

Tarea 1

Autor: Lukas Pavez
Profesor: Pablo Guerrero P.
Auxiliar: Matías Torrealba A.
Ayudantes: Gabriel Chandía G.
Gaspar Ricci
Fecha de entrega: 16/10/2018

1. Introduccion

Se pide implementar un puzzle de 2x2 en donde el usuario puede intercambiar el valor de la posición en la que se ubica con un joystick con la posición en la que se ubica el 0, para lograr esto se dividió el problema en 3 bloques:

1. Bloque de Decodificación
2. Bloque de Control
3. Bloque Memoria del Tablero

2. Bloque de Decodificación

Se partió implementando un circuito que recibe una entrada de 2 bits y entrega una salida de 7 bits, donde la salida corresponde a un numero (1, 2, 3 o apagado) en un display de 7 segmentos. Para lograrlo se construyó la siguiente tabla de verdad:

Tabla 1: Decodificador 2x7

x_1	x_0	y_6	y_5	y_4	y_3	y_2	y_1	y_0
0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	0
1	0	1	0	1	1	0	1	1
1	1	1	0	0	1	1	1	1

Donde x_1x_0 es la entrada de 2 bits y $y_6...y_0$ es la salida que debería dar cada entrada. Para pasar la tabla a circuito se hizo un mapa de Karnaugh para cada y_i :

Tabla 2: Mapas de Karnaugh decodificador 2x7

	<table><tr><td>x_1/x_0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x_1/x_0	0	1	0	0	0	1	1	1		<table><tr><td>x_1/x_0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr></table>	x_1/x_0	0	1	0	0	0	1	0	0	
x_1/x_0	0	1																				
0	0	0																				
1	1	1																				
x_1/x_0	0	1																				
0	0	0																				
1	0	0																				
y_6 :			y_5 :																			

	<table><tr><td>x_1/x_0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x_1/x_0	0	1	0	0	0	1	1	0		<table><tr><td>x_1/x_0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x_1/x_0	0	1	0	0	0	1	1	1		<table><tr><td>x_1/x_0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr></table>	x_1/x_0	0	1	0	0	1	1	0	1	
x_1/x_0	0	1																															
0	0	0																															
1	1	0																															
x_1/x_0	0	1																															
0	0	0																															
1	1	1																															
x_1/x_0	0	1																															
0	0	1																															
1	0	1																															
y_4 :		y_3 :		y_2 :																													

	<table><tr><td>x_1/x_0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x_1/x_0	0	1	0	0	1	1	1	1		<table><tr><td>x_1/x_0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x_1/x_0	0	1	0	0	0	1	1	1	
x_1/x_0	0	1																				
0	0	1																				
1	1	1																				
x_1/x_0	0	1																				
0	0	0																				
1	1	1																				
y_1 :		y_0 :																				

Con los mapas obtenidos, se calcularon las ecuaciones para cada y_i :

- $y_6 = x_1$
- $y_5 = 0$
- $y_4 = x_1 \neg x_0$
- $y_3 = x_1$
- $y_2 = x_0$
- $y_1 = x_1 \vee x_0$
- $y_0 = x_1$

El circuito obtenido se puede ver en la Figura 1.

Luego, utilizando 4 de estos decodificadores, se construyó el bloque de Decodificación, que se puede ver en la Figura 2.

3. Bloque de Control

El Bloque de Control se divide en 3 partes: circuito vicinity para encontrar si 2 posiciones son vecinas, CursorControl que toma de entrada a un joystick y mueve la posición del cursor y un circuito selector_2x4 que selecciona en cual display esta el cursor.

3.1. Vicinity

Este circuito recibe 2 posiciones de 2 bits y arroja en 1 bit si son vecinas o no, la tabla de verdad de este circuito es la siguiente:

Tabla 3: Tabla de verdad circuito vicinity

x_1	x_0	y_1	y_0	out
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
x_1	x_0	y_1	y_0	out
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Tabla 4: Mapa de Karnaugh circuito vicinity

x_1x_0 / y_1y_0	00	01	11	10
0 0	0	1	0	1
0 1	1	0	1	0
1 1	0	1	0	1
1 0	1	0	1	0

Del mapa de Karnaugh se puede obtener la ecuación para out:

$$\text{out} = \neg x_1 \neg x_0 \neg y_1 y_0 \vee \neg x_1 \neg x_0 y_1 \neg y_0 \vee \neg x_1 x_0 \neg y_1 \neg y_0 \vee \neg x_1 x_0 y_1 y_0 \vee x_1 x_0 \neg y_1 y_0 \vee x_1 x_0 y_1 \neg y_0 \vee x_1 \neg x_0 \neg y_1 \neg y_0 \vee x_1 \neg x_0 y_1 y_0$$

El circuito construido se puede ver en la Figura 3.

3.2. CursorControl

Este circuito tiene 2 entradas de 4 bits que representan a un joystick y una salida de 2 bits que representa una posición en el tablero. También hay una entrada de 1 bit que representa al reloj.

El joystick tiene un rango que va de 1 a 15, en el eje x aumenta de izquierda a derecha y en el eje y aumenta de arriba hacia abajo, por lo que se definió un rango para indicar en que momento se mueve el cursor: si el eje x es menor a 4, entonces se mueve hacia la izquierda, si es mayor a 12, entonces se mueve hacia la derecha. Con el eje y es similar, menor a 4 es arriba y mayor a 12 es abajo. Para esto se utilizaron 4 comparadores, y las salidas de los comparadores entran a un circuito llamado magic que tiene 4 entradas de 1 bit (las salidas de los comparadores) y una entrada de 2 bits que representa a la posición anterior del joystick, y tiene una salida de 2 bits que conecta a un registro y luego a la salida del circuito CursorControl, que representa la actual posición del joystick.

Se puede ver el circuito implementado en la Figura 4.

Para construir el circuito magic se hizo la siguiente tabla de verdad:

Tabla 5: Tabla de verdad circuito magic

x_3	x_2	x_1	x_0	q_1	q_0	d_1	d_0
0	0	1	0	0	0	0	1
0	1	0	0	0	0	0	0
x	x	x	x	0	0	0	0
0	0	0	1	0	1	0	0
0	1	0	0	0	1	1	1
x	x	x	x	0	1	0	1

x_3	x_2	x_1	x_0	q_1	q_0	d_1	d_0
1	0	0	0	1	0	0	0
0	0	1	0	1	0	1	1
x	x	x	x	1	0	1	0
1	0	0	0	1	1	0	1
0	0	0	1	1	1	1	0
x	x	x	x	1	1	1	1

Donde $x_3x_2x_1x_0$ son los bits que indican arriba, abajo, derecha, izquierda (en ese orden), q_1q_0 representan la antigua posición del joystick y d_1d_0 la nueva posición.

A partir de la tabla se crearon los mapas de Karnaugh para d_1 y d_0 :

Tabla 6: Mapa de Karnaugh circuito magic d_1

$x_3x_2x_1/x_0q_1q_0$	000	001	011	010	110	111	101	100
000	0	0	1	1	1	1	0	0
001	0	0	1	1	1	1	0	0
011	0	0	1	1	1	1	0	0
010	1	1	1	1	1	1	0	0
110	0	0	1	1	1	1	0	0
111	0	0	1	1	1	1	0	0
101	0	0	1	1	1	1	0	0
100	0	0	0	0	1	1	0	0

Tabla 7: Mapa de Karnaugh circuito magic d_0

$x_3x_2x_1/x_0q_1q_0$	000	001	011	010	110	111	101	100
000	0	1	1	0	0	0	0	0
001	1	1	1	1	0	1	1	0
011	0	1	1	0	0	1	1	0
010	0	1	1	0	0	1	1	0
110	0	1	1	0	0	1	1	0
111	0	1	1	0	0	1	1	0
101	0	1	1	0	0	1	1	0
100	0	1	1	0	0	1	1	0

De los mapas de Karnaugh se obtienen las siguientes ecuaciones:

- $d_1 = \neg x_3x_2\neg x_1\neg x_0 \vee \neg x_3x_2\neg x_1q_1 \vee x_3x_1q_1 \vee x_2\neg x_1q_1 \vee \neg x_3q_1 \vee x_0q_1$
- $d_1 = \neg x_3\neg x_2x_1\neg x_0 \vee \neg x_3x_1x_0q_0 \vee x_2x_0q_0 \vee x_3x_0q_0 \vee \neg x_0q_0$

Los circuitos implementados se pueden ver en la Figura 5 (d_1), Figura 6 (d_0), Figura 7 (circuito magic).

3.3. selector_2x4

Este circuito tiene una entrada de 2 bits y una salida de 4 bits (en la salida hay un bit que es 1 y el resto es 0), tiene la siguiente tabla de verdad:

Tabla 8: Tabla de verdad circuito selector

x_1	x_0	y_3	y_2	y_1	y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

A partir de la tabla se obtienen los mapas de Karnaugh:

Tabla 9: Mapas de Karnaugh selector

y_3 :	x_1/x_0	0	1
	0	0	0
	1	0	1
y_2 :	x_1/x_0	0	1
	0	0	0
	1	1	0
y_1 :	x_1/x_0	0	1
	0	0	1
	1	0	0
y_0 :	x_1/x_0	0	1
	0	1	0
	1	0	0

Donde se obtienen las siguientes ecuaciones:

- $y_3 = x_1x_0$
- $y_2 = x_1\neg x_0$
- $y_1 = \neg x_1x_0$
- $y_0 = \neg x_1\neg x_0$

Se puede ver el circuito implementado en la Figura 8.

3.4. Bloque de Control

Finalmente se toman los 3 circuitos y se juntan, este bloque tiene 5 entradas, 2 de 4 bits que representan al joystick, 1 de 2 bits que representa la posición del 0 y 2 entradas de 1 bit, una es el reloj y la otra es un botón. El circuito tiene 3 salidas, una de 4 bits que representa el led del display en el que esta el cursor, una salida de 2 bits que representa la posición del cursor y una salida de 1 bit que indica si hay que intercambiar.

El joystick y el reloj se conectan al circuito CursorControl, la salida de CursorControl se conecta a la salida de 2 bits, también se conecta al circuito selector y al circuito vicinity, las salidas del circuito selector se conectan a un splitter y del splitter a la salida de 4 bits. Al circuito vicinity también se le conecta la entrada de 2 bits (posición del 0), y su salida se conecta a una compuerta and, junto con la otra entrada de 1 bit (botón), el resultado de la compuerta se conecta a la salida de 1 bit (intercambiar).

Se puede ver el bloque implementado en la Figura 9.

4. Bloque Memoria del Tablero

La memoria del tablero tiene 4 partes: un intercambiador (exchanger) que recibe las posiciones e intercambia la posición de 2 valores, los otros 2 quedan en 0, un banco de memoria (RegisterBank) que va guardando los valores de cada posición del tablero, un selector que selecciona las posiciones que se van a intercambiar y un circuito que calcula la posición del 0 (ZeroCalc).

4.1. exchanger

Este circuito tiene 6 entradas de 2 bits, 4 son los valores de cada posición y 2 son las posiciones a intercambiar.

Las 4 entradas se conectan a 2 mux en paralelo, y en la entrada select de un mux se conecta una de las entradas de 2 bits, y en el otro se conecta la otra entrada, así los valores que entrega cada mux es el que se encuentra en las posiciones seleccionadas, ahora se conectan las salidas de los mux a un demux cada una, y en la entrada select del demux se conecta la posición contraria a la que se uso en el mux correspondiente (si en un mux se conectó p1, entonces en el demux se conecta p2, p1 y p2 son las posiciones recibidas por el circuito), luego conecto las salidas correspondientes de los demux a compuertas or (la salida 00 de un demux con la 00 de el otro), y las compuertas las conecto con las salidas en orden (la salida 00 va a ser la que se obtuvo con las comparaciones 00 de los demux), esto va a entregar los valores cambiados de las posiciones deseadas, y en las que no se cambiaron quedaron en 0 (no importa el valor).

La implementación del circuito se puede ver en la Figura 10.

4.2. RegisterBank

Este circuito corresponde a la memoria del circuito principal, aquí se guarda que valor hay en cada posición del tablero.

Tiene 10 entradas:

- 4 de 2 bits que representan valores a guardar.
- 4 de 1 bit que indican si un valor se guarda o no en el registro.
- 2 de 1 bit que representan a un reloj y a un reset

Este circuito tiene 4 registros de 2 bits (1 para cada posición), en la entrada Q de cada uno se conecta la salida de un mux, a cada mux se le conectan 2 cosas: una de las entradas de 2 bits y una constante que representa el valor inicial, y en el bit de selección se conecta la entrada reset, para que cuando se presione el botón reset se elija como valor la constante inicial. En la entrada enable de cada registro se conecta la salida de una compuerta or, esta compuerta recibe 2 cosas: una de las entradas de 1 bit y la entrada reset, esto es para que cuando se reciba reset se guarde en el registro la constante inicial, y cuando se reciba el select se guarde el valor recibido.

Las salidas D de los registros son las salidas del circuito RegisterBank.

Se puede ver el circuito implementado en la Figura 11.

4.3. selector

Este circuito recibe 2 entradas de 2 bits (índices del tablero) y entrega 4 salidas de 1 bit, 2 de estas salidas van a ser 1 (las posiciones a cambiar) y las otras 2 son 0.

Para implementarlo se utilizan 2 circuitos selector_2x4 (construidos en la sección 3.3) en paralelo, y las salidas correspondientes de cada uno se conectan a una compuerta or, y el resultado a una salida del circuito, esto es para entregar 1 cuando uno de los selector_2x4 entregue 1, así van a haber 2 salidas 1 y 2 salidas 0.

Se puede ver el circuito implementado en la Figura 12.

4.4. ZeroCalc

Este circuito tiene 3 entradas de 2 bits y una salida de 2 bits, que entrega el índice de la entrada que tiene el valor 0 (si hay mas de una entrada con valor 0 va a entregar un or entre los índices de esas entradas, y si el cero no esta en ninguna de las 3 entradas entonces entrega 00).

En este circuito se utiliza un comparador '=' por entrada, y se compara la entrada con 0, la salida de el comparador va a un bit selector de un mux, y en el mux están conectados un 0 y una constante que representa el índice de la entrada que se esta comparando, luego las salidas de los 3 mux van a una compuerta or, y la salida de la compuerta va a la salida del circuito, por lo que si el 0 está en una de las entradas, el índice de esa entrada va a pasar por el mux y por el or a la salida, y si no está en ninguna de las 3, el or entrega 00, que sería el índice de la posición en la que se encuentra el 0.

Se puede ver el circuito implementado en la Figura 13.

4.5. BoardMemory

Finalmente, se arma el circuito principal de memoria del tablero, este circuito tiene 4 entradas: una de 2 bits que representa la posición del joystick, una de 1 bit que indica si se intercambia o no, y otras 2 de 1 bit que son reloj y reset. También tiene 5 salidas de 2 bits, 4 que indican los valores de cada posición del tablero y una que indica en que posición esta el 0.

Partiendo con un selector, en una de sus entradas se conecta la posición del cursor, y cada una de sus salidas se conecta a una compuerta and, junto con la entrada exchange que indica si se intercambia o no, luego las salidas de las compuertas se conectan a las 4 entradas de 1 bit del circuito RegisterBank, ahora, las salidas del RegisterBank van a las salidas del circuito principal y también se conectan a las 4 entradas de un circuito exchanger y un circuito ZeroCalc, la salida del ZeroCalc va a la salida que indica la posición del 0 y también va a las entradas de los circuitos selector y exchanger. En el circuito exchanger se conecta la entrada de posición del cursor, y las salidas van a las 4 entradas del RegisterBank.

Así el RegisterBank parte con las posiciones iniciales y se las entrega al exchanger y el exchanger le entrega las posiciones intercambiadas al RegisterBank, y solo se intercambian cuando el selector

lo indica.

Se puede ver el circuito implementado en la Figura 14

5. Circuito Principal

Para armar el circuito principal se ingresaron los siguientes elementos de input:

- 1 joystick
- 1 reloj
- 2 botones
- 4 displays de 7 segmentos

Se conectó el joystick, 1 botón y el reloj al circuito ControlBlock, la salida de 4 bits para posición del cursor se conectó a un splitter y se conectó a los 4 displays. El circuito ControlBlock recibe la posición del 0 desde el circuito BoardMemory, y BoardMemory recibe las entradas exchange y cursor_pos desde ControlBlock, también usan el mismo reloj, además, BoardMemory se conecta a otro botón para resetear y poner los valores del tablero en los valores iniciales.

Finalmente, las 4 salidas de BoardMemory se conectan a las 4 entradas de DecoBlock, y las salidas de DecoBlock se separan con splitters para conectarse cada una a un display de 7 segmentos.

Se puede ver el circuito implementado en la Figura 15, también se puede ver el tablero con posiciones cambiadas a través del joystick y el botón en la Figura 16.

6. Anexos

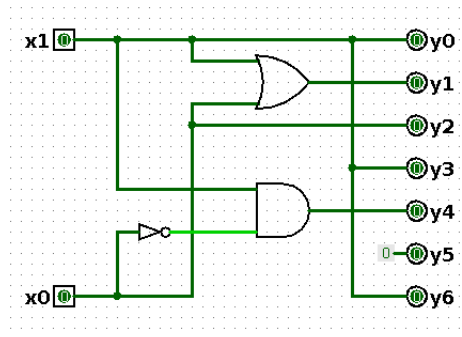


Figura 1: Decodificador 2 x 4

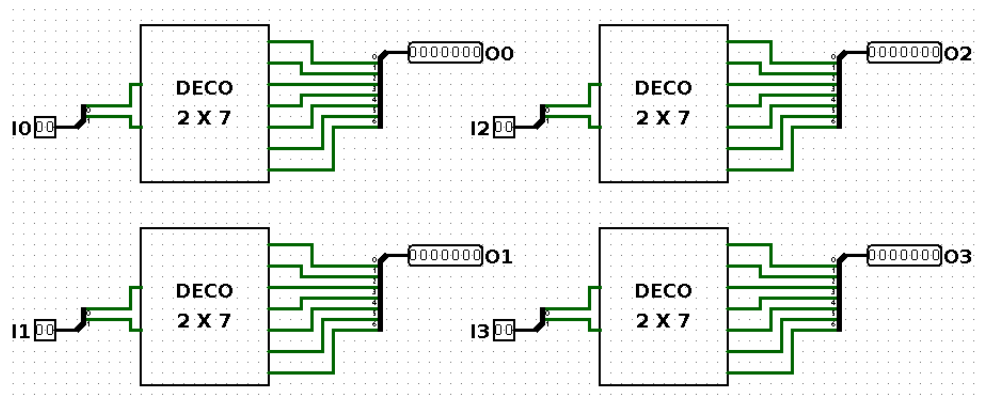


Figura 2: Bloque de decodificadores

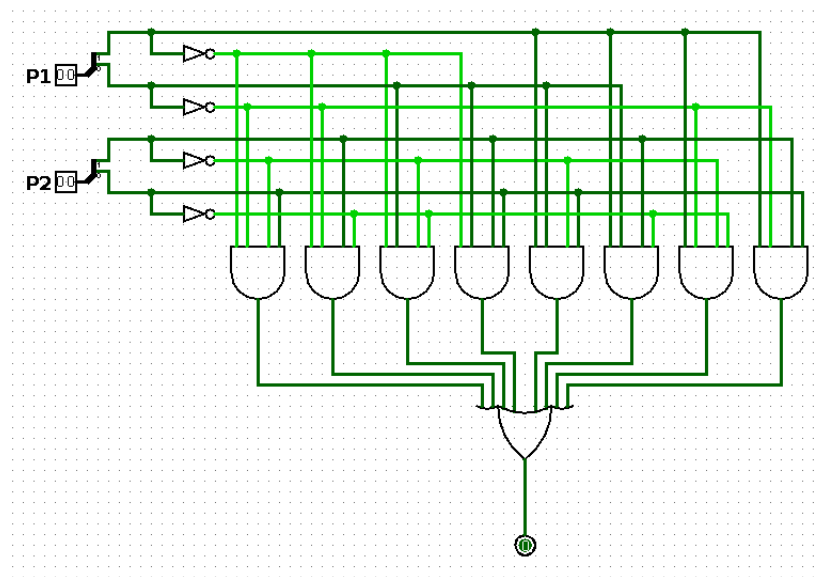


Figura 3: Detector de vecindad

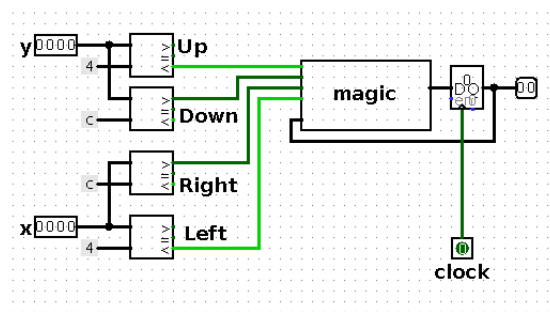


Figura 4: Circuito de control para el cursor

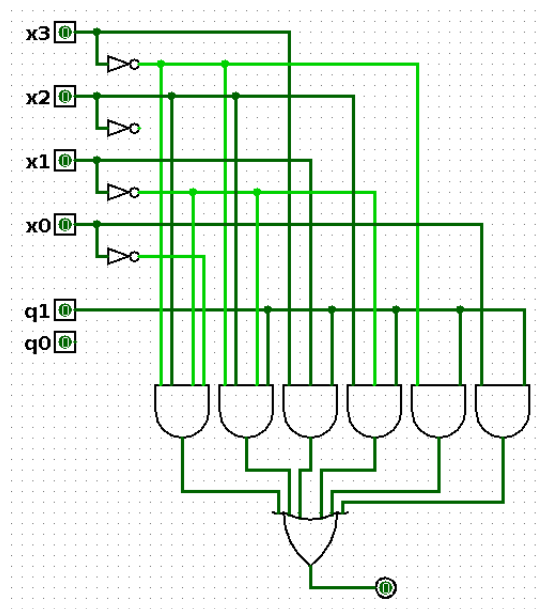


Figura 5: Circuito magic bit 1

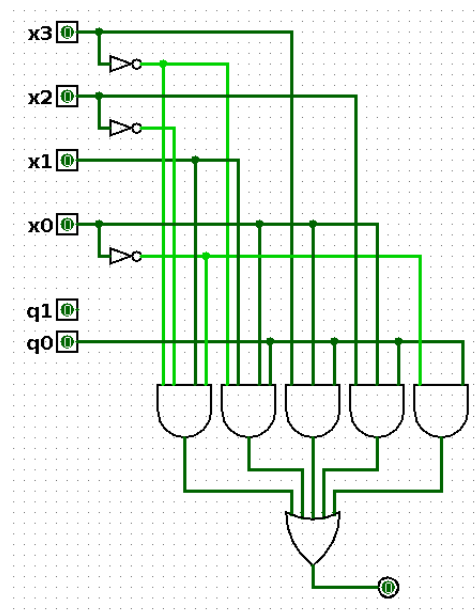


Figura 6: Circuito magic bit 0

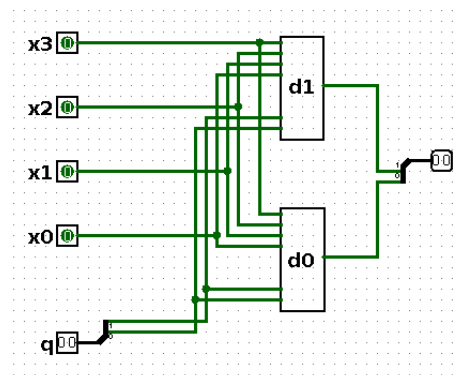


Figura 7: Circuito magic

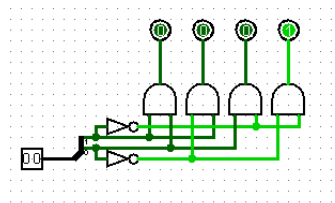


Figura 8: Circuito selector 2 x 4

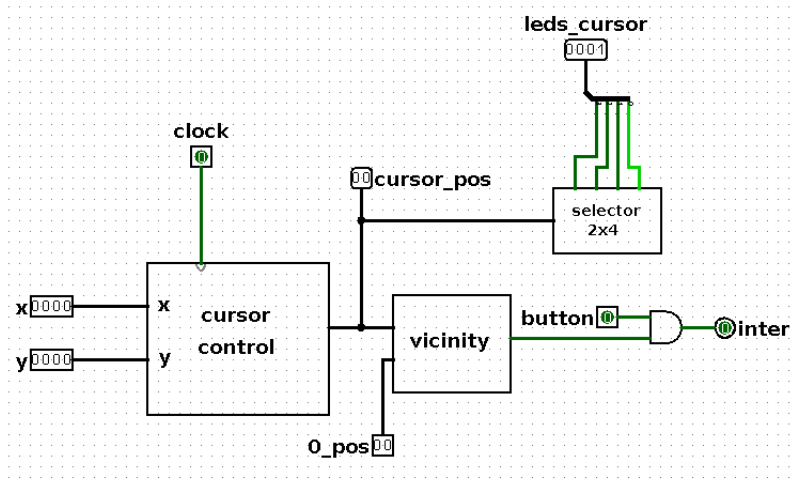


Figura 9: Bloque control

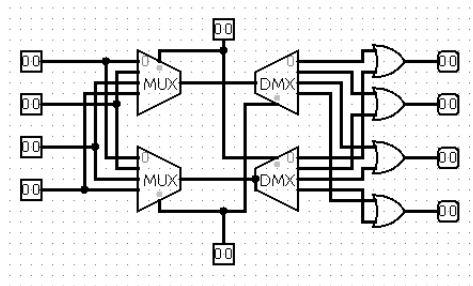


Figura 10: Circuito intercambiador

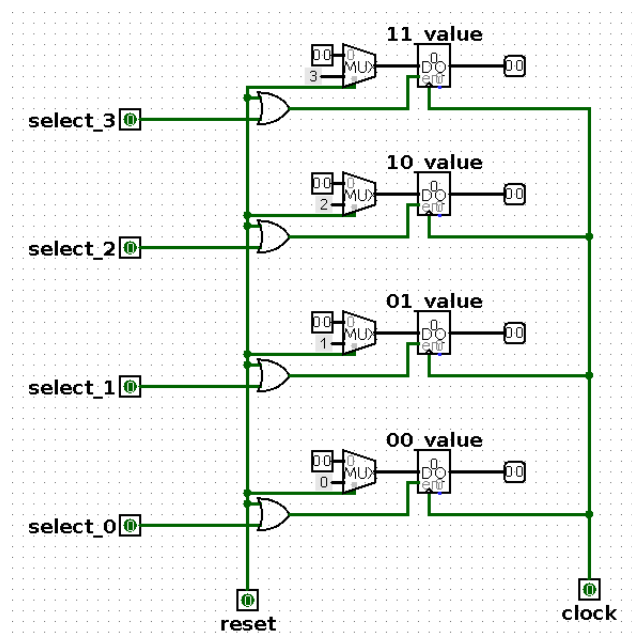


Figura 11: Banco de registros

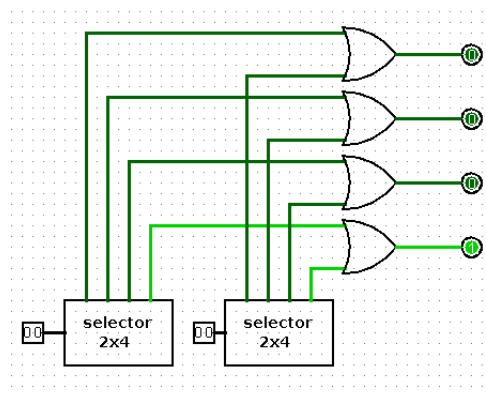


Figura 12: Circuito para el selector

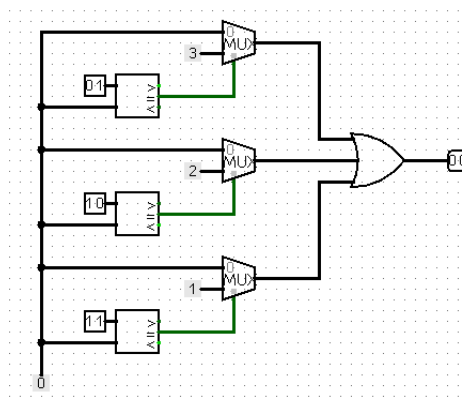


Figura 13: Circuito que calcula la posición del 0

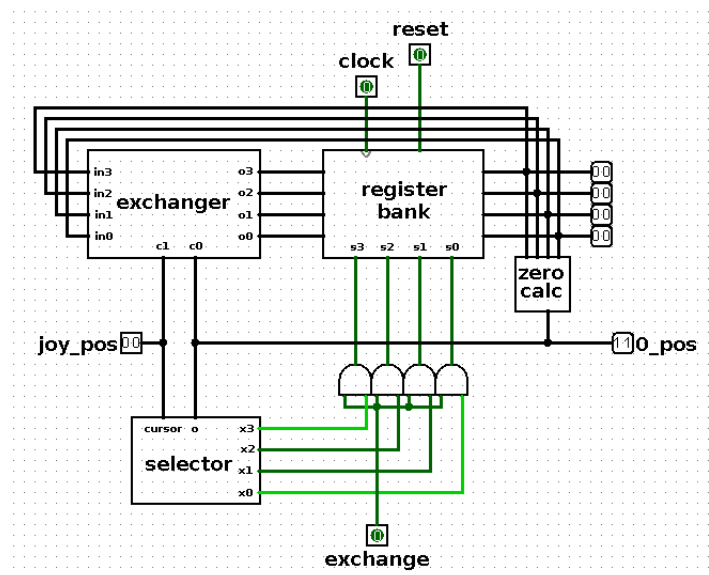


Figura 14: Memoria del Tablero

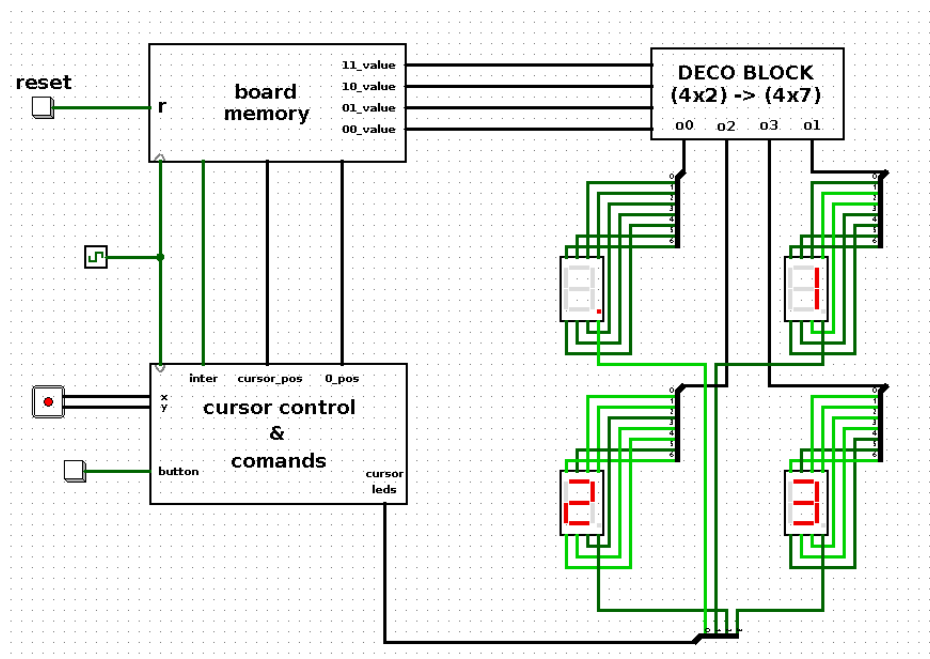


Figura 15: Circuito principal

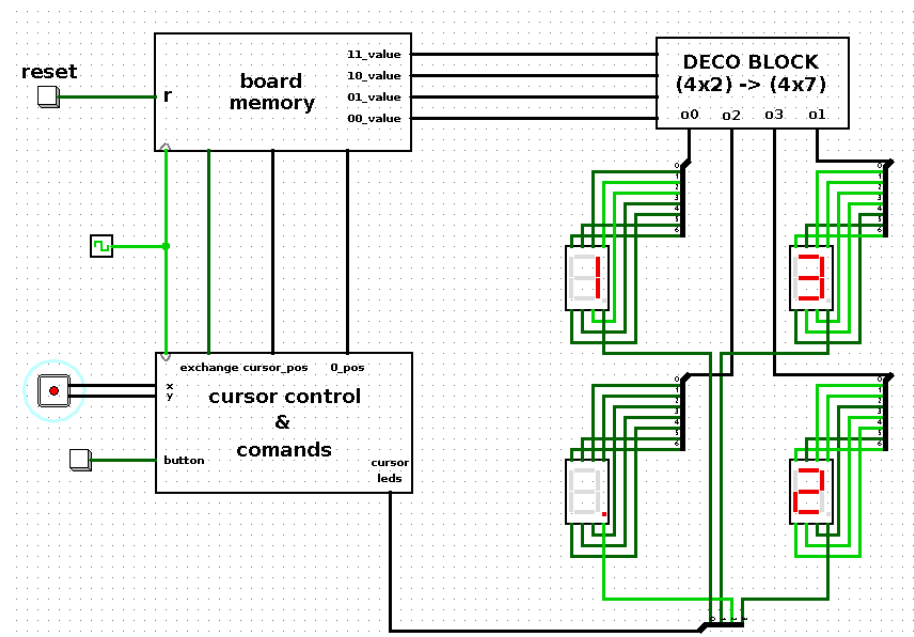


Figura 16: Circuito principal después de cambiar posiciones