

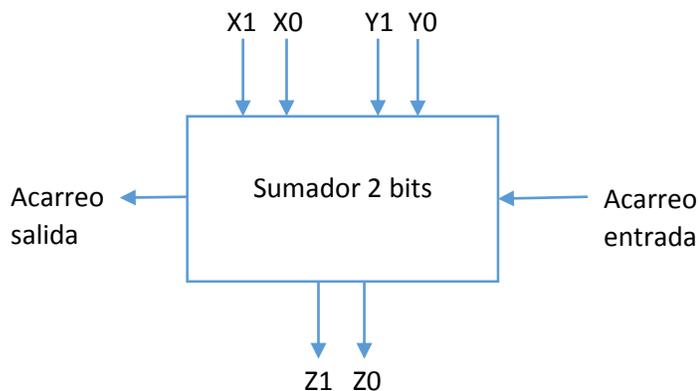
Taller #2: Circuitos Combinacionales

Un circuito combinacional es un circuito digital que implementa una o varias funciones de conmutación, tal que las salidas del circuito en cada instante de tiempo dependen única y exclusivamente de las señales de entrada en aquel mismo instante.

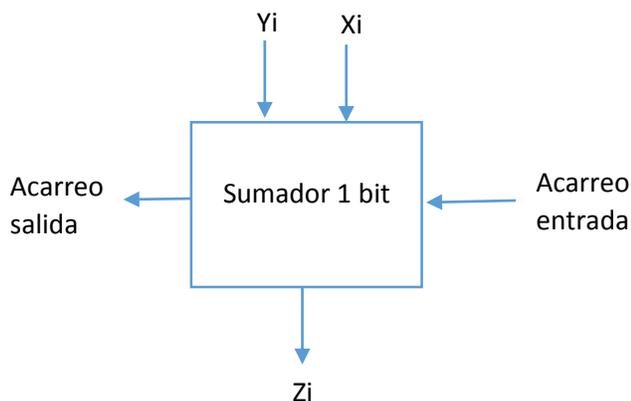
PARTE I

Los siguientes pasos detallan como realizar un circuito combinacional en Logisim de un sumador de 2 bits.

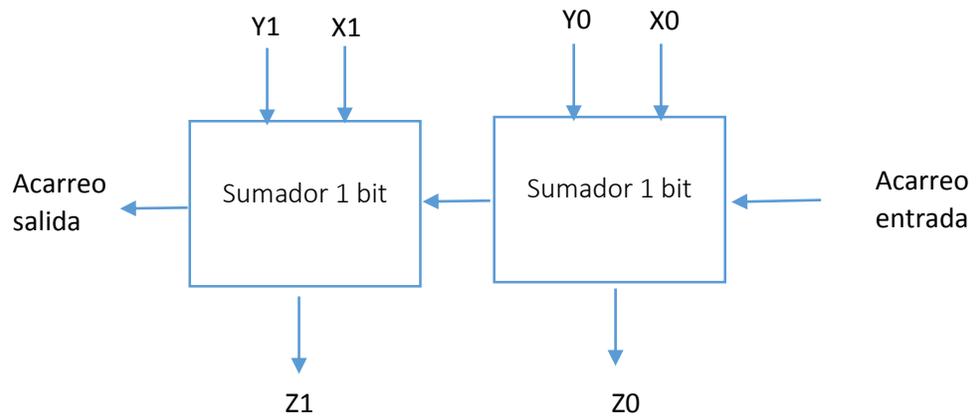
- 1) Para realizar un sumador de 2 bits hay que tomar en cuenta que tendrá 2^2 entradas es decir, contara con cuatro entradas, como se muestra a continuación:



- 2) La forma más fácil de realizar este circuito combinacional es separándolo en dos sumadores de un bit, esto hace el circuito modular, es decir, si se necesita un sumador de 4 bits solo se tienen que unir cuatro de un solo bit, quedando así el sumador de 1 bit.



3) Y para implementar el sumador de 2 bits uniremos dos sumadores de 1 bit como el anterior:



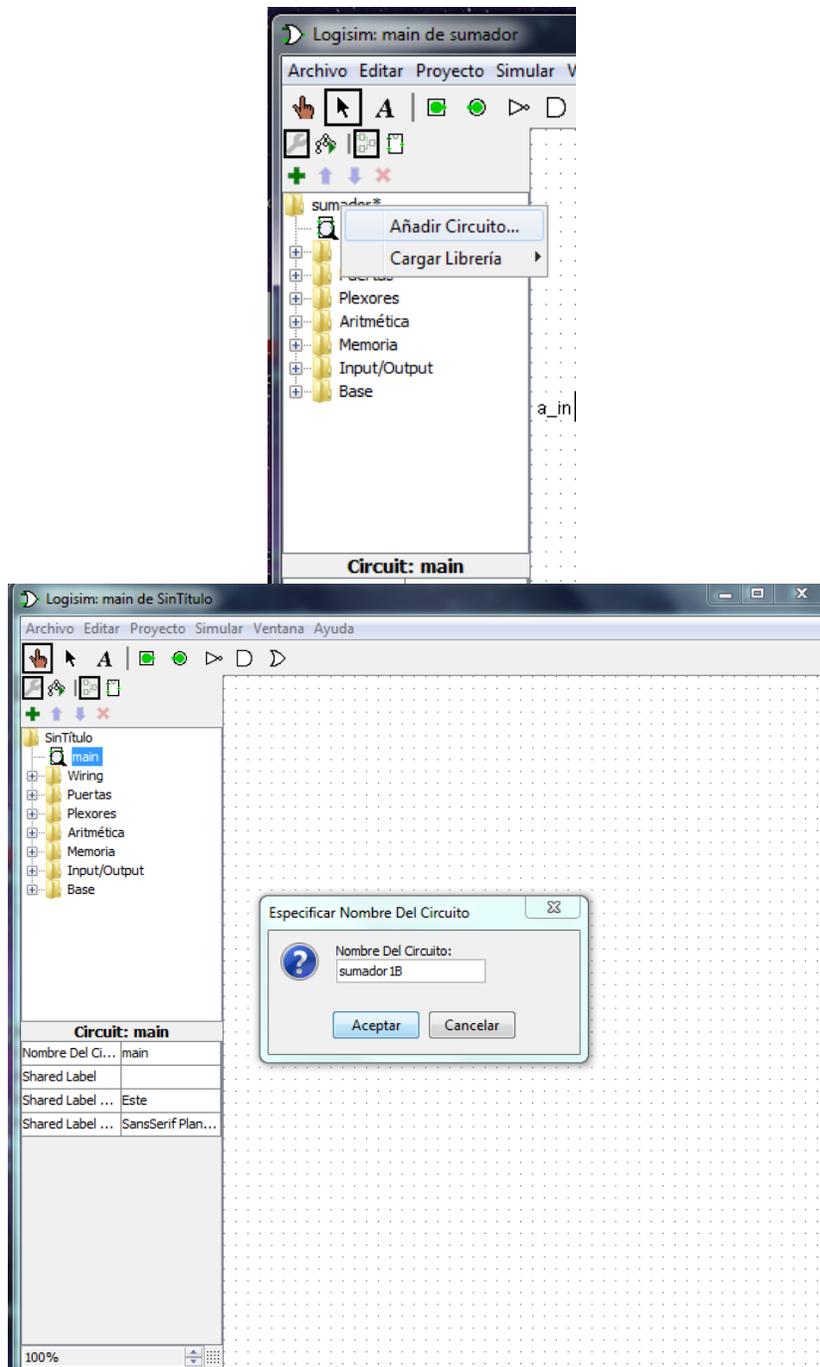
4) La tabla de verdad para un sumador de 1 bit es la siguiente:

Dónde a_{in} es el acarreo de entrada y a_{out} el de salida.

(Recordemos que $x = 1, y = 1$ y $a_{in} = 0$ genera una salida $Z = 0$ con $a_{out} = 1$ y para $x = 1, y = 1$ y $a_{in} = 1$ genera una salida $Z = 1$ con $a_{out} = 1$)

x	y	a_in	a_out	z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

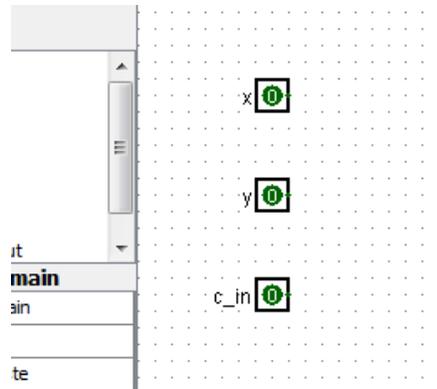
5) El primer paso en logisim es darle clic a la carpeta que estamos creando y seleccionar "Añadir Circuito", en este caso de nombre le colocamos Sumador1B



- 6) Ahora para realizar el circuito para a_{out} vamos a formular la función booleana asociada, recordemos que esto la suma de los minterms de la función.

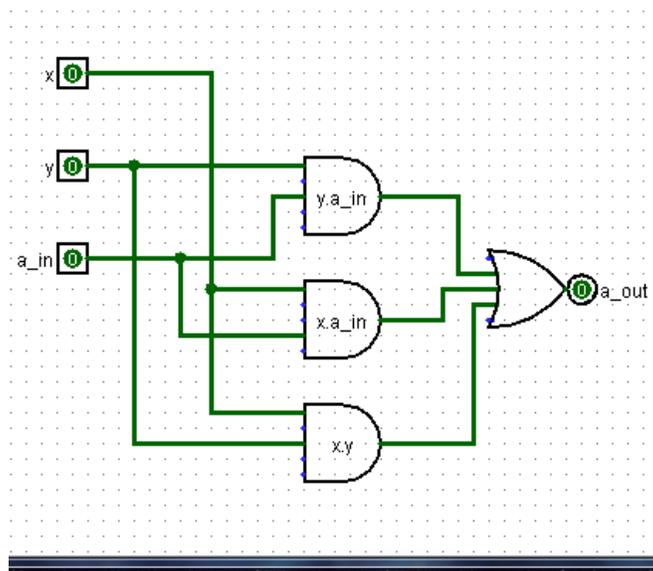
Entonces $a_{out} = \bar{x}ya_{in} + x\bar{y}a_{in} + xy\bar{a}_{in} + xya_{in}$, minimizando la función nos quedaría $a_{out} = ya_{in} + xa_{in} + xy$

Para representar esta función booleana en logisim necesitamos colocar las tres entradas: x , y y a_{in} .

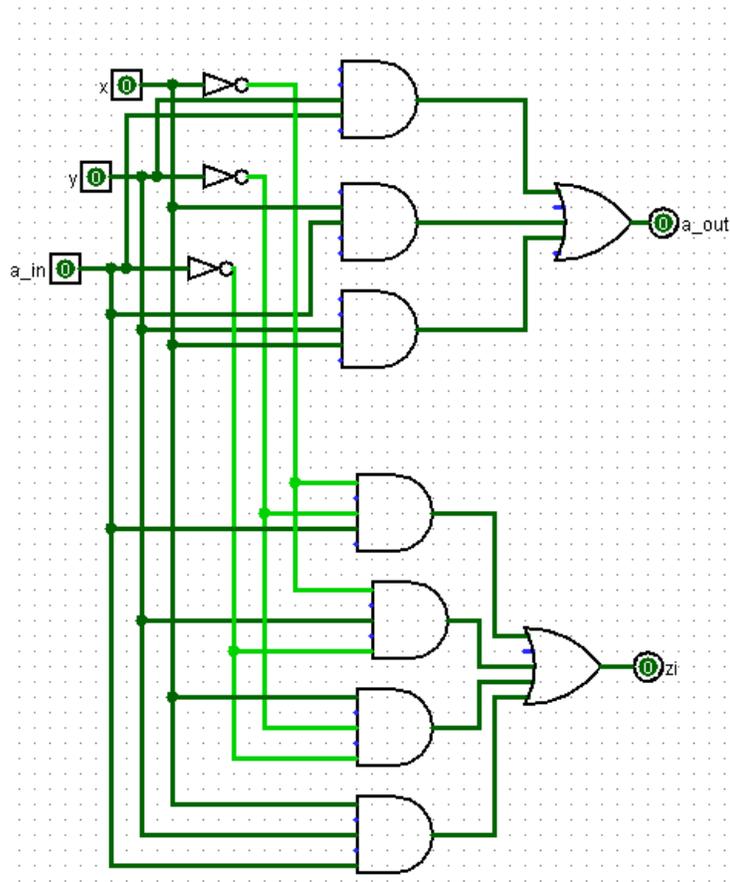


Y tres compuertas AND debido a la suma de productos de la función que tiene tres multiplicaciones booleanas, para la suma utilizaremos un OR.

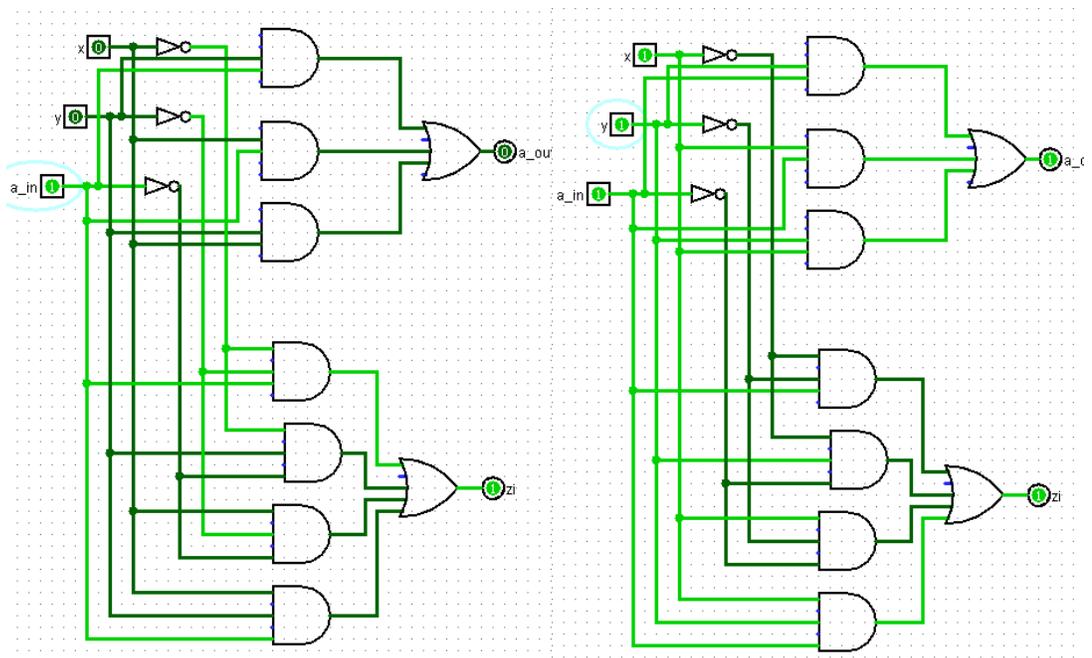
La función booleana para el resultado de a_out quedaría de la siguiente manera:



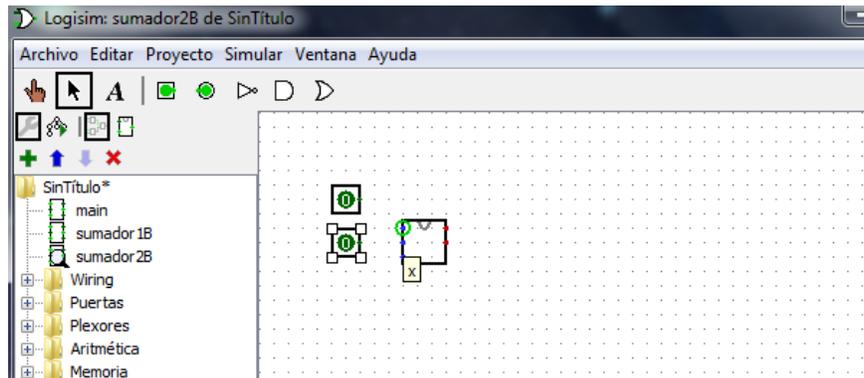
- 7) Para la salida Zi realizaremos el mismo proceso, generamos la función booleana que es la siguiente: $z_i = \bar{x}\bar{y}a_{in} + \bar{x}y\bar{a}_{in} + x\bar{y}\bar{a}_{in} + xy a_{in}$ y realizamos el circuito lógico combinacional asociado a la función booleana de zi.



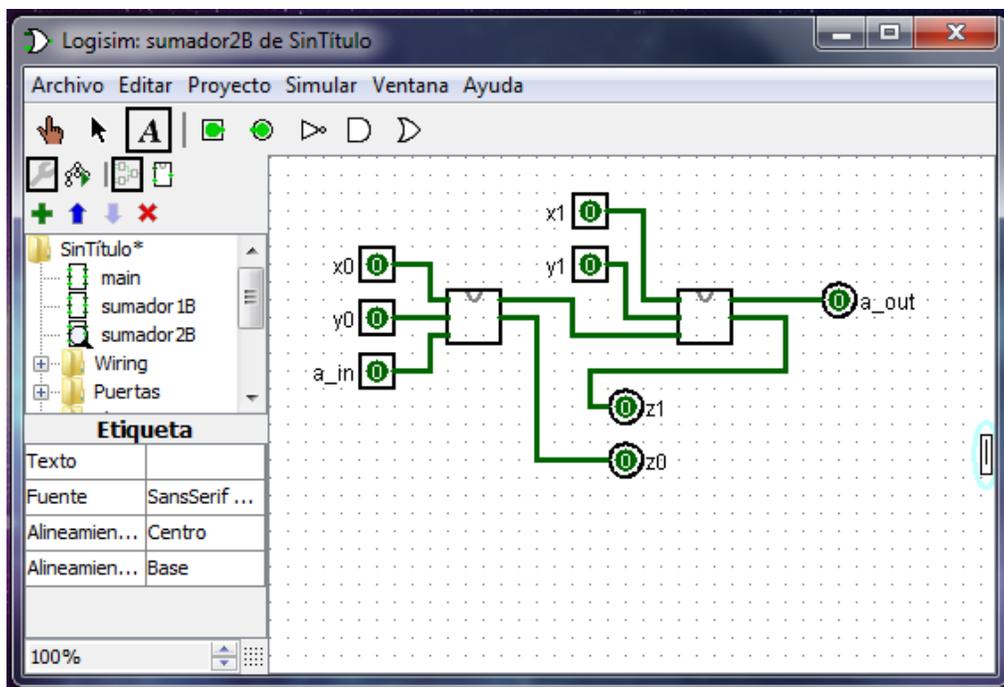
8) Podemos probar el circuito como se aprendió en el taller 1 y ver que la salida corresponde a la tabla de verdad que hicimos en un principio.



- 9) Ahora añadimos un nuevo circuito, que es el sumador de 2 bits, lo añadimos como se realizó en el paso 5.
- Al tener el lienzo del sumador2B, lo que haremos será seleccionar un sumador1B y arrastrarlo hasta el lienzo, nos quedara de la siguiente manera:



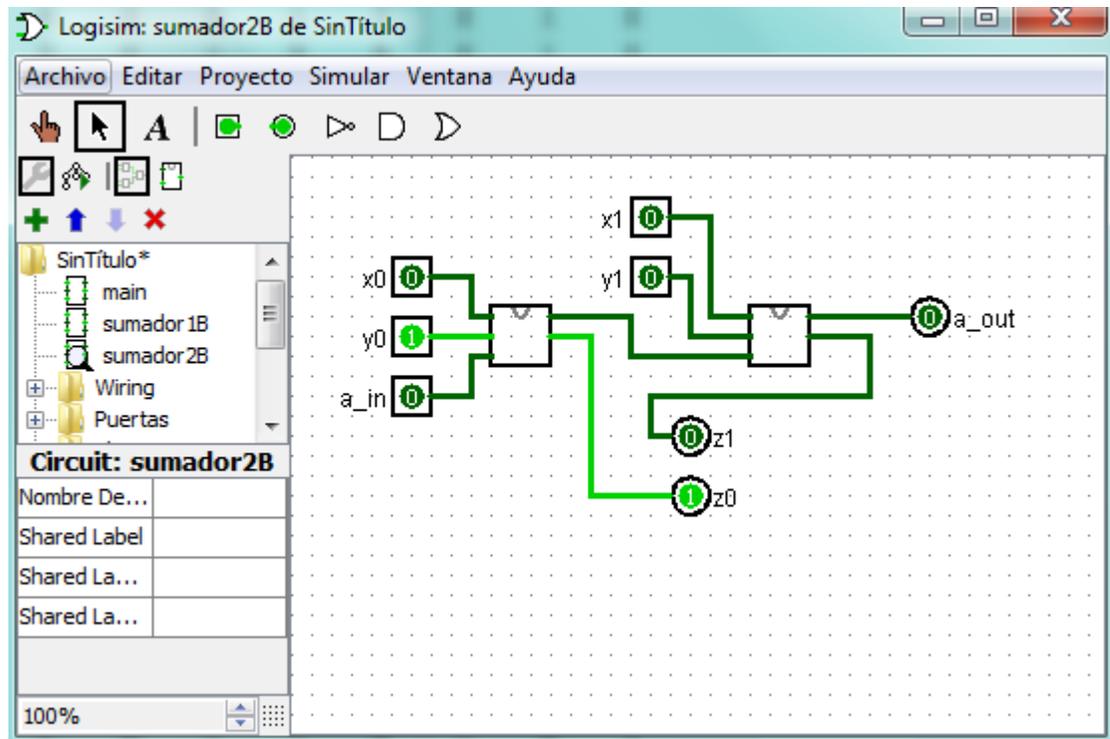
Si movemos el cursor sobre el sumador1B que acabamos de colocar podremos ver que se indica las entradas (x,y) y el acarreo de entrada que cuando se realiza un sumador de 2 bits, el segundo sumador tendrá de acarreo de entrada el acarreo de salida del primero.



Como la imagen anterior nos quedaría un sumador de 2 bits realizado con dos sumadores de 1 bit. La tabla de verdad asociada a este sumador es la siguiente:

x1	x0	y1	y0	a_in	a_out	z1	z0
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1
0	0	0	1	0	0	0	1
0	0	0	1	1	0	1	0
0	0	1	0	0	0	1	0
0	0	1	0	1	0	1	1
0	0	1	1	0	0	1	1
0	0	1	1	1	1	0	0
0	1	0	0	0	0	0	1
0	1	0	0	1	0	1	0
0	1	0	1	0	0	1	0
0	1	0	1	1	0	1	1
0	1	1	0	0	0	1	1
0	1	1	0	1	1	0	0
0	1	1	1	0	1	0	0
0	1	1	1	1	1	0	1
1	0	0	0	0	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	0	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	0	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	0	1	0	1
1	0	1	1	1	1	1	0
1	1	0	0	0	0	1	1
1	1	0	0	1	1	0	0
1	1	0	1	0	1	0	0
1	1	0	1	1	1	0	1
1	1	1	0	0	1	0	1
1	1	1	0	1	1	1	0
1	1	1	1	0	1	1	0
1	1	1	1	1	1	1	1

La tabla anterior nos servirá para poder testear el circuito combinacional que acabamos de realizar. Por ejemplo la suma de $x_0 = 0$, $y_1 = 1$, $x_1 = 0$, $y_1 = 0$ y $a_{in} = 0$ es igual a 01 con acarreo 0, como se puede ver en la figura es la salida que da el circuito con esos valores de entrada.



Ejercicio

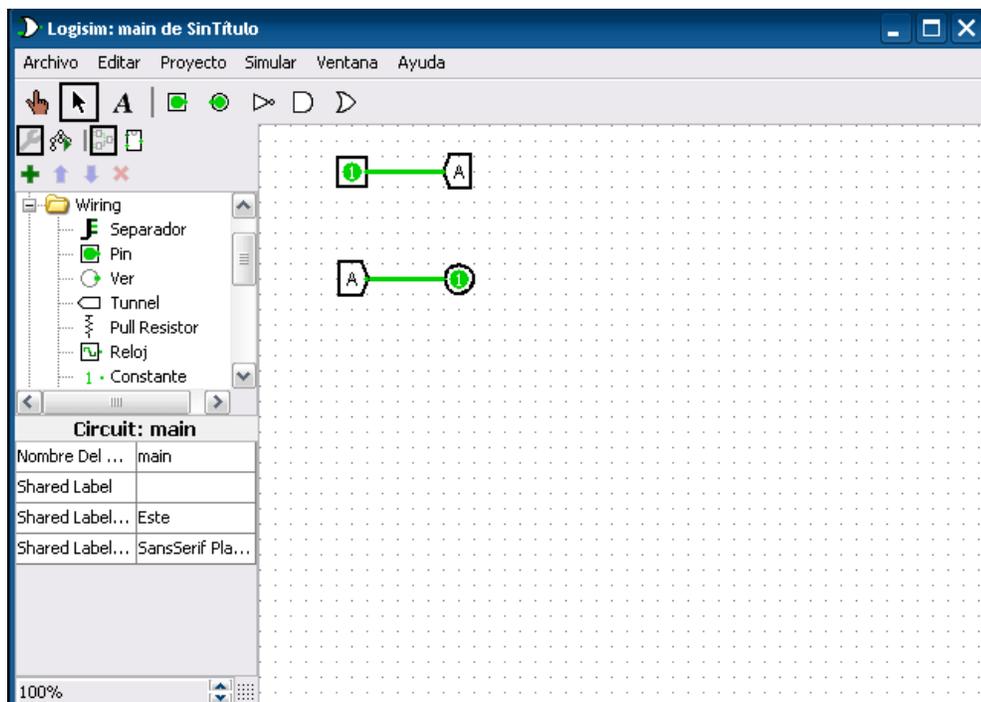
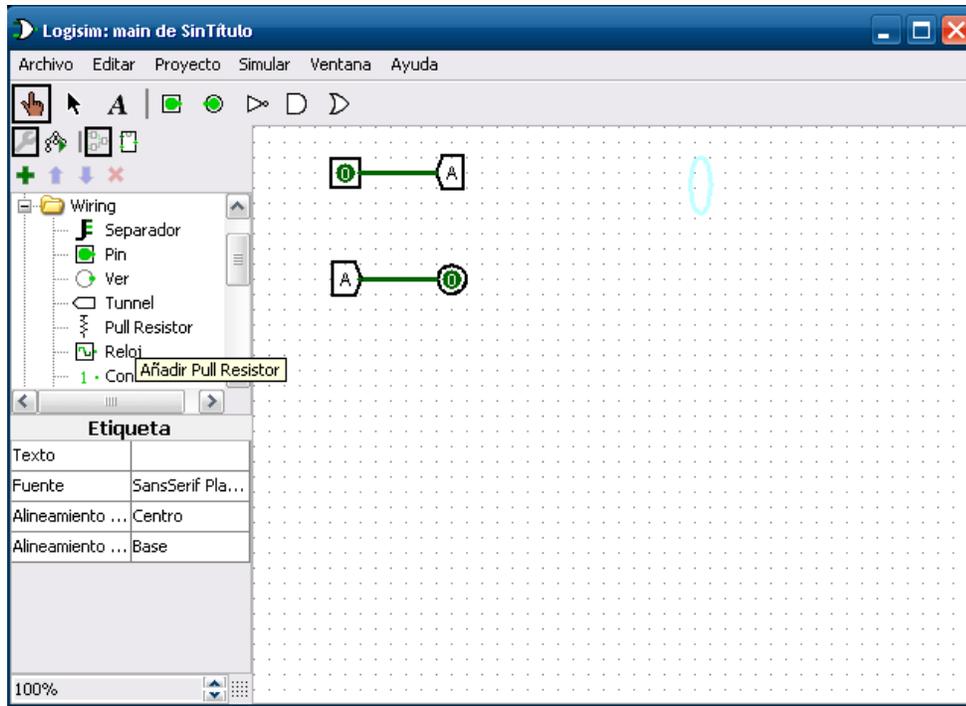
- 1- Realice un sumador de 4 bits.
- 2- Realice el sumador de 1 bit con puertas NAND, NOR o XOR.

PARTE II

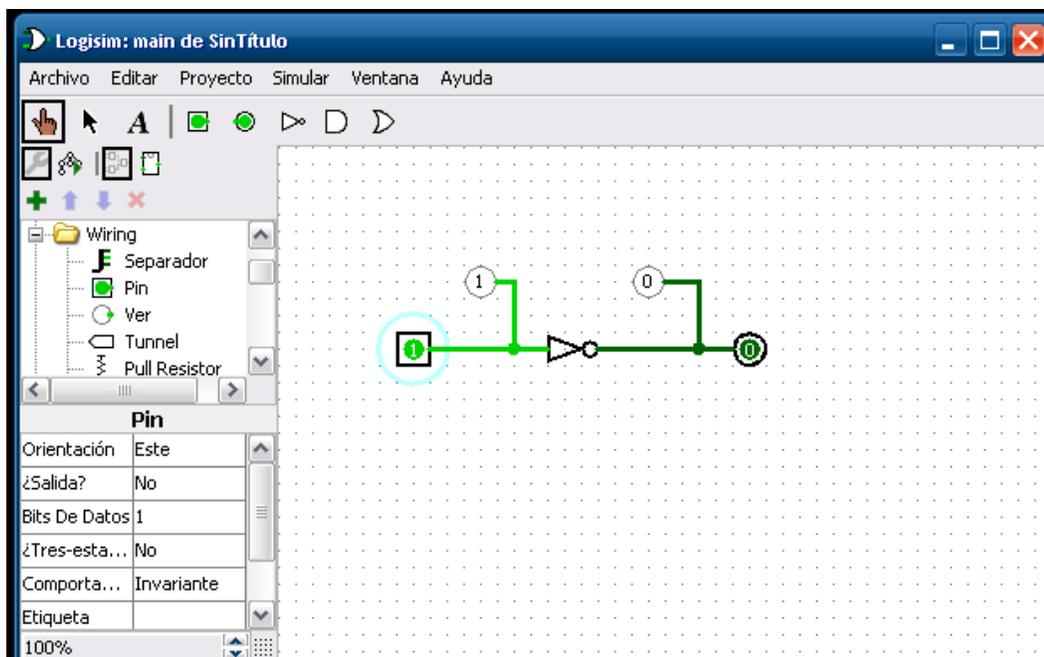
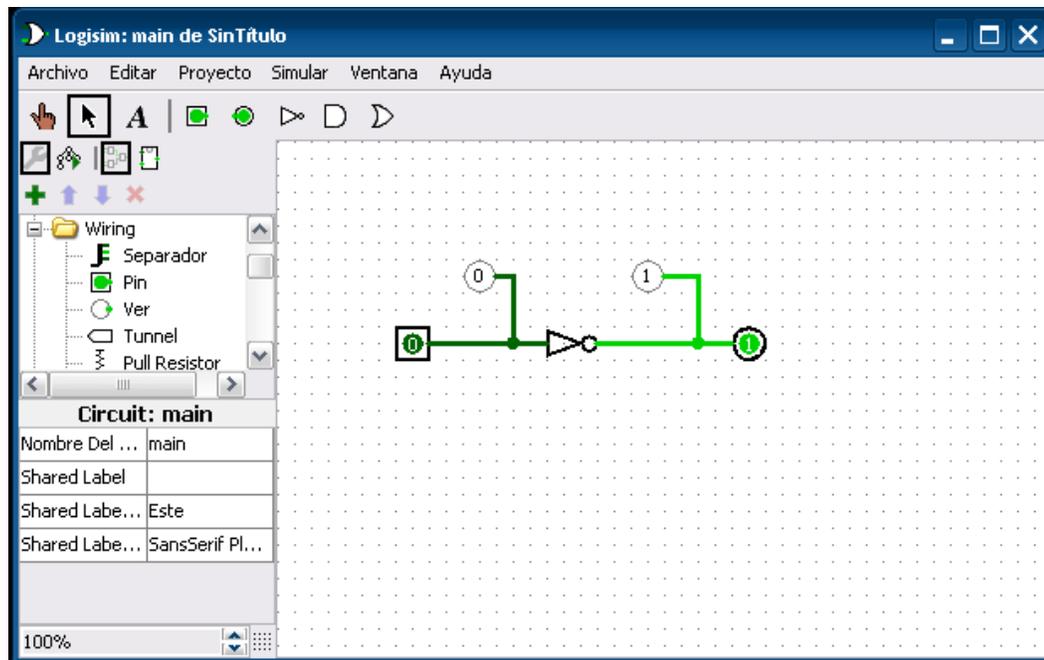
Adicionalmente, existen componentes para hacer más sencilla la elaboración de los circuitos combinatoriales.

El primero se denomina túnel y se utiliza para evitar el cableado de más y enredado que se puede generar al hacer un circuito muy grande. Lo vamos a encontrar en el menú de la izquierda *Wiring* -> *Tunnel*, a este componente le vamos a poner una etiqueta y todos los que tengan la misma etiqueta van a generar la entrada que se le asigne. Su funcionamiento se puede ver en la imagen siguiente:

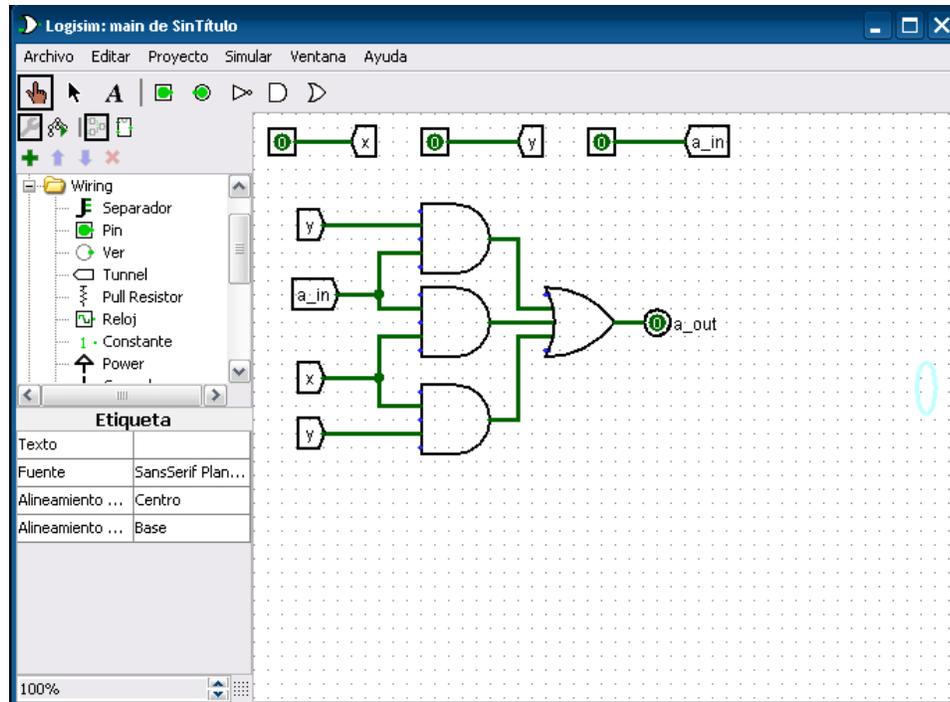
Usando un túnel con la etiqueta A, este propaga la entrada que se le asigne en algún momento dado.



También existe un componente llamado Ver que encontramos en el menú izquierdo en *Wiring* -> *Ver*, que sirve para poder ver o testear el valor que esté pasando por un cable, entrada o salida. El ejemplo se puede ver en la imagen.

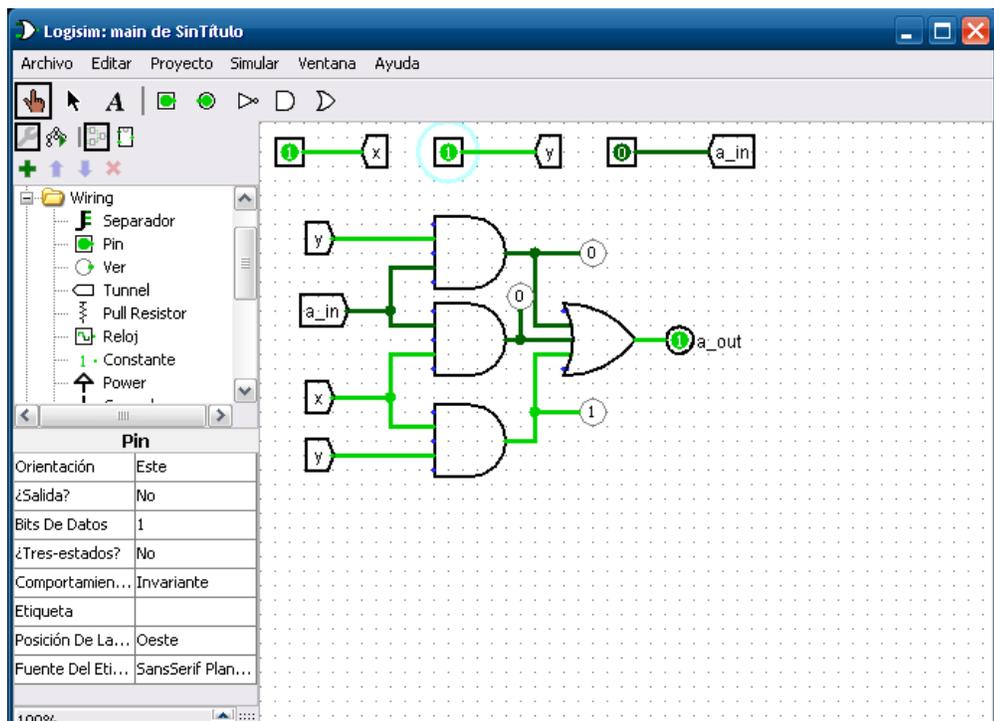


Ahora vamos a crear el circuito combinacional para la salida a_out haciendo uso de los túneles:

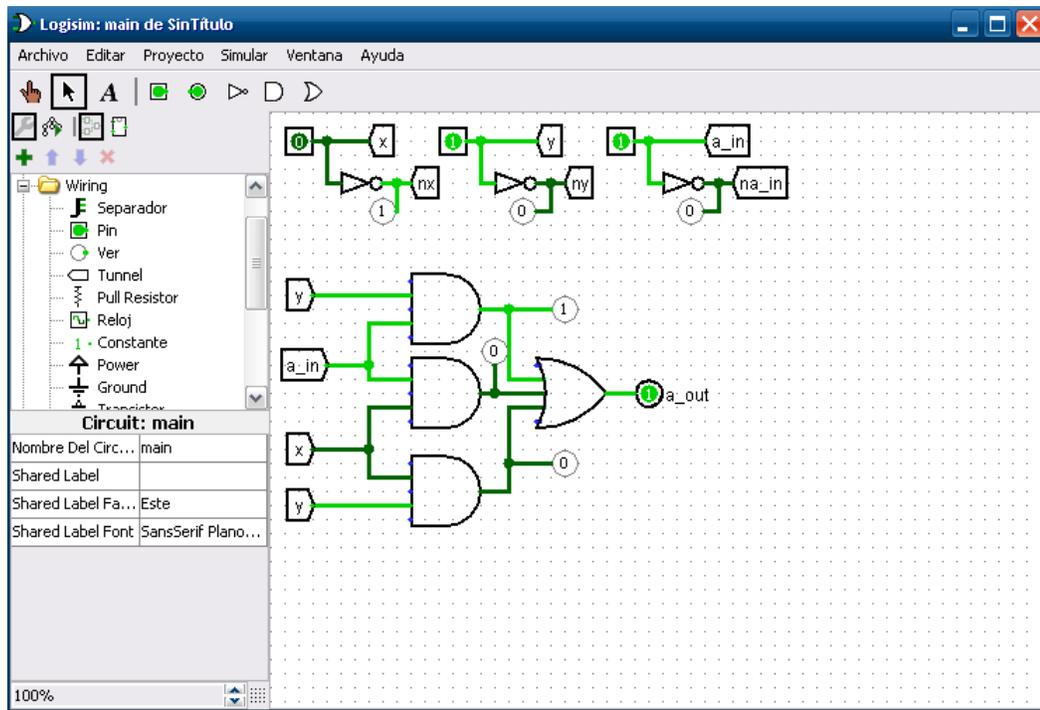


En la imagen anterior se puede ver que el circuito se vuelve más entendible ya que no existen cables que se solapan, de esta manera se organiza mejor el circuito, y se evitan errores, también se tiene un control centralizado de las variables.

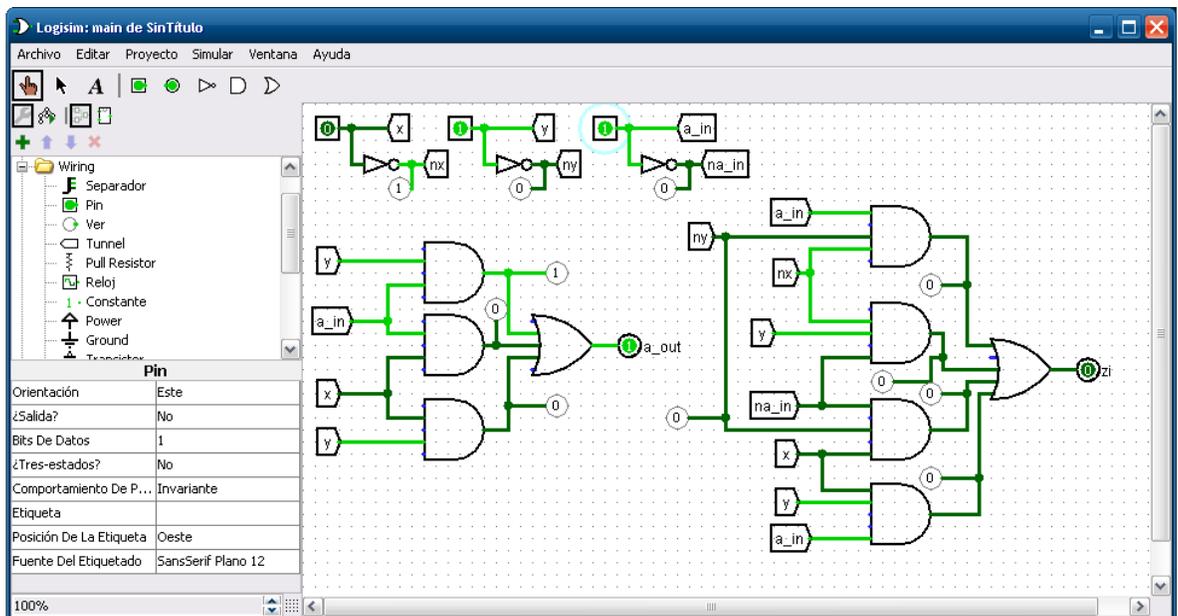
A este circuito vamos a agregarle el componente Ver, para poder testear el circuito y ver qué valor tienen las salidas de cada and.



Para crear el circuito de z_i , se necesitan las variables negadas, para esto utilizaremos túneles, donde cada variable tendrá su negado expresado con una n delante del nombre, también pondremos unos componentes Ver para demostrar su funcionamiento mejor.



Por último crearemos la salida z_i con los componentes nuevos. Con esto podemos notar que el circuito queda mucho más limpio y ordenado, evitando el solapamiento de cables y siendo más fácil el entendimiento del circuito.



Ejercicio

Agregar estos componentes al circuito que creo en el ejercicio 2 de la parte 1.

GDOECII