Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computadores
Fundamentos de Arquitectura en Computadores

Bitacora Proyecto 1: Lógica Combinatoria: Calculadora tomógrafo

Profesor: Luis Alberto Chavarría Zamora

Estudiantes:
Gabriel Vargas López
Fabiola Meléndez Sequeira

I Semestre 2025
Marzo 2025

Semana 1 y 2

Fecha inicio: 11 Marzo 2025
Fecha finalización: 25 Marzo 2025

Fecha: 11 de marzo

Asignación del proyecto 1.

Fecha: 14 de marzo

Primera reunión por medio de discord. Se discute lo que se trabajará en las primeras semanas y se llega al acuerdo de utilizar las primeras 2 semanas para profundizar en los conceptos a trabajar. Además se deberán repasar los conceptos vistos en clase como mapas de Karnaugh, tablas de verdad, sumas de productos, productos de suma, entre otros.

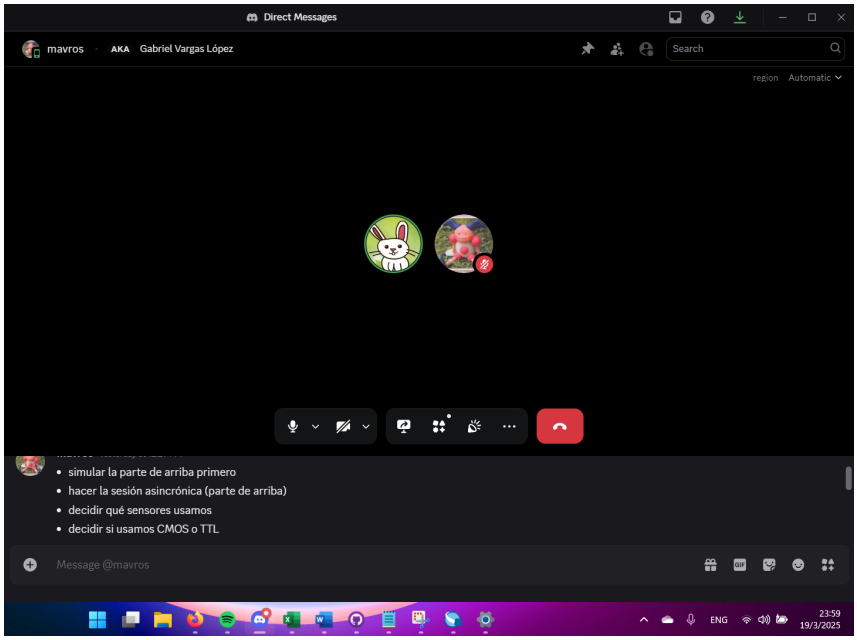

Figura 1: Primera reunión

Fecha: 23 de marzo

Anteriormente señalado, en estas primeras semanas los estudiantes se dedicaron a entender cada uno de los puntos a desarrollar, investigar so-

bre cada uno de los compontentes y comprender la integración total del circuito.

Dentro de los puntos más importantes que se trataron fueron:

- ¿Que es un decodificador?

  Un decodificador tiene N entradas y 2^n salidas. Activa exactamente una de sus salidas dependiendo de la combinación de entrada.

- ¿Como pasar de 4 a dos bits?

  Para pasar de 2^n entradas a N salidas, se debe de utilizar un codificador. La función del codificador es la contraría del decodificador, por lo que entonces su proceso va de la misma mano.

- ¿Que es un BCD?

  Es una manera de representar los números binarios en decimal. Cada número en decimal se puede representar con 4 bits en binario. [3]

- ¿Como representar números de 2 bits en un 7 segmentos?

  El 7 segmentos está dividido por 7 LEDs, cada uno representado por una letra como se observa en la imagen [2]

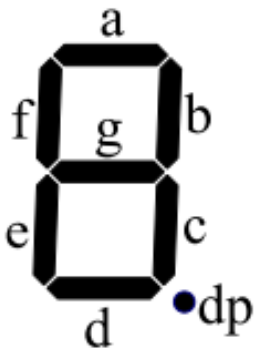

Figura 2: Letras del display del 7 segmentos

Cada una de estas letras tiene asignada un bit con el que se enciende cuando es 0. En la tabla se puede demostrar cómo denotar cada

número del 0 al 3, la que son los números que se utilizaron en el taller [1]

Decimal	g	f	e	d	c	b	a
0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	1
2	1	0	0	0	0	0	0
3	1	0	0	0	0	0	0

Cuadro 1: Representación de números en un 7 segmentos

- ¿Como se puede sumar en un decodificador?
  Se debe obtener la tabla de verdad de las entradas y su resultado, luego obtener el Mapa de Karnaugh para obtener la expresión lógica de cada bit de la respuesta.

Semana 3

Fecha inicio: 25 Marzo 2025
Fecha finalización: 1 Abril 2025

Fecha: 24 de marzo

Se realiza una segunda reunión para asignar las tareas de cada uno de los integrantes del grupo. Se acuerda que Fabiola comenzará trabajando en el codificador, una vez listo y probado, Gabriel tomará este como referencia para que el decodificador pueda realizar la suma. Al ser esta una de las partes más confusas se acuerda realizar una segunda reunión para que ambos desarrollen la interpretación de las salidas para poder comparar resultados. Por ultimo, Fabiola tendrá a cargo la transformación de los valores al display de 7 segmentos.

Además, se plantean las tablas de verdad del codificador de 4 a 2 bits y del decodificador de la suma en un documento de excel, para faciliar su visualización y tenerlo a mano fácilmente.

Fecha: 25 de marzo

Una vez realizada la etapa de investigación, inicia el desarrollo del codificador para las entradas.

Empezando con el desarrollo del codificador de 4 a 2 bits empezaron a surgir dudas, como por ejemplo, ¿qué pasa cuando el número de la entrada esté fuera del rango de la salida? A partir de esto surge la idea del desarrollo de un codificador de 4 a 2 bits con prioridad, de manera que los valores solo funcionen dentro de un rango, mientras los otros valores son ignorados.
Despues de consultar, se llegó a la conclusión de que no se debían tomar en cuenta estos casos y que se debían más bien establecer las entradas de la manera en la que pareciera más conveniente trabajar.

Tomando en cuenta lo anterior, se plantea la tabla vista en [2].

E3	E2	E1	E0	Y1	Y0
0	0	0	0	0	0
1	0	0	0	0	1
1	1	0	0	1	0
1	1	1	1	1	1

Cuadro 2: Codificador de 4 a 2

Luego, a través del producto de suma y su síntesis según[2], se obtienen los siguientes resultados:

- Producto de suma de Y1:

  $(E3 \vee E2 \vee E1 \vee \neg E0) \wedge (\neg E3 \vee E2 \vee E1 \vee E0)$
  $= E2 + E1 + E0$

- Producto de suma de Y0:

  $(E3 \vee E2 \vee E1 \vee E0) \wedge (\neg E3 \vee \neg E2 \vee E1 \vee E0)$
  $= (E3 \oplus E2) \vee E1 \vee E0$

Para poder verificar que su funcionamiento si estuviera realizandose de manera correcta, se simuló en la herramienta Quartus para también así obtener el circuito sintetizado.

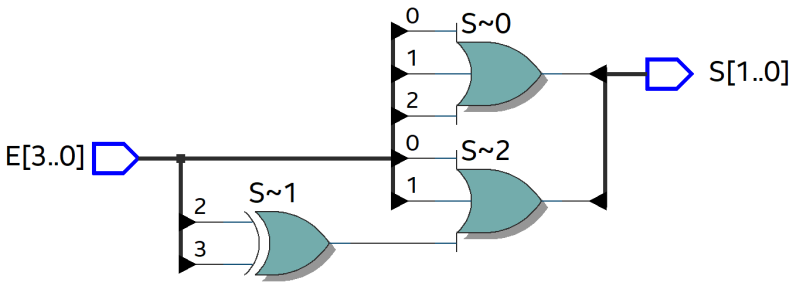

Figura 3: Circuito Codificador Sintetizado

Fecha: 26 de marzo

Posteriormente, se realizó entonces la simulación en Tinkerkad para probar su funcionamiento.

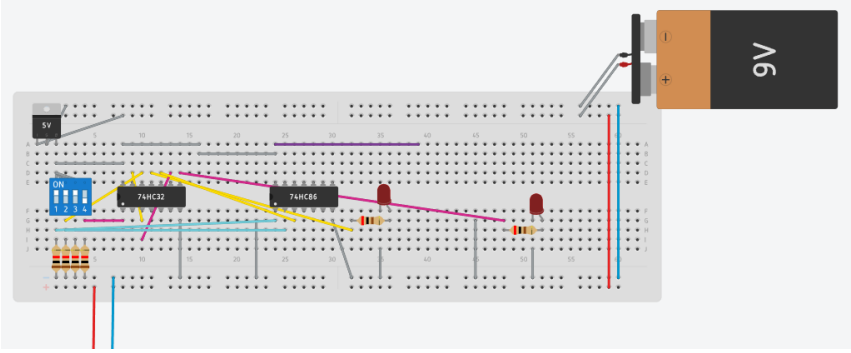

Figura 4: Simulación del Circuito Codificador

Una vez establecidas estas entradas se comienza a montar la tabla para que el decodificador pueda sumar.

Acumulado		Entrada		Resultado		Representación
A1	A0	V1	V0	R1	R0	en decimal
0	0	0	0	0	0	0+0=0
0	0	0	1	0	1	0+1=1
0	0	1	0	1	0	0+2=2
0	0	1	1	1	1	0+3=3
0	1	0	0	0	1	1+0=1
0	1	0	1	1	0	1+1=2
0	1	1	0	1	1	1+2=3
0	1	1	1	0	0	1+3=0
1	0	0	0	1	0	2+0=2
1	0	0	1	1	1	2+1=3
1	0	1	0	0	0	2+2=0
1	0	1	1	0	1	2+3=1
1	1	0	0	1	1	3+0=3
1	1	0	1	0	0	3+1=0
1	1	1	0	0	1	3+2=1
1	1	1	1	1	0	3+3=2

Cuadro 3: Tabla de verdad para el decodificador

Se realizan los mapas de Karnaugh teniendo como base la tabla [3] para obtener los bits del resultado, R1 siendo el más significativo y R0 siendo el menos significativo.

A1A0 \V1V0	V1V0			
A1A0	00	01	11	10
00	0	0	1	1
01	0	1	0	1
11	1	0	1	0
10	1	1	0	0

Cuadro 4: Mapa de Karnaugh para R1

A1A0 \V1V0 A1A0	V1V0 00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	0

Cuadro 5: Mapa de Karnaugh para R0

Siguiendo un procedimiento similar al de la tabla 2 se aplica suma de producto a las tablas de Karnaugh para facilitar el proceso de obtención de los componentes lógicos

- Suma de producto de R1:

$$R1 = \neg A0 \wedge (A1 \oplus V1) \vee \neg A1 \wedge A0(V1 \oplus V0) \vee A1 \wedge A0(V1 \odot V0)$$

- Suma de producto de R0:

$$R0 = (\neg A0 \wedge V0) \vee (A0 \wedge \neg V0) = A0 \oplus V0[XOR]$$

Para poder verificar cada uno de estos resultados se monta en Quartus para así probar todas las posibles entradas con todas las posibles salidas y también que el programa ayude a sintetizar al maximo el circuito correspondiente.

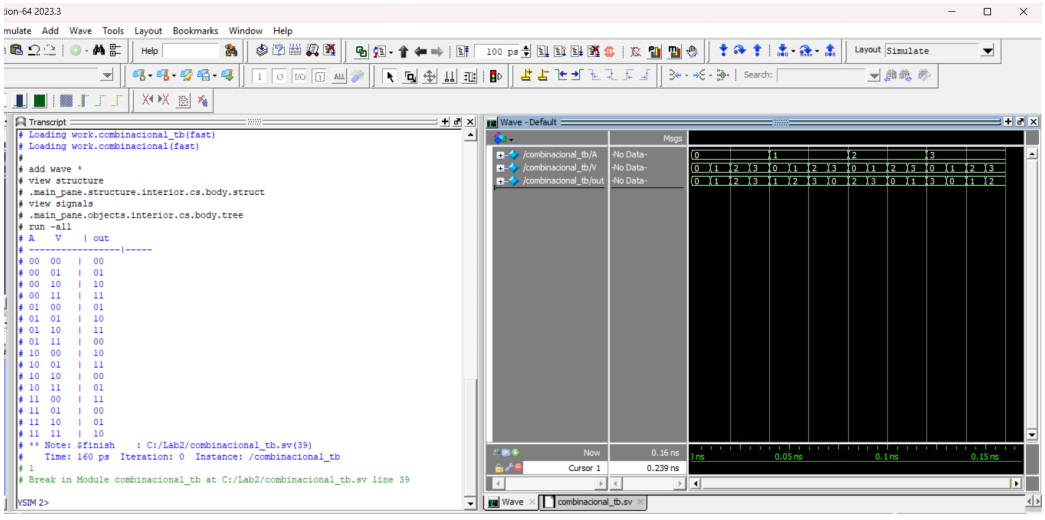

Figura 5: Test Bench para Combinaciones en el Decodificador

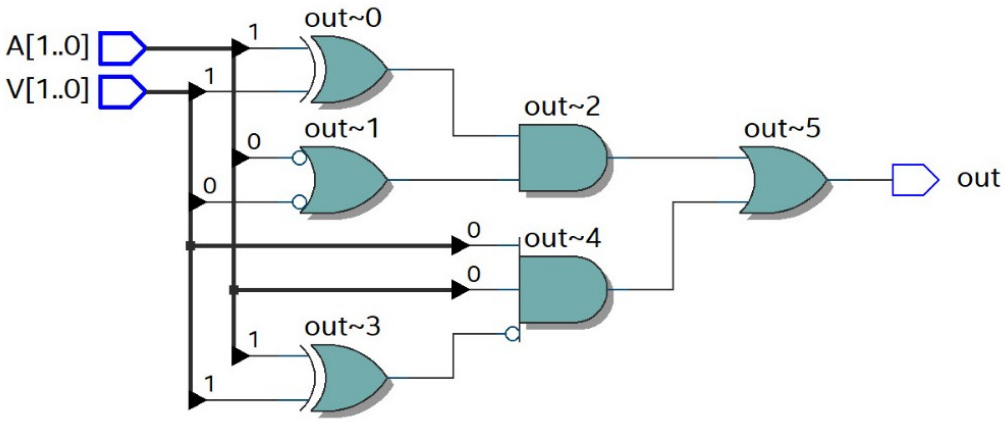

Figura 6: Diagrama lógico de S1

Fecha: 27 de marzo

Teniendo el circuito sintentizado de S1 y sabiendo que el S0 se puede obtener simplemente con un XOR, se procede a la emulación en Tinkercad para tener una mejor visualización del circuito y garantizar que el circuito funcione, ademas de preveer el daño de los componentes e instrumentos a utilizar.

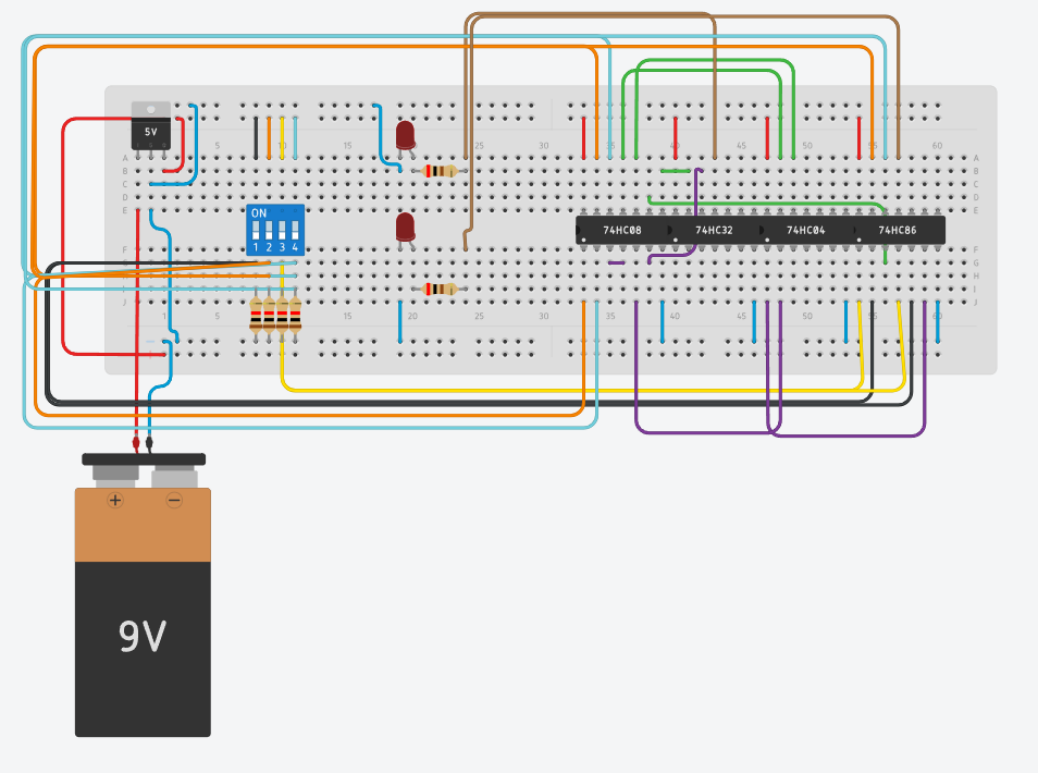

Figura 7: Representación del decodificador en Tinkercad

Para la integración del circuito con BCD, se utilizó la herramienta de Tinkercad para montarlo y una vez montado se hizo la unión del mismo con el codificador para verificar que cada uno de los valores esperados se estuviera representando de manera correcta.

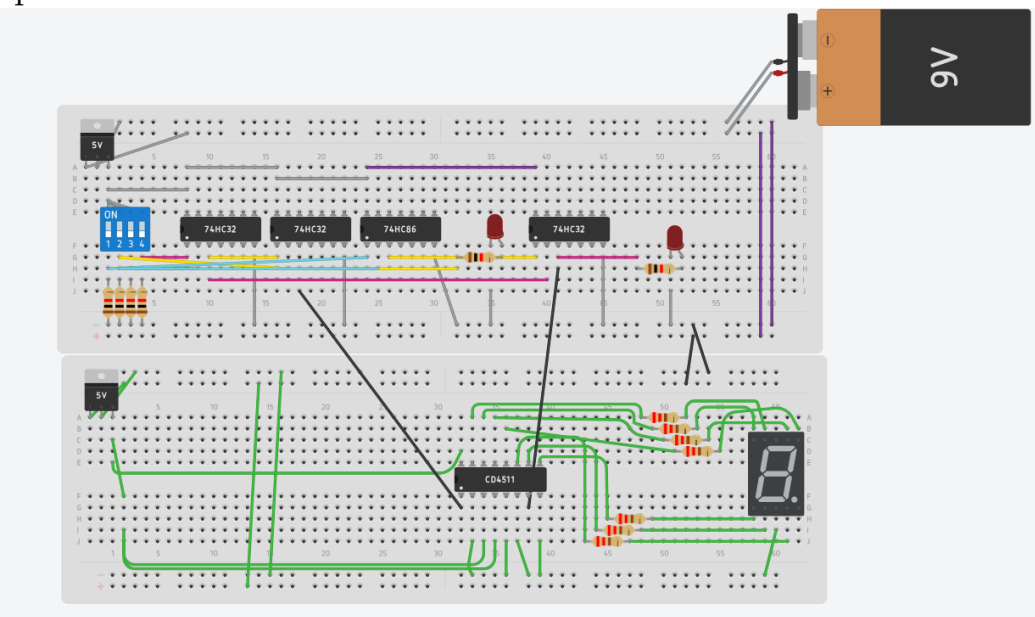

Figura 8: Integrado de BCD con el codificador

Fecha: 28 y 29 de marzo

Una vez simulados y comprobados cada uno de los circuitos se procede a hacer la compra de cada uno de los componentes. De lo anterior surgieron bastantes inconvenientes ya que en las electronicas muchos de los componentes estaban agotados, por lo que la busqueda de cada uno de ellos se volvió un poco tediosa.

Fecha: 30 y 31 de marzo

Con cada uno de los componentes comprados se procede a montar el circuito completo en la protoboard[9]. A la hora de montarlo, se encontraron distintos inconvenientes que tuvieron que ser atendidos, dificultando y atrasando el proceso.
Entre ellos se pueden mencionar:

1. La costumbre de trabajar con compuertas TTL y pasar a trabajar con compuertas CMOS provocó una alteración en la perspectiva de los pines. El nuevo órden de los pines de las compuertas a lidiar no se tomó en cuenta a la hora del montaje, lo que provocó un inconveniente al armar y corregir el circuito, invirtiendo más tiempo de lo esperado en esto.

2. Compra de compuertas equivocadas

3. Falta de cables

4. Confusión de tierras comunes

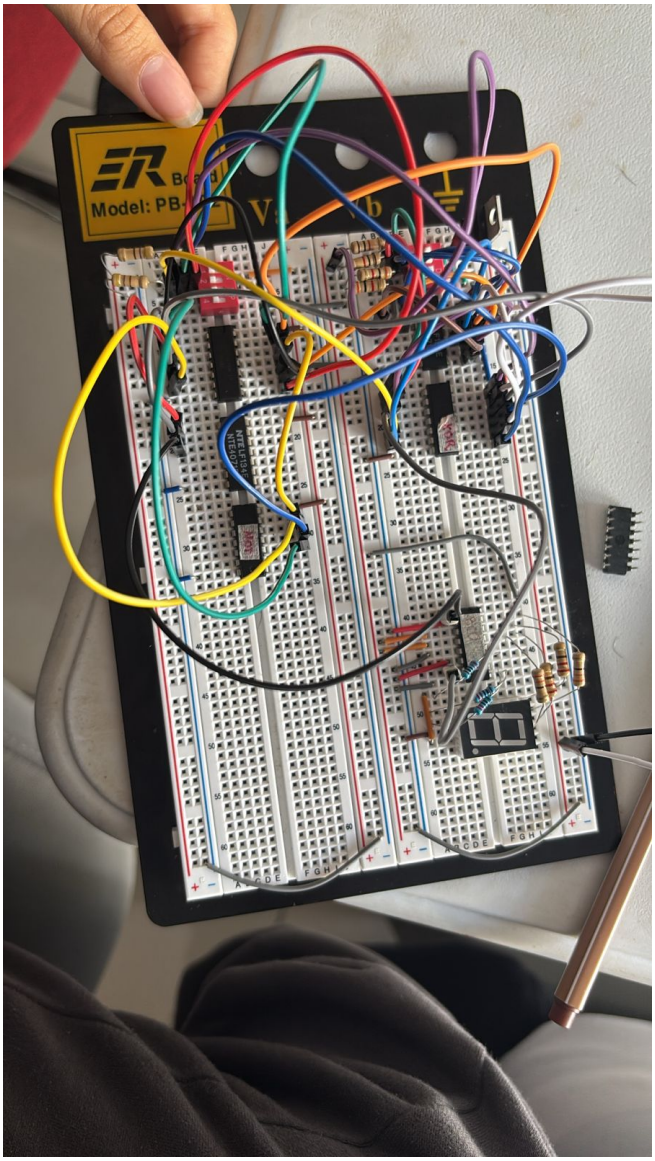

Figura 9: Circuito con Decodificador y switches

Fecha: 1 de abril

Se agrega el botón de la suma que anteriormente se había pasado por alto.

Fecha: 3 de abril

Este día se realizó una nueva reunión para revisar cada uno de los puntos restantes y así poder dividir equitativamente cada una de las tareas.

Se tomaron en cuenta nuevos problemas a resolver, como el cambio de switches a sensores, la implementación del flip-flop en el acumulado y la implementación de un motor en el desacople.

Fecha: 4 de abril

Se realiza una investigación de los flip flops y se simula un prototipo funcional en tinkercad que guarde un registro de la suma de un circuito similar al creado anteriormente.

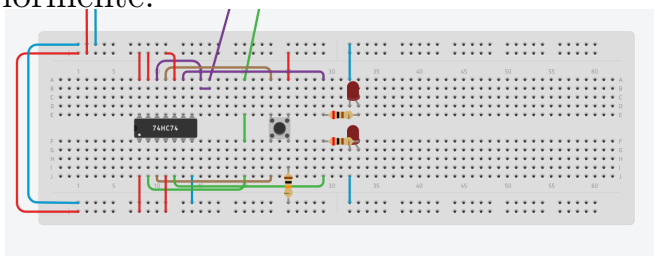

Figura 10: Flip-flop en tinkercad

Al presionar el botón (conectado al reloj del flip flop) se almacena lo que venga en la entrada D a la salida Q, como se puede observar en el siguiente diagrama

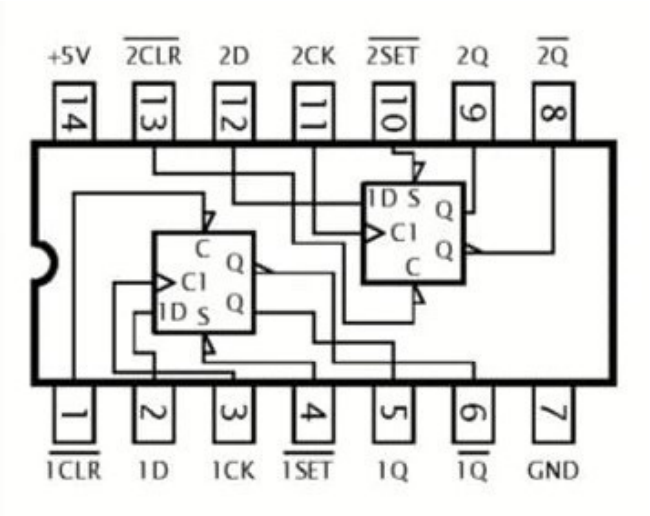

Figura 11: Datos del chip de flip-flop

Cabe destacar que el diagrama presente en al figura 11 es el diagrama del flip-flop de Tinkercad. El flip-flop utilizado en la entrega final tendrá una estructura diferente, como se verá luego.

Fecha: 5 de abril

Se termina de comprender el proposito total del decodificador 2 y se monta una tabla en excel para trabajarla [6]. Al querer que el activador funcionara en solo ciertos rango se pensó en activarlo por conveniencia según el flip-flop en los rangos de 0 y 3 por lo que entonces se llegó al uso de una compuerta xnor.

E1	E0	Y
0	0	1
0	1	0
1	0	0
1	1	1

Cuadro 6: Decodificador 2

La salida del decodificador deberá entonces estar conectada al accionador pero se deberá pasar por un desacople primeramente por el consumo de energia que tendrá el accionador. Para este desacople se utilizará un transistor, especificamente un transistor NMOS para intentar anular cualquier fuga de corriente lo mayor posible. Por conveniencia, se utiliza un motor DC como accionador y con la ayuda de la herramienta de tinkerkad se simula:

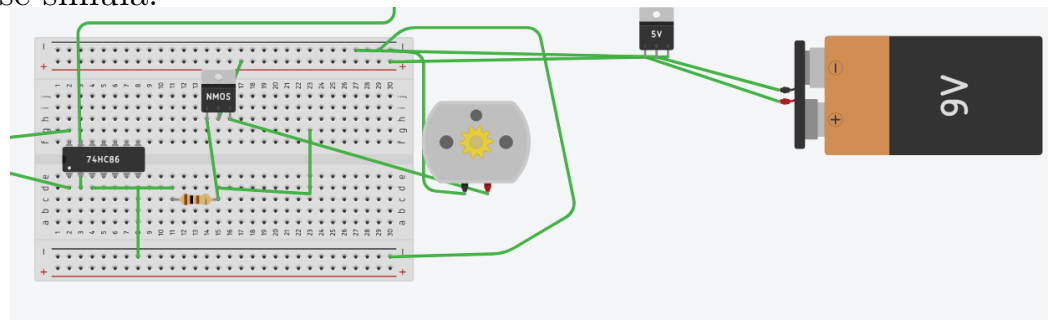

Figura 12: Decodificador 2 con desacople y motor

Fecha: 7 de abril

Se realiza la compra de los componentes y se comienza el ensamblaje

Al conectar el flip-flop se encuentra con el inconveniente de la gran diferencia entre diagramas, comparando con 13

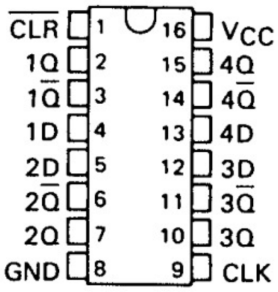

Figura 13: Decodificador 2 con desacople y motor

Esto hacía creer, cuando había un error, que era debido a esta diferencia. Finalmente la adaptación al nuevo flip-flop fue exitosa y logró cumplir su función de almacenamiento.

Fecha: 8 de abril

Al implementar los sensores, tomando en cuenta que al activarse cada sensor debe ser indicado que este fue activado con un LED, se encuentra otro inconveniente.
Finalmente se opta por la implementación de transistores BJT NPN para poder cumplir con el requisito estipulado anteriormente.

Referencias

[1] Matthew B. Akanle y Victoria Oguntosin. "A digital indicator system with 7-segment display". En: *Journal of Physics Conference Series* 1299.1 (ago. de 2019), pág. 012139. DOI: 10.1088/1742-6596/1299/1/012139. URL: https://doi.org/10.1088/1742-6596/1299/1/012139.

[2] *Boolean Algebra Calculator - eMathHelp*. URL: https://www.emathhelp.net/calculators/discrete-mathematics/boolean-algebra-calculator/.

[3] David Money Harris y Sarah L. Harris. *Digital Design and Computer Architecture*. Elsevier, ene. de 2013.