



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE
CC4301 ARQUITECTURA DE COMPUTADORES

Tarea 1

Profesor:
Pablo Guerrero

Auxiliar
Pablo Polanco

Alumno:
Gabriel Azócar C.

Fecha:
18 de Octubre de 2016

Introducción

Este informe tiene como objetivo explicar la implementación de un Pacman simplificado. Para esto, se crearon diferentes circuitos para lograr el resultado esperado, los cuales se explicarán en las siguientes secciones.

Cálculo de direcciones

Se tiene un Joystick el cual nos entrega dos valores: un valor que describe sus movimientos verticales y otro que describe los horizontales. A partir de estos, codificaremos las cuatro direcciones que necesitamos: arriba, abajo, izquierda y derecha. La codificación será la siguiente:

1. Izquierda: $y_1y_0 = 00$
2. Derecha: $y_1y_0 = 01$
3. Arriba: $y_1y_0 = 10$
4. Abajo: $y_1y_0 = 11$

Analizando los valores del joystick, la tabla de valores queda de la siguiente manera (con solo los valores que nos interesan):

h_1	h_0	v_1	v_0	y_1	y_0
0	1	1	0	0	0
1	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	1	1

Tabla 1: Tabla de valores de codificación de las direcciones del joystick

Luego, para crear el circuito que finalmente implementaremos, creamos las tablas de Karnaugh de y_1 y y_0

Para y_1 :

$v_1v_0 \backslash h_1h_0$	00	01	11	10
00	x	x	x	x
01	x	x	x	1
11	x	x	x	1
10	x	0	0	x

Tabla 2: Tabla de Karnaugh para y_1

Para y_0 :

$v_1v_0 \backslash h_1h_0$	00	01	11	10
00	x	x	x	x
01	x	x	x	0
11	x	x	x	1
10	x	0	1	x

Tabla 3: Tabla de Karnaugh para y_0

Luego, nos queda que:

1. $y_1 = h_1v_0$
2. $y_0 = h_1v_1$

De esta forma, el circuito que codifica las direcciones a partir de los datos entregados por el joystick que como muestra la Figura 1.

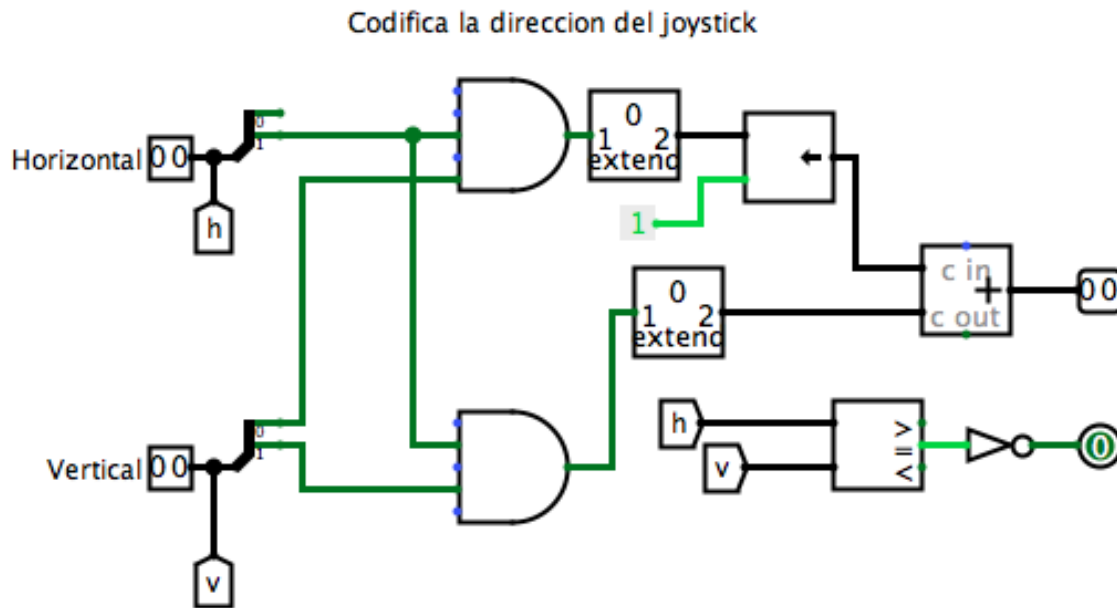


Figura 1: Circuito que codifica las direcciones a partir de los datos entregados por el joystick

Cuando el joystick está al centro, entrega $h_1h_0 = 10$ y $v_1v_0 = 10$. Por lo tanto, si h y v son iguales, se debe entregar la última dirección válida ingresada. Esta comparación nos servirá en el siguiente circuito a implementar. En la Figura 1 se muestra que calculamos y_1 , lo movemos al segundo bit y lo sumamos a y_0 para que el circuito nos entregue y_1y_0 .

Ahora, creamos el circuito que nos entrega la dirección hacia donde se está moviendo el Pacman, en base a la codificación que acabamos de hacer y la condición de que el joystick nos entregue direcciones válidas.

La implementación de este circuito se muestra a continuación.

Entrega la direccion del joystick

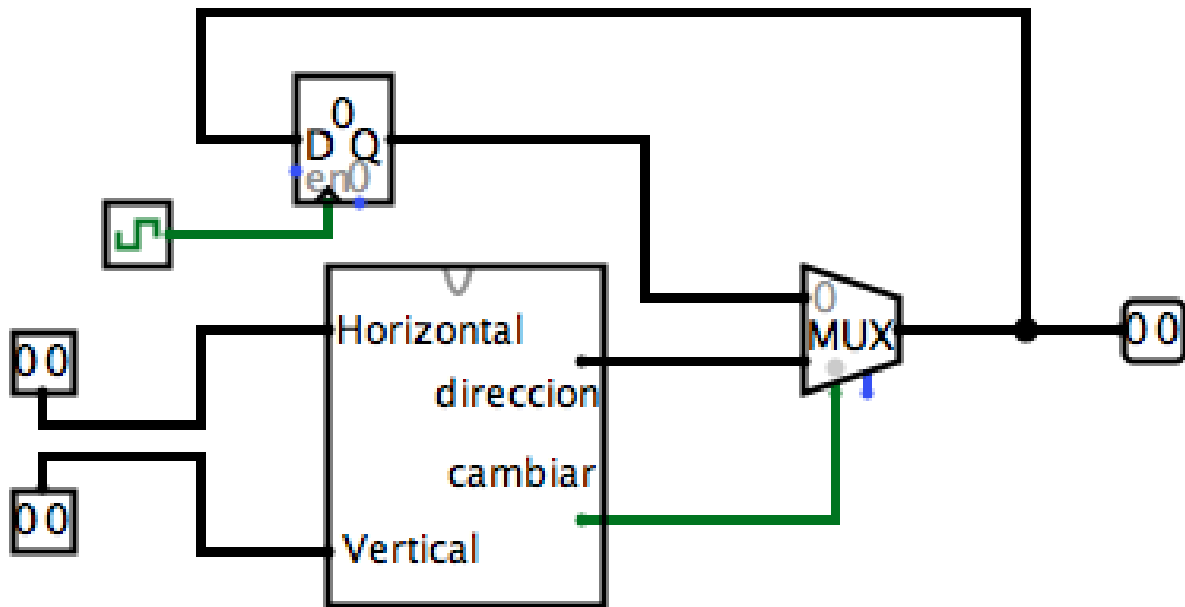


Figura 2: Circuito que entrega la dirección que el joystick nos indica.

Podemos ver que dependiendo de la condición de que h y v sean diferentes, se entrega la última dirección válida ingresada o la dirección que se está entregando actualmente.

Cálculo de posiciones

Crearemos un subcircuito que es capaz de guardar la posición actual del Pacman, e ir actualizándola dependiendo de la dirección actual del joystick. Además, debe tener un botón de reinicio que haga volver al Pacman a una posición dada. Este circuito se muestra en la Figura 3.

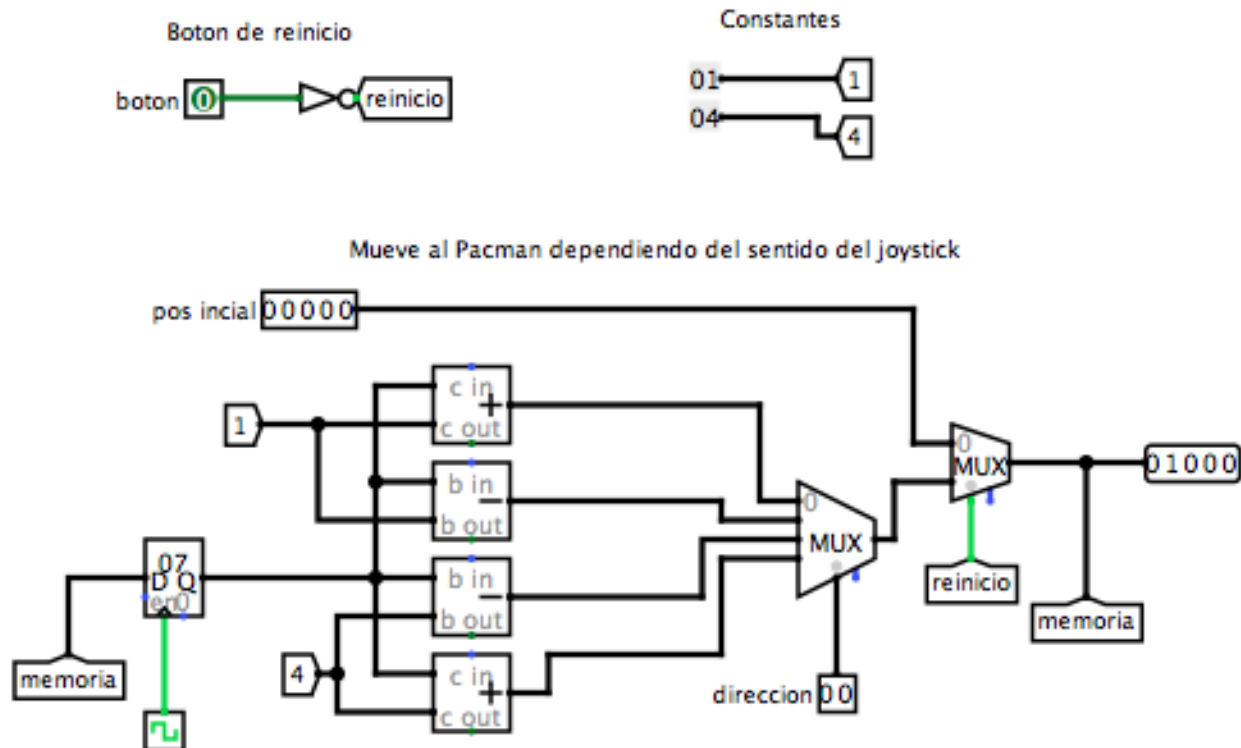


Figura 3: Circuito que entrega a posición actual del Pacman.

El circuito calcula la posición de la siguiente forma: Para moverse una casilla en la matriz se debe:

1. Hacia la izquierda (00): Se debe sumar 1 a la posición actual.
2. Hacia la derecha (01): Se debe restar 1 a la posición actual.
3. Hacia arriba (10): Se debe restar 4 a la posición actual.
4. Hacia abajo (11): Se debe sumar 4 a la posición actual.

Por eso, dependiendo de la dirección del joystick que se entrega, se devuelve la nueva posición siguiendo las reglas antes descritas. Al final, se agrega un nuevo mux, donde se entrega la posición actual o la posición inicial del Pacman (si es que se presiona el botón de reinicio).

Cálculo de matriz de Pepitas de oro

Se implementa un circuito que define las posiciones de las pepitas de oro. Se le entrega una matriz inicial de posiciones y esta se modifica según la posición del Pacman, ya que si el Pacman está en una casilla la cual tiene una pepita, esta última desaparece. El circuito implementado es el siguiente:

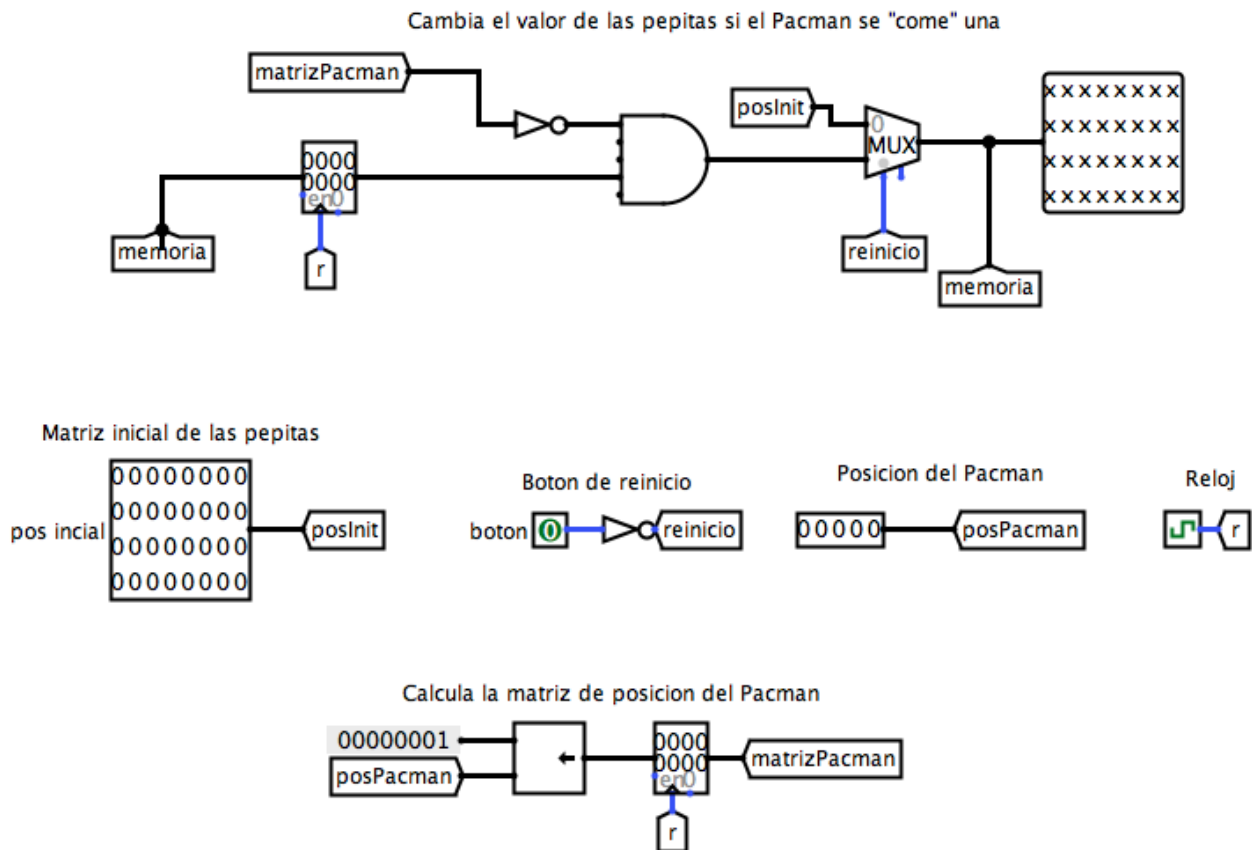


Figura 4: Circuito que describe las pepitas de oro.

Cálculo de celdas

Se implementará un circuito que calcule una matriz de estado de celdas que contenga la información de: en qué celda está el Pacman, en qué celda está el fantasma y en qué celdas hay pepitas de oro. Para esto, codificaremos los estados de las celdas como:

1. Celda vacía: $y_1y_0 = 00$
2. Pepita de oro: $y_1y_0 = 01$
3. Fantasma: $y_1y_0 = 10$
4. Pacman: $y_1y_0 = 11$

Para hacer esto, se usarán dos matrices de 32 bits, donde en una estará la información de y_1 y en la otra la información de y_0 .

El circuito implementado es el siguiente:

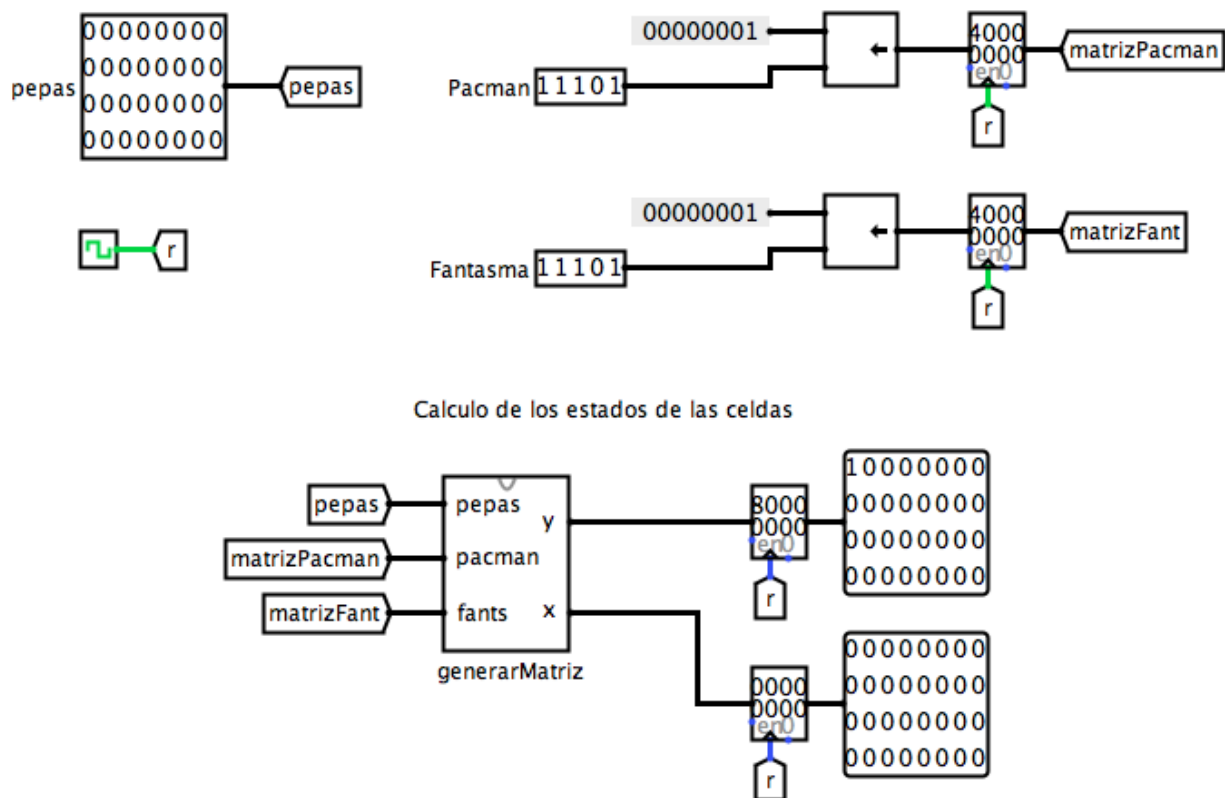
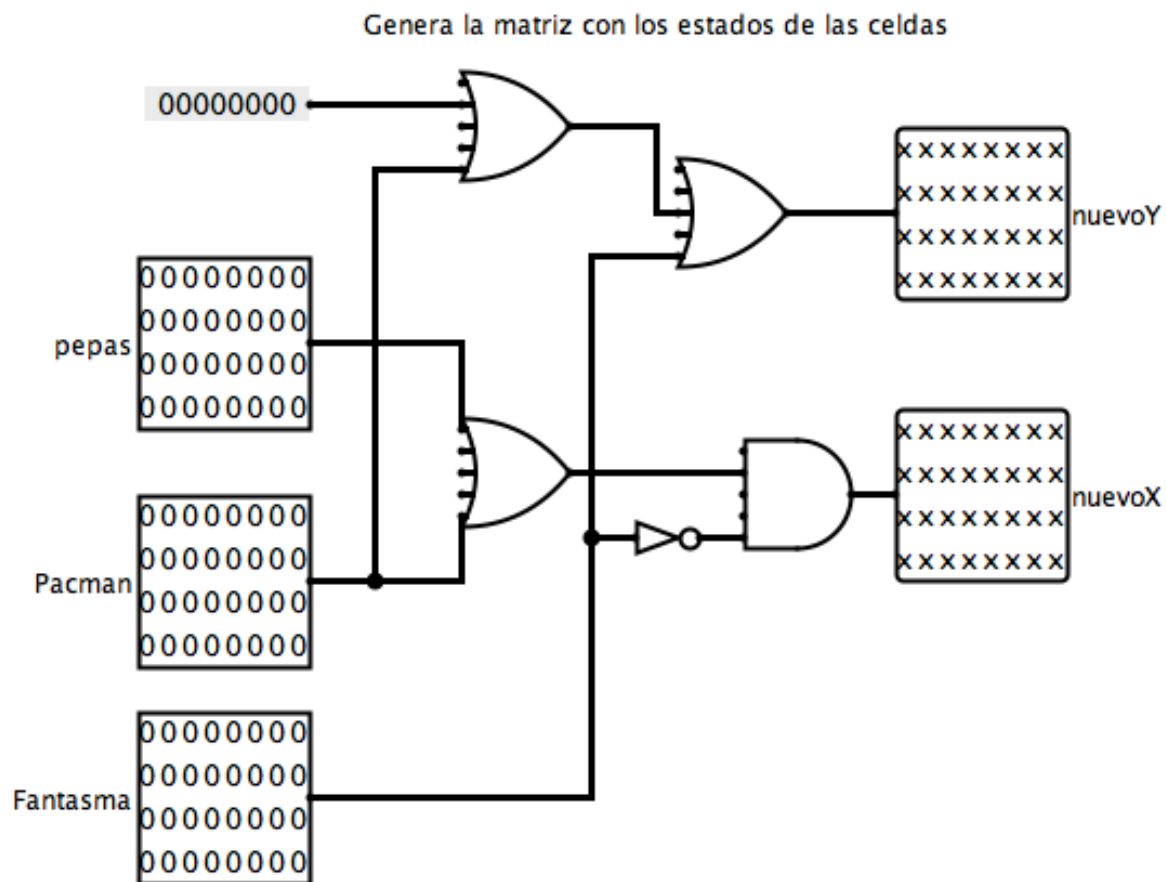


Figura 5: Circuito que describe las celdas y sus estados.

Genera la matriz con los estados de las celdas



Los que hace este circuito es fusionar las tres matrices (donde está el Pacman, donde está el Fantasma y donde están las pepitas). Para lograr copiar las pepitas, solo debemos copiar sus 1 en la matriz X. Para copiar el Pacman solo debemos copiar sus 1 en ambas matrices. Y para copiar el Fantasma debemos copiar su 1 en la matriz Y y dejar en 0 su posición en la matriz X.

Detección de reinicio

Se crea un circuito capaz de detectar tres situaciones:

1. Que el botón de reinicio esté presionado.
2. Que Pacman y Fantasma estén en la misma posición.
3. Que las pepitas de oro se acaben.

El circuito recibe tres entradas, las cuales avisan si alguna de estas situaciones ha ocurrido. La decisión de "ha ocurrido." "no ha ocurrido" la toma el circuito de lógica de juego, que viene en la siguiente sección.

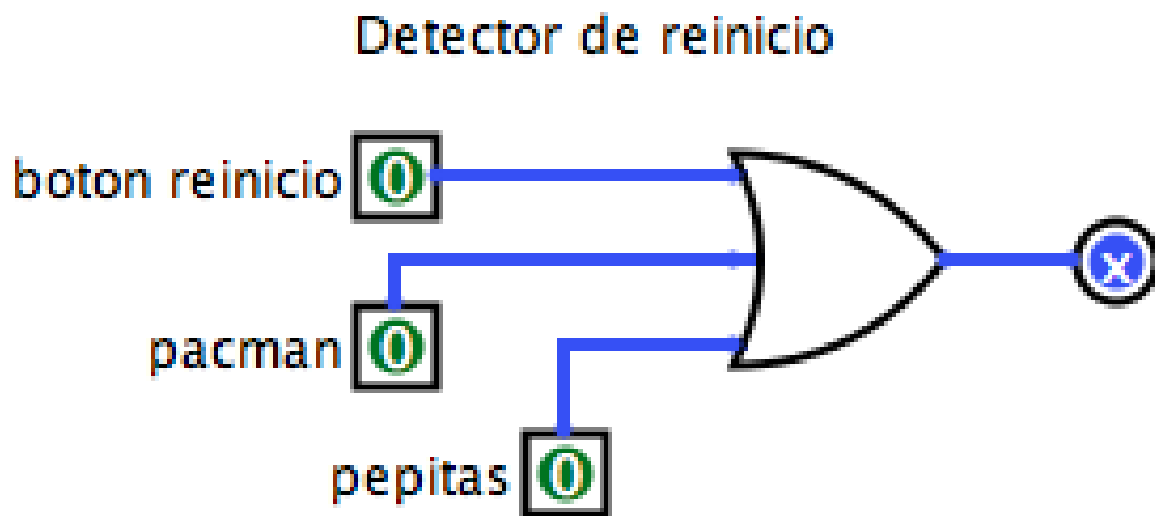


Figura 7: Circuito que avisa el reinicio.

Lógica de juego

Este circuito se encarga de crear las matrices que describen los estados de las celdas a través de las direcciones entregadas por los joystick (manual y aleatorio). El circuito se muestra en la siguiente figura:

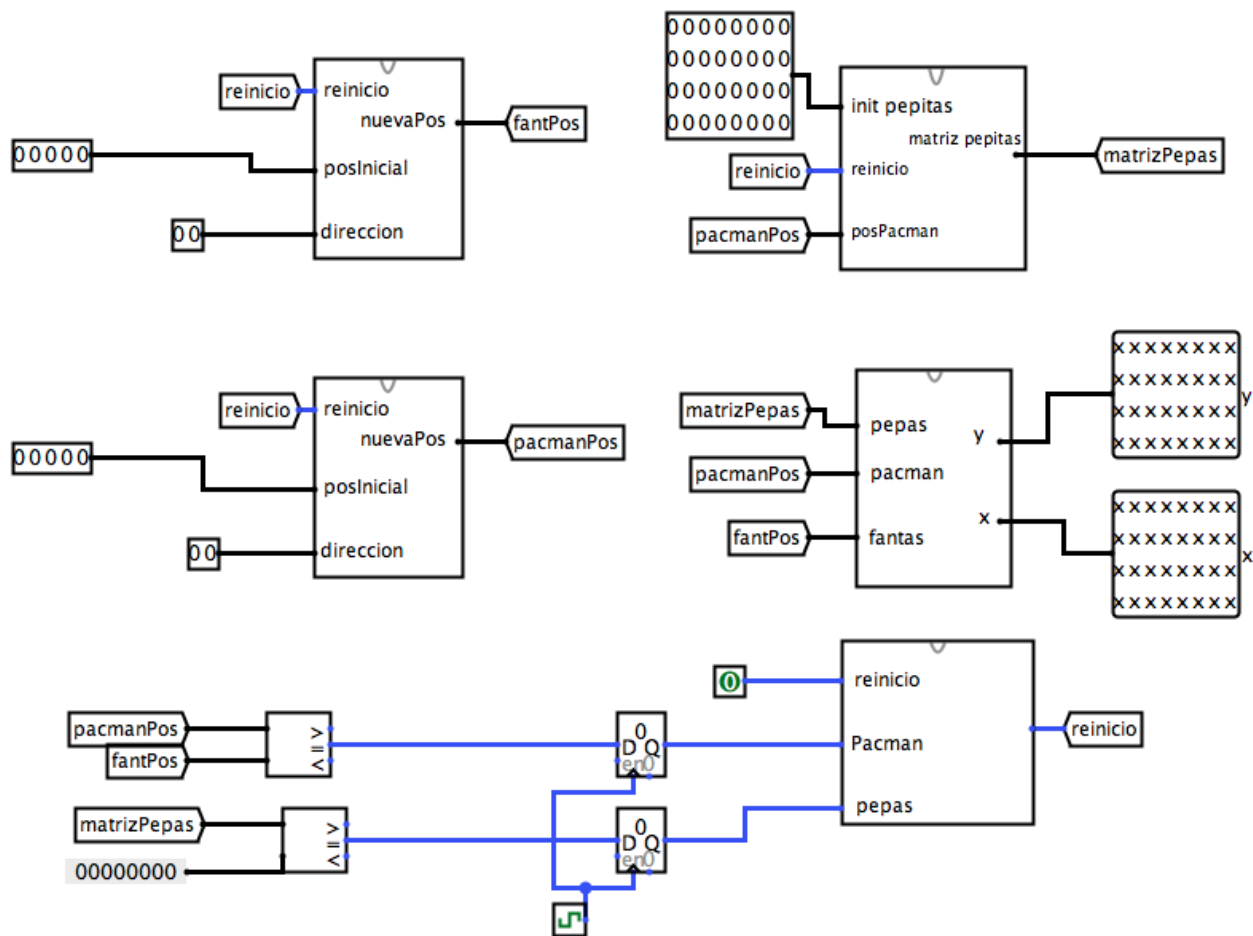


Figura 8: Circuito que describe la lógica del juego

Tenemos 5 subcircuitos:

1. Circuito que calcula la posición del Fantasma: Como su nombre lo indica, calcula la posición del fantasma a partir de la dirección actual de su joystick. Si el reinicio está activo, se define como la posición inicial que también es dada.
2. Circuito que calcula la posición del Pacman: funciona de la misma forma que el anterior.

3. Circuito que calcula la matriz de pepitas de oro: Calcula los lugares donde hay pepitas de oro. Si es que el Pacman pasa por la misma posición de donde hay una pepita, este circuito se encarga de actualizar ese valor y eliminar esa pepita. Por eso necesita la posición actual del Pacman como entrada.
4. Circuito que calcula las celdas de 64 bits: Se le entrega la matriz de pepitas, la posición del Pacman y la posición del Fantasma y utiliza el subcircuito de la Figura 6.
5. Circuito de reinicio: Entrega como entradas un comparador entre las posiciones del Pacman y el Fantasma, un comparador entre las Pepitas y el número cero, y por última el valor actual del botón de reinicio. Luego usa el circuito de la Figura 7 para tomar la decisión.

Renderizado y matriz de LEDs

Lamentablemente, no pude idearmelas para generar lo pedido. Sin embargo hice varios intentos, del cual es más cercano a mostrar algo fue lo que se muestra en la Figura 9 y 10.

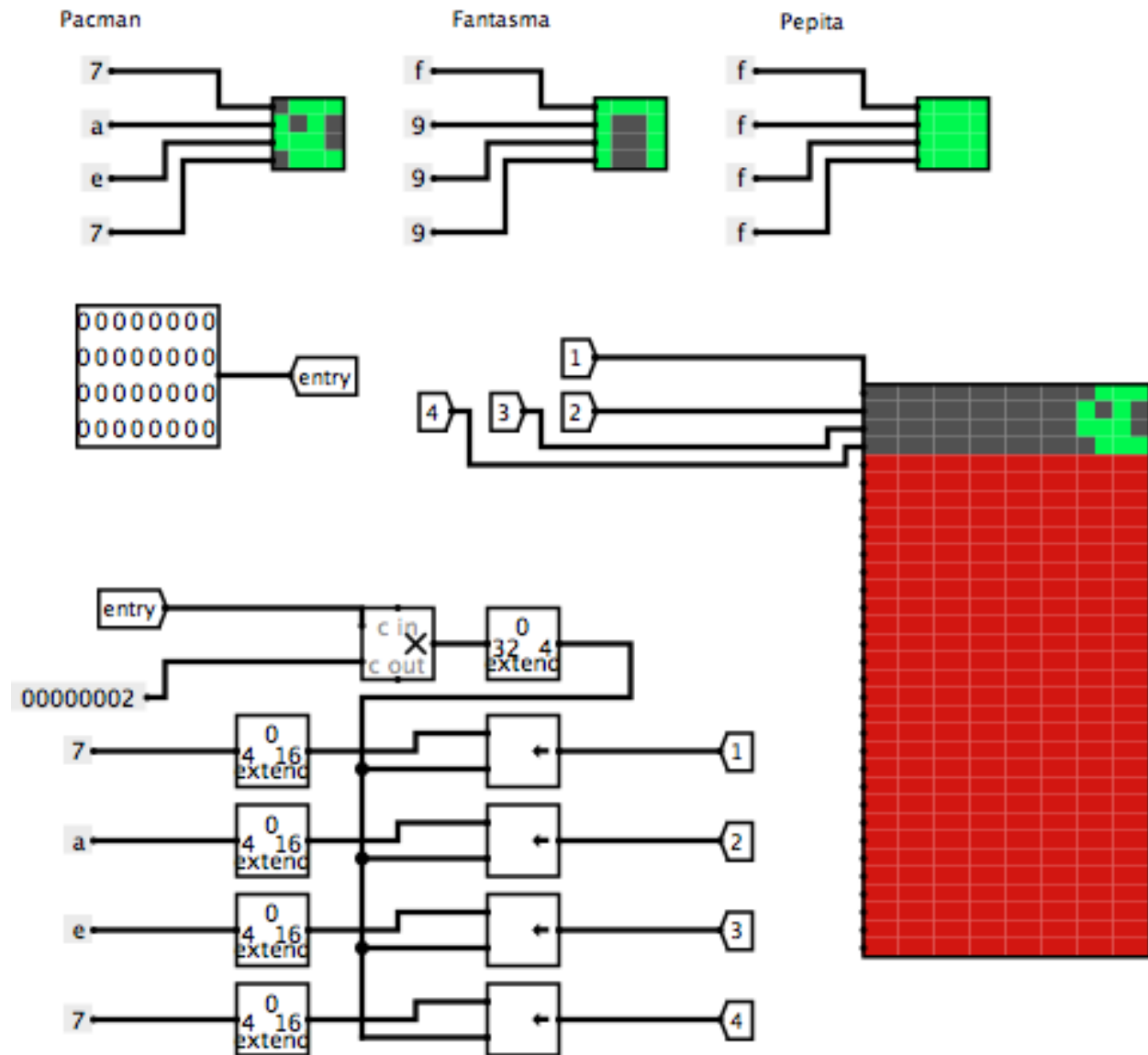


Figura 9: Pacman en su posición inicial.

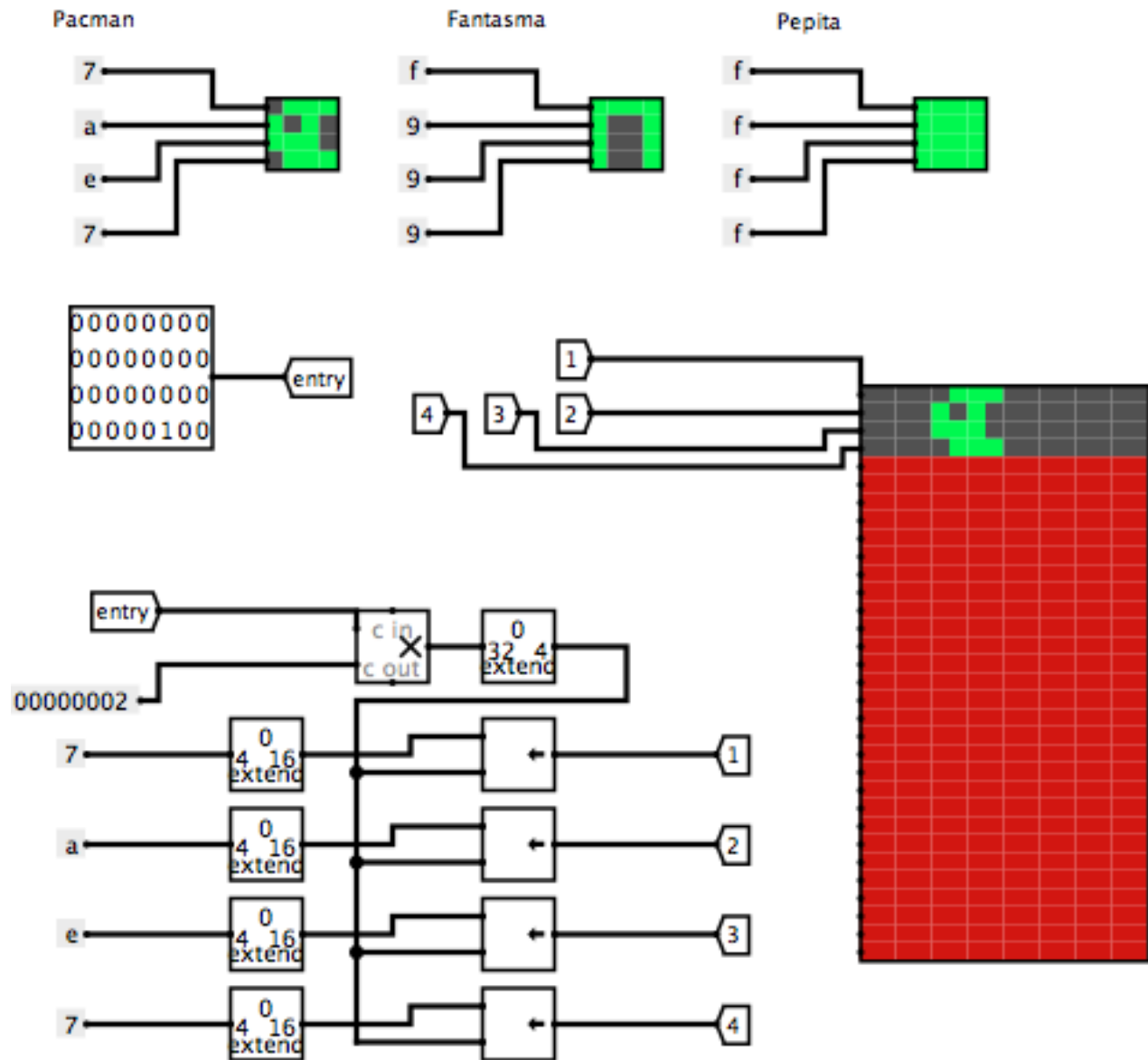


Figura 10: Pacman en la tercera posición hacia la izquierda.

Circuito principal

En la siguiente figura se muestra el circuito principal del juego. Como no se pudo implementar una visualización LED, se implementó una visualización simplificada con dos matrices LED que muestran la matriz X e Y antes descritas en este informe. Este circuito muestra que las funcionalidades implementadas funcionan como se espera.

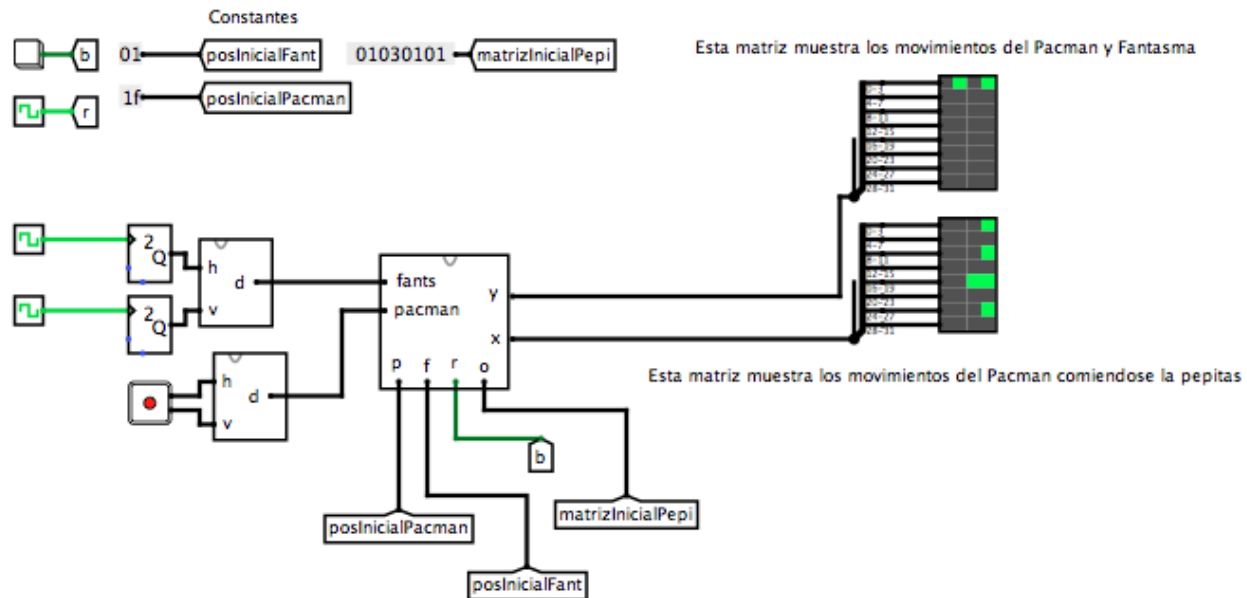


Figura 11: Pacman en la tercera posición hacia la izquierda.

En la matrices LED, arriba se visualizan dos pixeles encendidos. El de la izquierda es el Fantasma, y el de la derecha es el Pacman. Siguiendo con la codificación antes descrita (Pacman: $y_1y_0 = 11$), en la matriz de abajo, la posición del Pacman es el pixel encendido que está en la misma posición en ambas matrices. Los demás pixeles encendidos en la matriz de abajo son las pepitas. Si se simula el juego en Logisim, se aprecia que el reinicio y los movimientos del joystick funcionan perfectamente.