

Fundamentos de los computadores

Práctica 3

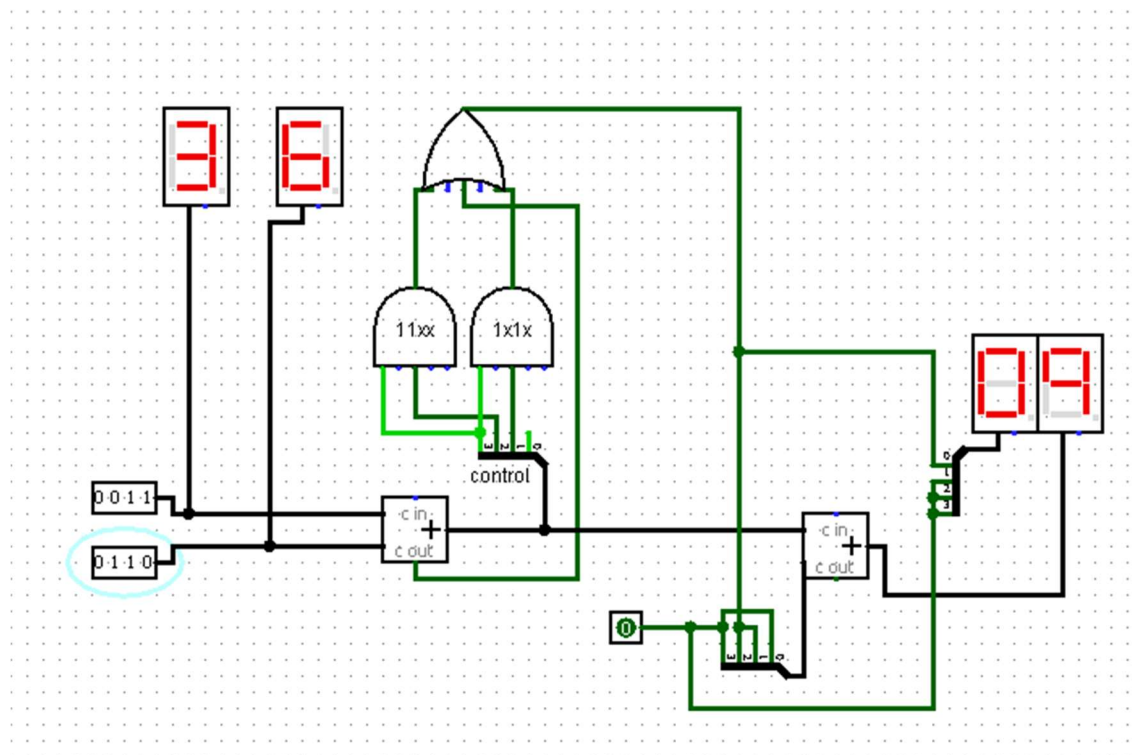
1) Diseña un sumador BCD de 4 bits. Explica su funcionamiento. Impleméntalo con el circuito sumador dotando a las entradas y salida de 4 bits y las puertas lógicas necesarias. Verifica el diseño efectuando las sumas de prueba que se indican. Muestra los operandos y los resultados con displays con entrada hexadecimal (Hex Digit Display).

- 0011 + 0110
- 1000 + 0101
- 1000 + 1001

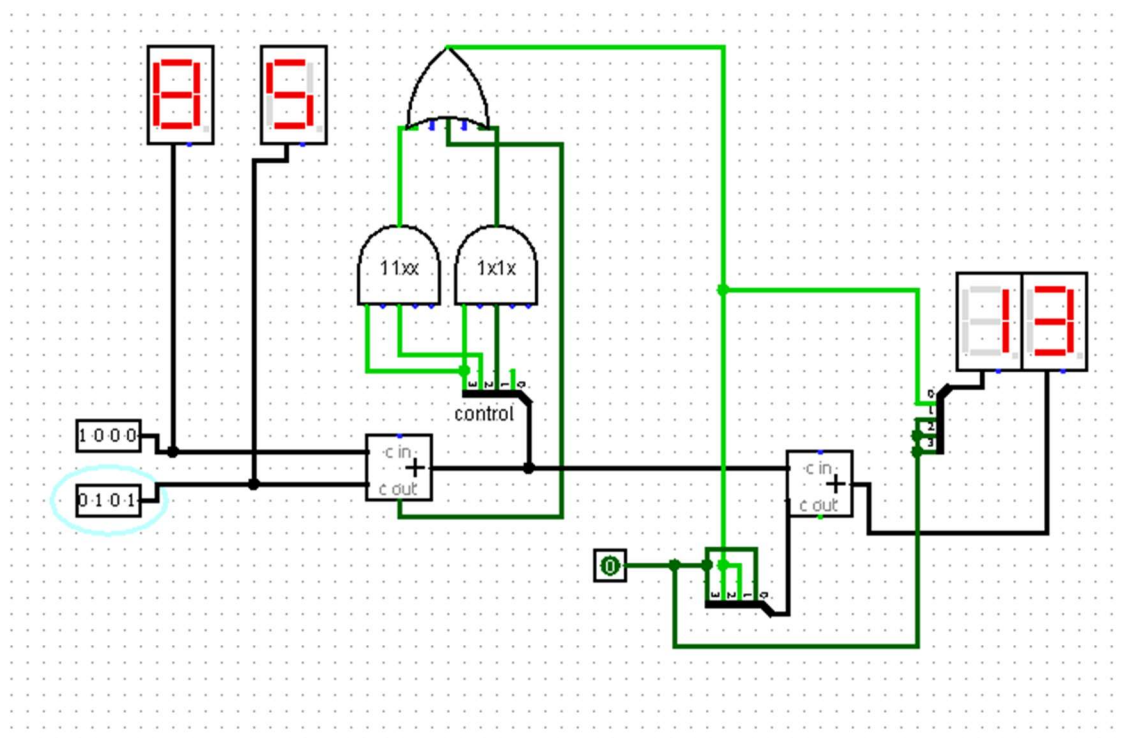
Este circuito tiene como entrada dos números de 4 bits, con displays de entrada hexadecimal para cada uno, luego esos dos números se suman con un sumador de 4 bits y como es en formato BCD, si el resultado es mayor que 9 (1001), a este mismo se le suma 6 (0110), de lo contrario el resultado se deja como esta (Se le suma 0) y aparece en el display con entrada hexadecimal.

Ejemplos:

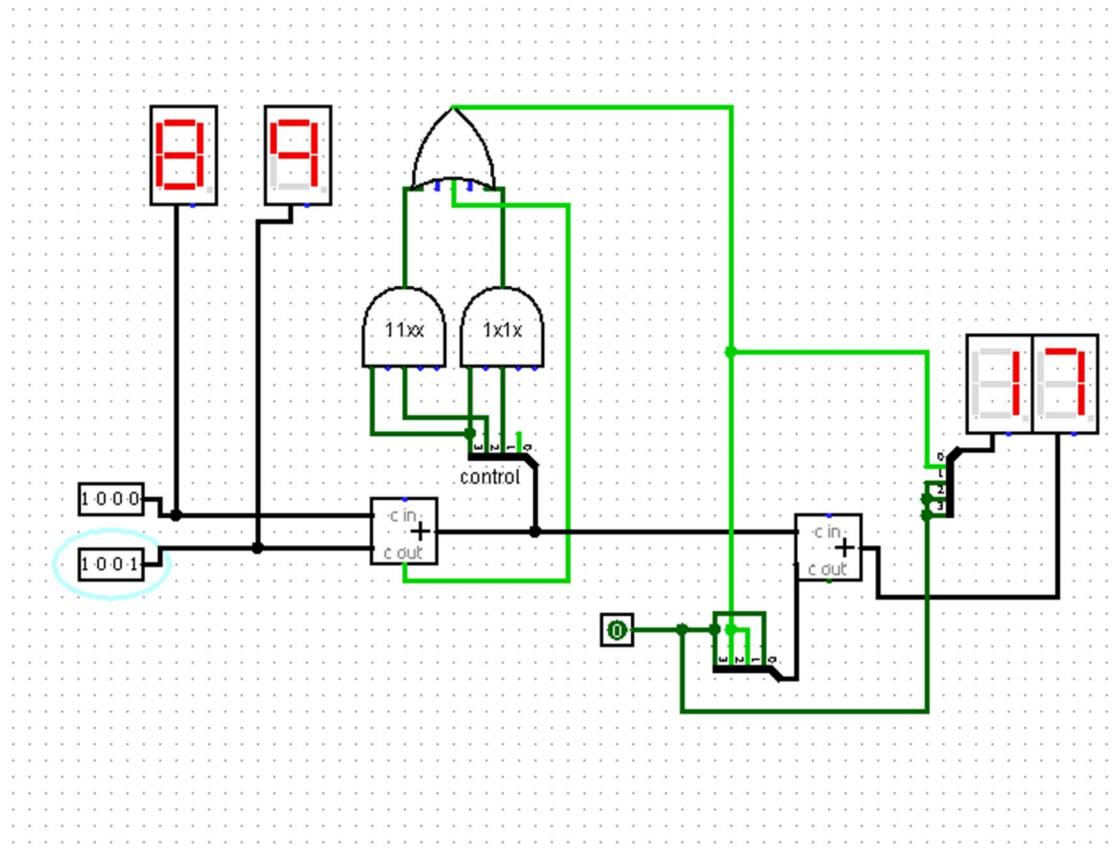
- 0011 + 0110



b. $1000 + 0101$



c. $1000 + 1001$



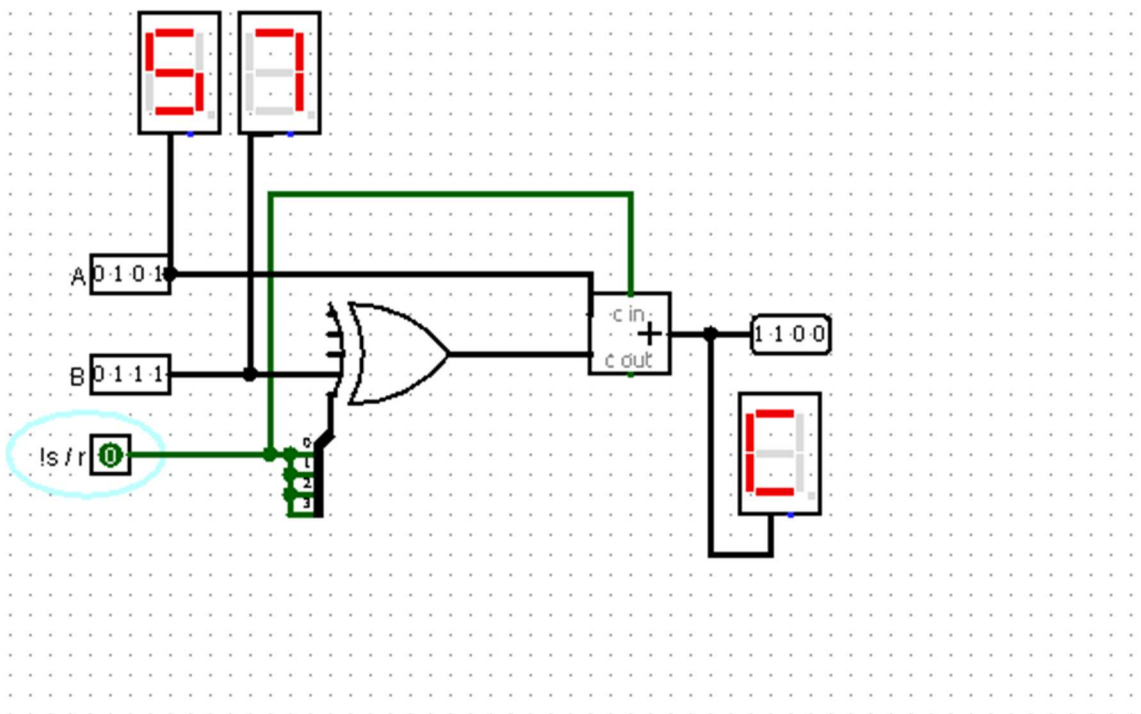
- 2) Diseña un sumador-restador de 4 bits en complemento a 2 basado en el circuito estudiado en clase. Explica la misión que tienen las puertas XOR en el circuito y como actúa la señal SR. Indica el rango de funcionamiento. Impleméntalo con un sumador de 4 bits. Verifica el dispositivo efectuando las sumas de prueba que se indican. Muestra los resultados con visualizadores y comenta los resultados obtenidos en cada caso indicando si son correctos o no.
- 5+7
 - 2-6
 - 8+4
 - 7-5

Este circuito tiene como entrada 2 números de 4 bits, y una entrada !S/R (si estamos sumando la entrada es 0, y si estamos restando es 1), la cual junto con la puerta XOR sirve para invertir el segundo número y sumar 1 como acarreo, es decir, pasarlo a complemento a 2, en caso de la resta. En caso de la suma ambas entradas se quedan igual, se suman, y se conecta a un display de entrada hexadecimal y una salida, que al ser de 4 bits en complemento a 2, solo tendrá un rango de funcionamiento de -8 a +7.

Ejemplos:

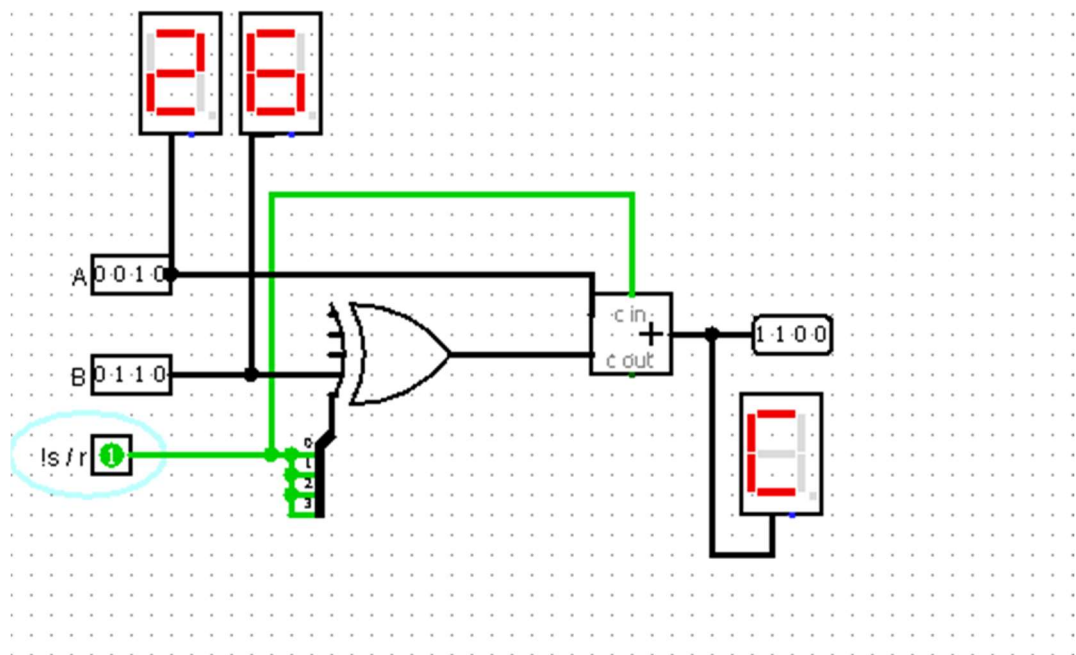
- 5+7=12

Como el resultado esta fuera del rango de funcionamiento [-8,+7], el resultado que muestra es erroneo.



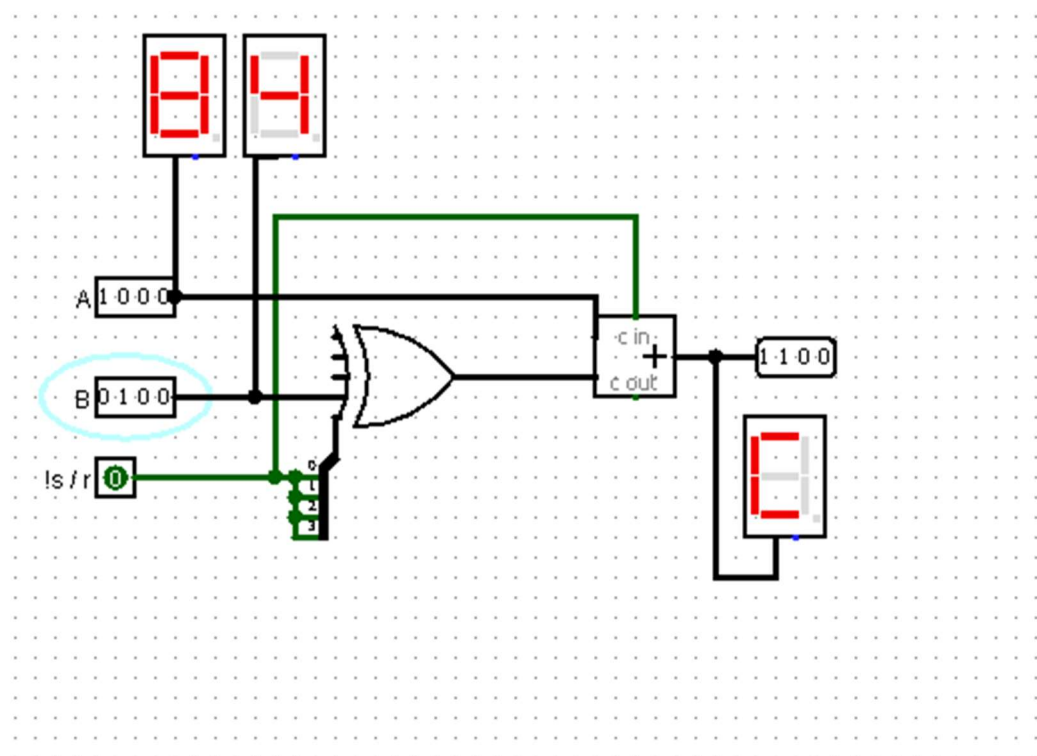
- 2-6=-4

Como el resultado está dentro del rango de funcionamiento [-8,+7], se representa correctamente en la salida de 4 bits, pero no en el display hexadecimal, ya que el resultado es en complemento a 2.



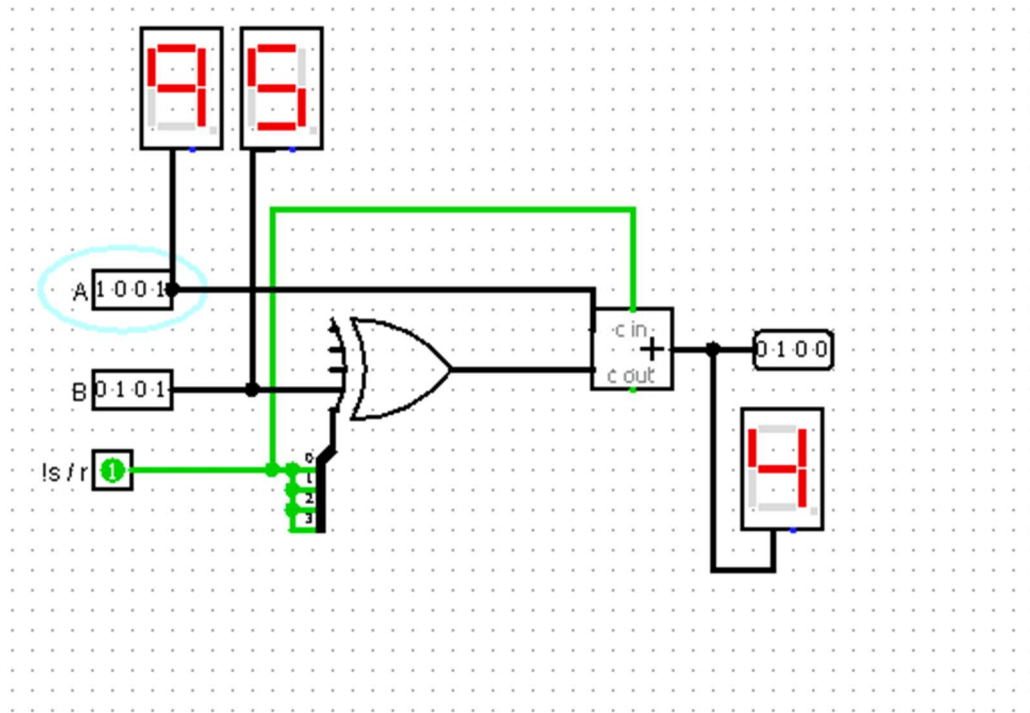
c. $-8+4=-4$

Lo mismo que en el apartado anterior, el resultado está dentro del rango de funcionamiento $[-8,+7]$, y se representa correctamente en la salida de 4 bits, pero no en el display hexadecimal, ya que el resultado es en complemento a 2, y el display no lo tiene en cuenta.



d. $-7-5=-12$

Como el resultado esta fuera del rango de funcionamiento $[-8,+7]$, el resultado que muestra es erróneo.



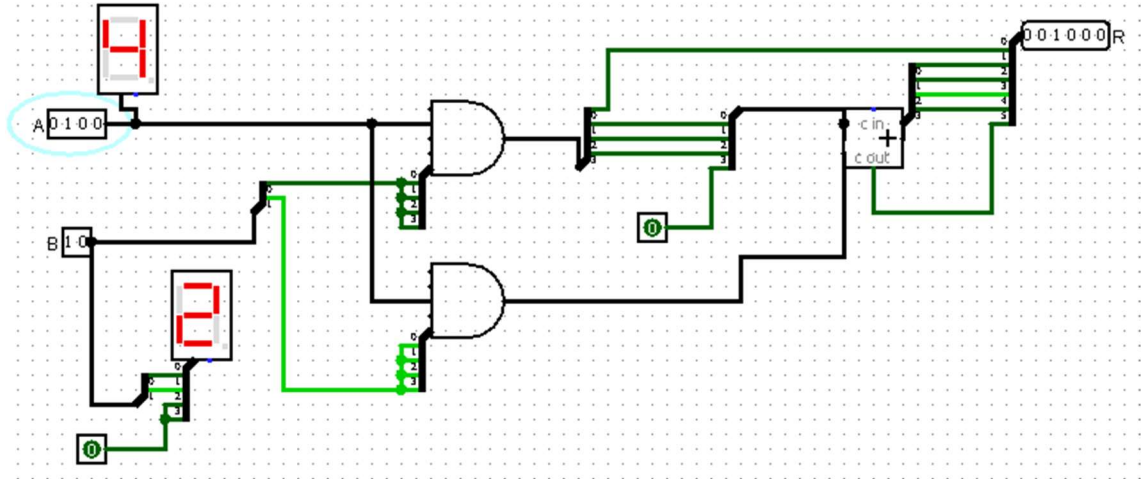
3) Diseña un multiplicador de dos números binarios: uno de 4 bits y el otro de 2 bits, $A(A_3,A_2,A_1,A_0)$ y $B(B_1,B_0)$. Impleméntalo con circuitos sumador de 4 bits y las puertas lógicas que sean necesarias. Verifica el funcionamiento con dos ejemplos.

Este circuito recibe una entrada A de 4 bits, y una entrada B de 2 bits. La entrada A pasa por 2 puertas AND, en las cuales pasa solo 1 valor de la entrada B conectado a un separador de 4 bits. Luego el resultado de la primera puerta $A \times B_0$ se conecta a otro separador y el dígito en A_0 pasa directamente al resultado (a R_0), mientras que los otros dígitos se suman con el resultado de la otra puerta, y dan como resultado a una salida de 6 bits (R).

Ejemplos:

a. $4 \times 2 = 8$

$$000100 \times 000010 = 001000 = 8$$



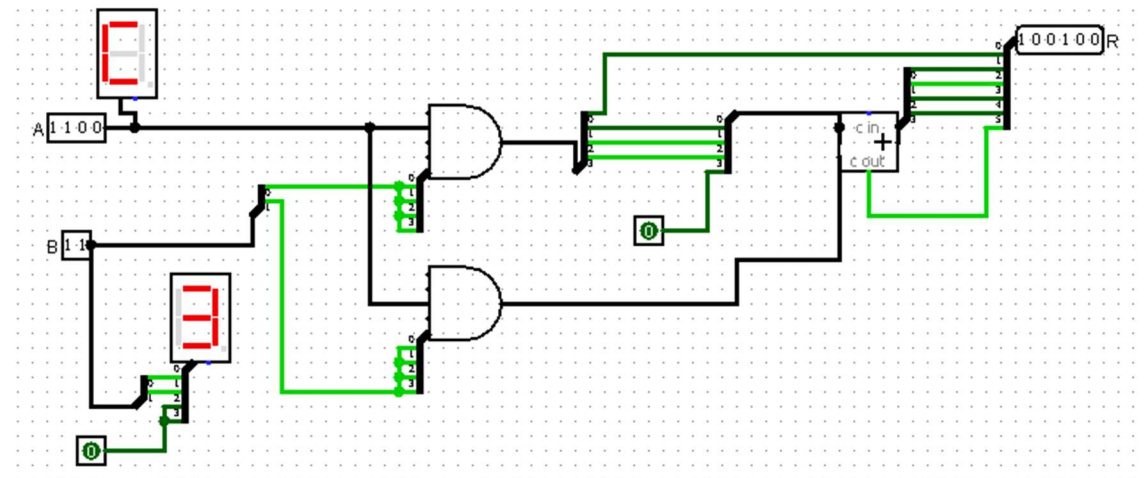
b. $12 \times 3 = 32$

$$001100 \times 000011 =$$

$$001100 \times 000010 + 001100 \times 000001 =$$

$$011000 + 001100 =$$

$$100100 = 32$$



- 4) Se desea diseñar un circuito indicador de rango de temperaturas que disponga de dos salidas, una constituida por un led rojo y la otra por uno verde. El led verde se activara cuando la temperatura de entrada se encuentre en el rango $[-3, +4]$ °C, y el rojo cuando se alcancen los 6 grados tanto positivos como negativos. La información sobre la temperatura la recibe el circuito codificada con cuatro bits binarios en complemento a 2.
Escribe su tabla de verdad e implementa dicho circuito utilizando un solo decodificador en logisim. Verifica su correcto funcionamiento.

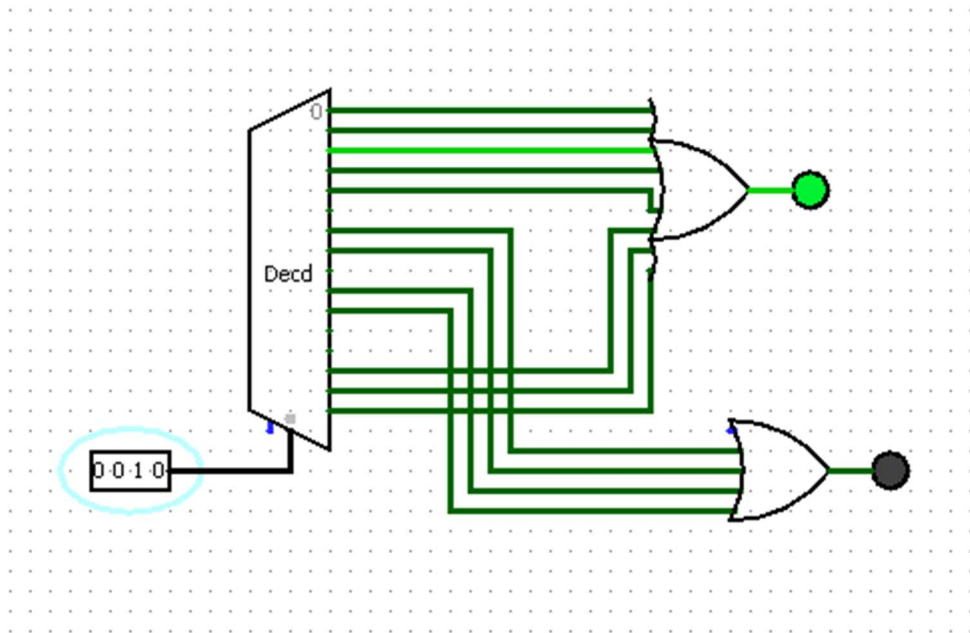
Este circuito recibe una entrada de 4 bits (La temperatura), que pasa por un solo decodificador, y dependiendo del valor de entrada, se encenderá un LED verde o un LED rojo.

Para determinar que salidas del decodificador se conectan a cada LED utilizamos la siguiente tabla de verdad.

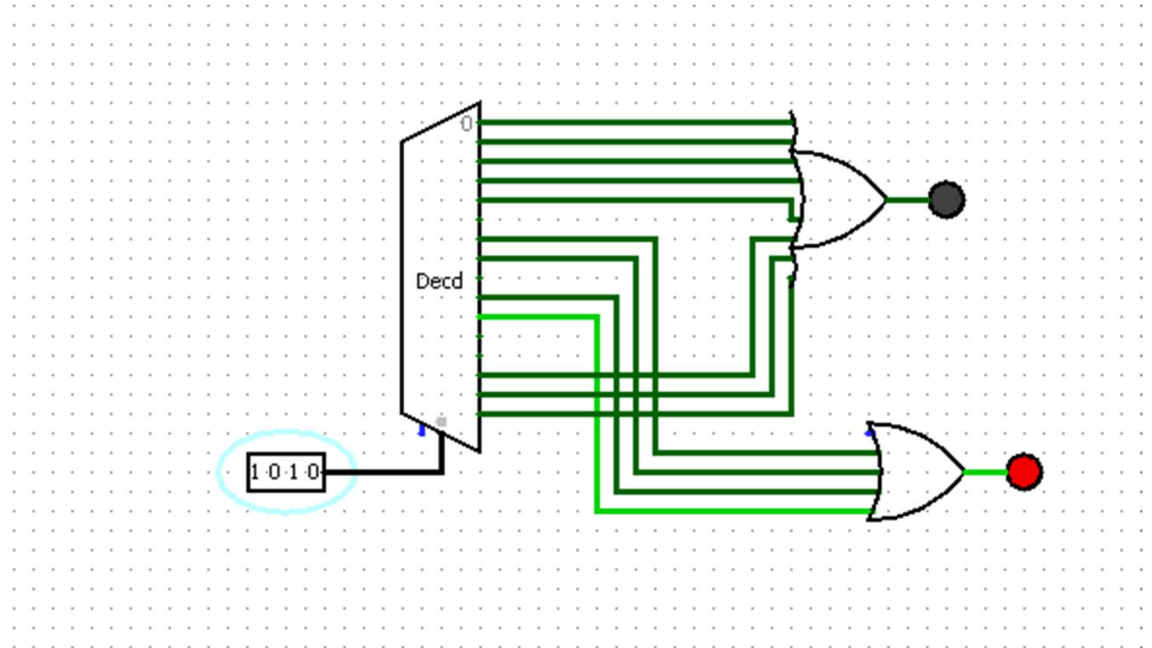
	a	b	c	d	S	°C
0	0	0	0	0	V	+0
1	0	0	0	1	V	+1
2	0	0	1	0	V	+2
3	0	0	1	1	V	+3
4	0	1	0	0	V	+4
5	0	1	0	1		
6	0	1	1	0	R	+6
7	0	1	1	1	R	+7
8	1	0	0	0		
9	1	0	0	1	R	-7
10	1	0	1	0	R	-6
11	1	0	1	1		
12	1	1	0	0		
13	1	1	0	1	V	-3
14	1	1	1	0	V	-2
15	1	1	1	1	V	-1

Ejemplos:

- a. 2 °C (0010) -> Led verde



- b. -6 °C (1010) -> Led rojo



- 5) Las acciones de una compañía están repartidas en cinco lotes según los siguientes porcentajes: A=23%, B=11%, C=15%, D=32%, E=19%. Las decisiones se toman por mayoría y cada accionista dispone de un interruptor particular en la mesa de juntas de tal forma que si no se acciona (0=voto en contra) y si se acciona (1= voto a favor). Diseña e implementa con logisim un circuito que indique mediante el encendido de un led L cuando se aprueban las propuestas presentadas en la junta de accionistas. Escribe su tabla de verdad y utiliza un multiplexor de 4 líneas de selección para la implementación.

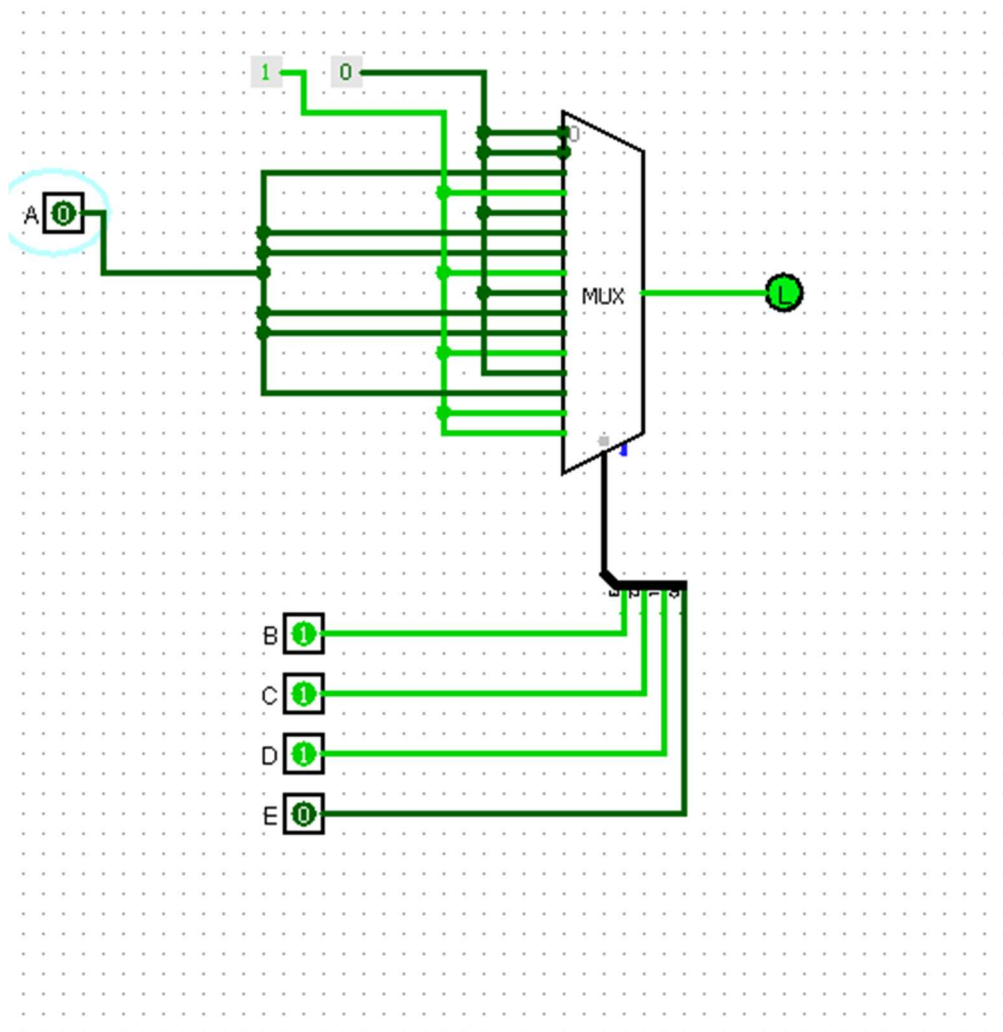
Este circuito recibe 5 entradas de 1 bit. Una de ellas se utiliza como entrada de datos en el multiplexor, solamente en los casos que el porcentaje de votos dependa de ese valor. Y las otras 4 entradas se conectan con un separador para conectar al multiplexor. Para los casos que el porcentaje sea menor de 50% se utiliza una constante 0, y para los que es mayor se utiliza una constante 1, y cuando la salida del multiplexor es 1, se enciende un led L.

Para que sea más fácil detectar con que valores se debería encender el led utilizamos la siguiente tabla, los valores marcados en amarillo son los que dependen de la primer entrada A.

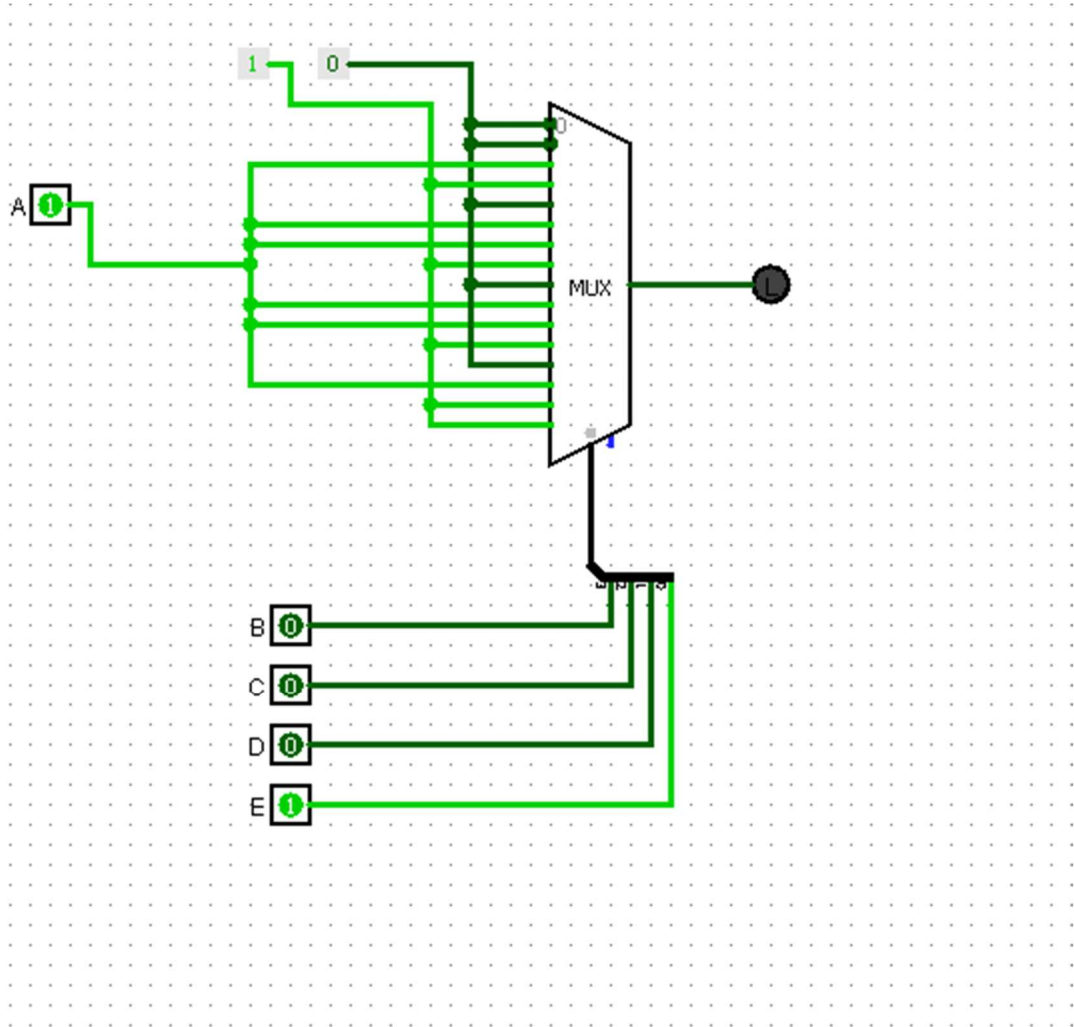
A	B	C	D	E	%	nº Entrada	Salida
0	0	0	0	0	0	0	0
0	0	0	0	1	19	1	0
0	0	0	1	0	32	2	0
0	0	0	1	1	51	3	1
0	0	1	0	0	15	4	0
0	0	1	0	1	34	5	0
0	0	1	1	0	47	6	0
0	0	1	1	1	66	7	1
0	1	0	0	0	11	8	0
0	1	0	0	1	30	9	0
0	1	0	1	0	43	10	0
0	1	0	1	1	62	11	1
0	1	1	0	0	26	12	0
0	1	1	0	1	45	13	0
0	1	1	1	0	58	14	1
0	1	1	1	1	77	15	1
1	0	0	0	0	23	16	0
1	0	0	0	1	42	17	0
1	0	0	1	0	55	18	1
1	0	0	1	1	74	19	1
1	0	1	0	0	37	20	0
1	0	1	0	1	56	21	1
1	0	1	1	0	70	22	1
1	0	1	1	1	89	23	1
1	1	0	0	0	34	24	0
1	1	0	0	1	53	25	1
1	1	0	1	0	66	26	1
1	1	0	1	1	85	27	1
1	1	1	0	0	49	28	0
1	1	1	0	1	68	29	1
1	1	1	1	0	81	30	1
1	1	1	1	1	100	31	1

Ejemplos:

a. 01110=58%



b. 10001=42%



6) Se dispone de dos señales digitales A y B. Cada una de ellas corresponde a un número en binario natural de dos bits ($A=A_1 A_0$ y $B=B_1 B_0$). Se desea construir un circuito combinacional que realice 4 funciones diferentes, según el valor que tomen dos señales de control G_1 y G_0 , tal y como se muestra en la siguiente tabla:

G_1	G_0	Función
0	0	Media redondeada por defecto
0	1	Número A
1	0	Número B
1	1	Media redondeada por exceso

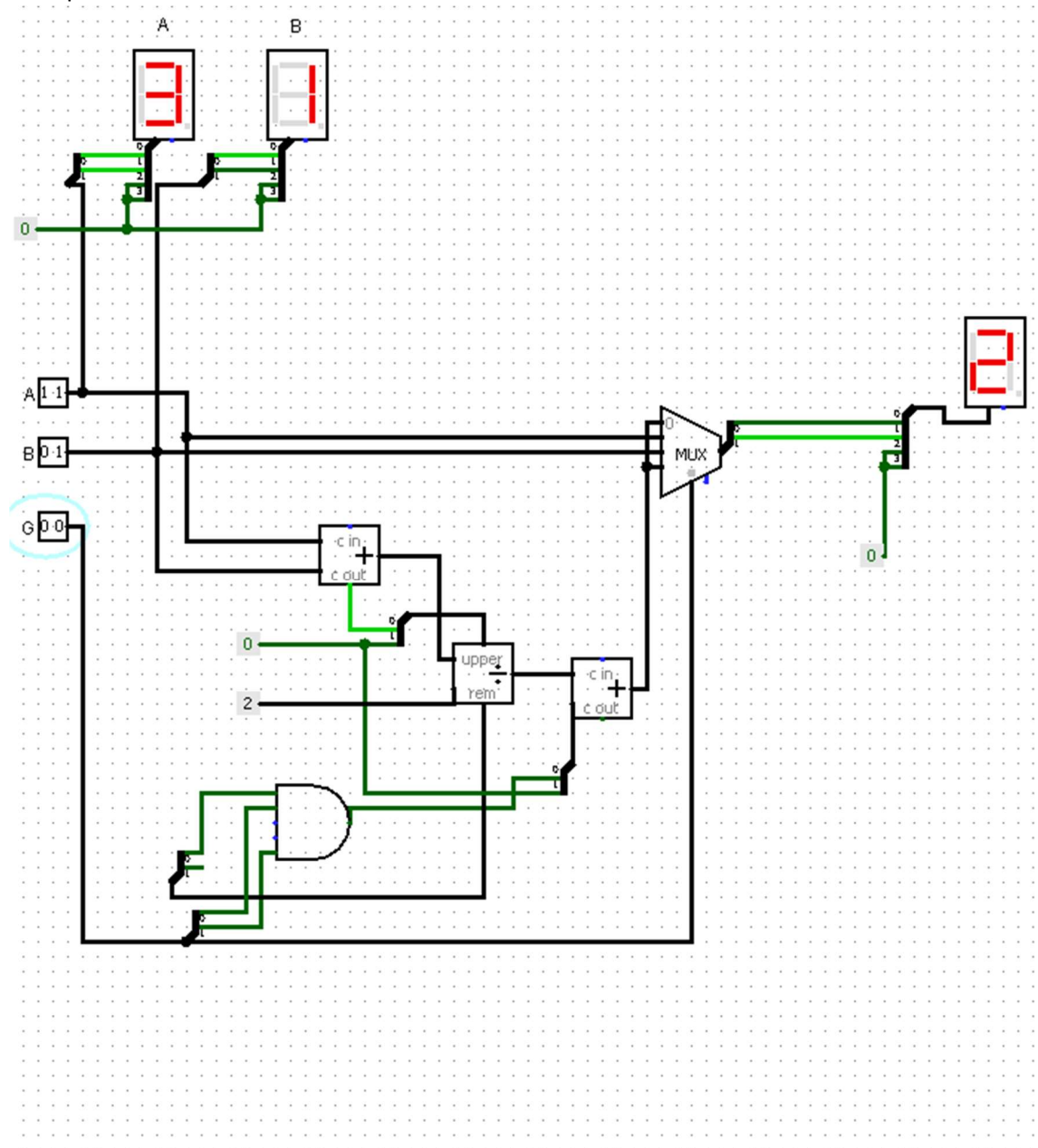
Indica los bloques combinacionales que utilizarías para la implementación de este circuito teniendo en cuenta que el resultado como los datos se mostraran en displays de 7 segmentos con entrada hexadecimal.

Realiza el diseño, impleméntalo en logisim, y comprueba el correcto funcionamiento.

Este circuito recibe 3 entradas de 2 bits, A, B y G. Con las entradas A y B conectadas a displays de entrada hexadecimal para que sea más fácil ver los valores. G está conectada al multiplexor, y dependiendo del valor de G la salida será una de las 4 entradas del multiplexor:

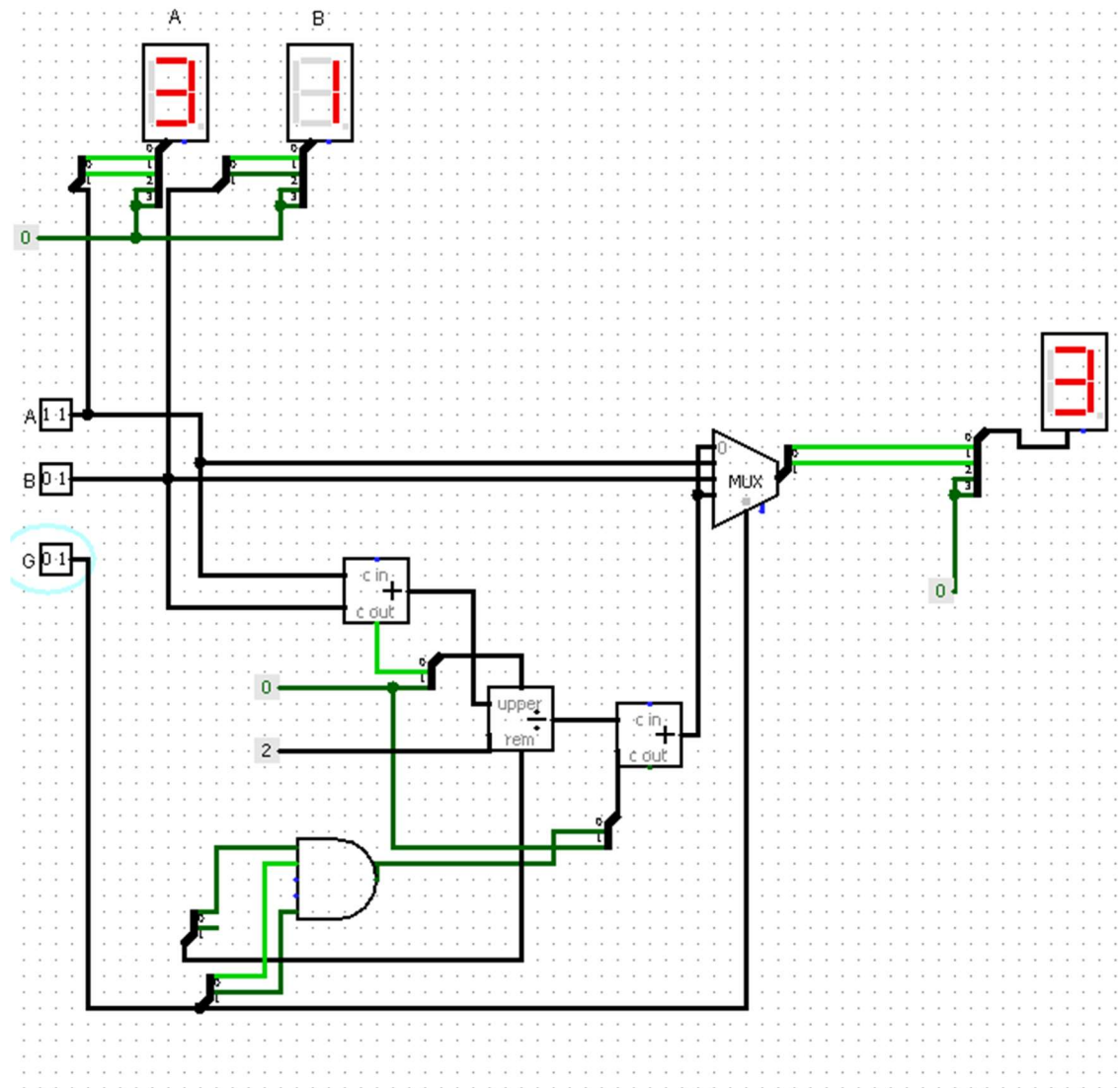
Entrada 0 (G= 0 0)

Para calcular la media redondeada por defecto, sumamos los valores A y B, y al resultado lo dividimos por 2, con el posible acarreo como tercer dígito. Luego el resultado pasa por un sumador, pero en este caso no se le suma nada y pasa al multiplexor.



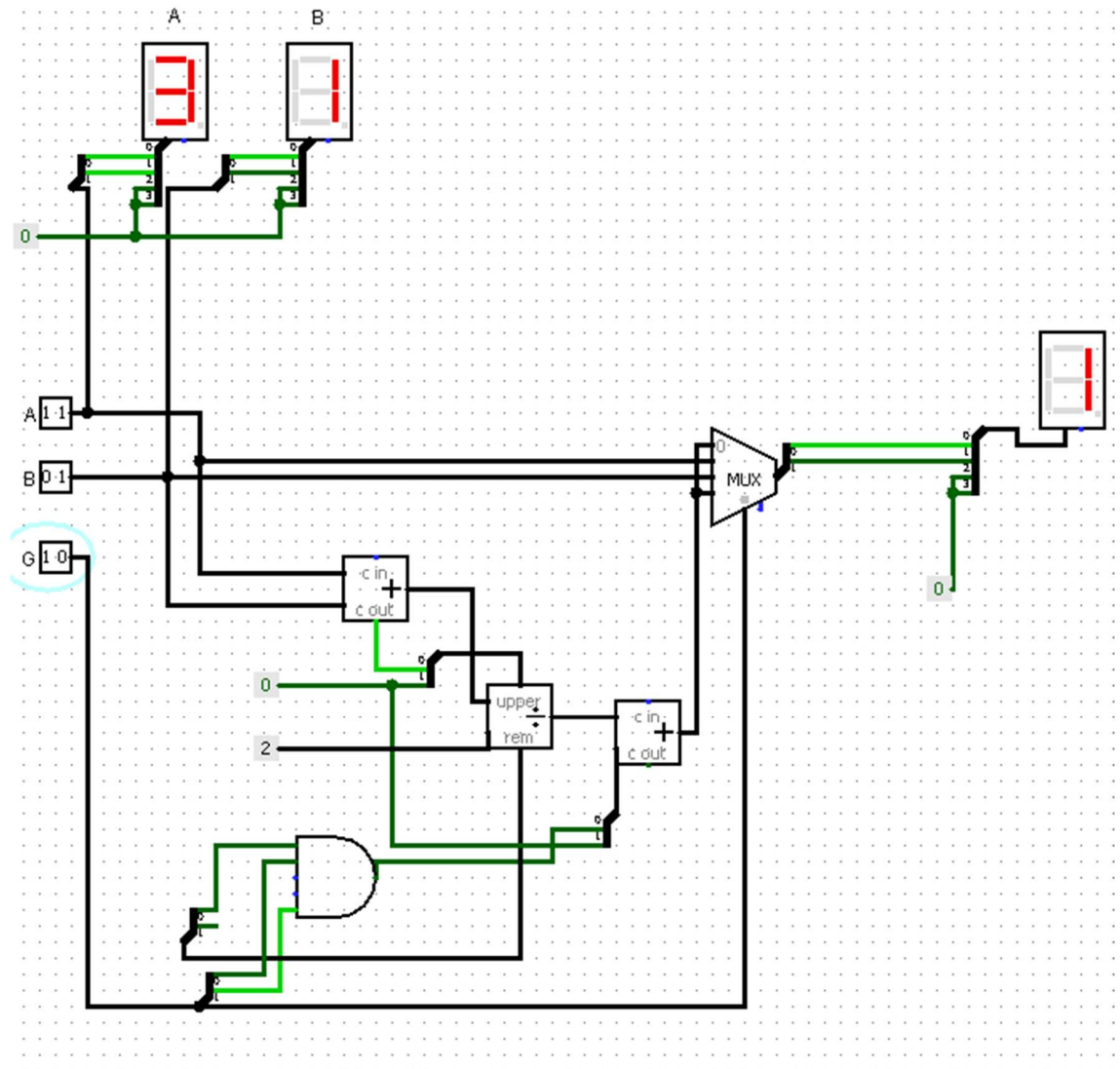
Entrada 1 (G= 0 1)

Directamente conectamos la entrada A a la entrada 1 del multiplexor.



Entrada 2 (G= 1 0)

Directamente conectamos la entrada B a la entrada 2 del multiplexor.



Entrada 3 (G= 1 1)

Para calcular la media redondeada por exceso hacemos el mismo proceso que en el primer caso de la división, solo que después de dividir le sumamos al resultado uno, y se suma gracias a la puerta AND a la que están conectadas los 2 dígitos de G, y el resto de la división, por ultimo se conecta al multiplexor.

