Este informe cumple el código de honor.

**PROJECT 01 |** BOLEAN LOGIC

**LISTADO DE CONCEPTOS**

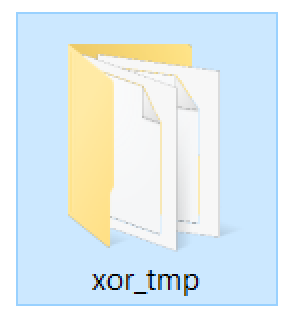
**1.a ¿Por qué es importante realizar los proyectos en orden?**

Es muy importante realizar los proyectos en orden, dado que el curso fue sido diseñado de tal forma que cada sesión fuera un bloque de construcción esencial para acceder al conocimiento de la siguiente sesión.

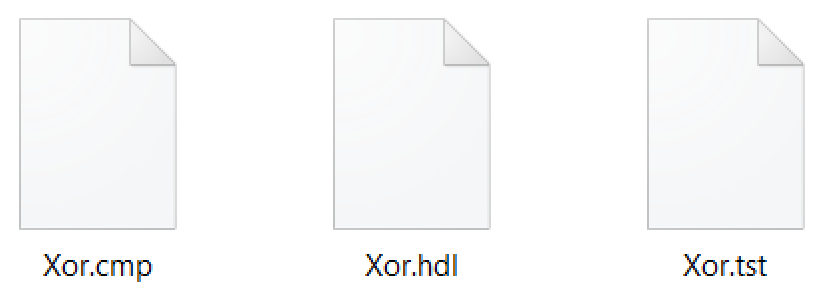
**1.b La implementación de la XOR requiere compuertas AND y OR. Cree un directorio temporal donde estén todos los archivos correspondientes a la XOR pero sin la implementación de la AND y la OR. ¿Funciona? Explique claramente por qué**

**Configurando espacio de pruebas**

Se creó la siguiente carpeta temporal para realizar la actividad propuesta.



En ella estaban contenidos únicamente los siguientes archivos.



Luego se corrió el Script **Xor.tst** para comprobar el funcionamiento del chip **Xor.hdl.**

* HardwareSimulator Xor.tst

Nótese que el script HardwareSimulator.bat fue incluido previamente en el path del sistema para poder ser accedido desde cualquier directorio del computador.

**Resultado**

El resultado de la comprobación fue exitoso, aun sin tener implementaciones de los chips **And, Or** y **Not,** en el directorio *xor\_tmp*

**Explicación**

HardwareSimulator busca las dependencias del chip Xor.hdl en el directorio actual sobre el cual se ejecuta el programa. En este caso busca los chips And, Or y Not pero al no estar estos en el directorio xor\_tmp HardwareSimulator busca una implementación por defecto en la carpeta del programa HardwareSimulatorFolder/builtInChips

**1.c ¿Qué pasa con los puertos de las compuertas o partes no conectados?**

Cuando el puerto de salida o entrada de un chip no está conectado a ningún otro puerto de entrada, no ocurre nada, por ejemplo, en el siguiente caso la única consecuencia será que el chip Not.hdl fallará los requerimientos que especifica el script Not.tst

CHIP Not {

IN in;

OUT out;

PARTS:

Nand(a=in, b=in);

}

HardwareSimulator considera los pines no conectados como válidos, internamente asume que el valor para el respectivo pin de entrada o salida es la función false(0).

**1. d-e ¿Si tengo un bus de salida de 16 bits cómo me refiero al bit de menor y mayor peso?**

El bit menos significativo de un bus cualquiera **b** puede ser accedido escribiendo **b[0]** y el bit más significativo **b[15].**

**1. f Indique los valores de *in, out, x, y* considerando el siguiente código.**

**// Foo.hdl**

**CHIP Foo {**

**IN in[8]**

**OUT out[8]**

**// Foo's body (irrelevant to the example)**

**}**

**// AnotherChip.hdl**

**//...**

**PARTS:**

**// v = 100**

**Foo(in[2..4]=v, in[6..7]=true, out[0..3]=x, out[2..6]=7);**

**//...**

Respuesta

in[8] = [0, 0, 1, 0, 0, 0, 1, 1]

x[4] = [out[0], out[1], out[2], out[3]]

y[5] = [out[2], out[3], out[4], out[5], out[6]]

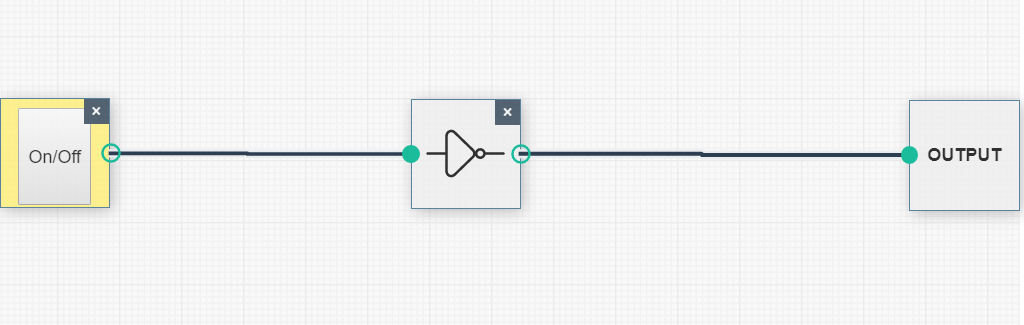
out[8] = [out[0], … , out[7]]

**EXPLICACION DE CHIPS**

El código de cada chip se explicará usando comentarios en el respectivo archivo \*.hdl de cada compuerta

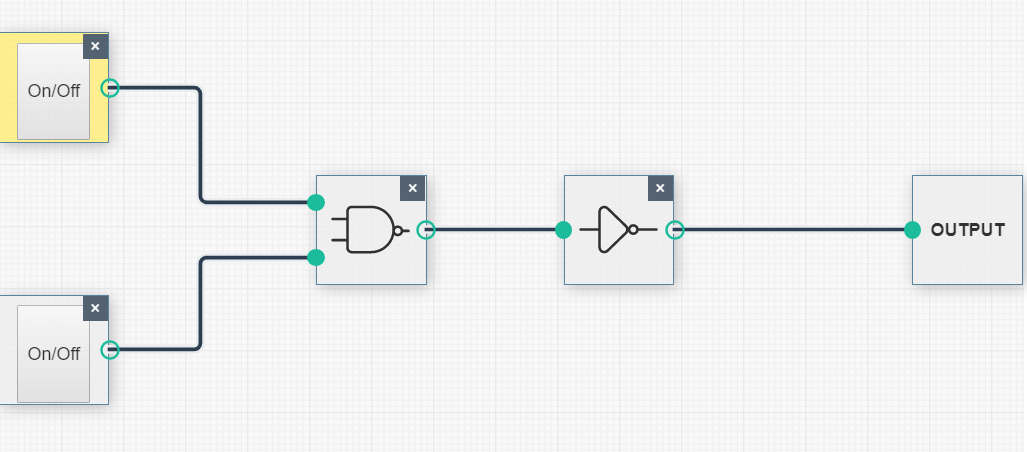
**Unary NOT**

La compuerta unary Not convierte un valor logico true o false a su respectivo contrario. A continuación se ilustra la implementación del chip usando logic-gate-simulator from Academo.org



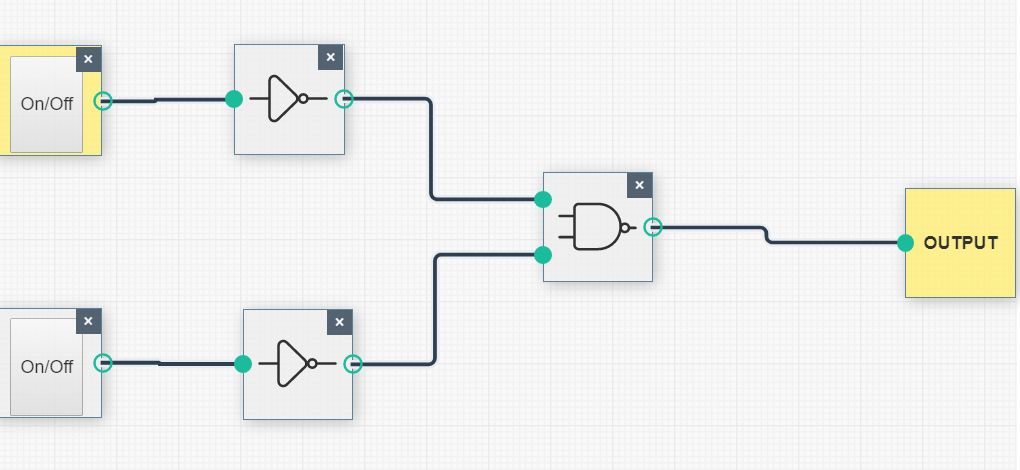
**AND**

La compuerta AND devuelve el valor lógico true solo si sus dos bits de entradas son true.



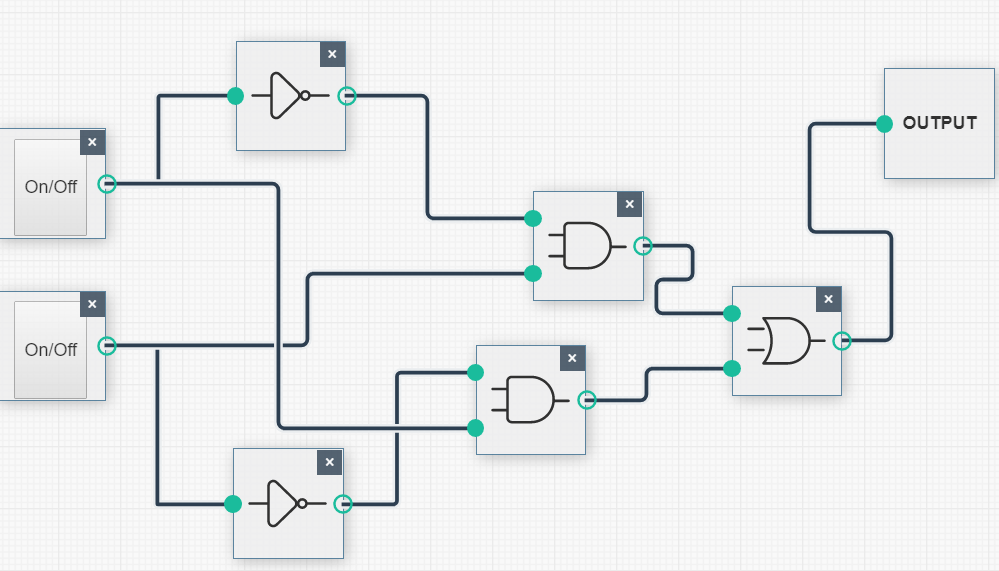
**OR**

La compuerta OR es verdadera cuando por lo menos uno de sus dos bits de entrada es 1.



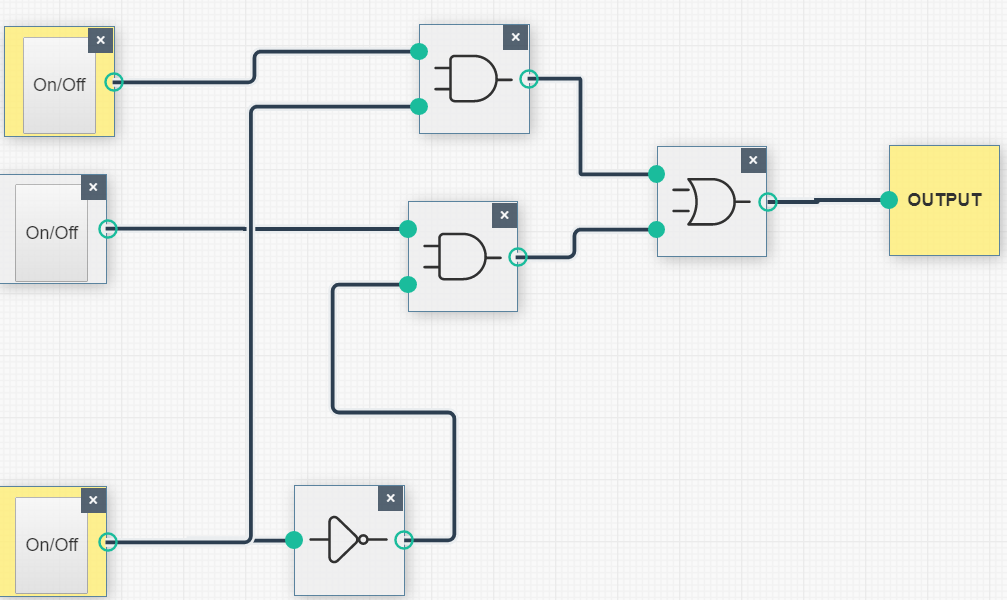
**XOR**

La compuerta Xor es verdadera cuando los valores de entrada son opuestos entre sí pero no iguales.



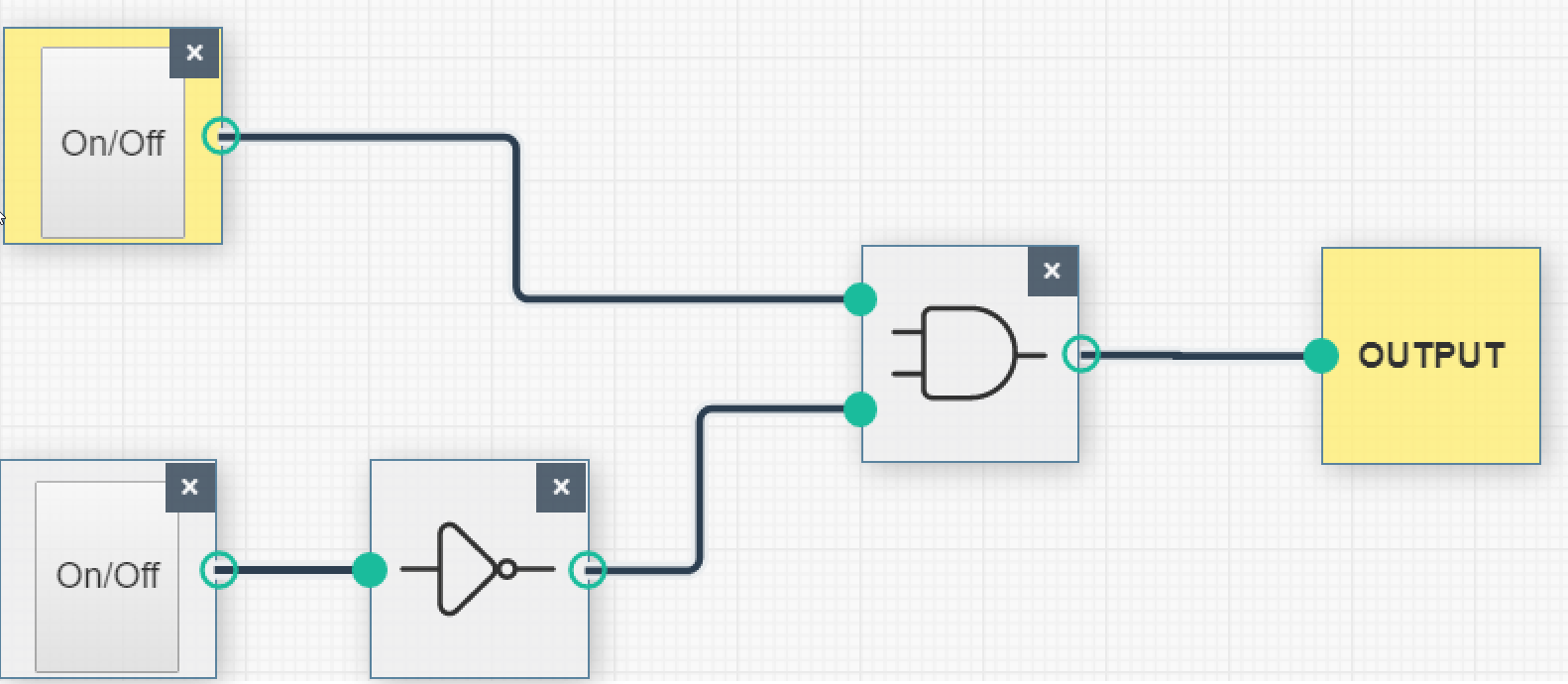
**MULTIPLEXOR**

El multiplexor de dos entradas tiene un selector de un bit y devuelve el valor de la primera entrada cuando el bit selector vale cero, en caso contrario devuelve el valor de la segunda entrada.

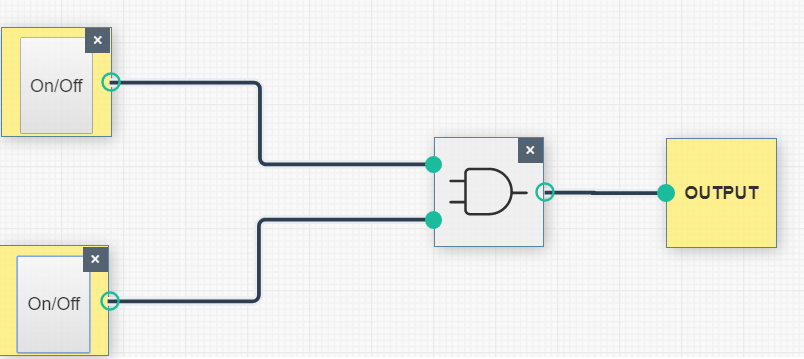


**DE-MULTIPLEXOR**

El de-multiplexor de dos canales recibe una única entrada y retransmite su valor a alguno de los dos canales de salida dependiendo del valor actual del bit selector.



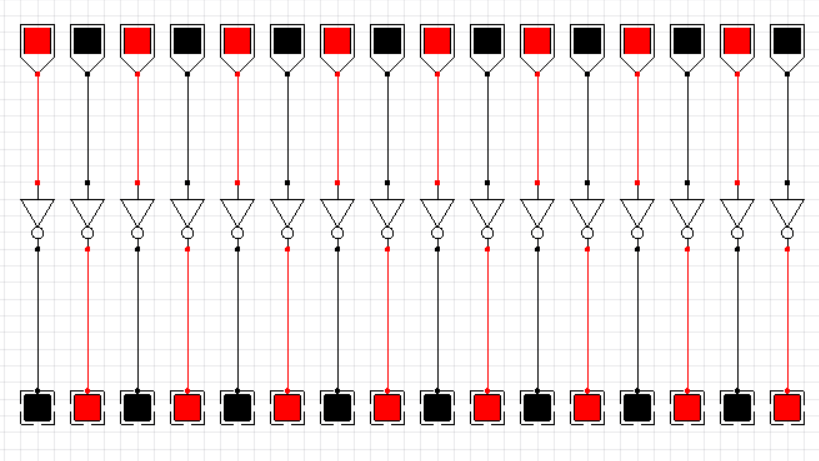
dmux enviando un 1 al canal A.



dmux enviando un 1 al canal B.

**NOT16**

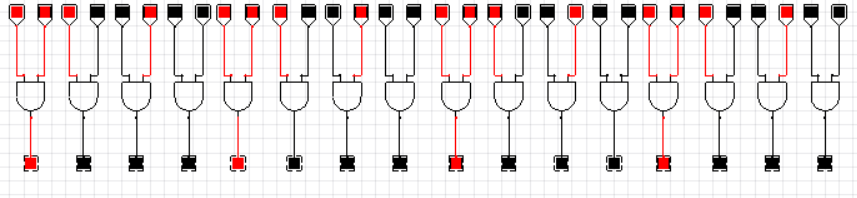
Devuelve 16 bits que representan la negación bit a bit del bus de entrada de 16 bits.

****

Donde rojo quiere decir encendido y negro apagado.

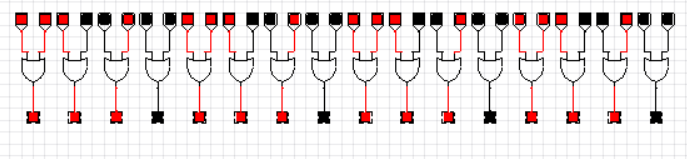
**AND16**

Realiza una comprobación And entre dos buses de entrada de 16 bits y devuelve un bus de 16 bits representando el resultado.



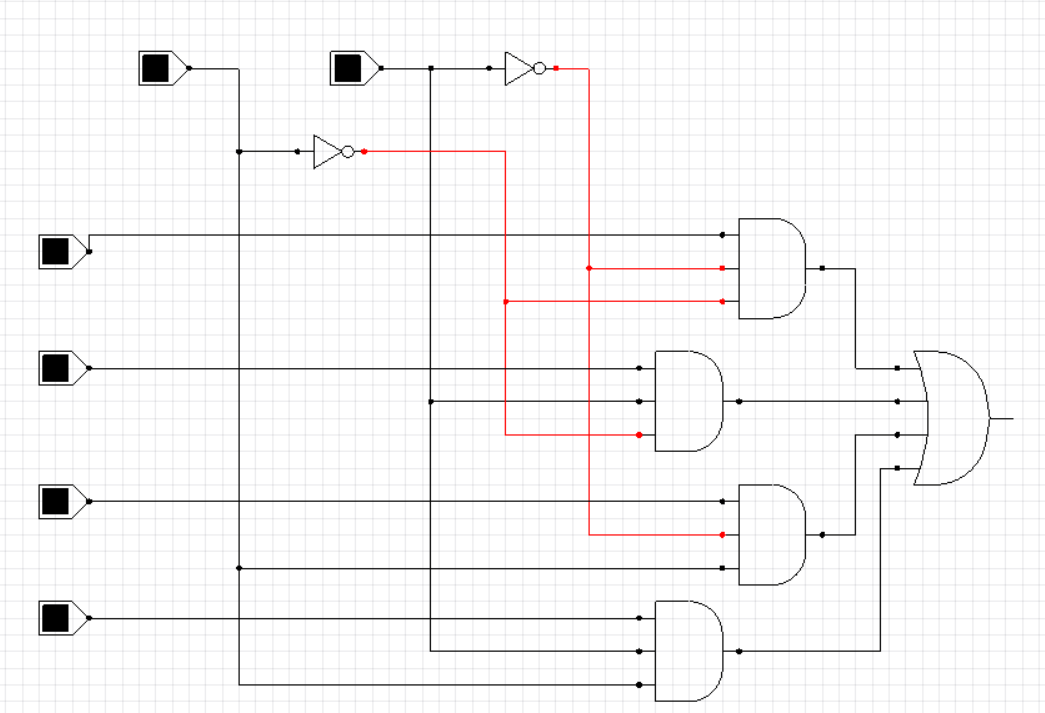
**OR16**

Realiza una comprobación Or entre dos buses de entrada de 16 bits y devuelve un bus de 16 bits representando el resultado.



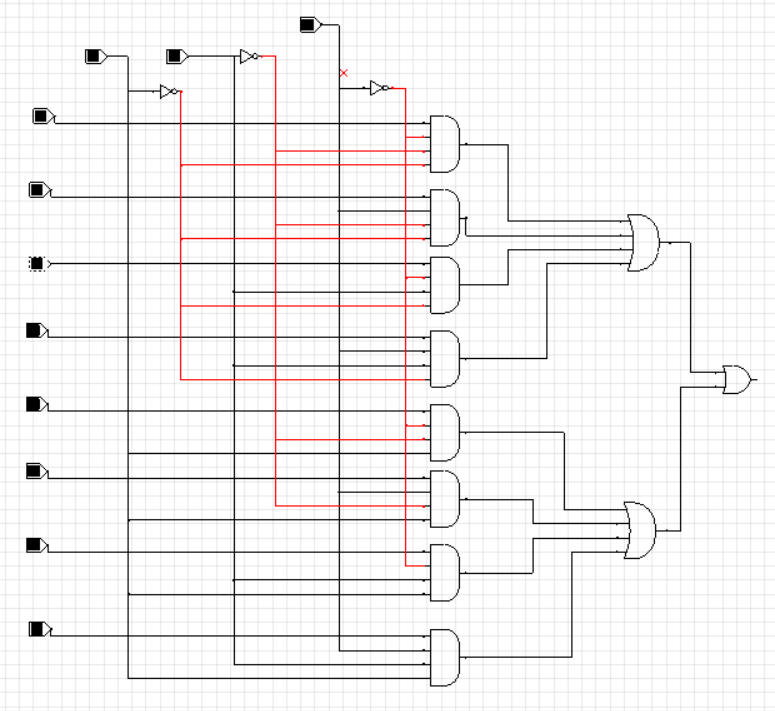
**Mux4Way16**

El multiplexor de cuatro entradas tiene un selector de dos bits y devuelve el valor de la primera entrada cuando los bits selectores valen 00, la segunda cuando los selectores valen 01, la tercera cuando los selectores valen 10 y la cuarta cuando los selectores valen 11

****

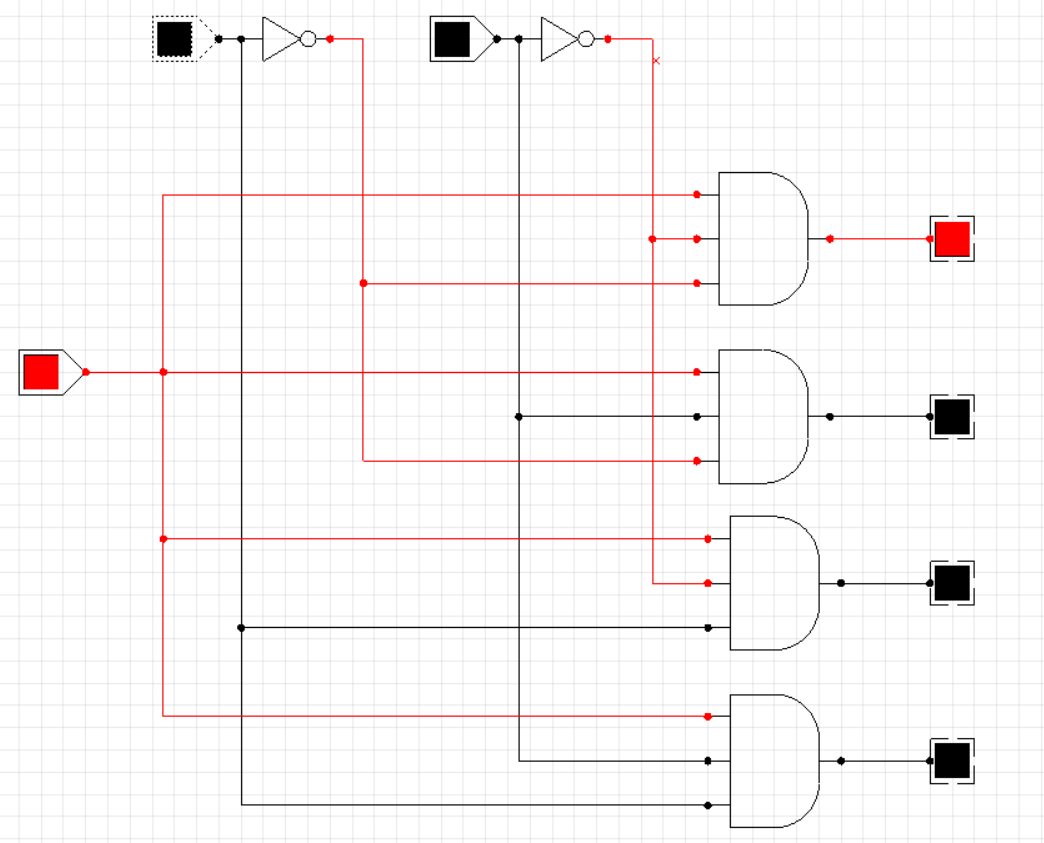
**Mux8Way16**

El multiplexor de ocho entradas tiene un selector de tres bits y devuelve el valor de la primera entrada cuando los bits selectores valen 000, la segunda cuando los selectores valen 001, la tercera cuando los selectores valen 010, la cuarta cuando los selectores valen 011, la quinta cuando los selectores valen 100, la sexta cuando los selectores valen 101, la séptima cuando los selectores valen 110 y la octava cuando los selectores valen 111.

****

**DMux4Way16**

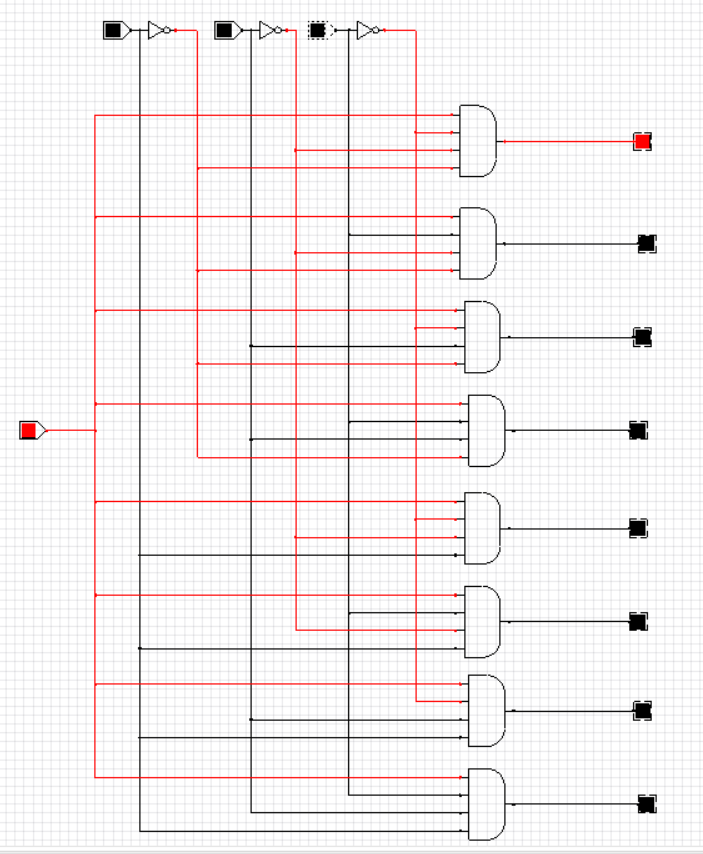
El de-multiplexor de cuatro canales recibe una única entrada y retransmite su valor a alguno de los cuatro canales de salida dependiendo del valor actual del bit selector.



En la imagen el bus de entrada está siendo retransmitido solo al canal más superior y a ningún otro dado que la configuración de los selectores es 00.

**DMux8Way16**

El de-multiplexor de ocho canales recibe una única entrada y retransmite su valor a alguno de los ocho canales de salida dependiendo del valor actual del bit selector.



En la imagen el bus de entrada está siendo retransmitido solo al canal más superior y a ningún otro dado que la configuración de los selectores es 000.