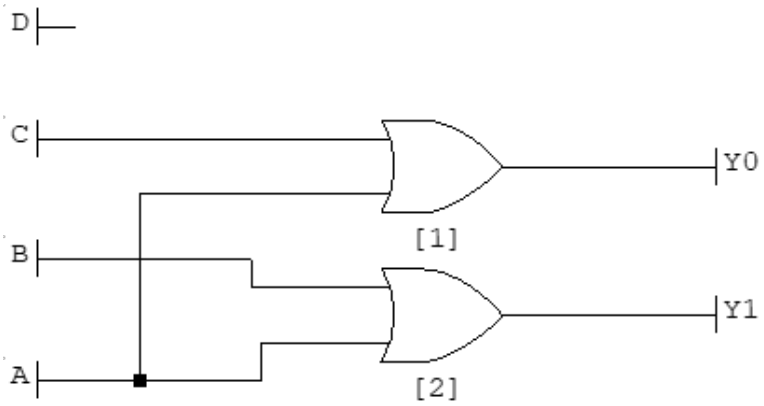
Figura 3. K-map para Y1.



Ecuaciones canónicas del encoder de 4 bits a 2 bits:

$$Y_1 = A + B$$
$$Y_0 = A + C$$

Figura 4. Circuito del encoder 4bits a 2bits.



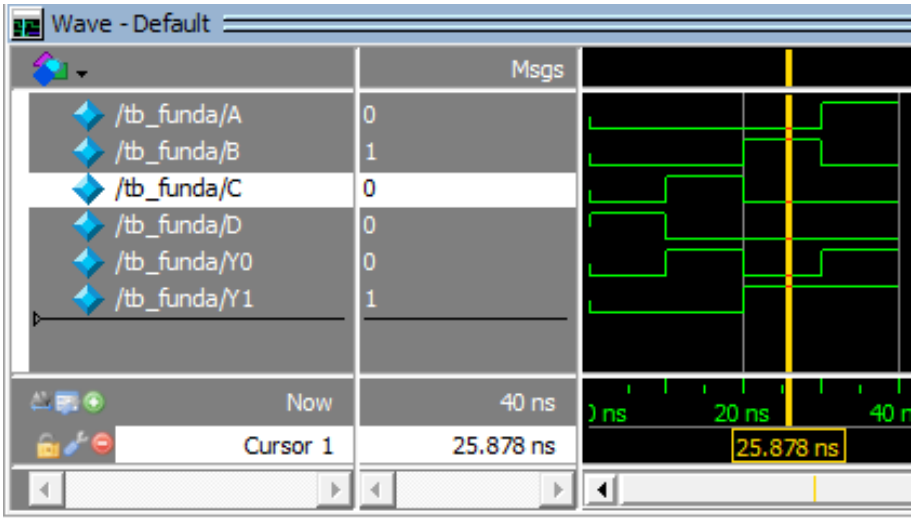
1.3. Fecha: 21/03/2025

Actividades:

1. Se verificó las ecuaciones canónicas del encoder de 4 bits a 2 bits en ModelSim en Quartus.
2. Se construyó la tabla de verdad para el sumador circular.
3. A partir de la tabla de verdad, se obtuvo la ecuación canónica con ayuda de *Logic Friday*.

Desarrollo y Resultados:
Para validar las ecuaciones canónicas, se simuló el circuito de la Figura 4 en ModelSim en Quartus 18.1. Los resultados son:

Figura 5. Simulación de la Figura 4.



Como se puede observar, cada salida coincide con la Tabla 1.

Por otra parte, se generó la tabla de verdad correspondiente al sumador circular y se extrajeron las ecuaciones lógicas necesarias para su implementación.

Tabla 2. Tabla de verdad del sumador circular de dos entradas de 2bits

AB	CD	Y1	Y0
00	00	0	0
00	01	0	1
00	10	1	0
10	11	1	1
01	00	0	1
01	01	1	0
01	10	1	1
01	11	0	0
10	00	1	0
10	01	1	1
10	10	0	0
10	11	0	1
11	00	1	1
11	01	0	0
11	10	0	1
11	11	1	0

Utilizando el software de *Logic Friday*, se simplificaron las ecuaciones de la Tabla 2:

Figura 6. Simplificación de ecuaciones del sumador circular.

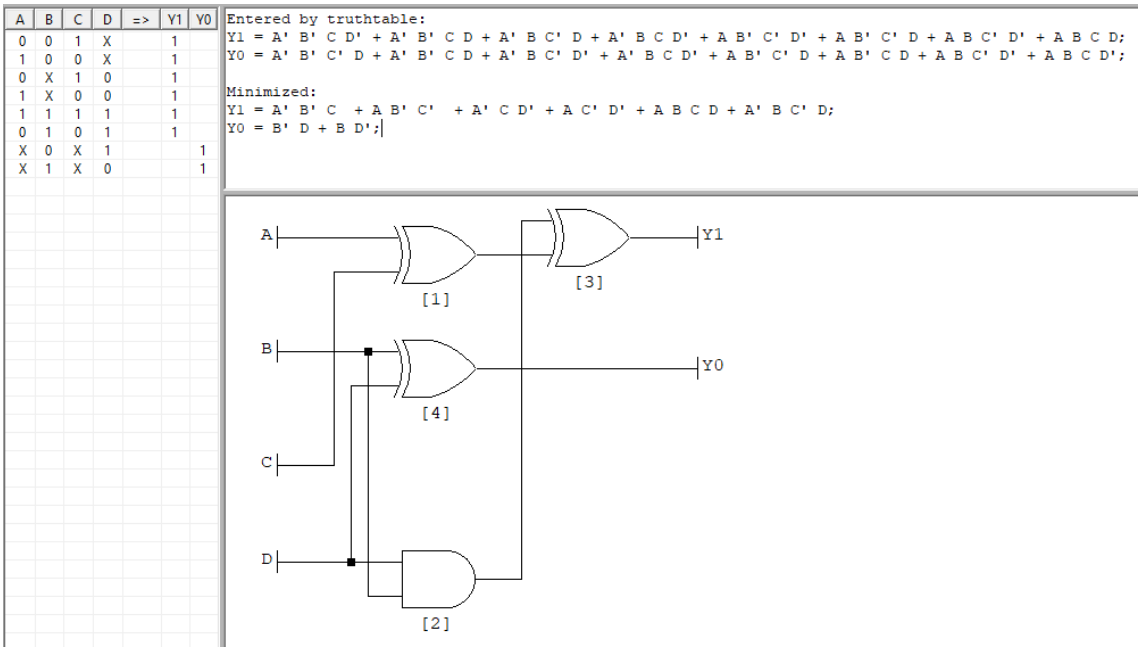
```
Minimized:
Y1 = A' B' C  + A B' C'  + A' C D' + A C' D' + A B C D + A' B C' D;
Y0 = B' D + B D';
```

Para Y0, la expresión se puede simplificar aún más con la propiedad de XOR, por lo que la ecuación canónica para Y0 es:

$$Y_0 = B \oplus D$$

Para Y1, es una expresión bastante grande, por lo cual se ocuparían varias compuertas. Sin embargo, el software también permite convertir una expresión booleana al equivalente en circuito lógico, especificando las compuertas que se utilizan:

Figura 7. Circuito de la Tabla 2.



Analizando la Figura 7, se puede obtener una expresión simplificada de Y1 en términos de XOR y AND:

$$Y_1 = (A \oplus C) \oplus (BD)$$

1.4. Fecha: 23/03/2025

Actividades:

1. Se comprobó la ecuación canónica obtenida previamente.
2. Se simuló las ecuaciones en ModelSim en Quartus.
3. Se diseñó el circuito completo, con el BCD encoder a 7 segmentos.

Desarrollo y Resultados:
Para confirmar la expresión, se analizó por medio de una tabla de verdad:

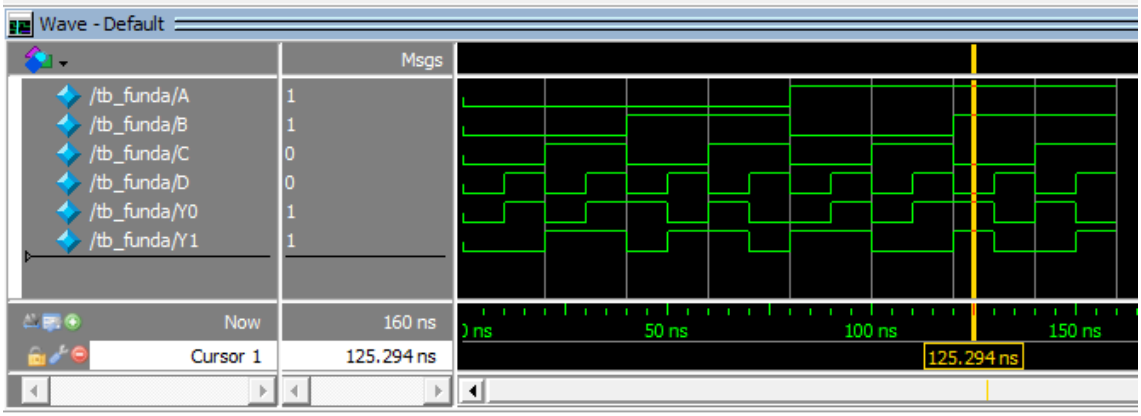
Tabla 3. Comprobación Y1

AB	CD	BD	$A \oplus C$	$(A \oplus C) \oplus (BD)$
00	00	0	0	0
00	01	0	0	0
00	10	0	1	1
10	11	0	1	1
01	00	0	0	0
01	01	1	0	1
01	10	0	1	1
01	11	1	1	0
10	00	0	1	1
10	01	0	1	1
10	10	0	0	0
10	11	0	0	0
11	00	0	1	1
11	01	1	1	0
11	10	0	0	0
11	11	1	0	1

Como se puede observar, la columna $(A \oplus C) \oplus (BD)$ es idéntica a la columna Y1 de la Tabla 2. Por lo tanto, queda comprobada la ecuación Y1.

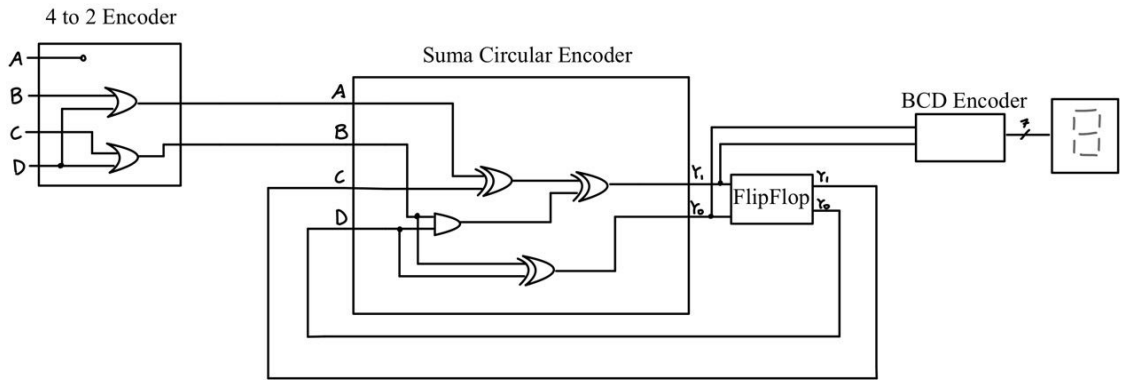
Además, para validar ambas ecuaciones, se simuló en ModelSim en Quartus 18.1. Para esto, se asignaron como entradas: A, B, C y D y a las salidas Y1 y Y0, se les asignaron las ecuaciones canónicas.

Figura 8. Simulación circuito de la Figura 7.



Analizando cada posible entrada, coinciden con las salidas de la Tabla 2.

Figura 9. Circuito final.



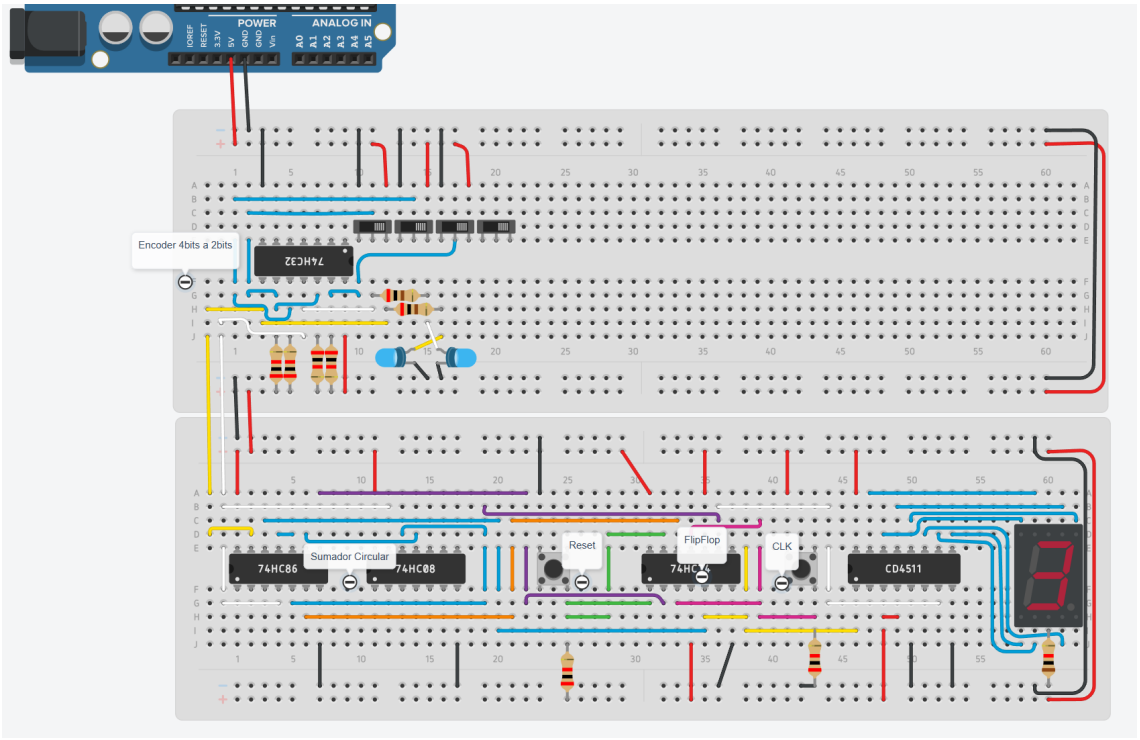
1.5. Fecha: 24/03/2025

Actividades:

1. Se diseñó el circuito de la Figura 9 en tinkercad.
2. Con ayuda del simulador, se probaron varios casos para verificar su correcto funcionamiento.

Desarrollo y Resultados:
Cada etapa del circuito montado en Tinkercad funcionó correctamente.

Figura 10. Circuito final en tinkercad.



1.6. Fecha: 25/03/2025

Actividades:

1. Se planificó cómo implementar el desacople utilizando transistores BJT NPN.
2. Se implementó el desacople utilizando transistores BJT NPN.

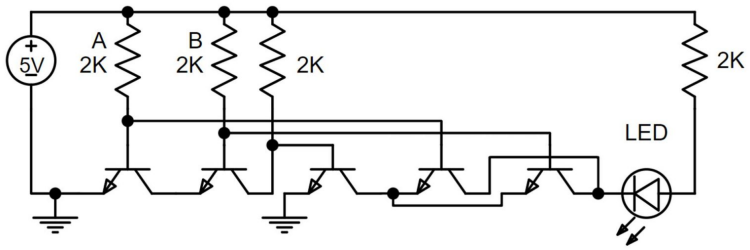
Desarrollo y Resultados:
Esta etapa habilita un actuador en dos valores lógicos. En este caso, se habilitará en el rango de 1_{10} a 2_{10} y se desactiva en 0_{10} y 3_{10}. Esto coincide con la operación XOR, con la siguiente tabla:

Tabla 4. TT XOR

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

De acuerdo con [1], se crea una compuerta XOR utilizando transistores BJT como en la siguiente figura:

Figura 11. Compuerta XOR con transistores BJT.



1.7. Fecha: 26/03/2025

Actividades:

1. Se implementó el desacople en tinkercad, junto al accionador.
2. Se integró al resto del circuito de la Figura 10.
3. Con el simulador, se probó si cumple con el funcionamiento descrito en la Tabla 4.

Desarrollo y Resultados:

Con la ayuda de la Figura 11, se diseña una compuerta XOR con BJT en tinkercad. Para este proyecto, se requiere un accionador, entonces se cambia el LED por un motor conectado en serie y una bateria para alimentar el motor.

Además, se modificó el circuito de la Figura 10, cambiando los switches por fotoresistencias.

El desacople funciona correctamente, donde el accionador se activa en el rango de 1_{10} a 2_{10} y se desactiva en 0_{10} y 3_{10}.

Figura 12. Implementación de la Figura 11 junto con un accionador.

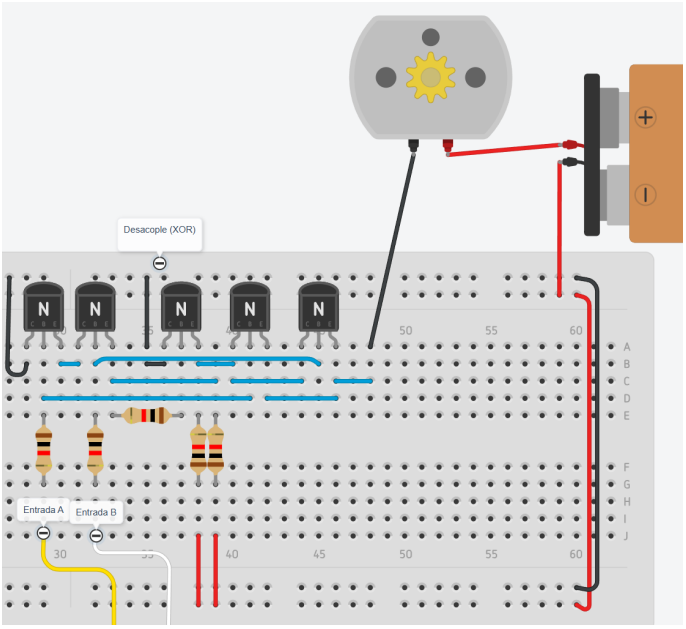
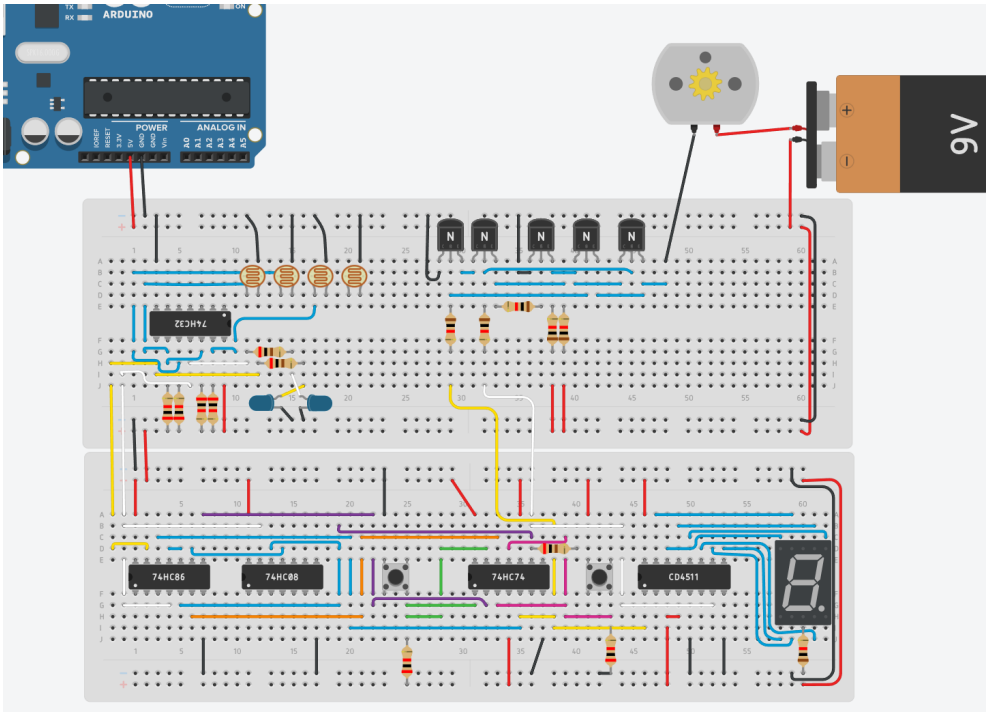


Figura 13. Circuito final junto con el desacople y accionador.



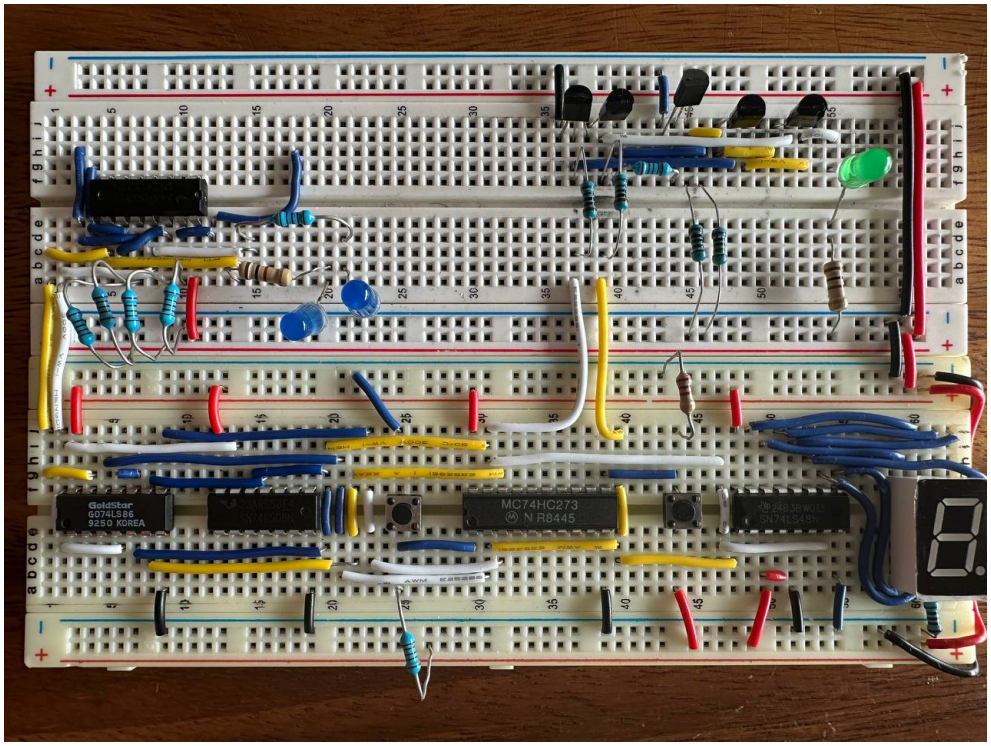
1.8. Fecha: 27/03/2025

Actividades:

1. Se compró los componentes necesarios para armar el circuito de la Figura 13.
2. Con los componentes, se armó el circuito.

Desarrollo y Resultados:

Figura 14. Circuito de la Figura 13 montado en protoboard sin motor.



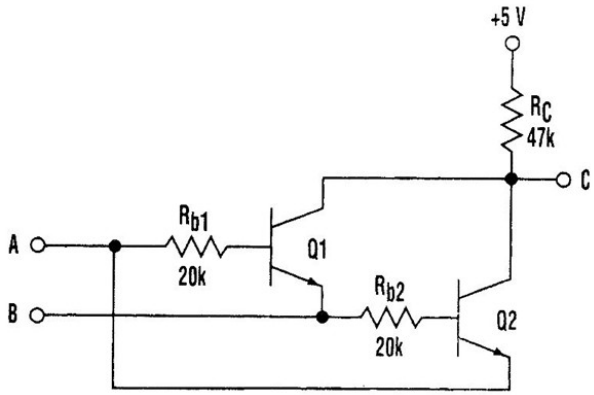
1.9. Fecha: 28/03/2025

Actividades:

1. Se modificó el circuito de desacople.
2. Se verificó el funcionamiento del circuito montado.

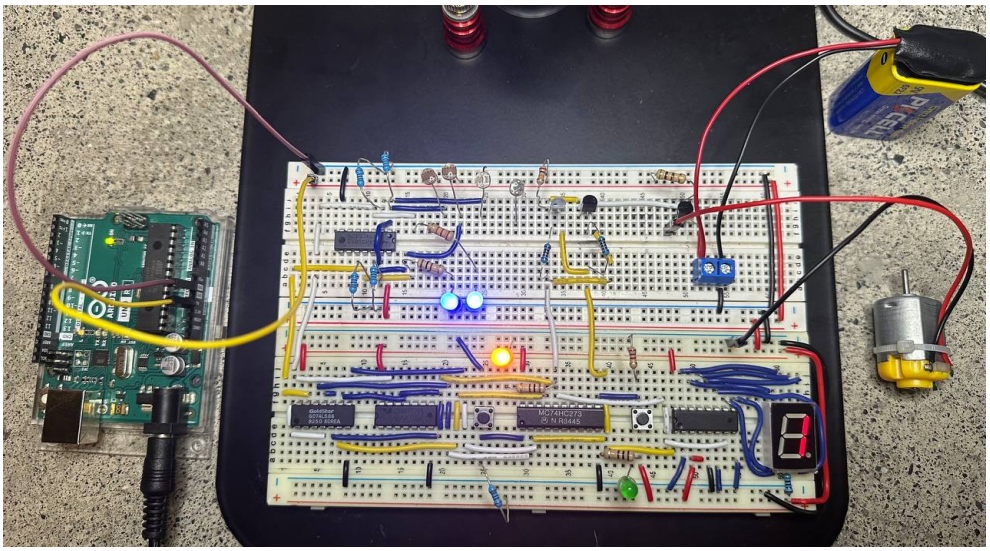
Se tuvo un error, donde el circuito de desacople de la Figura 11 daba la salida del voltaje en negativo, lo cual afectaba el control del motor. Por esta razón, se optó por utilizar otra implementación de XOR utilizando sólo dos transistores BJT. [2]

Figura 15. Circuito XOR con 2 BJT.



Implementando en la protoboard, quedó de esta manera:

Figura 16. Circuito montado con desacople de la Figura 15.



1.10. Fecha: 29/03/2025

Actividades:

1. Se realizaron ultimas pruebas en el circuito en protoboard.
2. Se realizó una revisión final de la documentación del proyecto.

2. Link al repositorio de la bitácora

https://github.com/JimF04/Bitacora_Proyecto1_CE1107

Referencias

[1] GS Network, *XOR Gate – Exclusive OR Gate Truth Table, Symbol, and Circuit Diagram*, Accessed: 25-Mar-2025, 2025. dirección: https://www.gsnetwork.com/xor-gate/.

[2] H. User, *Bipolar XOR Gate with Only 2 Transistors*, Accessed: 2025-03-31, 2018. dirección: https://hackaday.io/project/8449-hackaday-ttlers/log/150147-bipolar-xor-gate-with-only-2-transistors.