# 4. Operaciones lógicas / Función lógica

En el anterior capítulo hemos visto que es una tabla de la verdad y como se sacaba. Decíamos que la tabla de la verdad muestra todas las combinaciones posibles que se pueden dar de las entradas digitales para las salidas. Por otra parte, explicabamos la importancia que la variable binaria tiene en la electrónica digital.

En este apartado nos centraremos en las operaciones que podemos hacer con esas variables para obtener una función lógica, que básicamente son dos. La función de suma (cuando todas las entradas digitales son necesarias para activar la salida) y la función de multiplicación (donde una o varias entradas digitales, no todas, son necesarias para activar la salida).

Estas operaciones se resumen en la creación de las puertas lógicas, que hemos explicado anteriormente. Pero para entenderlo, lo haremos mucho más fácil.

Operaciones en el sistema binario

* pueden representar dos estados 1 y 0
* se representan por entradas y salidas

Imaginemos que tenemos dos entradas y una salida:

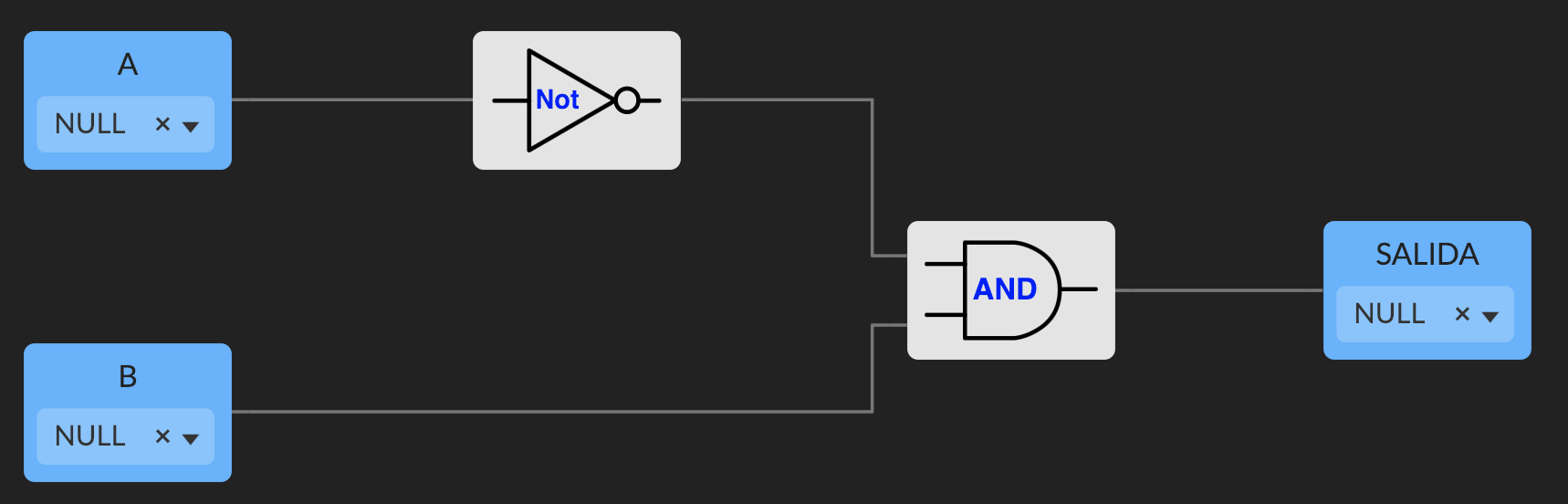
a - entrada 1 b - entrada 2 s - salida

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | SALIDA  (Función suma) | SALIDA  (Función multiplicación) |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 |

Como vemos en la tabla, para las entradas A y B tenemos diferentes salidas dependiendo qué función queramos darle. Al final, consiste en representar esta combinación entre entradas y salidas mediante álgebra, en una función lógica.

Sacar la tabla de la verdad de un circuito

Imaginemos el siguiente circuito:



para sacar la tabla de verdad lo primero necesitamos saber cuantas entradas y salidas hay. Sabemos que las entradas solo pueden tener dos estados: 1 y 0. Con eso, solo falta sacar el número de casos posibles que puede coger esas dos entradas.

= nº de casos posibles

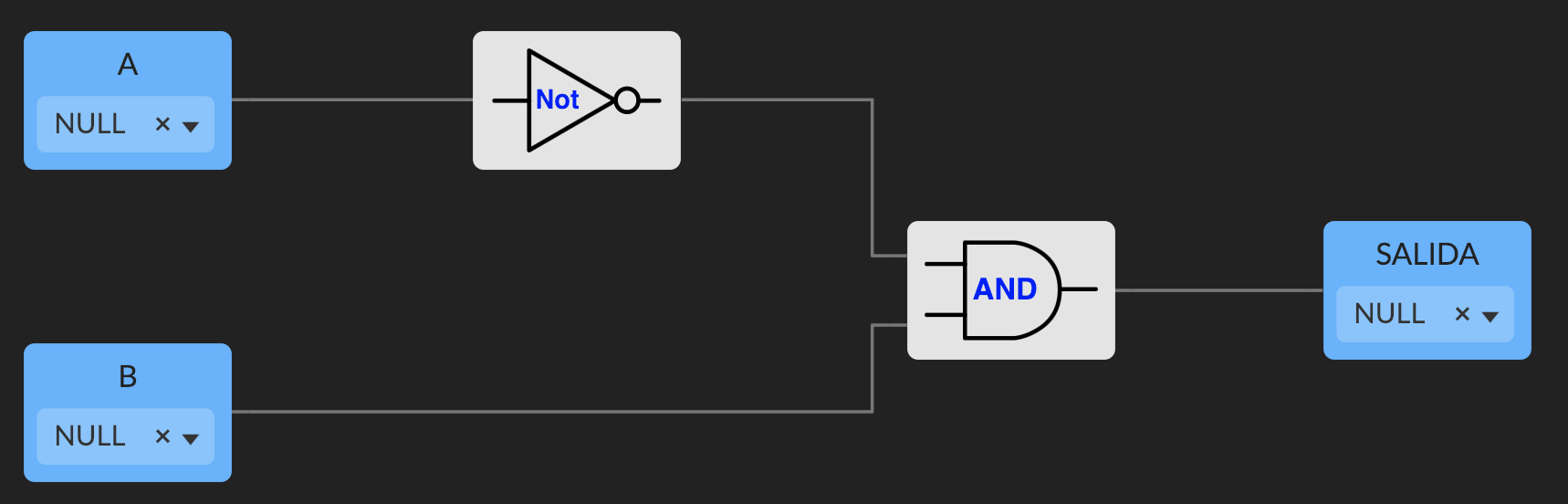
|  |  |  |
| --- | --- | --- |
| A | B | SALIDA |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

De esta sencilla forma obtenemos la tabla de la verdad partiendo de un circuito para poder sacar la función lógica.

Obtener la función lógica de un circuito

Para obtener la función lógica de un circuito, primero hay que saber cuáles son las puertas lógicas, cuales existen y cual es su función. Una vez sepamos identificarlas, podremos sacar la función desde el circuito.

Si seguimos con el ejemplo de anterior circuito:



Sacaremos la función a partir de las entradas y las puertas lógicas. Primero, tenemos dos entradas. Una se llama A y la otra B. Tenemos una puerta NOT, la cual invierte el valor de la entrada. Y por otro lado tenemos una puerta AND, la cual multiplica las entradas.

Funcion logica = SALIDA = A.B , donde A es negado

Obtener la función lógica desde la tabla de la verdad

Obtener la función lógica desde la tabla de la verdad es muy sencillo. Es conveniente saber algunos términos antes de realizar cualquier operación. En la tabla de la verdad, por cada combinación de entradas posible, se genera una salida. A esa salida le llamamos producto.

Para crear la función lógica partiendo desde la tabla de la verdad, tenemos dos opciones:

* La primera que llamamos minterm, suma de productos, con una lógica positiva suma todos los productos de 1.
* La otra, maxter, producto de sumas, con una lógica negativa suma todos los productos de 0.

Podemos utilizar cualquiera de las opciones para sacar la función, eso sí, la función será igual independientemente de cual utilizamos. Lo mejor, para sacar una función resumida, es utilizar el mínimo de productos que haya.

Con el ejemplo anterior:

|  |  |  |
| --- | --- | --- |
| A | B | SALIDA |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Función lógica (suma de productos) : A.B + A.B + A.B

Función lógica (producto de sumas) : A+B

Como vemos en el ejemplo, cuando utilizamos la suma de productos, tenemos en cuenta todos las salidas en 1. Por el contrario, en producto de sumas, tenemos en cuenta las salidas a 0.

Obtener la tabla de la verdad desde la función lógica

Es justo la acción contraria a lo que acabamos de ver. En el ejemplo anterior, utilizando la función de suma de productos, lo primero sería saber de cuántas entradas se compone la función. A y B. Después, basándonos en la función, sacar las salidas a 1 posibles. Estos serán los productos. Las demás serán 0.

Obtener el circuito lógico a partir de la función lógica

En este último paso veremos lo más sencillo, que es sacar el circuito a partir de la función.

Para ello nos basaremos en todo lo que hemos visto anteriormente. Sabemos que las funciones principales son la negación, la suma y la multiplicación. Simplemente tenemos que representarlas mediante puertas lógicas.

Método de Karnaugh

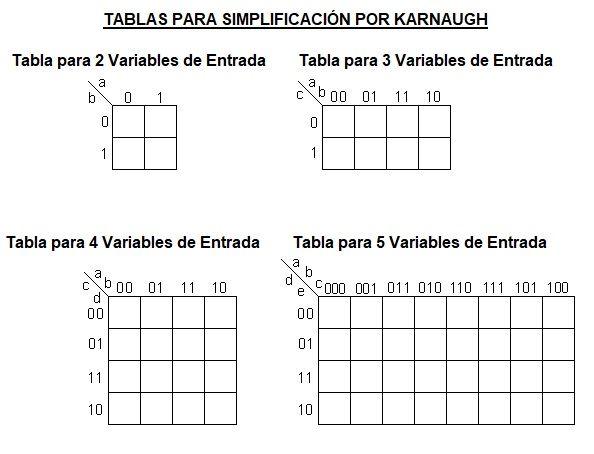
No podemos hablar de funciones lógicas sin tener en cuenta el método de Karnaugh. ¿Por qué de su importancia? A veces, la función lógica que obtenemos es muy larga o muy compleja. Es por eso que mediante este método podemos simplificar bastante esa función y sacar un circuito más reducido y eficiente.

¿Cómo funciona? Hay que seguir los siguiente pasos para hacer la simplificación:

1. De la tabla de la verdad - Sacaremos el mapa de Karnaugh
2. Agrupamos los productos lo máximo posible
3. Los nombraremos y sumaremos
4. Casos en los que el producto no está definido - Don't Cares

1 - Mapa de Karnaugh

Para sacar el mapa tendremos en cuenta todas las salidas posibles que obtenemos de la tabla de la verdad. Recordad que el número de salidas posibles es = nº de casos posibles.

Dependiendo del número de entradas y de salidas posibles que tengamos, obtendremos las dimensiones de la tabla:

Con el siguiente ejemplo lo veremos mucho más fácil:

Tabla de la verdad

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | SALIDA |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mapa de Karnaugh | | CD |  |  |  |
| 00 | 01 | 11 | 10 |
| AB | 00 | 0 | 1 | 0 | 1 |
|  | 01 | 0 | 1 | 0 | 0 |
|  | 11 | 0 | 0 | 1 | 0 |
|  | 10 | 0 | 0 | 0 | 1 |

2 - Agrupar los productos

Una vez que obtenemos el mapa, el siguiente paso es agrupar los productos en grupos lo mayor grandes posibles. Para ello, es fundamental cumplir las siguientes reglas:

* Coger 1 que no haya cogido antes
* Agruparlos
  + No pueden ser grupos en diagonal
  + Tienen que ser grupos de 2, 4, 8, o 16

Pondremos dos ejemplos para poder verlo mejor :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mapa de Karnaugh | | CD |  |  |  |
| 00 | 01 | 11 | 10 |
| AB | 00 |  | 1 |  |  |
|  | 01 | 1 |  |  |  |
|  | 11 | 1 |  | 1 | 1 |
|  | 10 |  |  | 1 | 1 |

Como vemos, en este mapa, hemos podido agrupar los productos en tres grupos. Uno de cuatro, otro de 2 y un 1 se nos ha quedado solo. Pero veamos otro ejemplo:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mapa de Karnaugh | | CD |  |  |  |
| 00 | 01 | 11 | 10 |
| AB | 00 | 1 |  |  | 1 |
|  | 01 | 1 |  | 1 | 1 |
|  | 11 | 1 |  | 1 | 1 |
|  | 10 | 1 | 1 |  | 1 |

En este caso, los hemos podido agrupar en grupos de 8, 4 y 1.

3 - Nombrarlos y sumarlos

Para finalizar, nombraremos cada uno de los grupos que hemos formado y los sumaremos. Recordar que cuanto más grandes sean los grupos mejor será a la hora de nombrarlos y sumarlos. La función que obtengamos estará más simplificada.

¿Cómo los nombramos?

Para responder a esta pregunta nos fijamos en el mapa. Si vemos las columnas y filas, las hemos ordenado según las entradas. Estas son las coordenadas que utilizamos para nombrar a los grupos. ¿Como? Muy fácil. Las entradas que se mantengan constantes en los grupos serán las que lo nombren.

Para verlos mejor, seguiremos con el ejemplo anterior:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mapa de Karnaugh | | CD |  |  |  |
| 00 | 01 | 11 | 10 |
| AB | 00 |  | 1 |  |  |
|  | 01 | 1 |  |  |  |
|  | 11 | 1 |  | 1 | 1 |
|  | 10 |  |  | 1 | 1 |

Función lógica simplificada - Karnaugh - S = B.C.D + A.B.C.D + A.C

4 - Don't Care

¿Qué son los Don't Care? En una tabla de la verdad son casos que jamás van a ocurrir, o que el estado que cogan, 0 o 1, es indiferente o no afecta a la salida. Dicho mal y pronto, son casos que no nos importa el estado en que estén.

¿Cómo lo representamos en el mapa?

Aprovechamos la ventaja que nos da, es decir, como nos da igual si su estado es 1 o 0, los utilizaremos para en el mapa hacer los grupos lo más grandes posibles.

Un ejemplo:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mapa de Karnaugh | | CD |  |  |  |
| 00 | 01 | 11 | 10 |
| AB | 00 |  | 1 | x | 1 |
|  | 01 | 1 |  | 1 | 1 |
|  | 11 |  | x |  |  |
|  | 10 | 1 |  |  |  |

En este ejemplo, vemos como tenemos dos Dont Cares. Bien, pues podemos utilizar uno de ellos para hacer dos de los grupos más grandes y así, simplificar más la función.

Teorema de De Morgan

Es un método que se utiliza para simplificar una función lógica. Se basa en las siguientes dos afirmaciones:

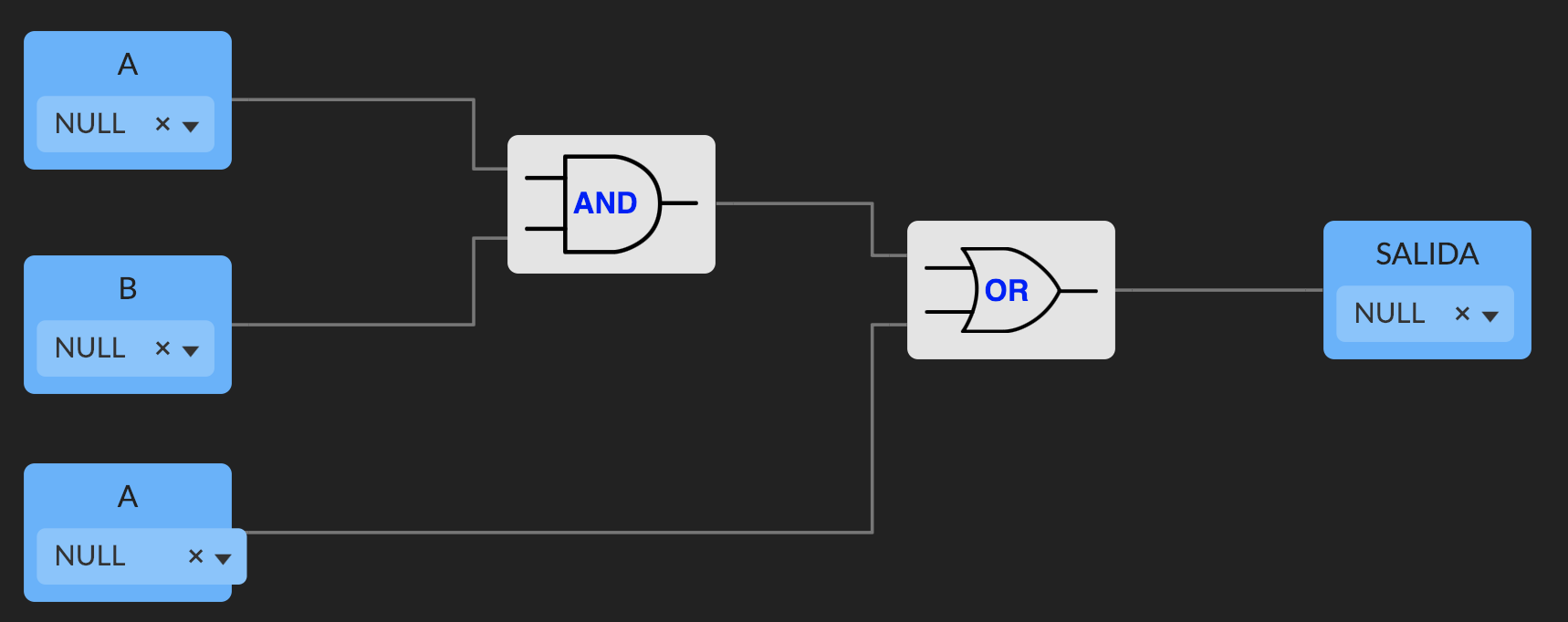
A . B = A + B

Para romper una función negada, solo basta con cambiar la operación

A + B = A . B

Simple y sencillo, no tiene más. Normalmente, este teorema es utilizado para convertir las funciones en puertas lógicas NAND y NOR. Mas que nada, por que con estos dos tipos de puertas, se puede construir cualquier circuito. Para entenderlo mejor, veremos el siguiente ejemplo:

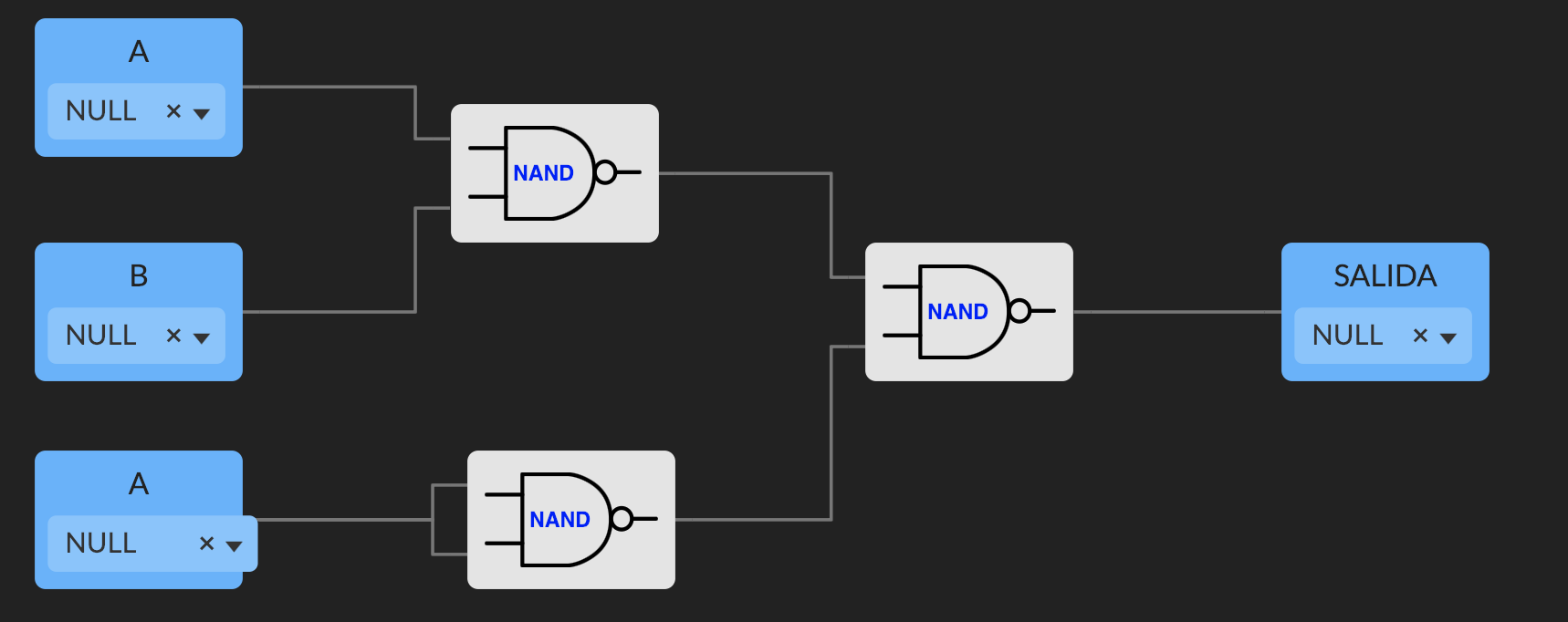
Imaginemos la siguiente función: F = (A . B) + C

Donde el circuito quedaría así :

Utilizando el teorema de De Morgan vamos a crear un inversor desde este mismo circuito utilizando sólo puertas NAND y NOR.

NAND

F = (A . B) + C = (A . B) + C = (A . B) . C



NOR

F = (A . B) + C = (A . B) + C = (A . B) + C

