Departamento de Ingeniería de Sistemas
Arquitectura de Computadores y Laboratorio -2021-1
Laboratorio No. 2
Diseño de circuitos secuenciales

UNIVERSIDAD DE ANTIOQUIA
Facultad de Ingeniería

Integrantes:

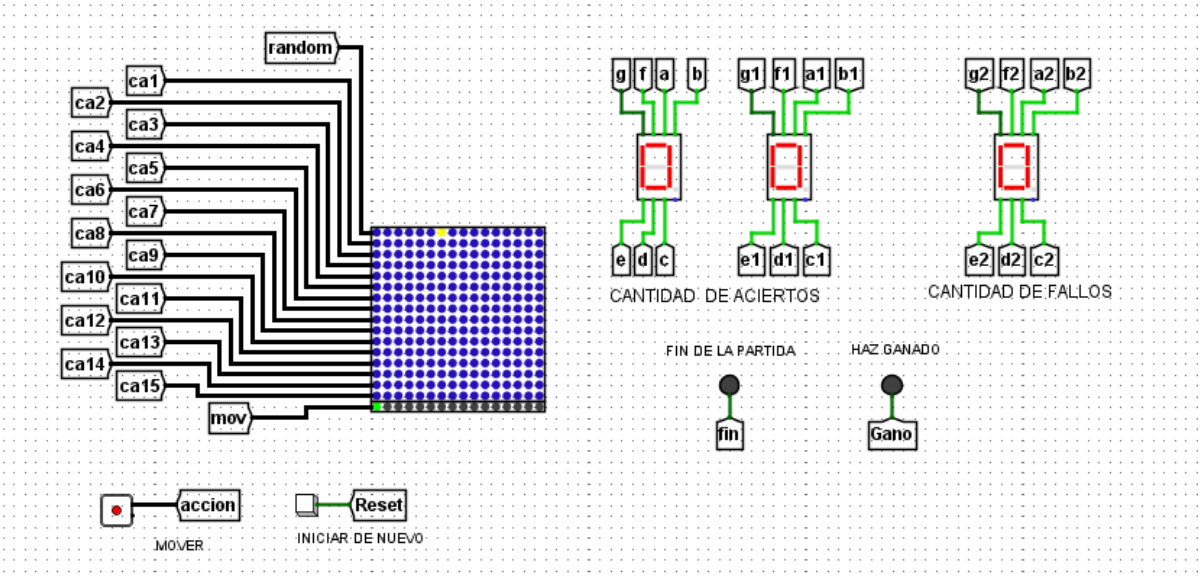
Juan Pablo Bedoya Sanchez CC. 1152691819
Valeria Granada Rodas CC. 1000763818

Objetivo:

- Emplear los conocimientos teóricos adquiridos en el curso en el proceso de diseño de sistemas secuenciales.
- Emplear herramientas de software para el diseño y la simulación de sistemas secuenciales

Descripción de la solución:

Se diseñó un sistema secuencial para un sencillo videojuego denominado “Catch a falling star” en el que el jugador tiene la posibilidad de mover horizontalmente un cursor ubicado en la parte inferior del escenario, buscando atrapar “estrellas” que caen verticalmente desde la parte superior del mismo.



Se empleó un estilo de diseño estructural y jerárquico, en el que se diseñan componentes básicos que luego son instanciados para crear otros más complejos y de mayor nivel en la jerarquía de diseño.

De acuerdo con lo aprendido en clase y mencionado en el enunciado del ejercicio planteado se dará prioridad al uso de **Flip flops JK** para la solución del problema.

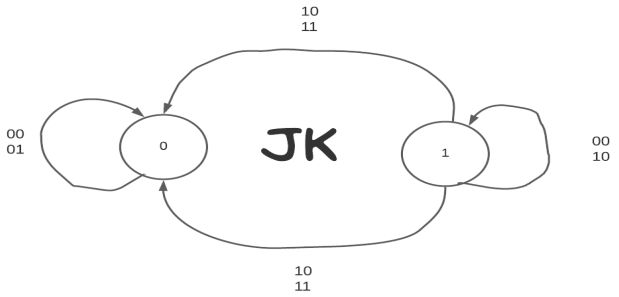
Acerca del juego:

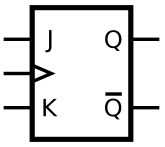
- Para ganar debes realizar 20 aciertos es decir atrapar 20 estrellas. Es importante destacar que cada 5 puntos cambiarás de nivel. Cada nivel aumentará la velocidad de caída de las estrellas, aumentando a su vez la dificultad del juego; finalmente, el juego se terminará si ganas o por el contrario dejas pasar más de 7 estrellas.
-

Informe:

- Para iniciar construiremos la tabla característica del comportamiento y diagrama de estados del flip flop JK

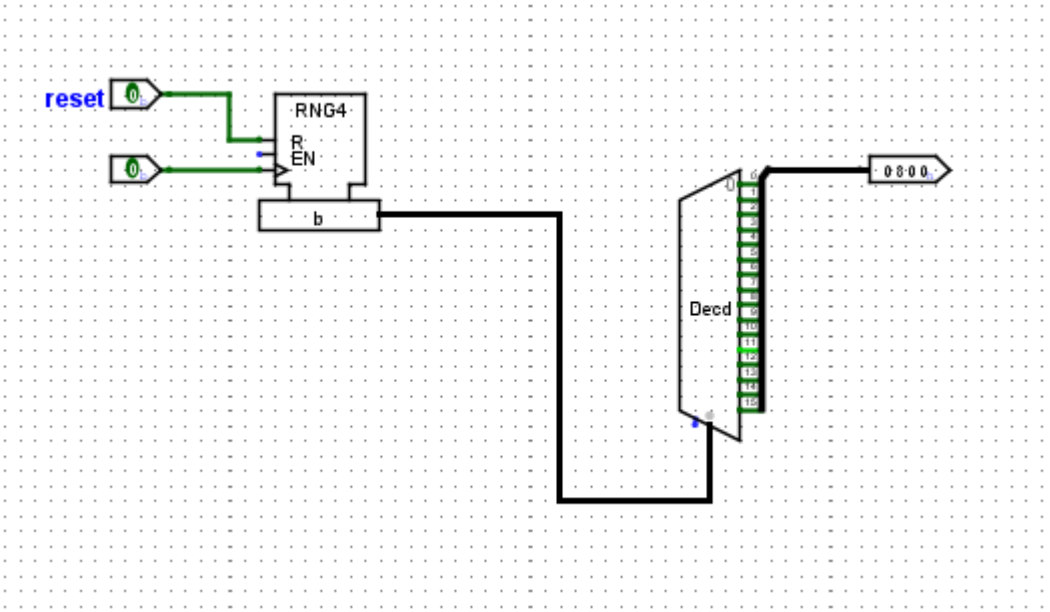
J	K	
0	0	q
0	1	0
1	0	1
1	1	q’



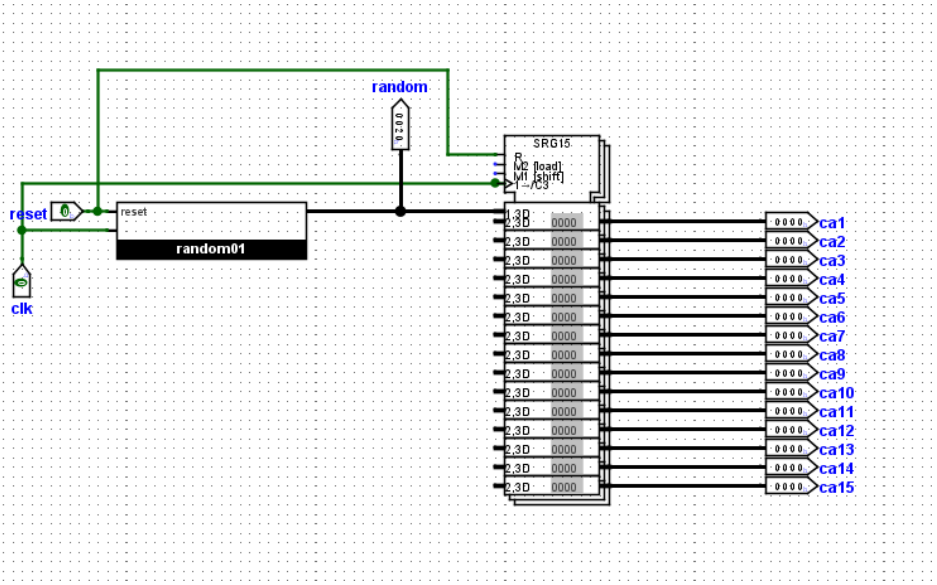


Componentes:

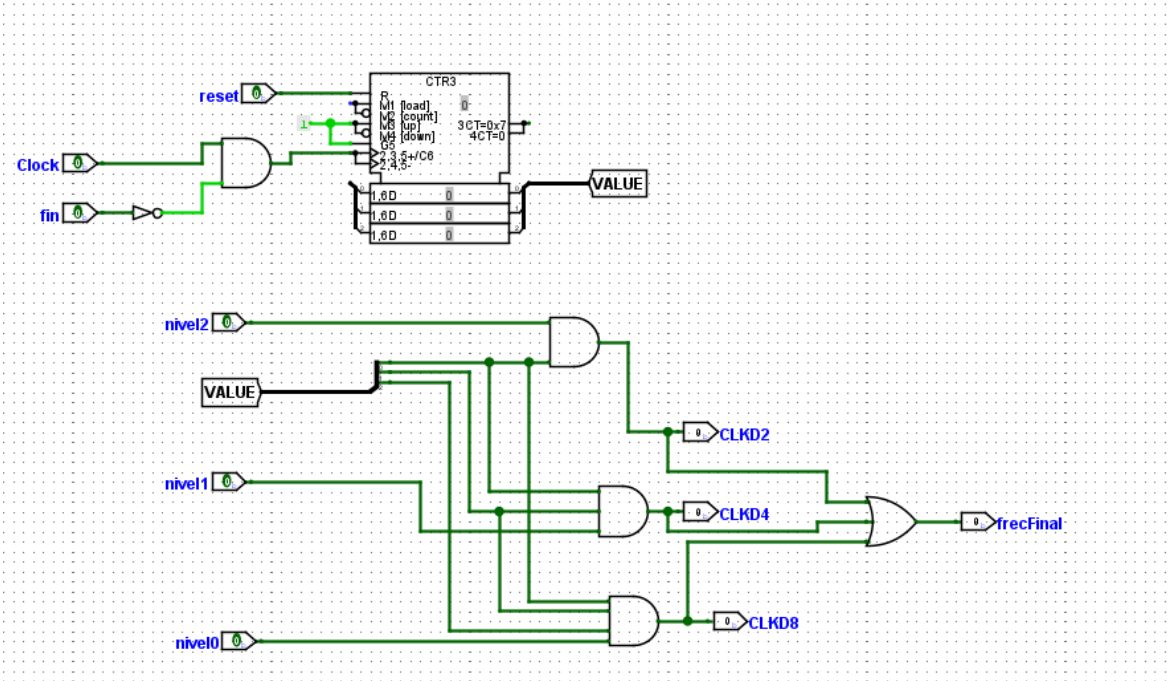
- **random01:**
  - Este componente se encarga de generar Bits aleatorios.
  -



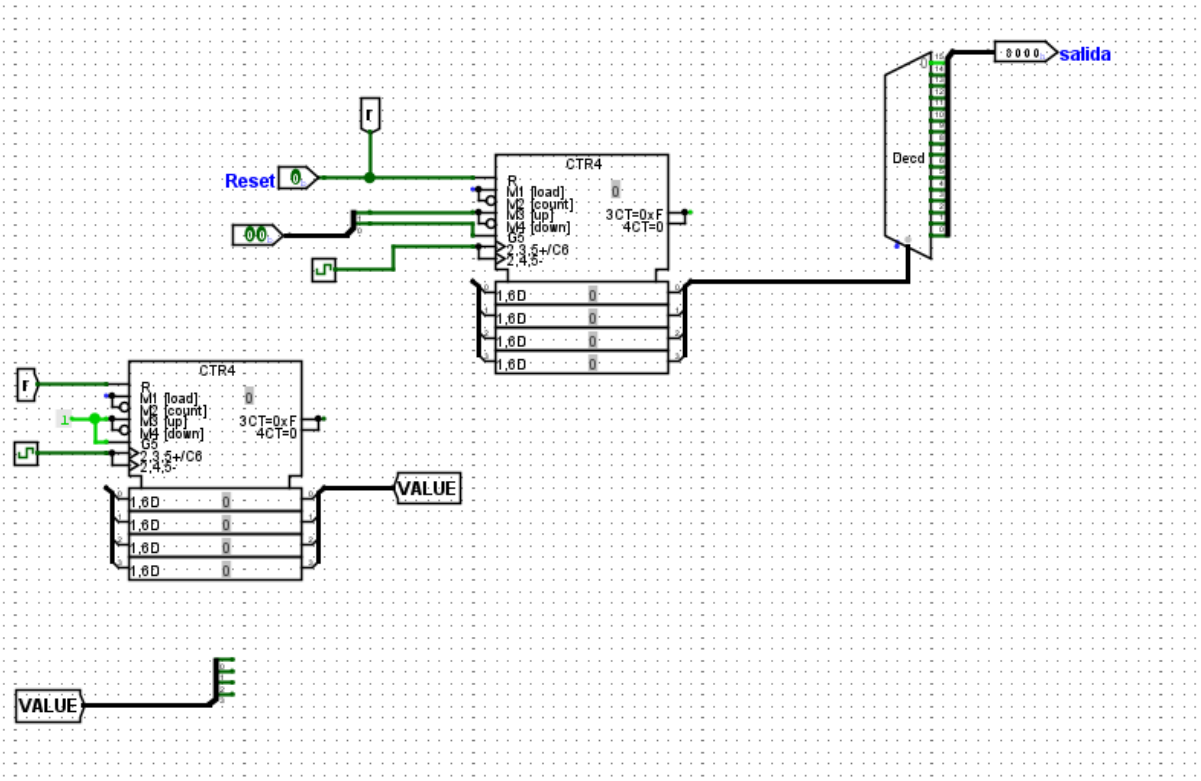
- **caidaEstrellas:**
    - Este componente usa el bloque random para provocar que la primera estrella en la malla de leds se ilumine de manera aleatoria y a su vez cuenta con un registro de desplazamiento el cual simula el desplazamiento de cada estrella que se va generando de acuerdo a la primera posición.



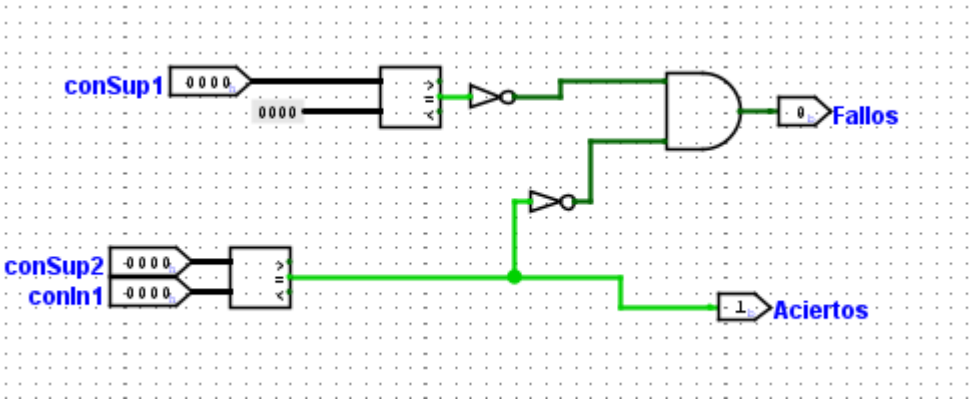
- **DivFrec:**
    - Este se encarga de recibir una señal de reloj y retorna 3 frecuencias diferentes de acuerdo a cada nivel del juego. Por ejemplo, el nivel 0 es la frecuencia indicada del reloj dividida entre 8.



- **movimiento:**
  ° Este es el encargado de generar el movimiento del mando para el personaje que atrapa las estrellas.

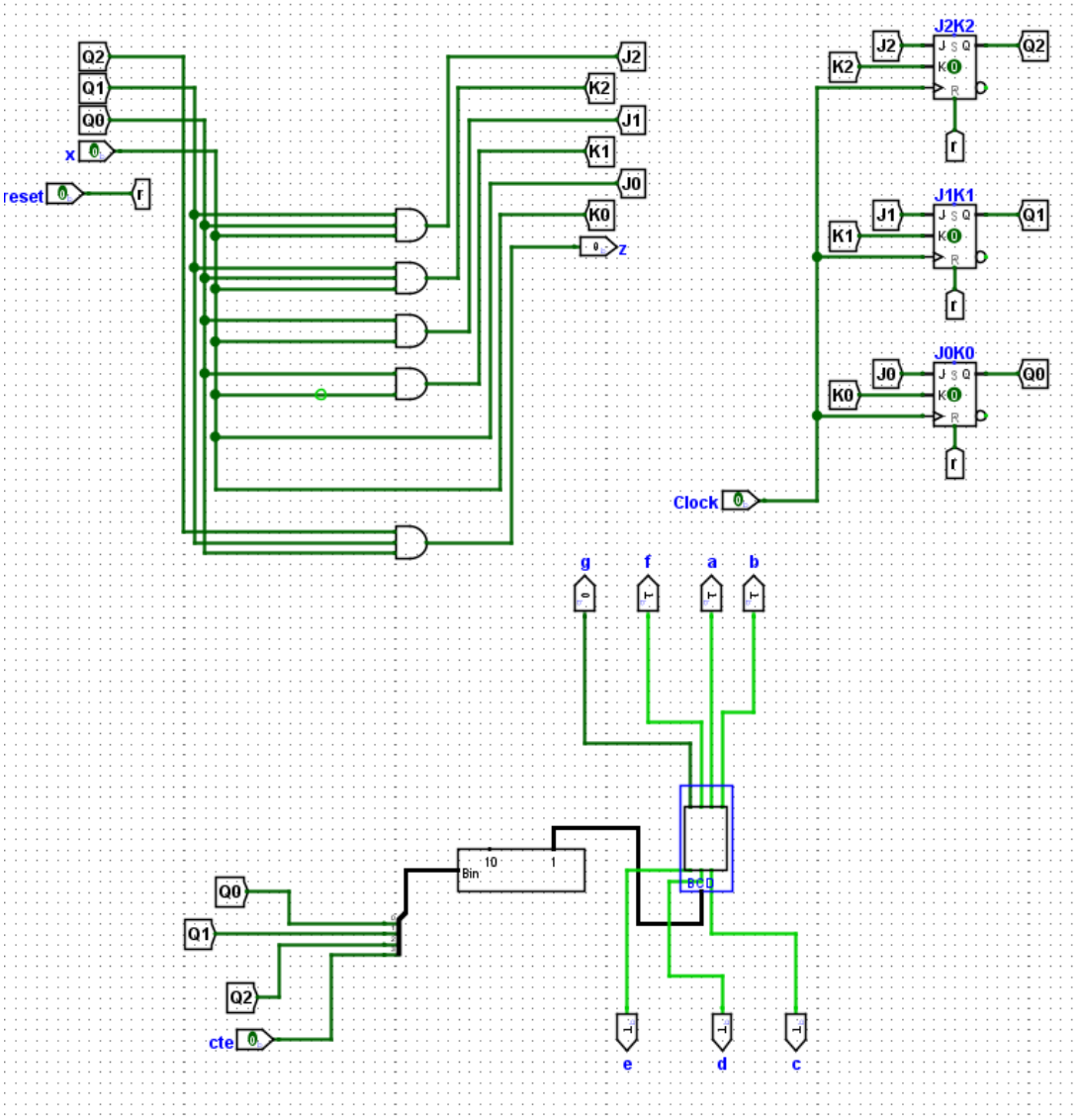


- **comparador:**
  - Este componente compara la última fila de la malla y la del mando para verificar si el momento fue un acierto o por el contrario un fallo.

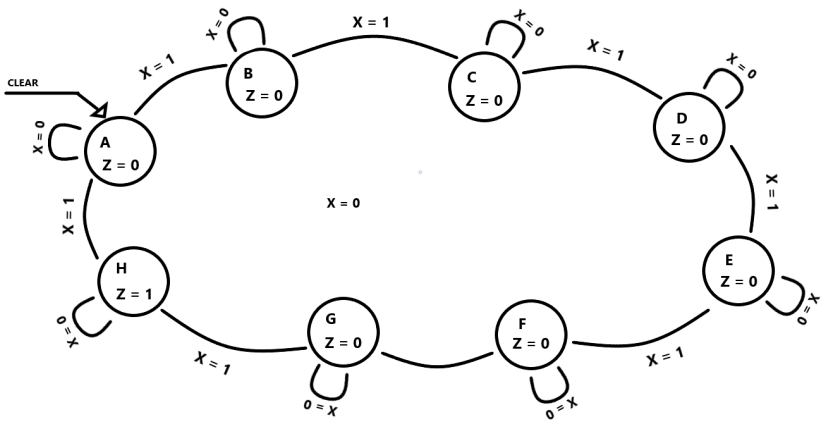


- **contadorDeFallas**
  - Hicimos uso de los flip flops JK para construir nuestro propio contador de fallas; para esto se hizo uno de las siguientes tablas de estados, diagramas de estados, diagramas

de excitación y mapas de karnaugh. De acuerdo al planteamiento del juego, fue necesario implementar 3 flip flops de este tipo. Adicionalmente, se agregó un convertidor de binario a BCD y luego otro de BCD a 7 segmentos para poder usar un display de 7 segmentos en la contabilización de las fallas.



- **Diagrama de estados:**



- **Tabla de estados y salidas:**

Estado Actual	Estado Siguiente		Salida
	X = 0	X = 1	
A	A	B	0
B	B	C	0
C	C	D	0
D	D	E	0
E	E	F	0
F	F	G	0
G	G	H	0
H	H	A	1

- **Código de Gray**

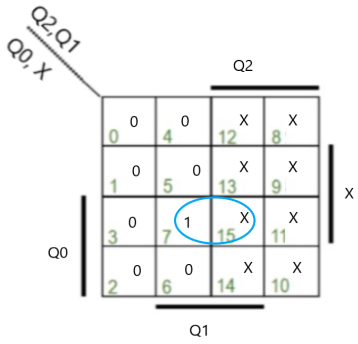
Código de Gray	
A	000
B	001
C	011
D	010

E	110
F	111
G	101
H	100

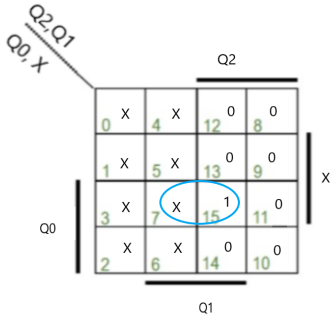
- **Tabla de estados y salidas extendida para determinar ecuaciones de excitación (de los flip flops JK) y salidas**

	Estado Actual Q2 Q1 Q0	X	Estado Siguiente Q2 Q1 Q0	Salida	J2K2	J1K1	J0K0	
A	000	0	000	0	0X	0X	0X	0
	000	1	001	0	0X	0X	1X	1
B	001	0	001	0	0X	0X	X0	2
	001	1	011	0	0X	1X	X1	3
D	010	0	010	0	0X	X0	0X	4
	010	1	110	0	0X	X0	1X	5
C	011	0	011	0	0X	X0	X0	6
	011	1	010	0	1X	X1	X1	7
H	100	0	100	0	X0	0X	0X	8
	100	1	000	0	X0	0X	1X	9
G	101	0	101	0	X0	0X	X0	10
	101	1	100	0	X0	1X	X1	11
E	110	0	110	0	X0	X0	0X	12
	110	1	111	0	X0	X0	1X	13
F	111	0	111	1	X0	X0	X0	14
	111	1	101	1	X1	X1	X1	15

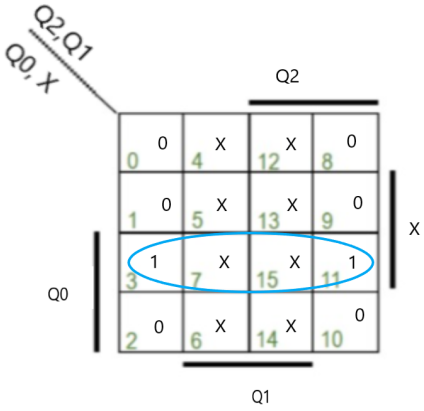
- **Mapas de karnaugh para encontrar las ecuaciones de excitación y salida**



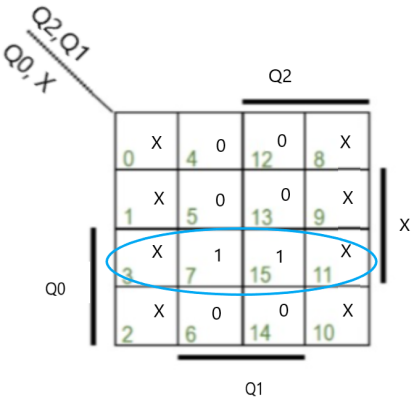
J2 = Q1 Q0 X



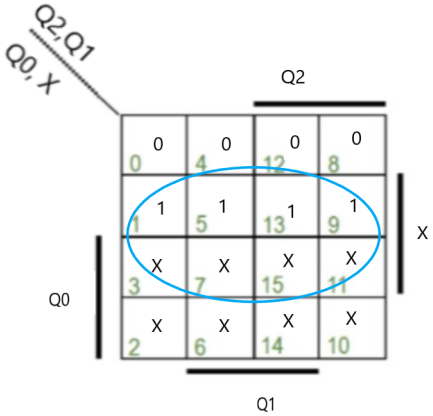
K2 = Q1 Q0 X



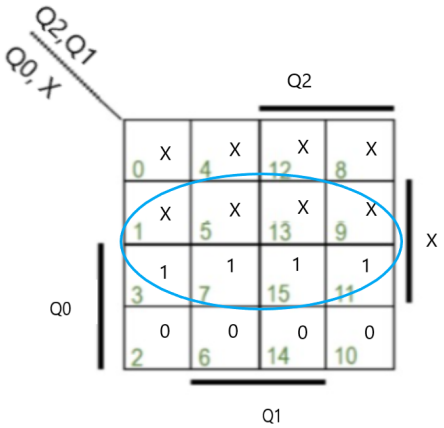
J1 = Q0 X



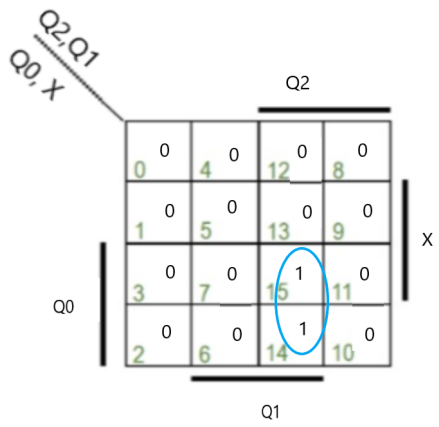
K1 = Q2 X



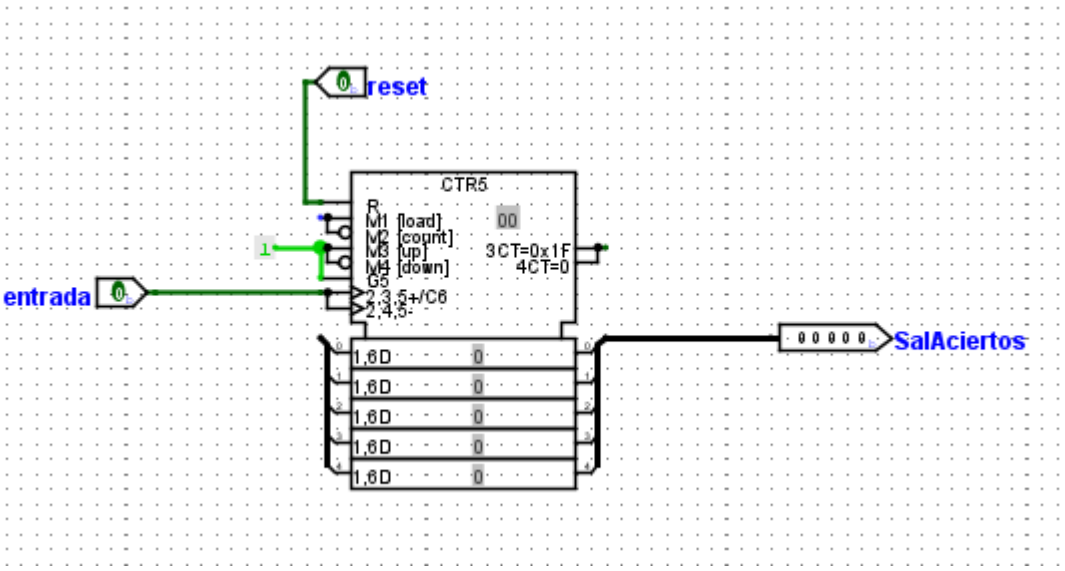
j0 = X



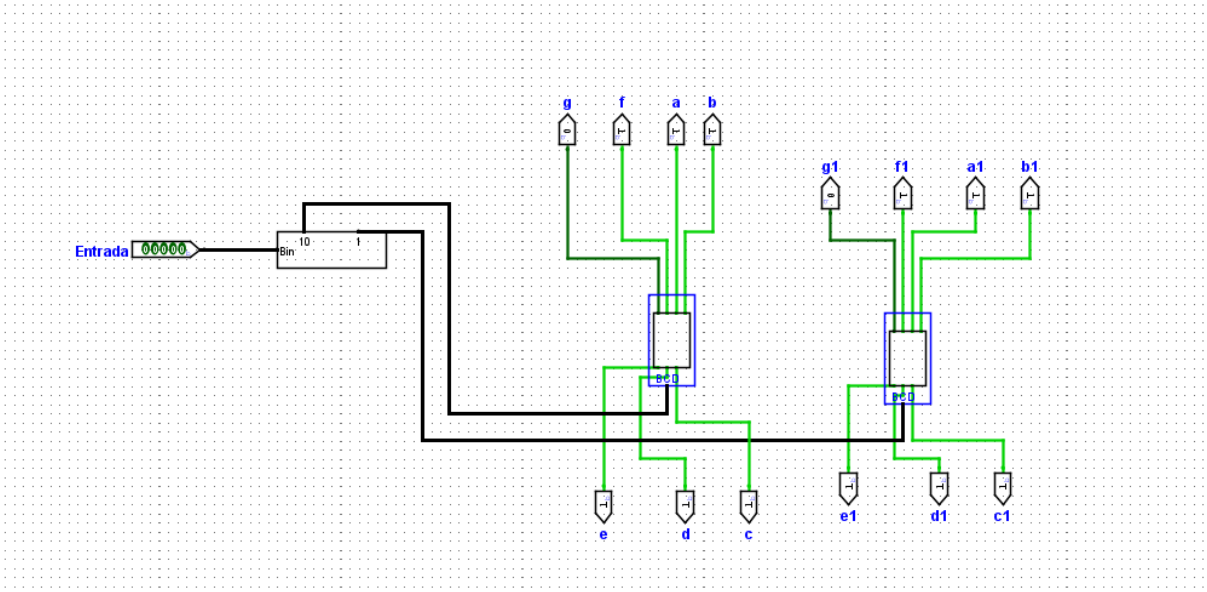
K0 = X



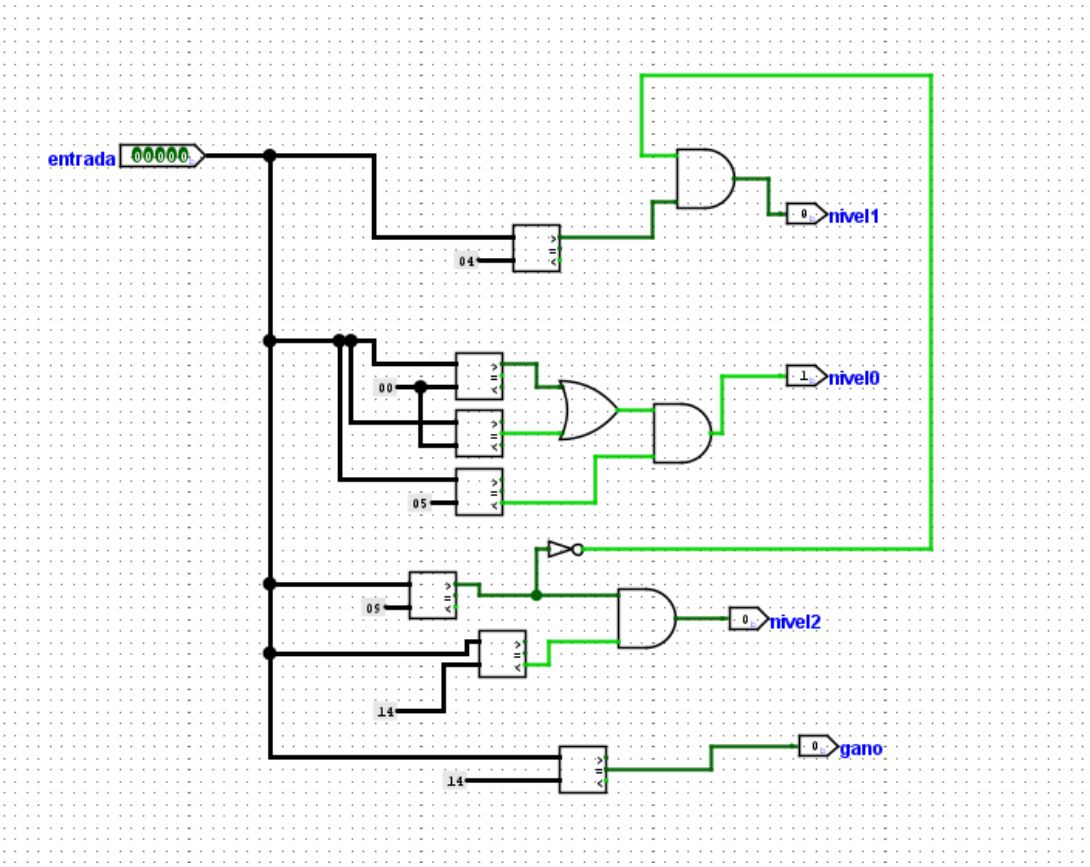
- **aciertos**:
  - Es el encargado de recibir una señal cada que hay un acierto y usar un contador para acumular los puntos y llevarlos al contador de aciertos.



- **contadorDeAciertos:**
    - Para este contador, se recibe la cantidad de aciertos a medida que pasa el juego, para luego convertirlos de binario a BCD y luego otro de BCD a 7 segmentos para poder usar dos displays de 7 segmentos en la contabilización de los aciertos.



- **Niveles:**
    - Es un componente que usa una función lógica que de acuerdo a las reglas del juego decide qué nivel activar o finalmente mostrar la victoria.



Conclusiones :

Con lo aprendido en clases sobre los Diseños de circuitos secuenciales se logró llevar a cabo satisfactoriamente la implementación de esta práctica, destacando el uso de flip flops y se evidenció la utilidad de ellos para la elaboración de circuitos que contienen componentes secuenciales.

Se facilitó el desarrollo en logisim gracias a la jerarquía de diseño implementada en el circuito final, separando responsabilidades de los componentes.

Presentamos algunas dificultades en la elaboración de las tablas, diagramas y mapas en el transcurso del desarrollo que nos retrasó un poco en el desarrollo del circuito, pero gracias al material de clases y grabaciones se pudo corregir los inconvenientes y culminar la práctica.

Agradecemos la respuesta oportuna a nuestras inquietudes y el acompañamiento en el transcurso de la práctica.

Github:

- https://github.com/Juanpablo733/Lab2Arq