



UNIVERSIDAD ORT URUGUAY

Facultad de Ingeniería

Arquitectura de Sistemas

OBLIGATORIO 1

Juan Manuel Varela 289733 (Grupo M3B)

Andrés Dalla Rizza 327421 (Grupo M3B)

Patricio Martino 242673 (Grupo M3B)

Noviembre 2023

Indice

Opción 1 - Logisim	3
CIRCUITO	4
Circuito completo:	4
Score:	5
Reloj	6
Pantalla	7
Generador Random	7
Grupo registros	8
Auto User	9
Movimiento	9
Choques	10
Final del muestreo del circuito.	11

Opción 1 - Logisim

Optamos por simular con logisim un juego de carreras de autos.

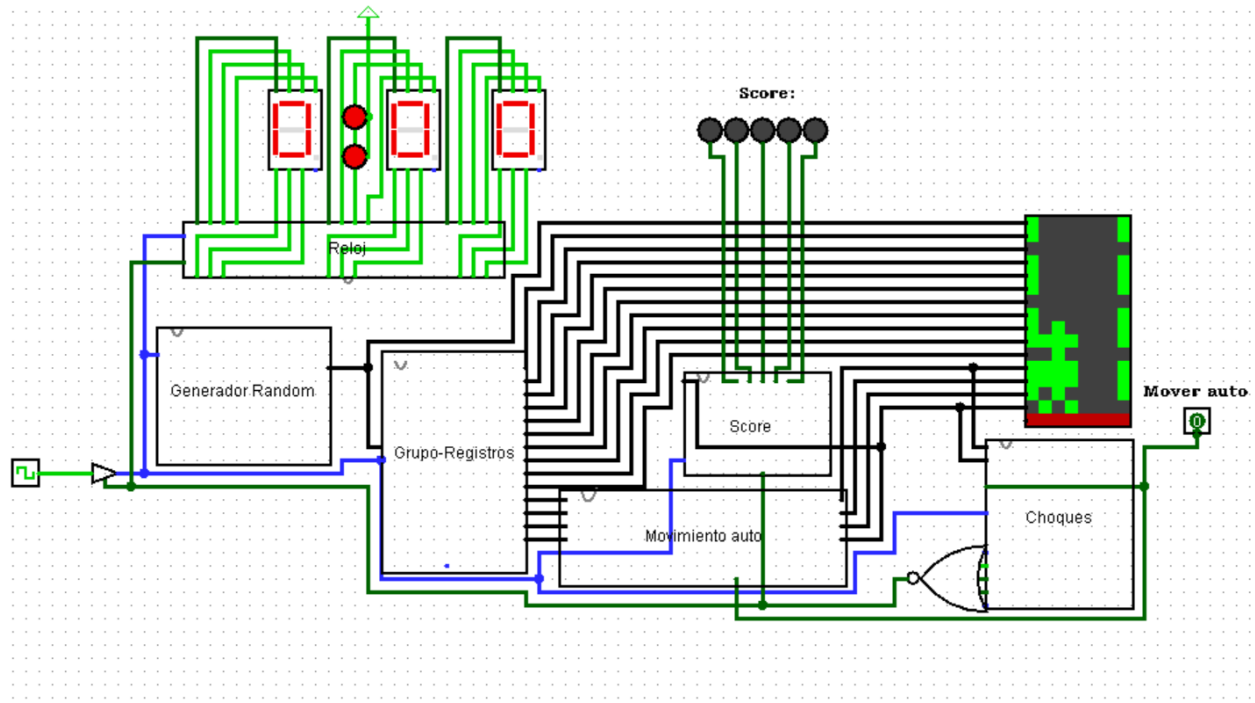
Como se menciona en la letra del obligatorio, y a medida qué se trabajó en el proyecto se definieron algunas reglas:

- El juego está acotado a una pantalla de 16 píxeles de alto por 8 de ancho.
- En la parte inferior de la pantalla existe un auto que tiene dos posiciones: sobre la izquierda o sobre la derecha:
- El score, aumenta cada vez que un auto es pasado. Este se debe mostrar con 5 leds en binario.
- El juego se detiene si el auto choca.
- Existe un reloj que arbitra los tiempos del juego, este se detiene cuando el juego termina, Este se debe mostrar en decimal, minutos y segundos.
- La generación de los autos contrarios se realiza mediante un random sugerido:
- Debe darse espacio entre auto y auto para que sea posible esquivarlo.
- La frecuencia del Tick del CLOCK debe setearse a 2 Hz. Cambiar esta frecuencia significa cambiar la velocidad del contador de tiempo, y la frecuencia con la que cambian los píxeles de la pantalla, haciendo que todo se acelere (si se aumenta la frecuencia) o decelere (si se decrementa).

CIRCUITO

A continuación explicamos las distintas partes del circuito, su funcionamiento y el razonamiento detrás de algunas decisiones qué se tomaron.

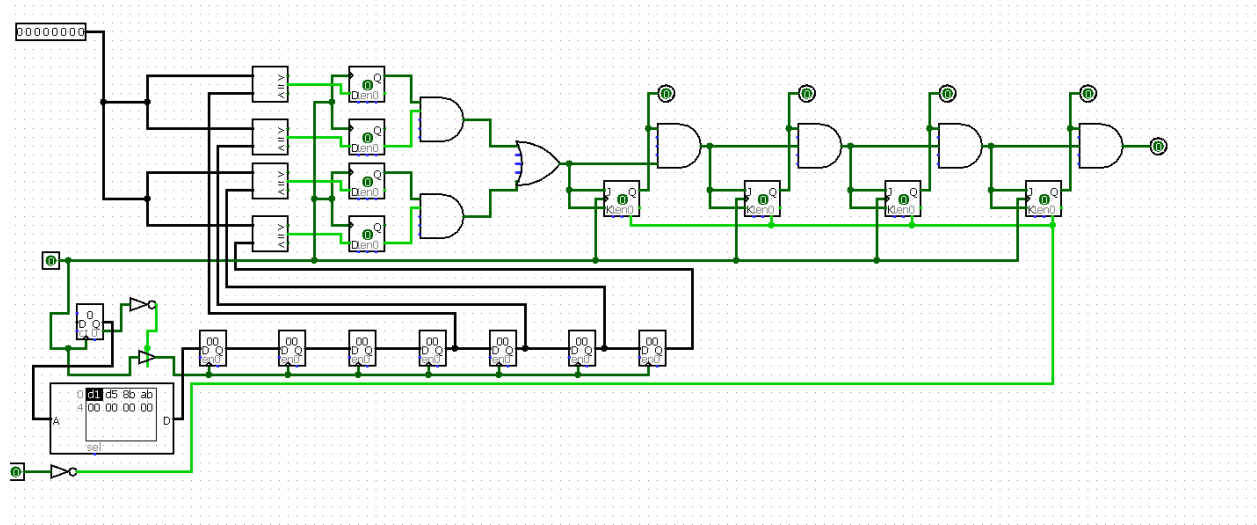
Circuito completo:



Como podemos notar, en el circuito se muestra de forma clara todos los elementos pedidos: el reloj, el score, y la entrada para mover el auto.

Para facilitar la lectura del circuito este fue dividido en módulos, separados por funcionalidad. A continuación se detalla cada módulo, y el razonamiento detrás de su construcción.

Score:



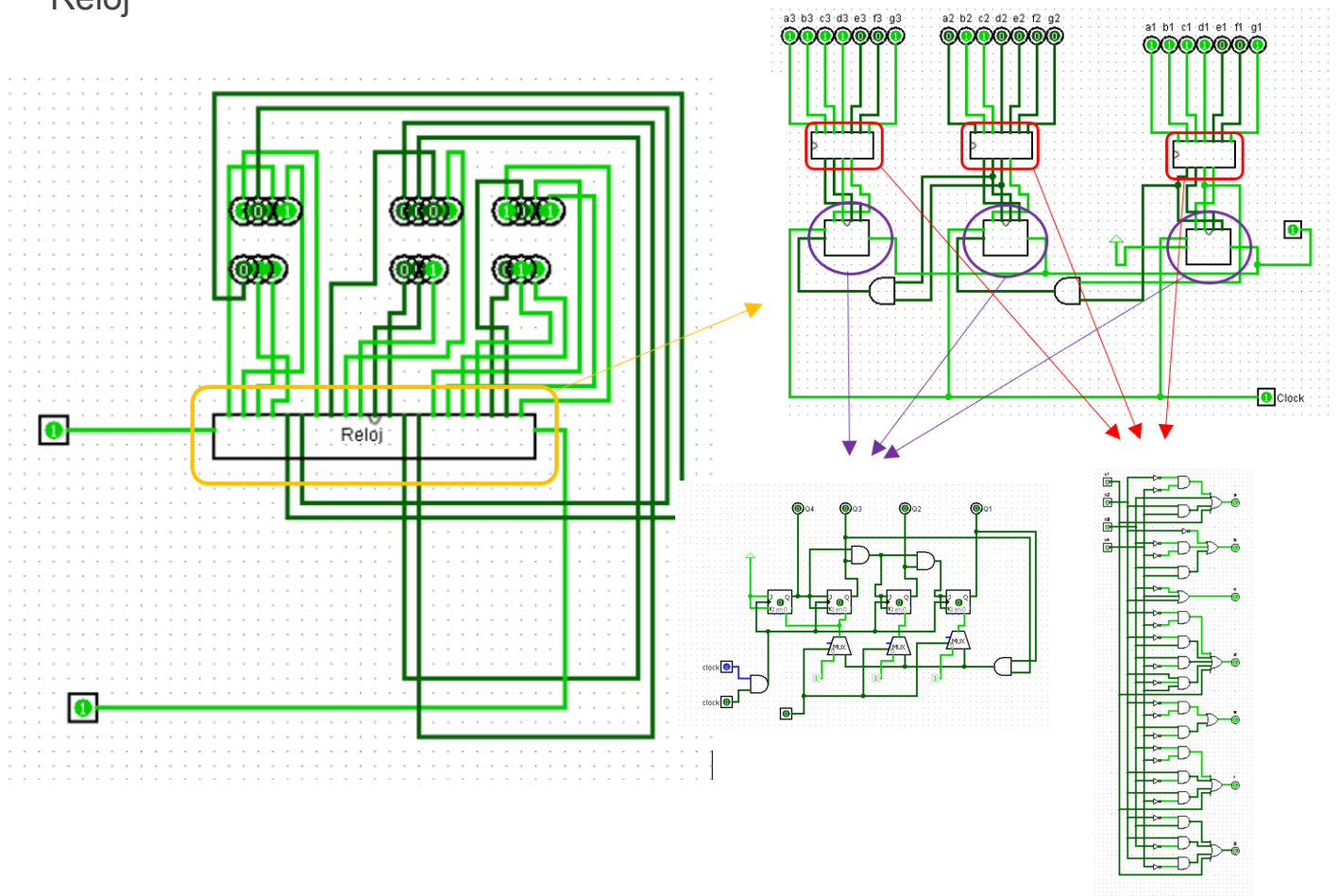
Utilizamos el circuito: Contador biestable con JK que se encuentra en aulas, bajo ‘Circuitos de Importancia’.

El Pin de arriba a la izquierda guarda el estado de los píxeles de la última fila de la pantalla (1 = prendido, 1 = apagado).

La ROM se utiliza para cargar los registros con cuatro valores importantes. El estado de los pixeles justo antes de que el auto enemigo termine de pasar por al lado del auto del usuario, y el estado de los pixeles cuando el auto enemigo termina de pasar por al lado del auto (Si el auto del usuario se encuentra del lado izquierdo, estos dos valores serían D5h y D1h respectivamente, o 11010101 y 11010001). Son cuatro valores para incluir ambos estados en ambas posiciones del auto.

Cuando el auto User está terminando de pasar el auto enemigo, se detecta ese valor y con un Comparator se prende un D Flip-Flop, si se detecta la siguiente posición, donde el auto User termino de pasar al auto enemigo, Se prende el otro Comparator, y el otro D Flip-Flop, lo cual envía la señal al circuito del contador biestable, aumentando en 1 el score. El score se muestra en binario de izquierda a derecha por medio de leds.

Reloj

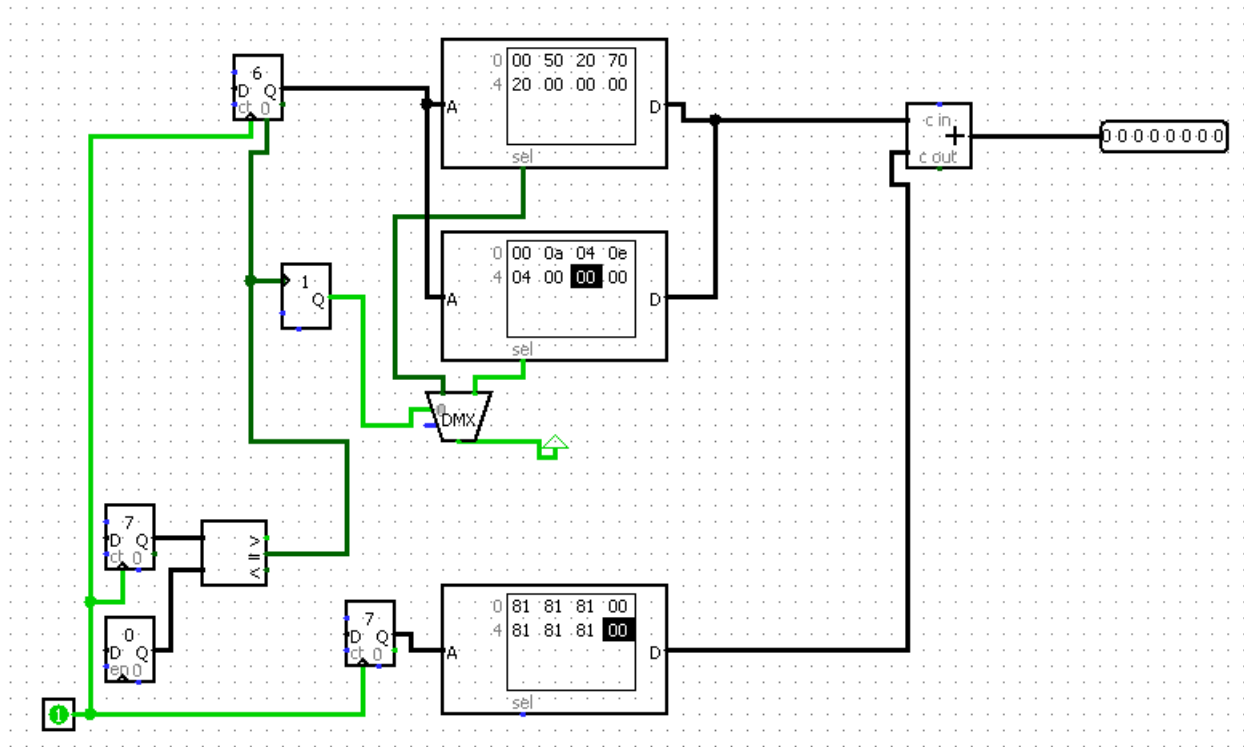


Es importante notar que el correcto funcionamiento del reloj depende de la frecuencia de Ticks que se setea. Para el funcionamiento esperado se sugiere configurarlo para 2 Hz.

Como podemos ver el reloj se compone por 3 módulos. El señalado en violeta es el encargado de ir aumentando los segundos y los minutos. El decodificador (señalado en rojo), se encarga de enviar las señales al codificador BCD de 7 segmentos. Ver como en todos los circuitos se conecta un cable para resetear el reloj una vez que el auto choca.

Pantalla

Generador Random



El siguiente circuito se dedica a generar todas las imágenes: las calles, los autos enemigos, los píxeles vacíos.

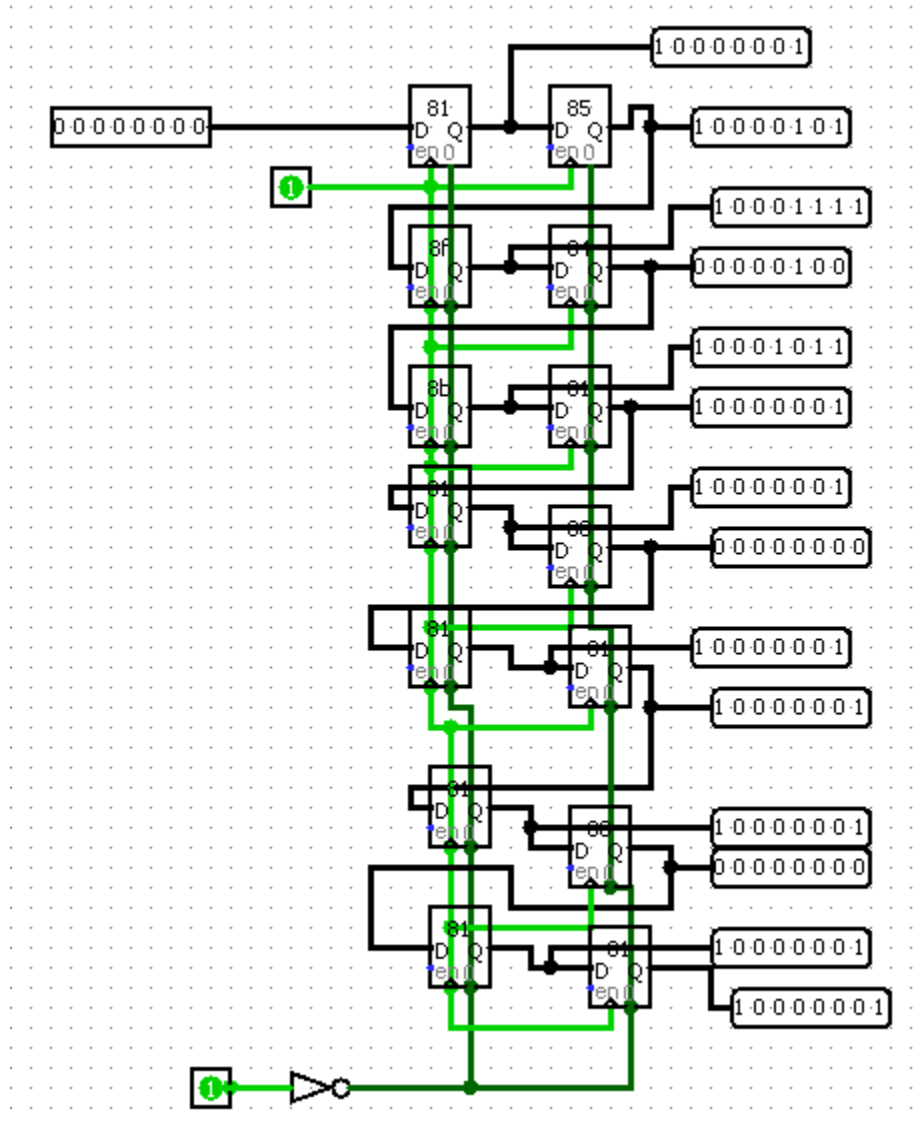
Hay tres ROM:

- La ROM de **más abajo** tiene guardados los valores de la calle. 81 h = 10000001 b.
- Las dos ROM restantes tienen guardadas los valores que representan las dos posiciones posibles de los autos enemigos: posición izquierda y posición derecha

Utilizando el circuito de Generador-Random de logisim, y un demultiplexor, se elige al azar cuál de las dos ROMs se ‘elige’ o ‘activa’. La forma de mostrar dos elementos a la vez (calle y auto enemigo) que ideamos fue sumando los bits de cada uno. Así se muestra en pantalla simultáneamente las dos figuras. Por lo tanto, los valores de la ROM activada luego se suman por medio de un sumador con los

valores de la ROM de abajo, la calle. El resultado se traslada al Pin, viajan a la fila de más arriba de la pantalla, y al Grupo-Registros generando la animación deseada.

Grupo registros



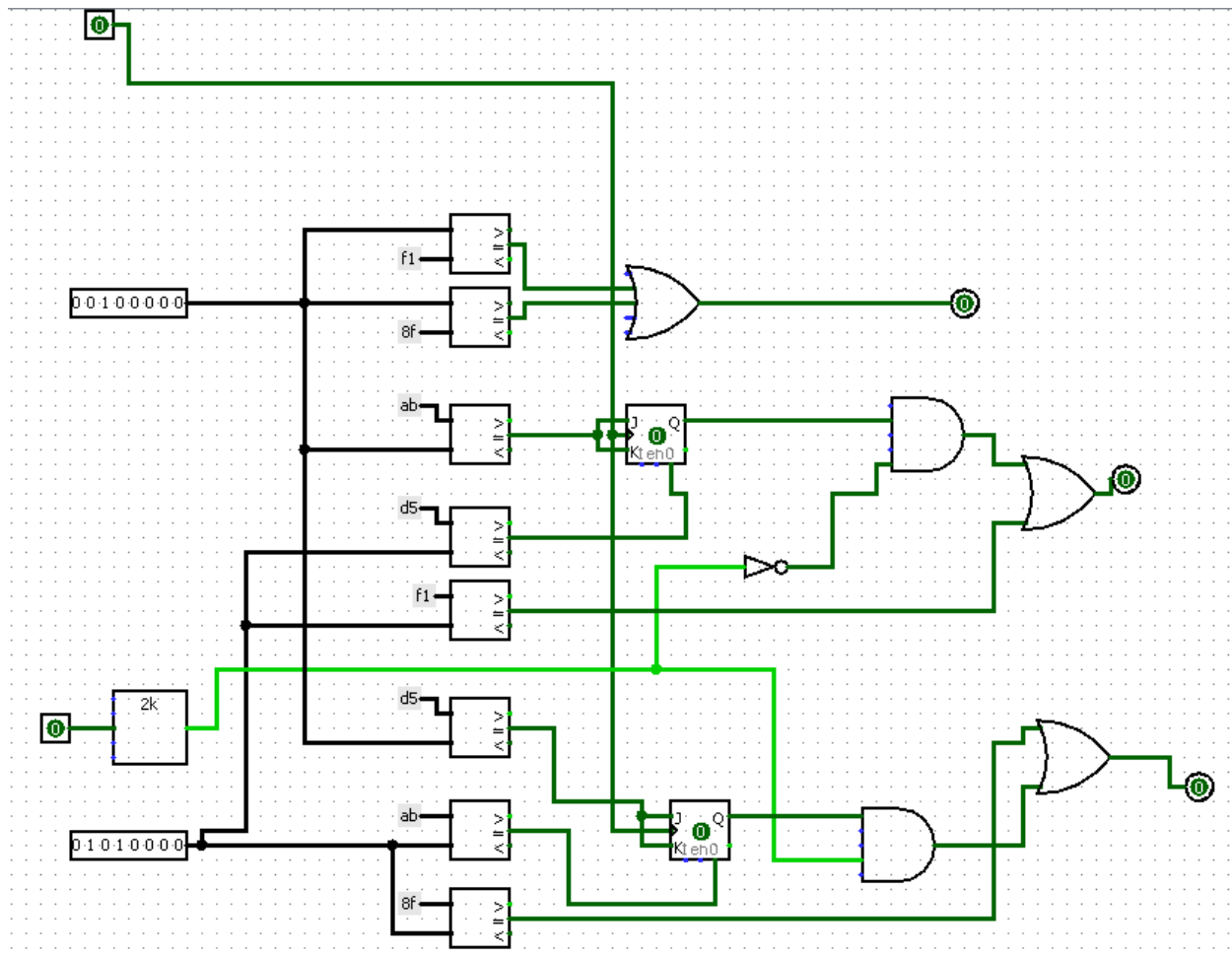
Una vez generada la primera fila de la pantalla por el Generador-Random, estos valores se deben trasladar hacia abajo para seguir generando nuevas filas de píxeles, y que así el juego cobre vida.

Al presionar el botón de movimiento del auto, el circuito envía la entrada de selección a los multiplexores, el cual decidirá el shifter para recibir la información

y pasársela al sumador. Cada Shifter contiene un registro conectado que indica la cantidad de bits de corrimiento que se desea

Los pines de la derecha son los encargados de recibir el resultado de la suma de 3 componentes: la calle y los autos enemigos (pin marcado en amarillo) con los datos que se reciben de los shifters, logrando que la información se despliegue de una forma conjunta en el array de leds

Choques



El circuito de Choques tiene un pensamiento similar al circuito de Score.

Para determinar si el Auto User choca, solo se necesita conocer los valores de los píxeles en las filas 1 y 4 comenzando de abajo. Es decir las filas de la cola y la cabeza del auto User. Utilizando Comparadores, cuando estas filas de píxeles equivalen alguno de los valores codificados en el circuito con Constantes (un valor

por cada choque posible), se determina que hubo un choque, y se prende una flag de choque, lo cual envía una señal a un buffer que corta la señal del reloj, haciendo que el sistema se detenga como es pedido.

Final del muestreo del circuito.