MIGUEL ANGEL ALVAREZ DIAZ

Calculadora Binaria de 3 Bits

El siguiente proyecto consiste en una Calculadora Binaria de 3 Bits conformada por 4 opciones de operaciones y/o herramientas. Las operaciones capaces de realizar de este circuito son suma, resta y multiplicación; la cuarta opción consiste en un contador automático de 4 bits, el cual depende de una señal de reloj para cambiar el valor mostrado de forma consecutiva. Para poder desplazarse entre cada una de las distintas opciones se realizó un Multiplexor 4 a 1.

Todo este proyecto fue desarrollado en la plataforma en línea WOKWI, en donde es posible observar el circuito de forma libre e incluso simularlo para interactuar con sus funciones. La idea de realizar una Calculadora Binaria de 3 Bits nació de la actividad crear un proyecto propuesto por el campamento LATINPRACTICE, el cual, lo recomendable es realizarse con herramientas libres de diseño VLSI, sin embargo, como algunos no poseemos una base sólida de estas herramientas, se nos dio la oportunidad de crear nuestro propio circuito utilizando compuertas lógicas en la plataforma de WOKWI.

https://wokwi.com/projects/373184190141669377

Contenido

Desarrollo del Circuito

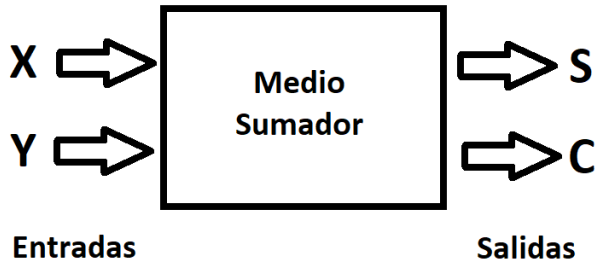
Sumador

https://wokwi.com/projects/373005455971337217

Para poder desarrollar el sumador de 3 bits, es necesario comprender como funciona un medio sumador y un sumador completo.

Medio Sumador

El medio sumador se conforma de dos bits de entrada (X, Y) y dos bits de salida. Las 2 salidas posibles del medio sumador son una salida denominada “S”, y un carry o acarreo denominado “C”, así como se muestra a continuación.

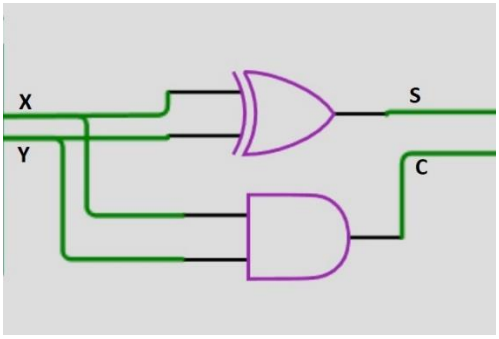


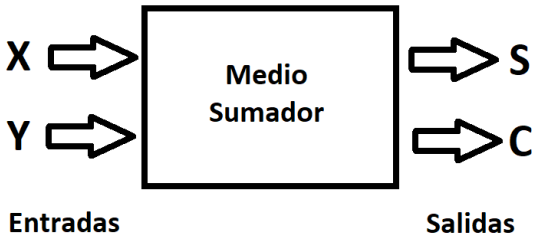
El carry o acarreo es el bit que se genera cuando ambas entradas (X, Y) están activadas y les llegan un bit a cada uno. Se podría decir que generan un 2 en binario (1 0), por lo tanto, la salida “S” no generaría un bit de salida.

Lo que caracteriza al medio sumador de su otra variante (sumador completo) es que solamente tiene dos bits de entrada y solo puede arrojar una sola salida a la vez, ya sea la salida “S” o la salida carry “C”, nunca las dos al mismo tiempo, así como se muestra en su tabla de verdad.

Medio Sumador				
X	Y	Suma (S)	Carry (C)	Mínima Expresión Booleana
0	0	0	0	S = (X' Y) + (X' Y) = X⊕Y
0	1	1	0	C = X Y
1	0	1	0	
1	1	0	1	

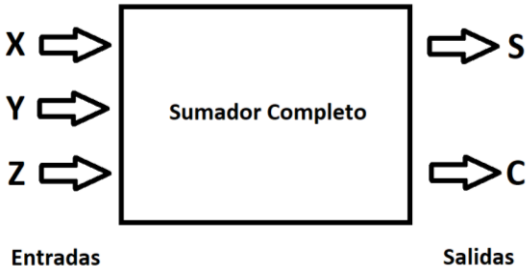
Con las expresiones booleanas de las salidas “S” y “C”, es posible generar un circuito con compuertas lógicas, por lo tanto, su circuito quedaría de la siguiente manera.



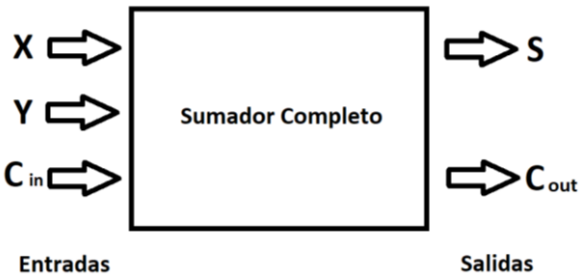


Sumador Completo

A diferencia del medio, el sumador completo se conforma de tres bits de entrada (X, Y, Z) y 2 bits de salida. Las 2 salidas posibles del medio sumador son una salida denominada “S”, y un carry o acarreo denominado “C”, así como se muestra a continuación.



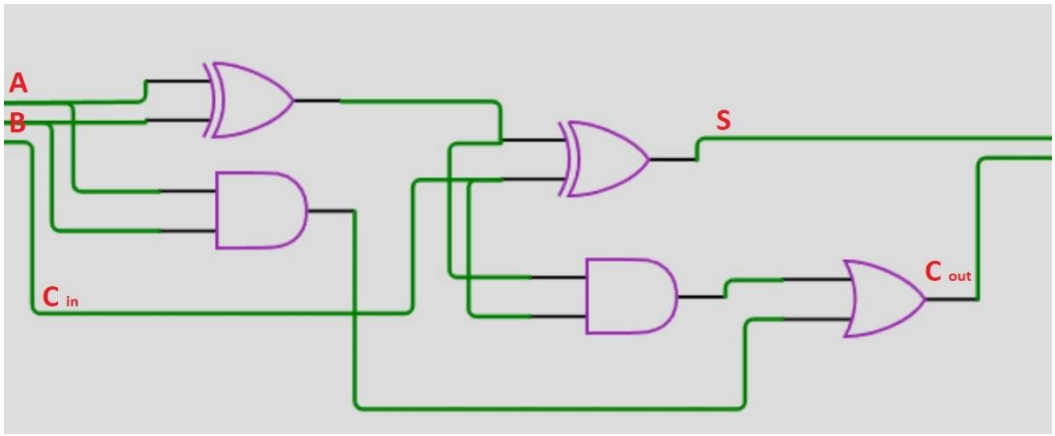
Cabe recalcar que otra forma de ver la entrada “Z” es como “C_{in}” y la salida “C” como “C_{out}”. Esto debido a que se pueden juntar varios sumadores para crear un sumador más grande con más entradas y salidas y está terminología permite una mejor comprensión de lo que se va a conectar. Esto se verá más adelante en el desarrollo del Sumador Binario de 3 Bits.



Lo que caracteriza al sumador completo de su otra variante (medio sumador) es que, debido a sus tres bits de entrada, puede arrojar ambas salidas a la vez, la salida “S” y la salida carry “C”, esto porque al sumar los tres bits de las entradas es como si se generara un número 3 en binario (1 1). También puede arrojar una sola salida a la vez, ya sea la salida “S” o la salida carry “C” si la suma lo permite, así como en el medio sumador. Se puede ver claramente esto en su tabla de verdad.

Sumador Completo (Es la combinacion de dos medios sumadores y una compuerta OR)					
X	Y	Z	Suma (S)	Carry (C)	Mínima Expresión Booleana
0	0	0	0	0	S = (X' Y' Z) + (X' Y Z') + (X Y' Z') + (X Y Z)
0	0	1	1	0	C = (X Y) + (X Z) + (Y Z)
0	1	0	1	0	
0	1	1	0	1	
1	0	0	1	0	
1	0	1	0	1	
1	1	0	0	1	
1	1	1	1	1	

Con las expresiones booleanas de las salidas “S” y “C”, es posible generar un circuito con compuertas lógicas, sin embargo, también se puede generar un sumador completo con dos medios sumadores y una compuerta lógica OR de la siguiente manera:



Se les recuerda que la entrada “Cin” es la entrada “Z” y la salida “Cout” es la salida “C”.

Sumador Binario de 3 bits

Ya que se sabe cómo funcionan y como se componen el medio sumador y el sumador completo, ya es posible realizar el Sumador Binario de 3 Bits.

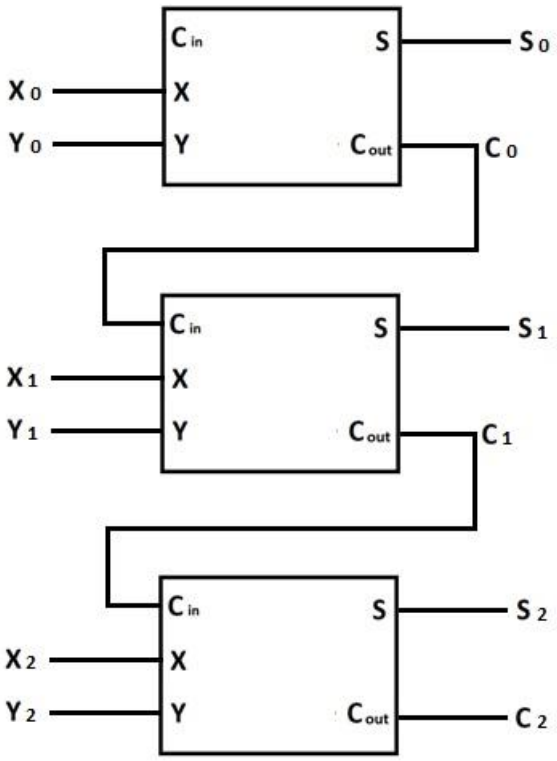
Características del sumador:

- Va a sumar 2 valores de entrada.
- Va a tener un sumando de 3 bits de entrada.
- Va a tener otro sumando de 3 bits de entrada.
- La salida va a tener en total 4 bits.
- El valor máximo de cada sumando es de 7 en binario (1 1 1).
- El valor máximo posible de resultado o salida es de 14 en binario (1 1 1 0).

Para realizar el Sumador Binario de 3 Bits, se tienen que utilizar 3 sumadores completos, los cuales van a ir interconectados de la siguiente manera:

1. Los sumadores completos se van a colocar en forma de lista (esto con el fin de tener un orden).
2. El carry o acarreo (C_{out}) del primer sumador va a ir conectado al carry de entrada (C_{in}) del siguiente sumador. Se repite lo mismo, pero con el segundo y el tercer sumador.
3. El carry de entrada (C_{in}) del primer sumador no va conectado a nada.

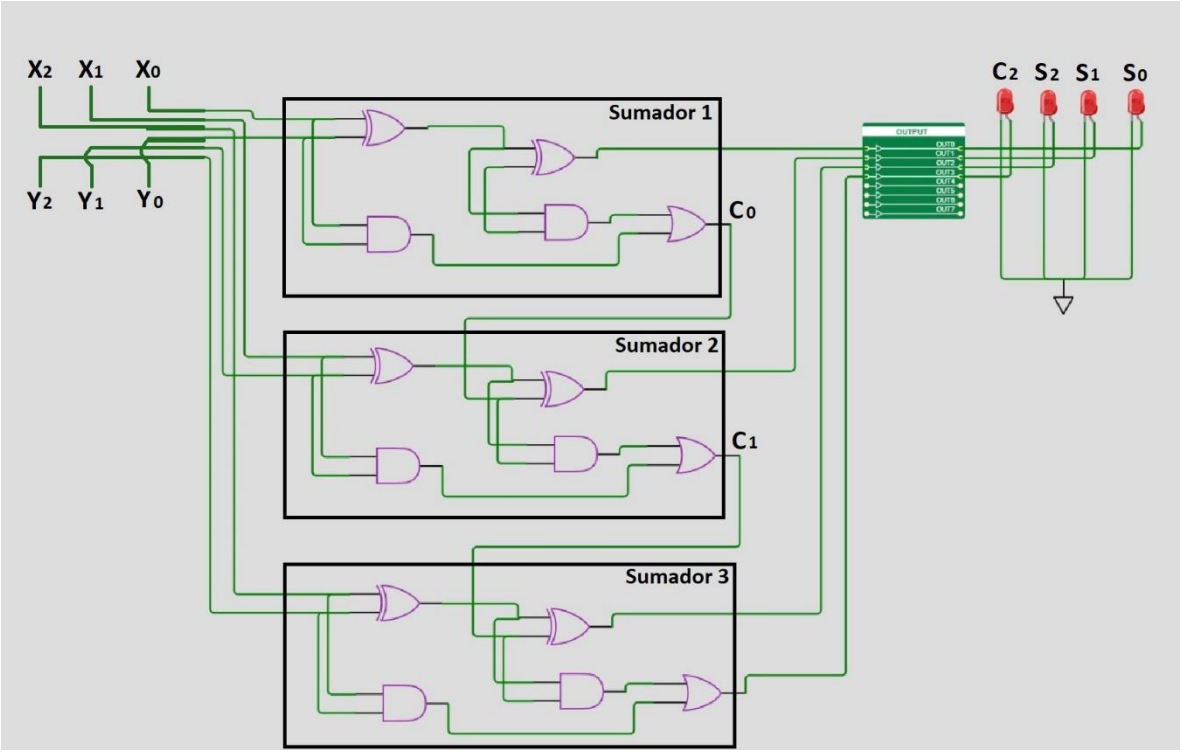
Por lo tanto, el Sumador Binario de 3 Bits quedaría de la siguiente manera:



En donde:

1. X_0, X_1 y X_2 representan el primer sumando, el cual X_0 es el bit de menor valor y X_2 es el bit de mayor valor.
2. Y_0, Y_1 y Y_2 representan el segundo sumando, el cual Y_0 es el bit de menor valor y Y_2 es el bit de mayor valor.
3. Las salidas S_0, S_1 y S_2 son los bits de salida que mostraran el resultado, en donde S_0 es el bit de menor valor.
4. El carry del último sumador (C_{out}) se puede considerar como una salida como si fuera un S_3, el cual será el bit de mayor valor.

Finalmente, el Sumador Binario de 3 Bits quedaría de la siguiente manera en la plataforma de WOKWI:



Restador

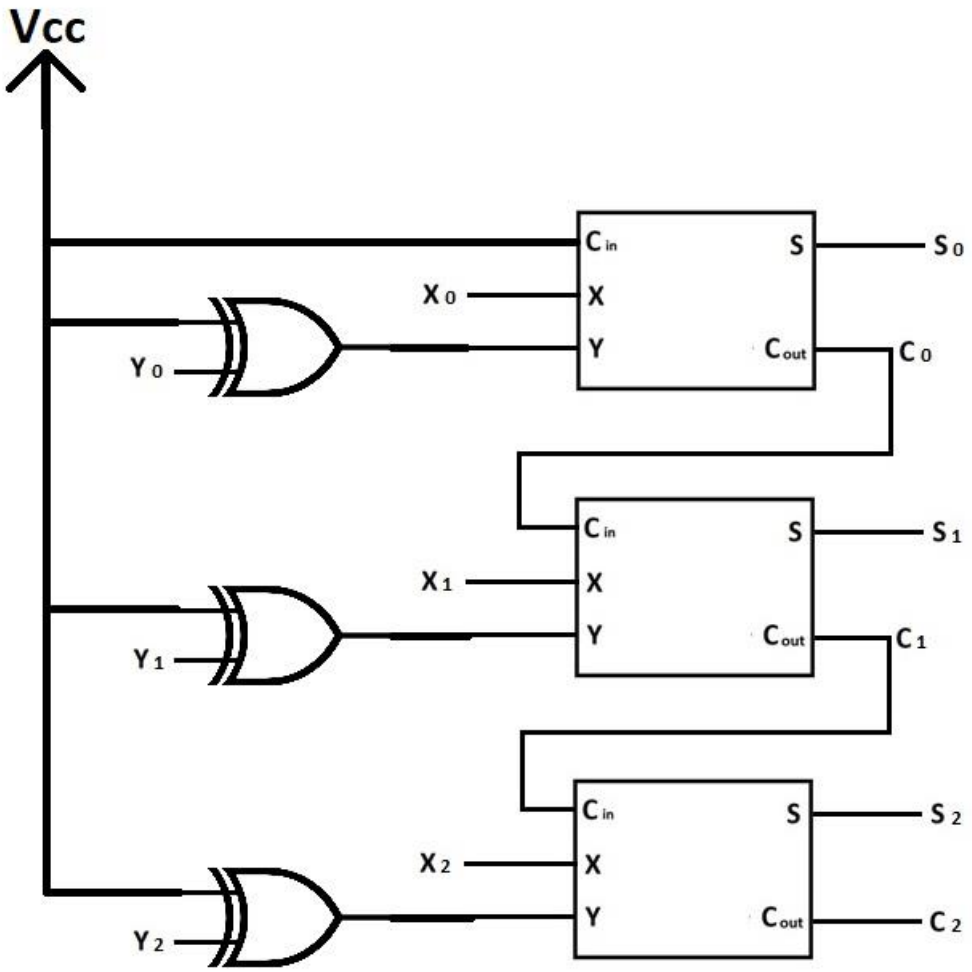
https://wokwi.com/projects/373085920015867905

Existe una forma de realizar un restador de una manera más sencilla a partir del circuito o diagrama de conexión de un sumador. Para este caso, se utilizará el Sumador Binario de 3 Bits para poder realizar un Restador Binario de 3 Bits. Las características del restador serán las siguientes:

- La resta se va a realizar entre dos valores de entrada.
- El minuendo (el número a restar) tendrá 3 bits de entrada.
- El sustraendo (el número que va a restar al minuendo) tendrá 3 bits de entrada.
- Los valores máximos del minuendo y sustraendo son de 7 en binario (1 1 1).
- Será posible obtener números negativos
- La salida va a tener en total 4 bits, de los cuales, el bit más significativo será el del signo.
- La diferencia máxima posible de resultado o salida es de -7 en binario (0 0 0 1 con complemento a 2).

Para obtener el restador a partir de un sumador, basta con agregar una compuerta lógica XOR a cada una de las entradas de los bits que conforman al sustraendo; una de las patas de las compuertas XOR serán los bits de entrada del sustraendo y las otras patas de las compuertas XOR deben ir conectadas al voltaje de alimentación Vcc; el carry de entrada (C_{in}) del primer sumador debe de ir conectado de igual manera al voltaje de alimentación Vcc.

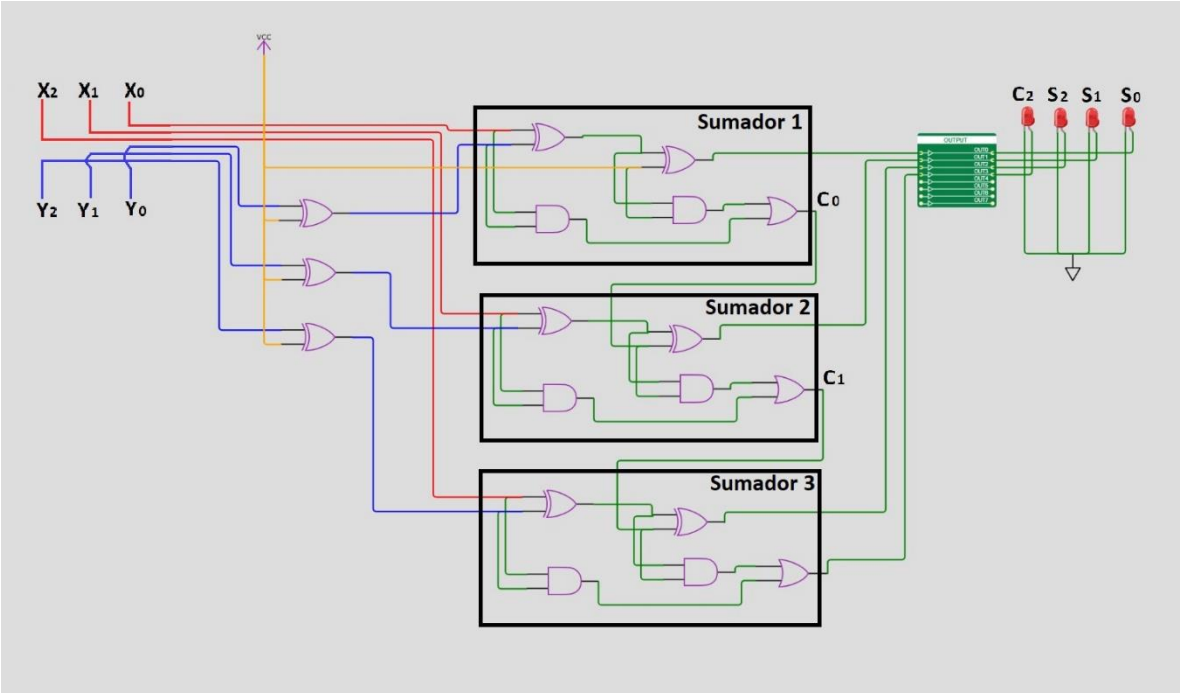
Por lo tanto, el Restador Binario de 3 Bits quedaría de la siguiente manera:



En donde:

1. X_0, X_1 y X_2 representan el minuendo, el cual X_0 es el bit de menor valor y X_2 es el bit de mayor valor.
2. Y_0, Y_1 y Y_2 representan el sustraendo, el cual Y_0 es el bit de menor valor y Y_2 es el bit de mayor valor.
3. Las salidas S_0, S_1 y S_2 son los bits de salida que mostraran el resultado, en donde S_0 es el bit de menor valor.
4. El carry del último sumador (C_{out}) se puede considerar como una salida como si fuera un S_3, el cual será el bit de signo.

Finalmente, el Restador Binario de 3 Bits quedaría de la siguiente manera en la plataforma de WOKWI:



Multiplicador

https://wokwi.com/projects/373066755436024833

Realizar una multiplicación entre dos bits en números binarios es realmente muy sencillo, ya que, como en una multiplicación normal, cada vez que se multiplica un número por cero el resulta seguirá siendo cero y en el caso de los bits solo tenemos dos números con los cuales trabajar, el uno y el cero.

Multiplicación de bits			
A	B	R	Mínima Expresión Booleana
0	0	0	R = A B
0	1	0	
1	0	0	
1	1	1	

La parte laboriosa empieza cuando los números a multiplicar tienen más de un bit de entrada, ya que, al igual que una multiplicación normal, después de multiplicar los dos valores numéricos, hay un momento en la multiplicación en donde se tienen que sumar los valores previamente multiplicados.

$$\begin{array}{r} 35 \\ \times\ 21 \\ \hline +\ 35 \\ +\ 70 \\ \hline 735 \end{array}$$

Multiplicación

Parte a sumar

Por lo tanto, para realizar el Multiplicador Binario de 3 Bits, se va a obtener su circuito de la misma manera que la multiplicación anterior; pero para eso, es necesario utilizar tanto el medio sumador y el sumador completo para realizar la parte de la suma de bits. Las características del multiplicador serán las siguientes:

- La multiplicación se va a realizar entre dos valores de entrada.
- El primer factor tendrá 3 bits de entrada.
- El segundo factor tendrá 3 bits de entrada.
- Los valores máximos de los factores son de 7 en binario (1 1 1).
- La salida va a tener en total 6 bits, ya que se deben de sumar los bits de los factores de entrada para poder obtener la cantidad necesaria de bits de salida y abarcar el número más grande de multiplicación.
- El valor máximo posible de resultado o salida es de 49 en binario (1 1 0 0 0 1).

Para realizar el Multiplicador Binario de 3 Bits, es un proceso más largo y laborioso que las operaciones anteriores, por lo tanto, se seguirán los siguientes pasos:

1. Se van a acomodar los dos números a multiplicar como en una multiplicación normal, de tal forma que los bits más significativos queden a la izquierda y los menos significativos a la derecha.

	B2	B1	B0
X	A2	A1	A0

2. Ya que se tienen acomodados los números a multiplicar, A0 va a multiplicar todos los números de la parte superior de derecha a izquierda, como en cualquier multiplicación.

	B2	B1	B0
X	A2	A1	A0
	A0*B2	A0*B1	A0 *B0

3. Se repite el mismo procedimiento, pero ahora A1 multiplicará a cada uno de los términos superiores y se recorrerán los resultados obtenidos una casilla a la izquierda.

		B2	B1	B0
	X	A2	A1	A0
		A0*B2	A0*B1	A0 *B0
A1*B2		A1*B1	A1 *B0	

4. Finalmente, A2 multiplicará cada uno de los términos superiores y los resultados obtenidos se recorrerán otra casilla más a la izquierda.

			B2	B1	B0
		X	A2	A1	A0
			A0*B2	A0*B1	A0 *B0
	A1*B2		A1*B1	A1 *B0	
A2*B2	A2*B1		A2 *B0		

5. Ya que se tienen todas las multiplicaciones, se proceden a sumar los términos en sus columnas correspondientes empezando de derecha a izquierda. Como se puede observar, **A0*B0** no tiene nadie con quien sumarse, por lo tanto, el resultado de esa multiplicación se pasa directo como el **resultado a mostrar**, el cual se denominará **S0**.

			B2	B1	B0
		X	A2	A1	A0
			A0*B2	A0*B1	A0 *B0
	A1*B2		A1*B1	A1 *B0	
A2*B2	A2*B1		A2 *B0		

(A0*B0) = S0

6. Las siguientes multiplicaciones por sumar son **A0*B1** con **A1*B0** y, algo que hay que dejar muy en claro es que, **cada vez que se cada vez que sumen dos términos, se va a utilizar un medio sumador**, y también debemos de recordar que **los medios sumadores solo pueden generar una salida "S" o un carry "C" nunca los dos al mismo tiempo**, pero esto se comprenderá mejor más adelante. Por lo tanto, la suma entre **A0*B1** con **A1*B0** genera una salida **S1** o un carry **C0**, quedando de la siguiente manera.

			B2	B1	B0
		X	A2	A1	A0
			A0*B2	A0*B1	A0 *B0
		A1*B2	A1*B1	A1 *B0	
	A2*B2	A2*B1	A2 *B0		
				(A0*B1 + A1*B0) = S1 ó C0	(A0*B0) = S0

7. Supongamos que de la suma anterior se genera el carry **C0**, eso quiere decir que se va a sumar con el siguiente conjunto de términos, los cuales son **A0*B2**, **A1*B1** y **A2*B0**. Lo primero que se tiene que hacer es sumar el carry **C0** con los primeros dos términos mencionados anteriormente, entonces **se sumarían C0 con A0*B2 y con A1*B1**, en esta ocasión se sumaran tres términos a la vez. También, algo que dejar muy en claro es que, **cada vez que se cada vez que sumen tres términos, se va a utilizar un sumador completo,** y también debemos de recordar que **los sumadores completos pueden generar tanto una salida "S" y un carry "C" al mismo tiempo**. Por lo tanto, la suma entre **C0 con A0*B2 y con A1*B1**, genera una salida **R0** o un carry **C1**. En esta ocasión la salida se colocó como **R0** (aunque puede tener el nombre que deseé el usuario) en lugar de **S2** ya que todavía queda un término por sumar en este conjunto de términos (**A2*B0**). La suma quedaría de la siguiente manera.

			B2	B1	B0
		X	A2	A1	A0
			A0*B2	A0*B1	A0 *B0
		A1*B2	A1*B1	A1 *B0	
	A2*B2	A2*B1	A2 *B0		
			(C0 + A0*B2 + A1*B1) = C1 y R0	(A0*B1 + A1*B0) = S1 ó C0	(A0*B0) = S0

8. Como se puede observar, de la suma anterior se generaron una salida **R0** y un carry **C1**. **Por el momento se guardará y no se utilizará el carry C1 ya que, como su nombre lo indica (carry o acarreo), pasa a sumarse con el siguiente grupo de términos (A1*B2 y A2*B1)**, entonces, la salida **R0** esta libre para utilizarse. La salida **R0** se sumará con **A2*B0**, por lo tanto, se utilizará un medio sumador y las salidas que genera esta suma son la salida **S2** ó el carry **C2**, quedando de la siguiente manera:

			B2	B1	B0
		X	A2	A1	A0
			A0*B2	A0*B1	A0 *B0
		A1*B2	A1*B1	A1 *B0	
	A2*B2	A2*B1	A2 *B0		
			(C0 + A0*B2 + A1*B1) = C1 y R0	(A0*B1 + A1*B0) = S1 ó C0	(A0*B0) = S0
			(R0 + A2*B0) = S2 ó C2		

9. Como se puede observar, ahora se tienen dos carrys (**C1** y **C2**) y ambos pasan a sumarse con el siguiente grupo de términos (**A1*B2 y A2*B1**), por lo tanto, se empezarán a sumar 3 términos en orden como se fueron generando. Entonces, se van a sumar **C1** con **C2** y con **A1*B2** y la suma de estos tres términos generan una salida **R1** y un carry **C3**.

			B2	B1	B0
		X	A2	A1	A0
			A0*B2	A0*B1	A0 *B0
		A1*B2	A1*B1	A1 *B0	
	A2*B2	A2*B1	A2 *B0		
		(C1 + C2 + A1*B2) = C3 y R1	(C0 + A0*B2 + A1*B1) = C1 y R0	(A0*B1 + A1*B0) = S1 ó C0	(A0*B0) = S0
			(R0 + A2*B0) = S2 ó C2		

10. Nuevamente, el carry **C3** se guarda para el siguiente grupo de términos (**A2*B2**) y la salida **R1** se va a utilizar en el grupo de término actual. Por lo tanto, se van a sumar la salida **R1** con **A2*B1** y la suma de estos dos términos da como resultado una salida **S3** o un carry **C4**.

			B2	B1	B0
		X	A2	A1	A0
			A0*B2	A0*B1	A0 *B0
		A1*B2	A1*B1	A1 *B0	
	A2*B2	A2*B1	A2 *B0		
		(C1 + C2 + A1*B2) = C3 y R1	(C0 + A0*B2 + A1*B1) = C1 y R0	(A0*B1 + A1*B0) = S1 ó C0	(A0*B0) = S0
		(R1 + A2*B1) = S3 ó C4	(R0 + A2*B0) = S2 ó C2		

11. Por segunda ocasión, se tienen dos carrys (**C3** y **C4**) y ambos pasan a sumarse con el último grupo de términos (**A2*B2**). Por lo tanto, la suma de **C3** con **C4** y **A2*B2** dan como resultado una salida **S4** y un carry **C5**. El carry **C5** pasaría al siguiente grupo de términos, pero como ya no queda nada con que sumar, pasa directamente como una salida, quedando de la siguiente manera:

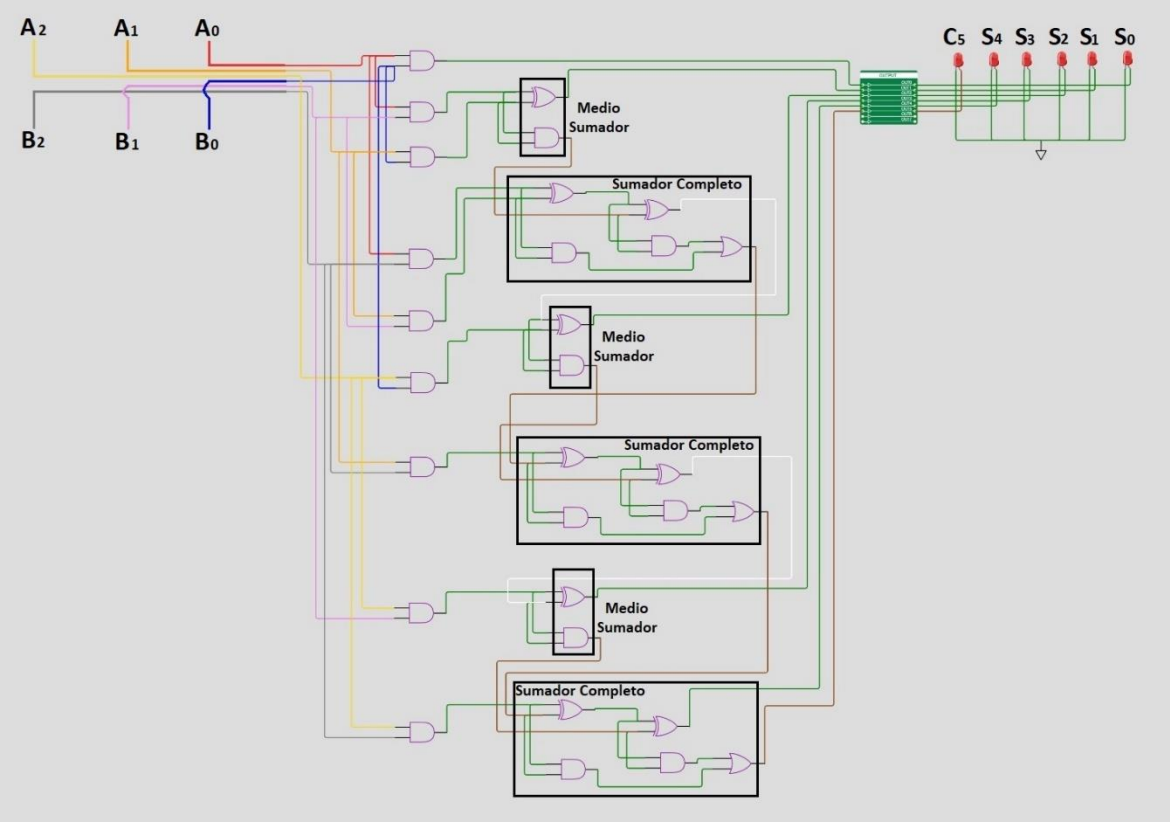
			B2	B1	B0
			X A2	A1	A0
			A0*B2	A0*B1	A0 *B0
		A1*B2	A1*B1	A1 *B0	
	A2*B2	A2*B1	A2 *B0		
C5	(C3 + C4 + A2*B2) = S4 y C5	(C1 + C2 + A1*B2) = C3 y R1	(C0 + A0*B2 + A1*B1) = C1 y R0	(A0*B1 + A1*B0) = S1 ó C0	(A0*B0) = S0
		(R1 + A2*B1) = S3 ó C4	(R0 + A2*B0) = S2 ó C2		

12. Finalmente, el Multiplicador Binario de 3 Bits quedaría de la siguiente manera, en donde:

- A_0, A_1 y A_2 representan el primer término a multiplicar, el cual A_0 es el bit de menor valor y A_2 es el bit de mayor valor.
- B_0, B_1 y B_2 representan el segundo término a multiplicar, el cual B_0 es el bit de menor valor y B_2 es el bit de mayor valor.
- Los términos marcados de color **MORADO** son las multiplicaciones previas antes de pasar a los sumadores. Para eso, se utilizan compuertas AND.
- Los términos marcados de color **ROJO** son las salidas del Multiplicador. **S0** es el bit de menor valor y **C5** el bit de mayor valor.
- Las sumatorias marcadas de color **AZUL** son los términos que requieren un sumador completo.
- Las sumatorias marcadas de color **VERDE** son los términos que requieren un medio sumador.

			B2	B1	B0
			X A2	A1	A0
			A0*B2	A0*B1	A0 *B0
		A1*B2	A1*B1	A1 *B0	
	A2*B2	A2*B1	A2 *B0		
C5	(C3 + C4 + A2*B2) = S4 y C5	(C1 + C2 + A1*B2) = C3 y R1	(C0 + A0*B2 + A1*B1) = C1 y R0	(A0*B1 + A1*B0) = S1 ó C0	(A0*B0) = S0
		(R1 + A2*B1) = S3 ó C4	(R0 + A2*B0) = S2 ó C2		
C5	S4	S3	S2	S1	S0

El Multiplicador Binario de 3 Bits quedaría de la siguiente manera en la plataforma de WOKWI:



Contador

https://wokwi.com/projects/373174994558429185

Para este caso, se van a realizar dos contadores, los cuales van a estar trabajando de manera diferente con una misma señal de reloj.

Contador Síncrono de 4 bits

El primer contador que se realizará a continuación consiste en un contador síncrono 4 bits, esto quiere decir que va a necesitar 4 Flip-Flops de tipo D (se puede hacer con otro tipo de Flip-Flop) y será síncrono porque todos estos Flip-Flops utilizan una misma señal de reloj para poder funcionar. Las características de este primer contador serán las siguientes:

- Utilizará 4 Flip-Flops tipo D.
- Tendrá una señal de reloj automática de 1 de frecuencia o en su defecto, puede utilizar un botón como señal de reloj.
- Para activarse, tendrá un switch que permita el paso de la señal de reloj.
- Será capaz de contar de cero hasta 15 en binario (1 1 1 1) y de ahí volverá a contar desde cero.

Para este caso, no será necesario utilizar la tabla de verdad del Flip-Flop de tipo D, pero lo que sí se va a requerir es su **Tabla de Transición** o también llamada **Tabla de Excitación**, de la cual se obtendrán los estados cambiantes del Flip-Flop con forme pasa un periodo de tiempo y de ahí realizar una tabla de verdad en la que se muestre como van a ir cambiando los bits de salida para obtener los números binarios en orden.

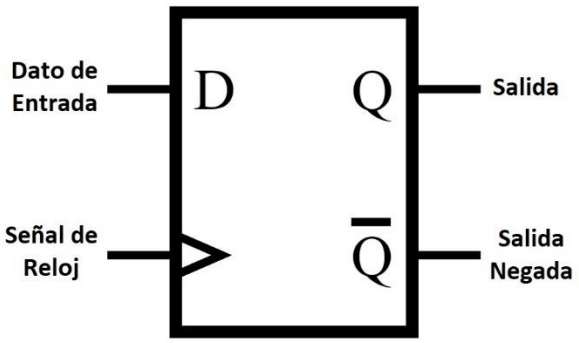


Tabla de Excitación del Flip-Flop D		
Q(t)	Q(t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

1. Como primer paso para realizar la tabla de verdad, se empezará haciendo una tabla donde se muestren todos los números posibles que se pueden crear con esos 4 bits de salida de los Flip-Flops, la cual se denominará **Estado Actual Q(t)** ya que, como su nombre lo indica, son los bits de salida previo a un cambio con forme al tiempo. A un lado se van a colocar las salidas futuras de los Flip-Flops denominada como **Estado Futuro Q(t+1)** y también se van a agregar la entrada de datos de cada uno de los Flip-Flops. Para identificar los Flip-Flops junto con sus entradas y salidas, se llamarán de la siguiente manera:

- Flip-Flop A: Entrada de datos: D_A; Salida de datos: Q_A
- Flip-Flop B: Entrada de datos: D_B; Salida de datos: Q_B
- Flip-Flop C: Entrada de datos: D_C; Salida de datos: Q_C
- Flip-Flop D: Entrada de datos: D_D; Salida de datos: Q_D

Por lo tanto, la tabla queda de la siguiente manera:

Contador Síncrono 4 Bits con Flip-Flop D											
Estado Actual Q(t)				Estado Futuro Q(t+1)				Flip-Flop A	Flip-Flop B	Flip-Flop C	Flip-Flop D
Q_A	Q_B	Q_C	Q_D	Q_A	Q_B	Q_C	Q_D	D_A	D_B	D_C	D_D
0	0	0	0								
0	0	0	1								
0	0	1	0								
0	0	1	1								

0	1	0	0								
0	1	0	1								
0	1	1	0								
0	1	1	1								
1	0	0	0								
1	0	0	1								
1	0	1	0								
1	0	1	1								
1	1	0	0								
1	1	0	1								
1	1	1	0								
1	1	1	1								

2. Para llenar la parte de **Estado Futuro Q(t+1)**, es necesario fijarse en los bits del **Estado Actual Q(t)** y colocar el número que se desea que le suceda, por ejemplo: Se tiene el número cero (0 0 0 0) en la parte de **Estado Actual Q(t)** y se desea que cambie a uno (0 0 0 1), por lo tanto, en la parte de **Estado Futuro Q(t+1)** se colocarán los bits del número uno (0 0 0 1). Si se desea que el número once (1 0 1 1) cambié a doce, en el **Estado Futuro Q(t+1)** se deben de colocar los bits de ese número (1 1 0 0). Para este caso todos los números son consecutivos, entonces, la tabla queda de la siguiente manera:

Contador Síncrono 4 Bits con Flip-Flop D											
Estado Actual Q(t)				Estado Futuro Q(t+1)				Flip-Flop A	Flip-Flop B	Flip-Flop C	Flip-Flop D
Q_A	Q_B	Q_C	Q_D	Q_A	Q_B	Q_C	Q_D	D_A	D_B	D_C	D_D
0	0	0	0	0	0	0	1				
0	0	0	1	0	0	1	0				
0	0	1	0	0	0	1	1				
0	0	1	1	0	1	0	0				
0	1	0	0	0	1	0	1				
0	1	0	1	0	1	1	0				
0	1	1	0	0	1	1	1				
0	1	1	1	1	0	0	0				
1	0	0	0	1	0	0	1				
1	0	0	1	1	0	1	0				
1	0	1	0	1	0	1	1				
1	0	1	1	1	1	0	0				
1	1	0	0	1	1	0	1				
1	1	0	1	1	1	1	0				
1	1	1	0	1	1	1	1				
1	1	1	1	0	0	0	0				

3. Para llenar las entradas de datos "D" de los Flip-Flops es necesario utilizar la **Tabla de Excitación** para observar como van cambiando las salidas Q del **Estado Actual Q(t)** al **Estado Futuro Q(t+1)**, por ejemplo, para llenar la primera casilla del Flip-Flop D, se tiene que observar cómo cambia Q_D del **Estado Actual Q(t)** al **Estado Futuro Q(t+1)**. Para este primer ejemplo, se puede observar que en el **Estado Actual Q(t)** esta en 0 y en el **Estado Futuro Q(t+1)** pasa a 1, por lo tanto, la entrada de datos "D" es 0 como indica la **Tabla de Excitación**.

Tabla de Excitación del Flip-Flop D		
Q(t)	Q(t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

Contador Síncrono 4 Bits con Flip-Flop D											
Estado Actual Q(t)				Estado Futuro Q(t+1)				Flip-Flop A	Flip-Flop B	Flip-Flop C	**Flip-Flop D**
Q_A	Q_B	Q_C	**Q_D**	Q_A	Q_B	Q_C	**Q_D**	D_A	D_B	D_C	**D_D**
			0				**1**				**0**

4. Siguiendo este método, se llenan todas las casillas de entrada de datos "D" del Flip-Flop D.

Contador Síncrono 4 Bits con Flip-Flop D											
Estado Actual Q(t)				Estado Futuro Q(t+1)				Flip-Flop A	Flip-Flop B	Flip-Flop C	**Flip-Flop D**
Q_A	Q_B	Q_C	Q_D	Q_A	Q_B	Q_C	Q_D	D_A	D_B	D_C	D_D
			0				**1**				**1**
			1				**0**				**0**
			0				**1**				**1**
			1				**0**				**0**
			0				**1**				**1**
			1				**0**				**0**
			0				**1**				**1**
			1				**0**				**0**
			0				**1**				**1**
			1				**0**				**0**
			0				**1**				**1**
			1				**0**				**0**
			0				**1**				**1**
			1				**0**				**0**
			0				**1**				**1**
			1				**0**				**0**

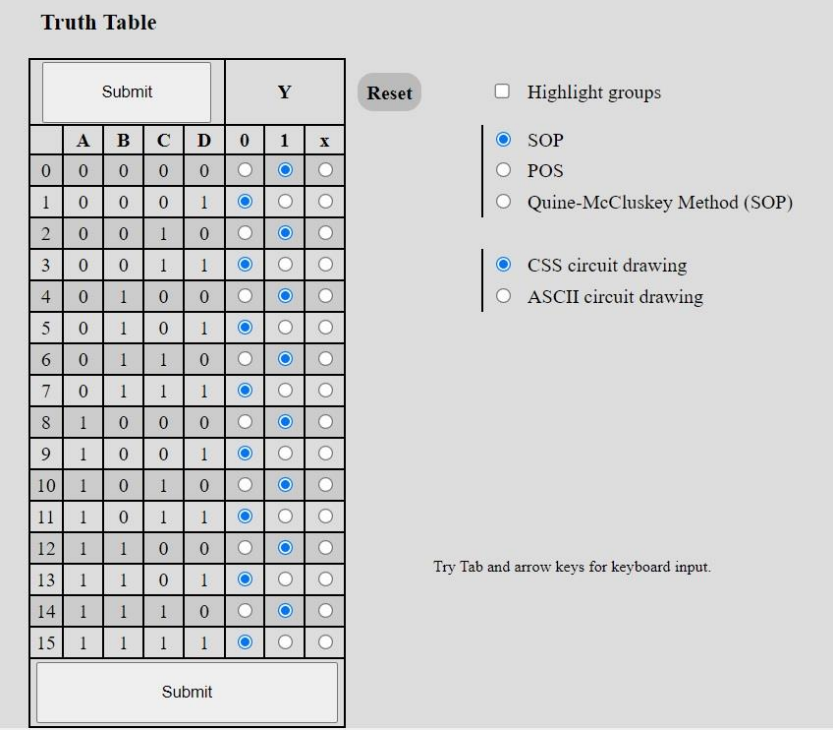
5. Se repite este mismo procedimiento con las otras entradas de datos de los Flip-Flops faltantes. La tabla de verdad queda de la siguiente manera:

Contador Síncrono 4 Bits con Flip-Flop D											
Estado Actual Q(t)				Estado Futuro Q(t+1)				Flip-Flop A	Flip-Flop B	Flip-Flop C	Flip-Flop D
Q_A	Q_B	Q_C	Q_D	Q_A	Q_B	Q_C	Q_D	D_A	D_B	D_C	D_D
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	1	0	0	1	1
0	0	1	1	0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	1	0	1	0	1
0	1	0	1	0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	1	0	1	1	1
0	1	1	1	1	0	0	0	1	0	0	0
1	0	0	0	1	0	0	1	1	0	0	1
1	0	0	1	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	1	1	0	1	1
1	0	1	1	1	1	0	0	1	1	0	0
1	1	0	0	1	1	0	1	1	1	0	1
1	1	0	1	1	1	1	0	1	1	1	0
1	1	1	0	1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0	0	0	0	0

6. El siguiente paso es obtener una expresión Booleana de cada entrada de datos “D”, para realizar esto, se puede utilizar Algebra Booleana o mapas de Karnaugh. Para este caso, se realizaron mapas de Karnaugh con la herramienta en línea de The Quine-McCluskey Solver. Para obtener la expresión Booleana de la entrada de datos “D_D” del Flip-Flop D se utiliza todas las salidas Q del **Estado Actual Q(t)**.

Contador Síncrono 4 Bits con Flip-Flop D											
Estado Actual Q(t)				Estado Futuro Q(t+1)				Flip-Flop A	Flip-Flop B	Flip-Flop C	Flip-Flop D
Q_A	Q_B	Q_C	Q_D	Q_A	Q_B	Q_C	Q_D	D_A	D_B	D_C	D_D
0	**0**	**0**	**0**								**1**
0	**0**	**0**	**1**								**0**
0	**0**	**1**	**0**								**1**
0	**0**	**1**	**1**								**0**
0	**1**	**0**	**0**								**1**
0	**1**	**0**	**1**								**0**
0	**1**	**1**	**0**								**1**

0	1	1	1								0
1	0	0	0								1
1	0	0	1								0
1	0	1	0								1
1	0	1	1								0
1	1	0	0								1
1	1	0	1								0
1	1	1	0								1
1	1	1	1								0



7. Utilizando este método, la mínima expresión Booleana de la entrada de datos "D_D" es: **$D_D = Q_D'$**.

8. Por lo tanto, las mínimas expresiones Booleanas de cada una de las entradas de datos son las siguientes:

Mínima Expresión Booleana
D (Data)
$D_A = Q_A Q_B' + Q_A Q_C' + Q_A Q_D' + Q_A' Q_B Q_C Q_D$
$D_B = Q_B Q_C' + Q_B Q_D' + Q_B' Q_C Q_D$
$D_C = Q_C' Q_D + Q_C Q_D'$
$D_D = Q_D'$

Contador Síncrono de 2 bits

El segundo contador que se realizará a consiste en un contador síncrono 2 bits, esto quiere decir que va a necesitar 2 Flip-Flops de tipo D (se puede hacer con otro tipo de Flip-Flop) y será síncrono porque todos estos Flip-Flops utilizan una misma señal de reloj para poder funcionar. Las características de este primer contador serán las siguientes:

- Utilizará 2 Flip-Flops tipo D.
- Tendrá una señal de reloj automática de 1 de frecuencia o en su defecto, puede utilizar un botón como señal de reloj.
- Para activarse, tendrá un switch que permita el paso de la señal de reloj.
- Aunque es capaz de contar hasta 3 en binario (1 1), solamente mostrará dos valores, el uno (0 1) y el dos (1 0), simulando como si se apagaran y se encendieran dos leds de manera intercalada.

Al igual que el contador anterior, se utilizará la **Tabla de Excitación** del Flip-Flop tipo D, de la cual se obtendrán los estados cambiantes del Flip-Flop con forme pasa un periodo de tiempo y de ahí realizar una tabla de verdad en la que se muestre como van a ir cambiando los bits de salida para obtener los números binarios.

1. Como primer paso para realizar la tabla de verdad, se empezará haciendo una tabla donde se muestren todos los números posibles que se pueden crear con esos 2 bits de salida de los Flip-Flops, la cual se denominará **Estado Actual Q(t)**. A un lado se van a colocar las salidas futuras de los Flip-Flops denominada como **Estado Futuro Q(t+1)** y también se van a agregar la entrada de datos de cada uno de los Flip-Flops. Para identificar los Flip-Flops junto con sus entradas y salidas, se llamarán de la siguiente manera:

- Flip-Flop X: Entrada de datos: D_X; Salida de datos: Q_X
- Flip-Flop Y: Entrada de datos: D_Y; Salida de datos: Q_Y

Por lo tanto, la tabla queda de la siguiente manera:

Contador Síncrono 2 Bits con Flip-Flop JK					
Estado Actual Q(t)		Estado Futuro Q(t+1)		Flip-Flop X	Flip-Flop Y
Q_X	Q_Y	Q_X	Q_Y	D_X	D_Y
0	0				
0	1				
1	0				
1	1				

2. Para llenar la parte de **Estado Futuro Q(t+1)**, es necesario fijarse en los bits del **Estado Actual Q(t)** y colocar el número que se desea que le suceda. Para este caso, se tiene el número uno (0 1) en la parte de **Estado Actual Q(t)** y se desea que cambie a dos (1 0), por lo tanto, en la parte de **Estado Futuro Q(t+1)** se colocarán los bits del número dos (1 0). Se tiene el número dos (1 0) en la parte de **Estado Actual Q(t)**, y para que cambie a uno (0 1) se deben de colocar los bits de salida en la parte de **Estado Futuro Q(t+1)**. Con estos dos cambios en la transición,

se dará el efecto de que los leds parpadean de forma intercalada. La tabla queda de la siguiente manera:

Contador Síncrono 2 Bits con Flip-Flop JK					
Estado Actual Q(t)		Estado Futuro Q(t+1)		Flip-Flop X	Flip-Flop Y
Q_X	Q_Y	Q_X	Q_Y	D_X	D_Y
0	0				
0	1	1	0		
1	0	0	1		
1	1				

3. Como se puede apreciar, hay casillas vacías en la parte de **Estado Futuro Q(t+1)**. Esas partes se van a llenar solamente con números ceros, de esta manera estos estados de transición no se ejecutarán en el resultado final.

Contador Síncrono 2 Bits con Flip-Flop JK					
Estado Actual Q(t)		Estado Futuro Q(t+1)		Flip-Flop X	Flip-Flop Y
Q_X	Q_Y	Q_X	Q_Y	D_X	D_Y
0	0	0	0		
0	1	1	0		
1	0	0	1		
1	1	0	0		

4. Al igual que el contador anterior, las entradas de datos “D” se van a llenar con la **Tabla de Excitación** del Flip-Flop tipo D, por lo tanto, la tabla de verdad resultante es la siguiente:

Tabla de Excitación del Flip-Flop D		
Q(t)	Q(t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

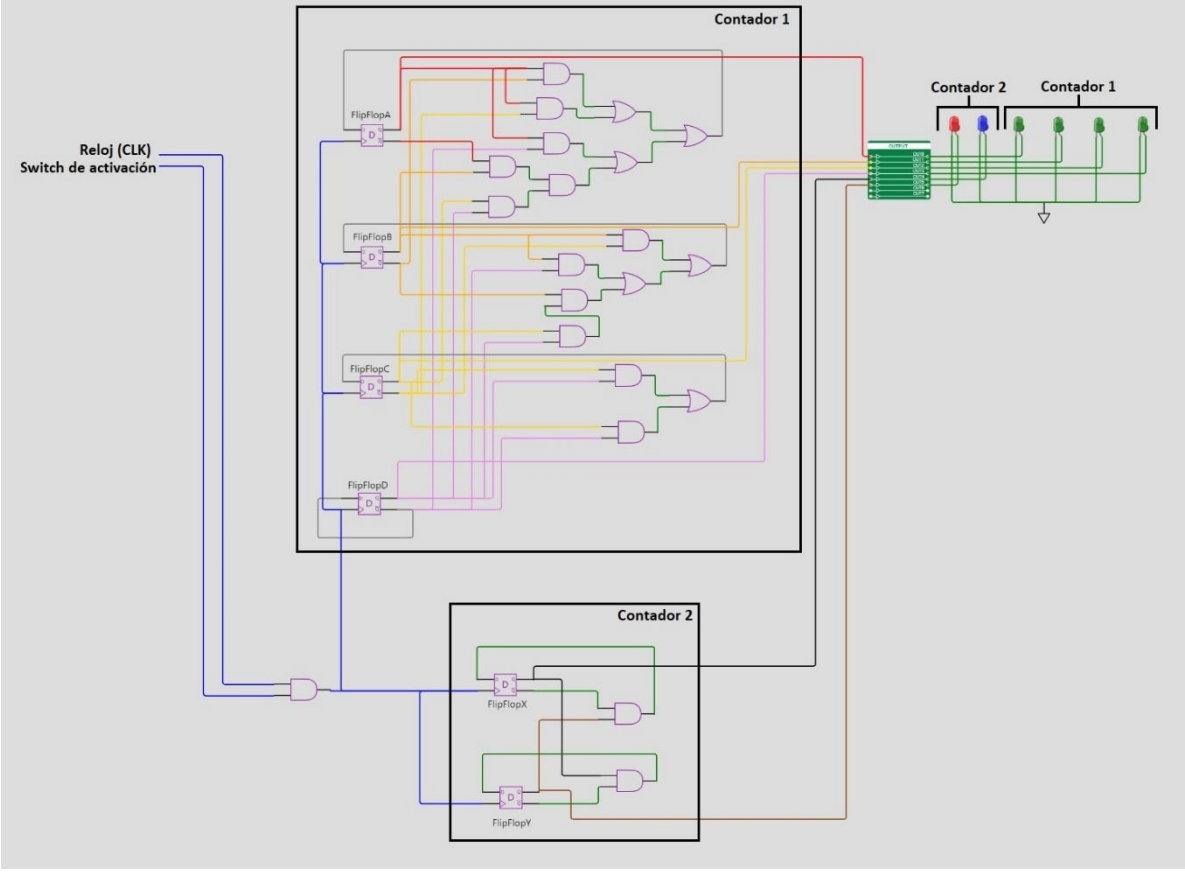
Contador Síncrono 2 Bits con Flip-Flop JK					
Estado Actual Q(t)		Estado Futuro Q(t+1)		Flip-Flop X	Flip-Flop Y
Q_X	Q_Y	Q_X	Q_Y	D_X	D_Y
0	0	0	0	0	0
0	1	1	0	1	0
1	0	0	1	0	1
1	1	0	0	0	0

5. El siguiente paso es obtener una expresión Booleana de cada entrada de datos “D”, para realizar esto, se puede utilizar Algebra Booleana o mapas de Karnaugh. Para este caso, se realizaron mapas de Karnaugh con la herramienta en línea de The Quine-McCluskey Solver. Se les recuerda que para obtener las expresiones Booleanas de las entradas de datos “D” se utiliza todas las salidas Q del **Estado Actual Q(t)**.

Mínima Expresión Booleana
D (Data)
$D_X = Q_X' Q_Y$
$D_Y = Q_X Q_Y'$

Diagrama de conexión de los contadores

Cabe recalcar que la salida Q del Flip-Flop A es el bit más significativo del contador, mientras que salida Q del Flip-Flop D es el bit menos significativo. También se puede apreciar que tanto la señal de Reloj como el Switch de activación están conectados a una misma compuerta AND, esto para que los contadores se activen cuando se encienda el Switch.



Multiplexor

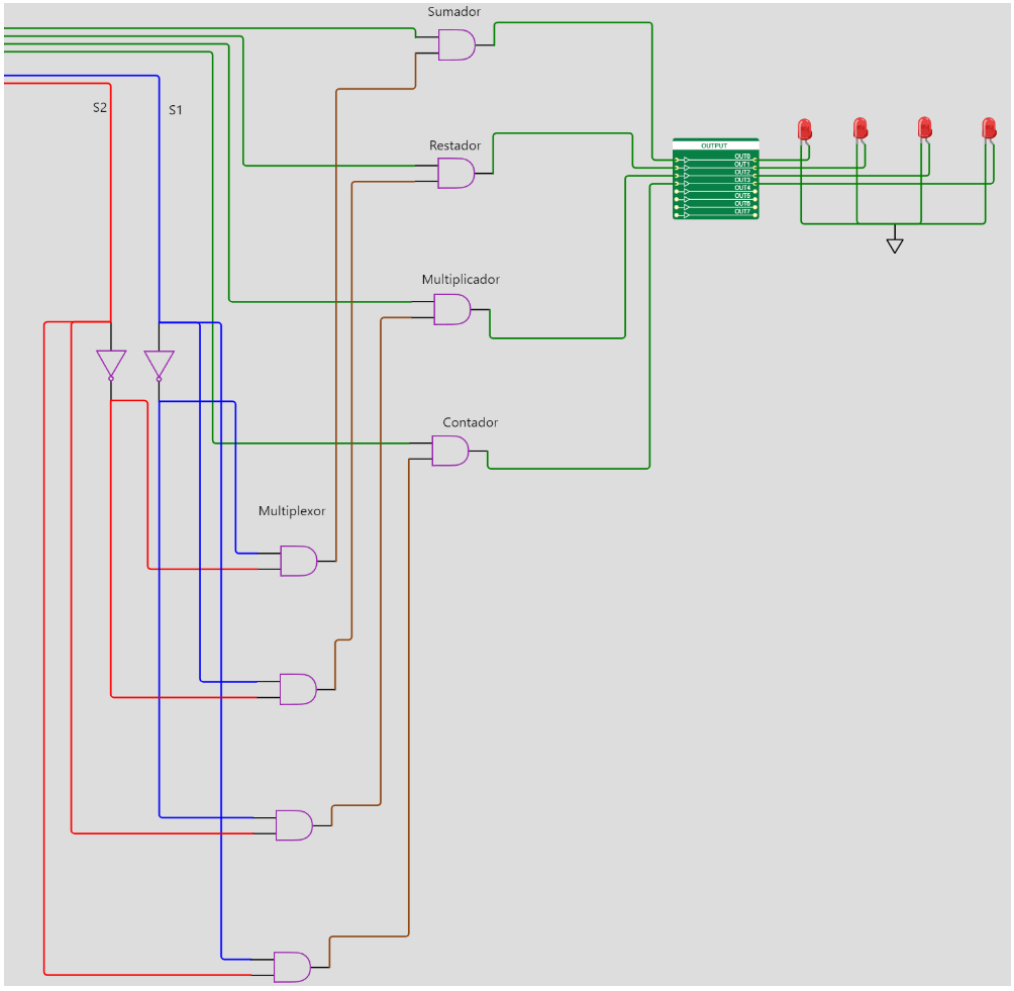
https://wokwi.com/projects/373078293675098113

Como último dispositivo por ver es el Multiplexor 4 a 1, ya que este mismo permitirá al usuario cambiar entre las diferentes opciones de la Calculadora Binaria de 3 bits cada vez que se introduzca una combinación de bits en las entradas del Multiplexor.

La tabla de verdad del Multiplexor 4 a 1 es la siguiente:

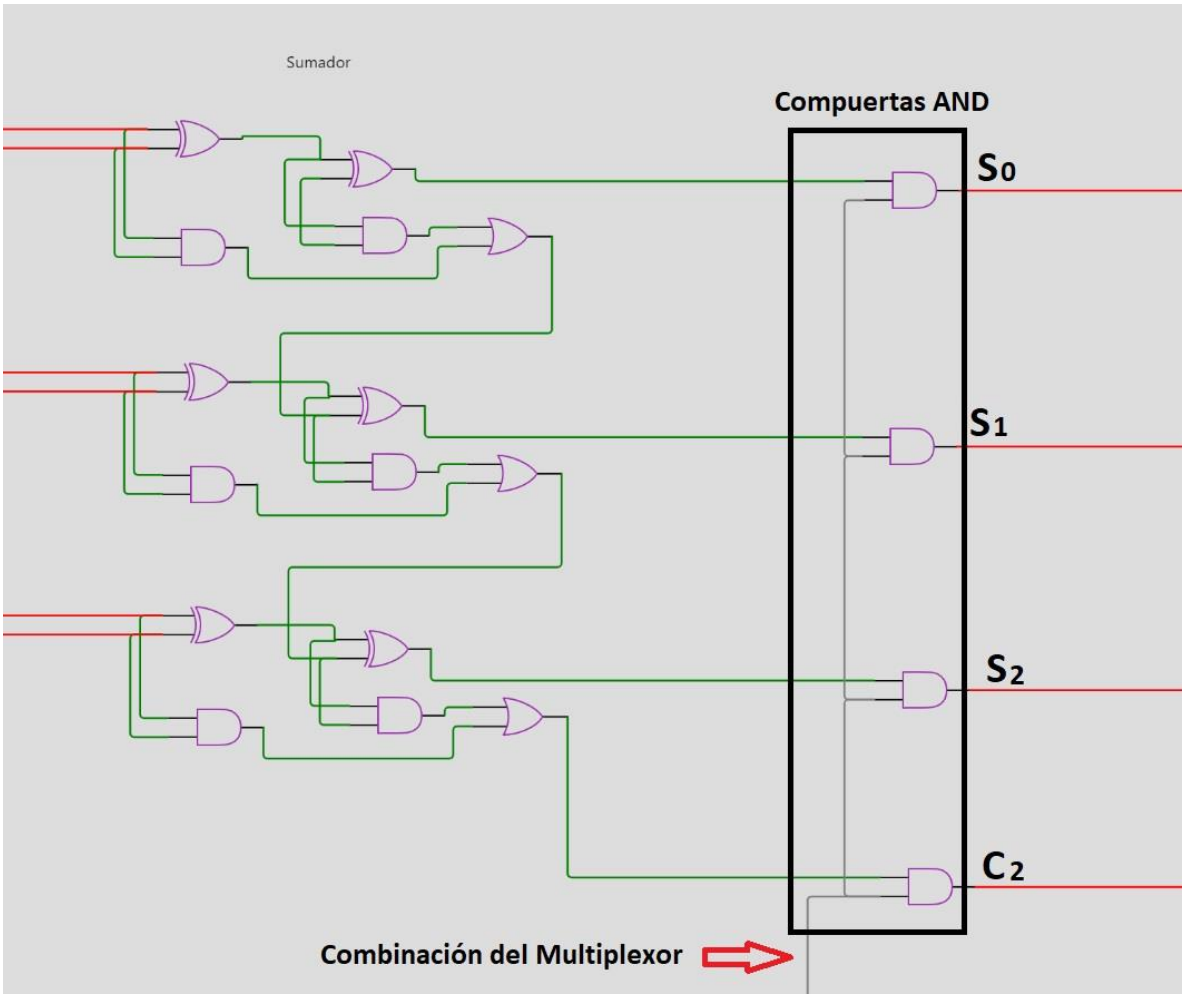
Multiplexor 4 a 1						
Sumador	Restador	Multiplicador	Contador	S2	S1	Y
1	0	0	0	0	0	Sumador
0	1	0	0	0	1	Restador
0	0	1	0	1	0	Multiplicador
0	0	0	1	1	1	Contador

Por lo tanto, el diagrama de conexión es el siguiente:



Para usarse de la manera correcta en cada una de las operaciones de la calculadora y poder cambiar entre opciones, se debe de agregar una compuerta AND en cada una de las

salidas de cada operación e ir conectada a su correspondiente salida del Multiplexor. Por ejemplo, para que la función de Suma de la calculadora funcione, cada una de sus salidas "S" debe de estar conectada a una compuerta AND (a una de sus entradas), la otra entrada de la compuerta AND debe de estar conectada a su combinación correspondiente del Multiplexor.



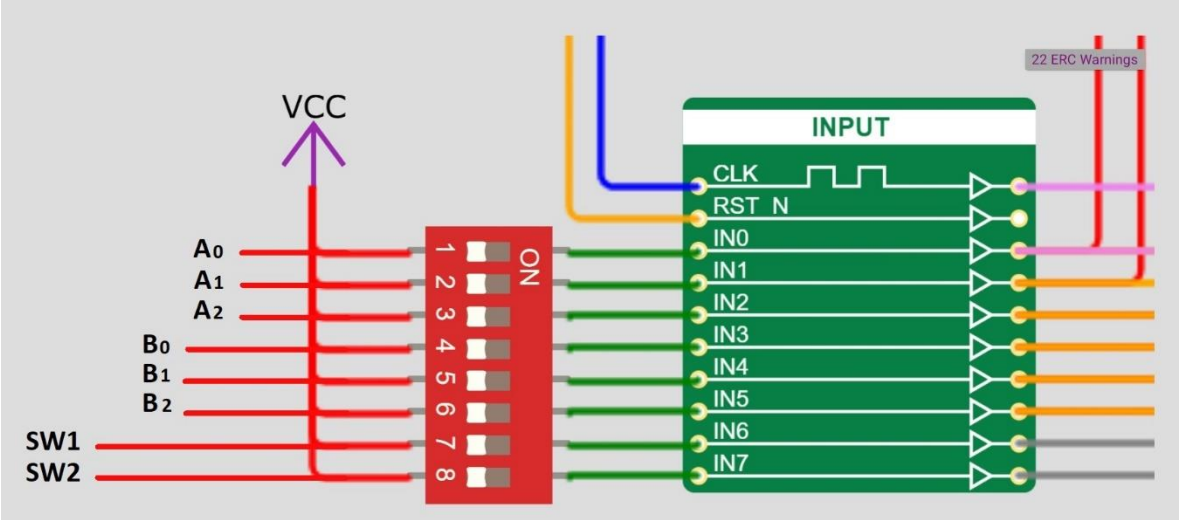
Como se puede apreciar, ahora las salidas de las compuertas AND toman el lugar de las salidas "S" del sumador. Se repite este mismo procedimiento con cada una de las operaciones y se tiene la calculadora armada.

Uso del Circuito y Ejemplos

Entrada de datos

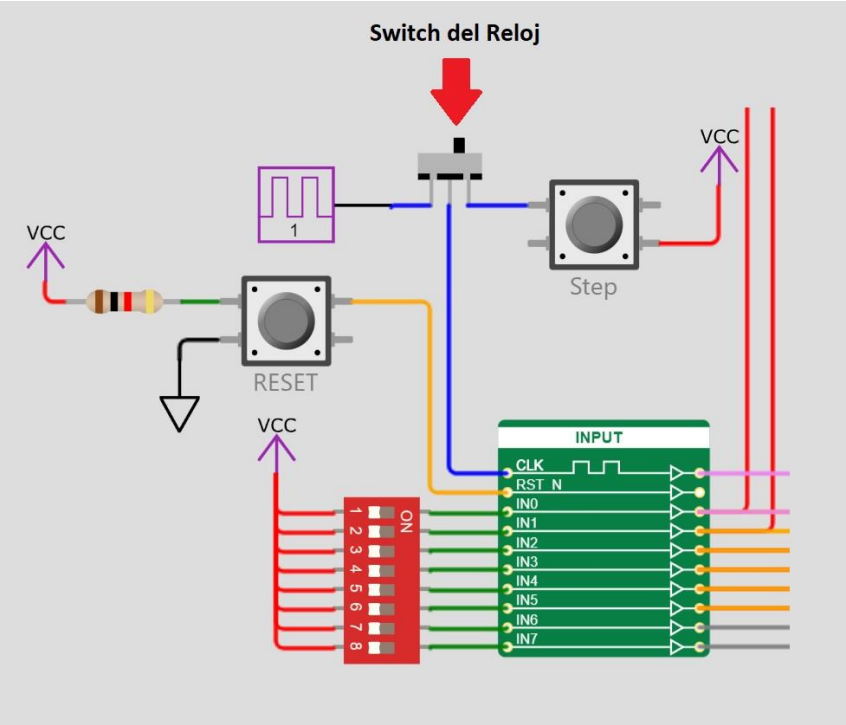
Para poder ingresar los bits de entrada de la calculadora se utilizará un Dip Switch con 8 entradas de datos.

- Las primeras 3 entradas corresponden al primer número “A” conformado por A_0, A_1 y A_2, el cual A_0 es el bit de menor valor y A_2 es el bit de mayor valor.
- Las siguientes 3 entradas corresponden al segundo número “B” conformado por B_0, B_1 y B_2, el cual B_0 es el bit de menor valor y B_2 es el bit de mayor valor.
- Las últimas 2 entradas corresponden al Multiplexor, el cual permitirá cambiar entre las 4 distintas opciones de la calculadora, donde SW1 es el bit de menor valor y SW2 el de mayor valor.



Señal de Reloj

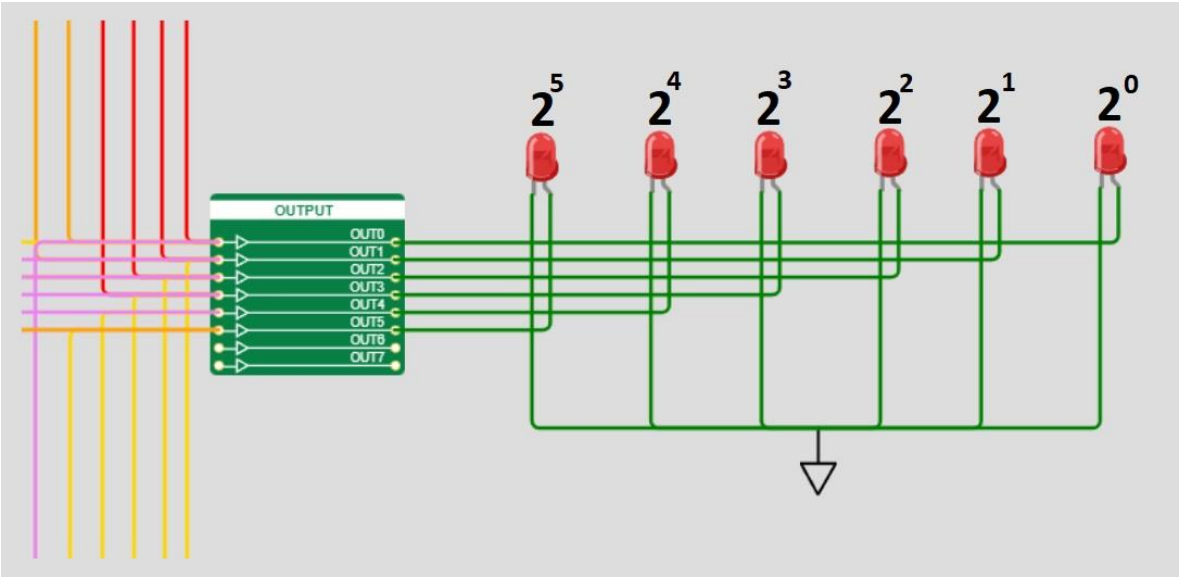
Para utilizar la señal de reloj se tienen dos opciones. La primera opción es usar una frecuencia de 1Hz para que envíe un flanco de subida de forma automática. La segunda opción es utilizar un push button para enviar la señal de reloj de forma manual. Para cambiar de función se tiene un switch deslizante para elegir entre estas dos opciones.



Salida de datos

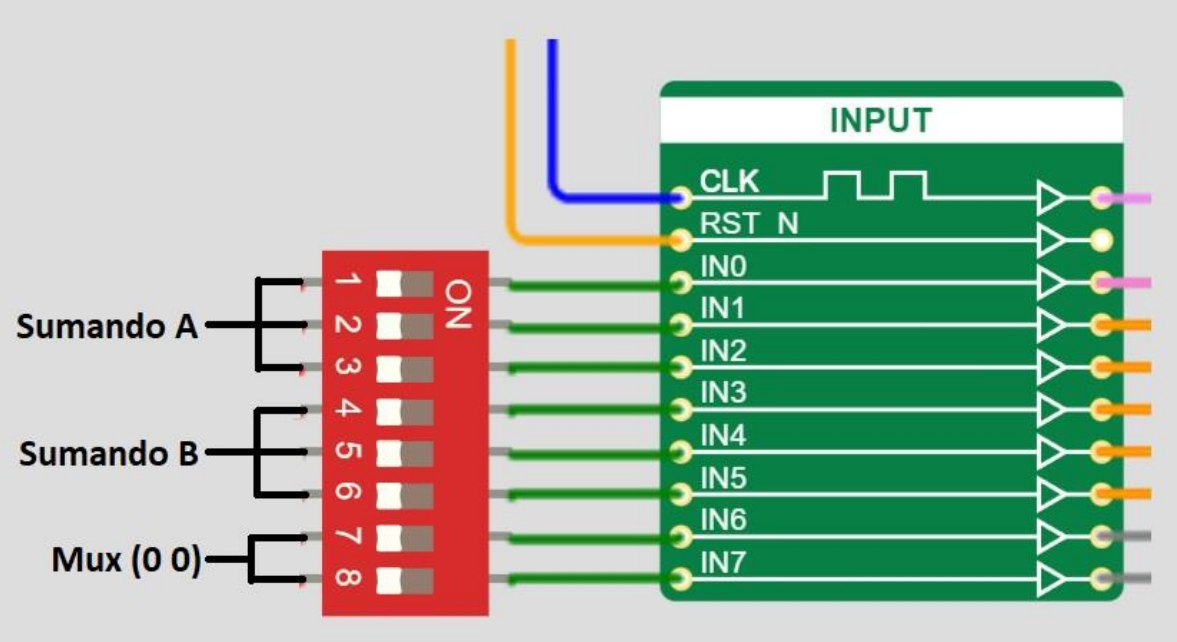
Para la salida de datos se utilizarán leds como indicadores. Cada vez que se prende un led significa que le llego un uno en binario, en caso de que esté apagado es un cero en binario. El bit de menor valor está posicionado en la parte derecha de todos los leds, mientras que el bit de mayor valor está posicionado del lado izquierdo de todos los leds.

El único caso en el que esto cambia es en el contador, pero esto se verá más a detalle más adelante.



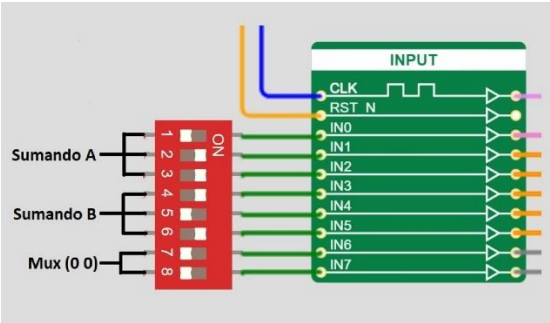
Sumador

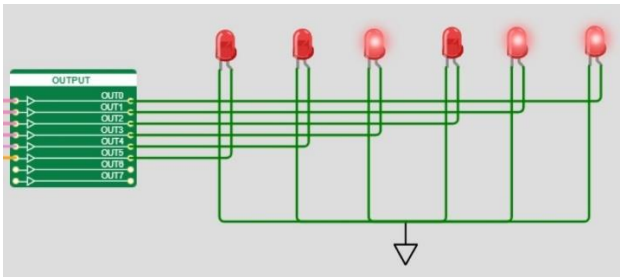
Para utilizar el Sumador Binario de 3 Bits, tanto el SW1 como el SW2 deben de estar en ceros (0 0) tal y como se muestra en la imagen. Al primer sumando (A) le corresponden las primeras tres entradas de datos y al segundo sumando (B) las siguientes tres entradas de datos.



Ejemplo 1:

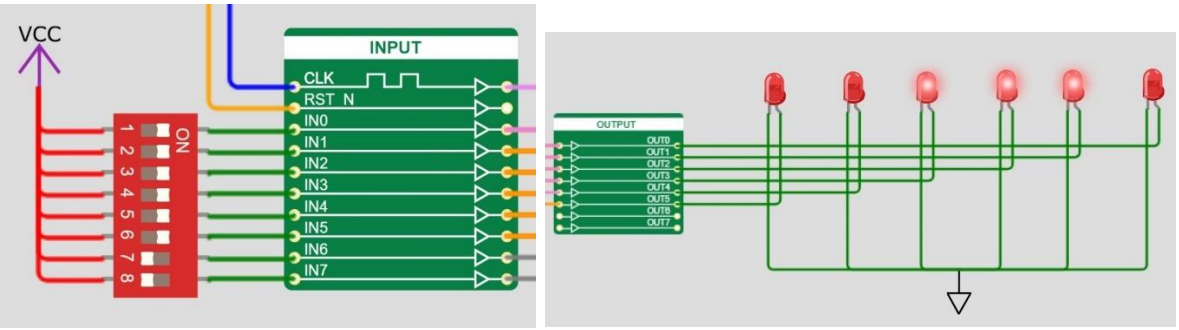
- Sumando A = **1 1 0** = 6 (decimal).
- Sumando B = **1 0 1** = 5 (decimal).
- Resultado = **1 0 1 1** = 11 (decimal).





Ejemplo 2:

- Sumando A = **1 1 1** = 7 (decimal).
- Sumando B = **1 1 1** = 7 (decimal).
- Resultado = **1 1 1 0** = 14 (decimal).

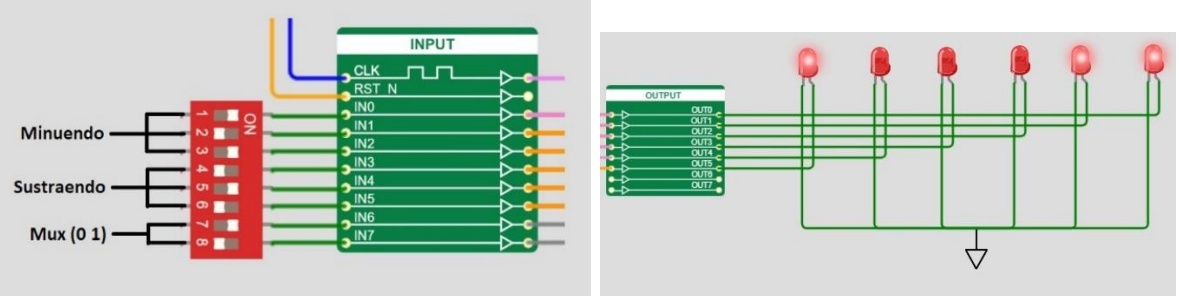


Restador

Para utilizar el Restador Binario de 3 Bits, el SW1 debe de estar activado y el SW2 debe de estar en cero (0 1) tal y como se muestra en la imagen. Al minuendo le corresponden las primeras tres entradas de datos y al sustraendo las siguientes tres entradas de datos. Se les recuerda que el bit de mayor valor es de signo, si está activado (1) significa que el valor es positivo y si está apagado (0) el resultado es negativo.

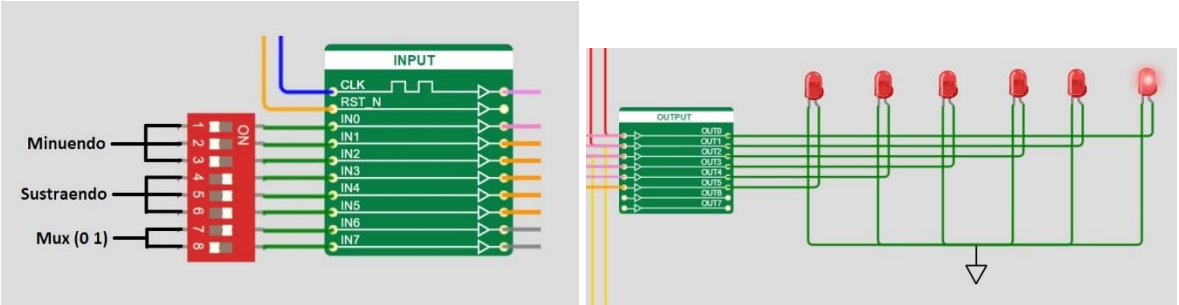
Ejemplo 1:

- Minuendo = **1 0 1** = 5 (decimal).
- Sustraendo = **0 1 0** = 2 (decimal).
- Resultado = **1 0 1 1** (el bit de mayor valor es el signo positivo) = 3 (decimal).
- Cabe recalcar que **el cuarto y quinto bit de derecha a izquierda no se cuentan**, pero se dejó este espacio entre bits para poder apreciar mejor el bit de signo.



Ejemplo 2:

- Minuendo = **0 0 0** = 0 (decimal)
- Sustraendo = **1 1 1** = 7 (decimal)
- Resultado = **0 0 0 1** (el bit de mayor valor es el signo negativo) = -7 (decimal)



Como se puede apreciar en este último ejemplo, el resultado es diferente a lo esperado. El resultado que se esperaría que apareciera es el 7 negativo, o sea **0 1 1 1**. Esto se debe a que el resultado mostrado se encuentra como Complemento a 2 y para pasarlo al resultado esperado se realiza la conversión a Complemento a 2 pero al revés, de la siguiente manera:

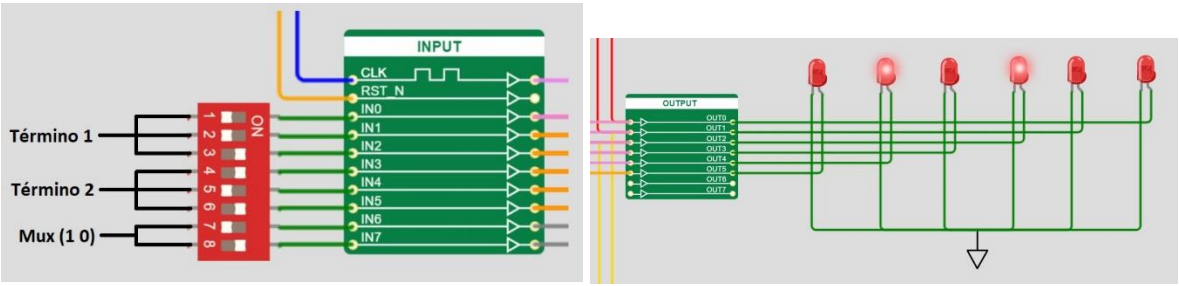
- El resultado obtenido fue **0 0 0 1.** A este número se le debe de restar uno al bit menos significativo, por lo tanto: **0 0 0 1 – 0 0 0 1.**
- El resultado de esta resta es **0 0 0 0**. Ahora todos los bits se deben de cambiar a su inverso, o sea los ceros se cambian a uno y los unos se cambian a ceros a excepción del bit de signo.
- El resultado de esta inversión sería **0 1 1 1**, dando como resultado el -7 en binario como normalmente se le encuentra.

Multiplicador

Para utilizar el Multiplicador Binario de 3 Bits, el SW1 debe de estar en cero y el SW2 debe de estar en uno (1 0) tal y como se muestra en la imagen. Al primer término le corresponden las primeras tres entradas de datos y al segundo término las siguientes tres entradas de datos.

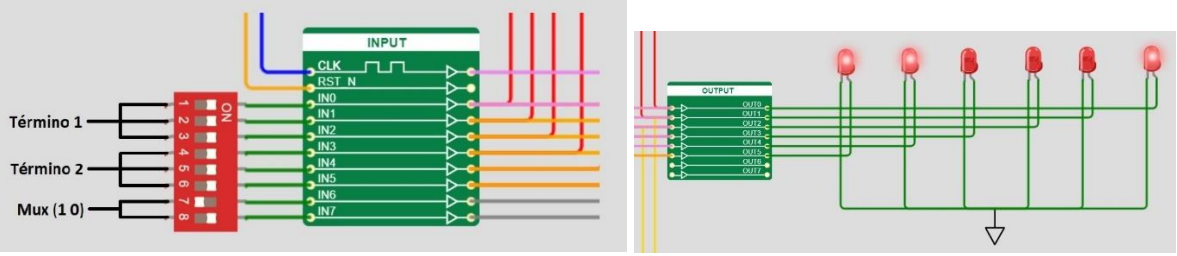
Ejemplo 1:

- Término 1 = **1 0 0** = 4 (decimal).
- Término 2 = **1 0 1** = 5 (decimal).
- Resultado = **0 1 0 1 0 0** = 20 (decimal).



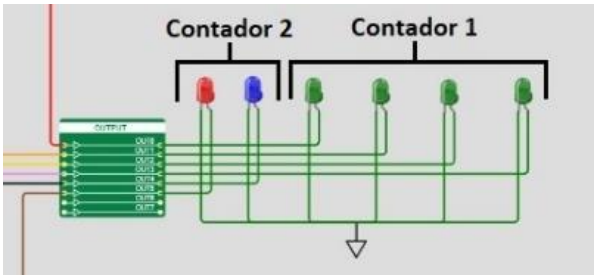
Ejemplo 2:

- Término 1 = **1 1 1** = 7 (decimal).
- Término 2 = **1 1 1** = 7 (decimal).
- Resultado = **1 1 0 0 0 1** = 49 (decimal).



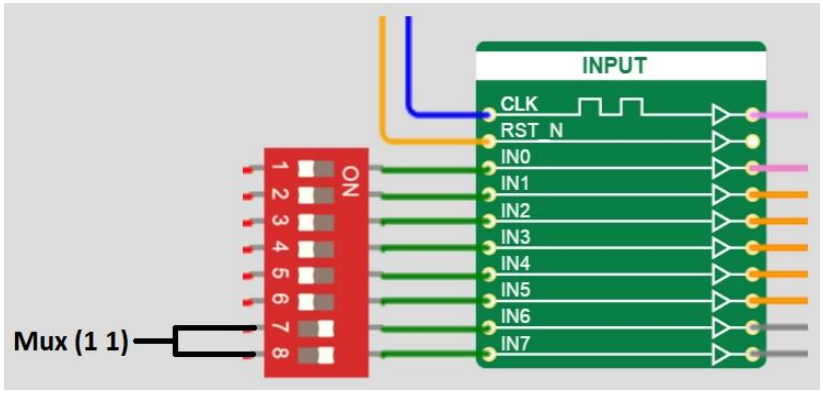
Contador

Para utilizar el Contador, el SW1 debe de estar en uno y el también SW2 debe de estar en uno (1 1) tal y como se muestra en la imagen. El contador realmente se compone de dos contadores; el contador de 4 bits utiliza los bits menos significativos y cuenta desde el cero al 15 en binario, mientras que el segundo contador utiliza los bits de mayor valor. Este último lo que va a hacer es ir alternando los leds en su encendido y apagado, eso quiere decir que cada vez que pase un periodo de tiempo, uno de los leds va a estar encendido y el otro apagado. Ambos contadores están conectados a la misma señal de reloj.

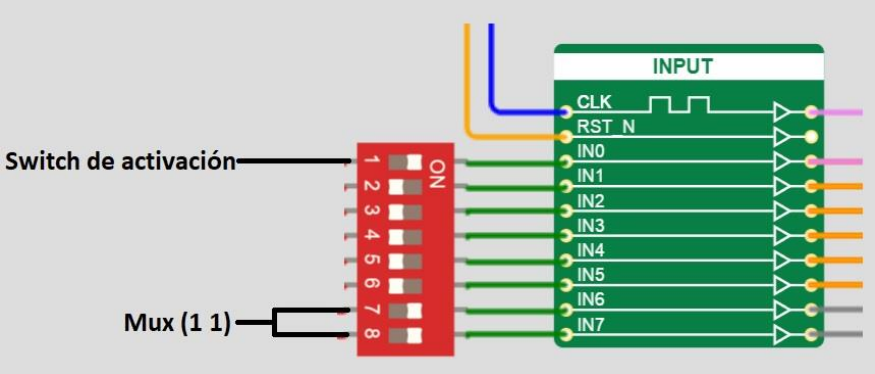


Ejemplo:

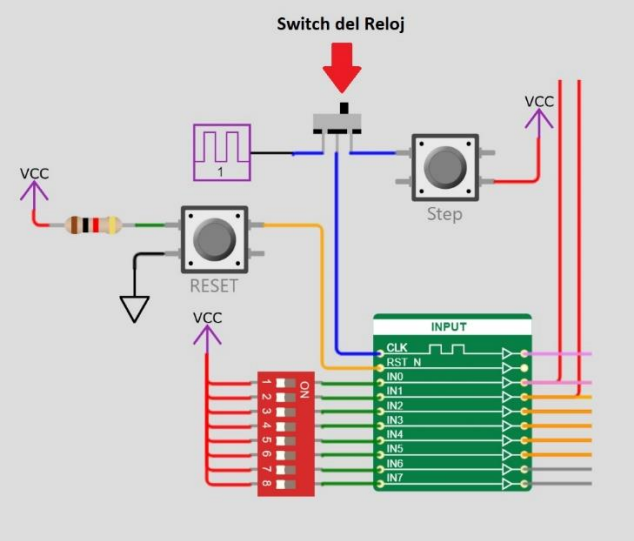
- Para utilizar el Contador, los switchs del Multiplexor deben de estar ambos activados (1 1).



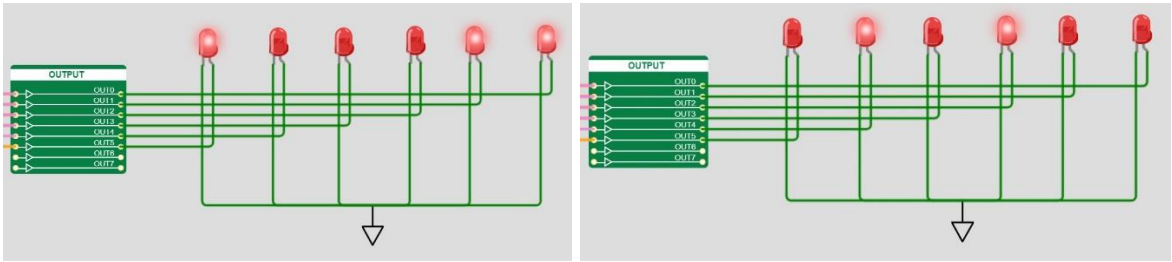
- Para activar el contador, se debe colocar en uno el primer switch.



- Hay un switch que permite usar una frecuencia de reloj preestablecida de 1Hz de frecuencia o se puede utilizar un push button como reloj y cambiar el contador a voluntad.



- Finalmente, el contador de 4 bits irá sumando de uno en uno hasta llegar hasta 15 en número binarios y empezar desde cero nuevamente. El contador de 2 bits irá intercalando el prendido de los leds.



- En ocasiones la simulación puede fallar y prender ambos leds del contador de 2 bits, esto debido al simulador, ya que en ocasiones puede tener algunos detalles a la hora de ejecutar algún circuito. Para solucionar esto, basta con reiniciar la simulación y se corregirá.