

**Universidad Nacional de Colombia Sede Medellín**

**Arquitectura de Computadores**

**Sumador de 4 Bits Usando Compuertas NAND**

**Integrantes:**

**Hinara Pastora Sánchez Mata**

**Juan Jose Ospina Erazo**

**Sebastian Aguinaga Velasquez**

**Luzarait Cañas Quintero**

**Semestre 2022-1s**

## Suma de Binario

Para sumar dos números en binario, se debe tener en cuenta la suma y el acarreo de dicha operación; esta se puede representar por la siguiente tabla:

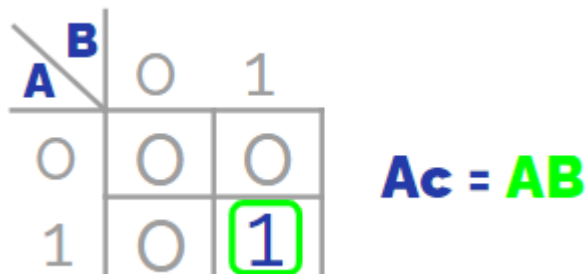
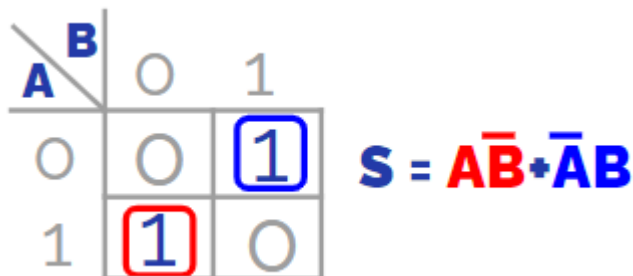
Tabla 1

A	B	Suma	Acarreo
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Acá vemos que el acarreo se genera cuando estamos sumando 1 mas 1, ya que esto nos da cero pero debemos llevar un 1 hacia la próxima 'columna'.

## Semisumador

A partir de la tabla 1 podemos realizar un semisumador (Half adder) y para ello primero realizaremos los mapas de karnaugh correspondientes a la suma y al acarreo



Con los cuales luego de aplicadas las leyes de Morgan y de involución para convertirlo a compuertas NAND, obtendremos:

$$S = A\bar{B} + \bar{A}B$$

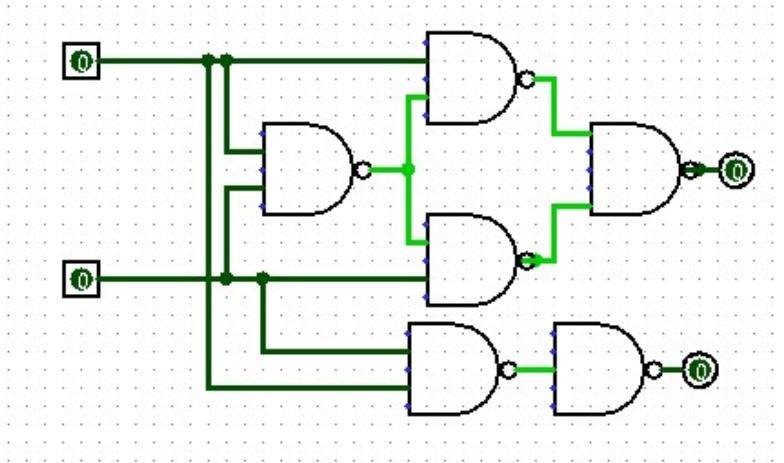
$$= \overline{(A \cdot \bar{A}B) \cdot (B \cdot \bar{A}\bar{B})}$$

$$Ac = AB$$

$$= \overline{\overline{AB}}$$

A partir de estos resultados implementaremos el circuito lógico correspondiente al semisumador:

### Circuito Half Adder



### Sumador completo

Para el sumador completo (Full adder) necesitaremos una nueva tabla de verdad que contemple los acarreo que se llevarán, para ello construimos la tabla de verdad para la suma:

Tabla 2

Entradas			Salidas	
A	B	C-in	Suma	C-out
0	0	0	0	0
0	0	1	1	0
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
---	---	---	---	---

En la tabla 2, A y B son los números que queremos sumar mientras que C-in es el acarreo inicial y C-out es el acarreo resultante.

Al realizar la Tabla 2 podemos construir el mapa de Karnaugh para la suma a continuación

### Mapa de Karnaugh para la Suma

BC A		00	01	11	10
0	0	0	1	0	1
1	1	1	0	1	0

$$S = A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC + \bar{A}B\bar{C} = (A \oplus B) \oplus C$$

Cuando creamos los grupos de unos, obtenemos como resultado final la ecuación de arriba. La cual simplificamos, haciendo uso de las leyes de Morgan y de involución hasta obtener la expresión de color azul oscuro.

También podemos construir el mapa de Karnaugh para el acarreo:

### Mapa de Karnaugh para el Acarreo

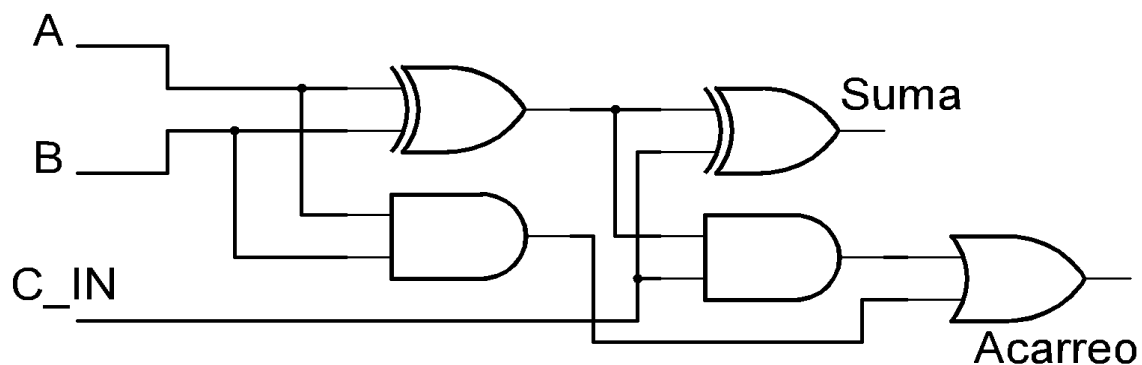
<b>A</b> \ <b>BC</b>	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$Ac = AC + AB + BC = AB + C(A \oplus B)$$

En este mapa, vemos que podemos hacer tres grupos cada uno compuesto por dos unos. La solución es la que se escribe en la ecuación de arriba, la cual se simplifica nuevamente hasta llegar a la expresión de azul.

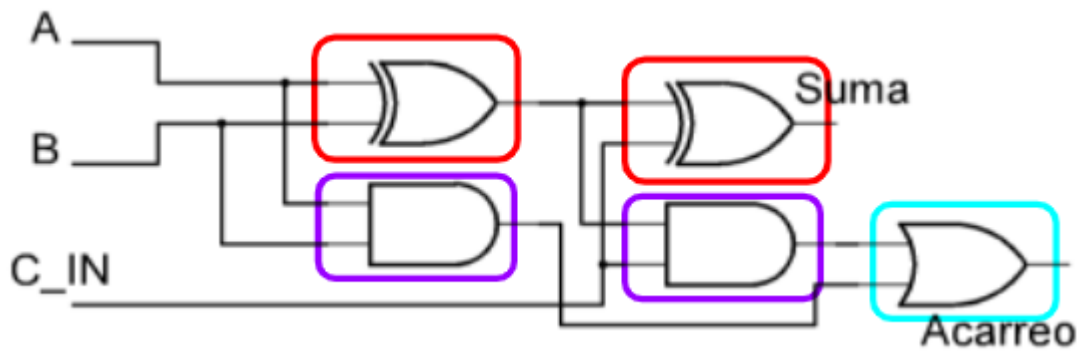
A partir de estos resultado podemos generar el circuito correspondiente para el full adder:

#### Circuito Full Adder



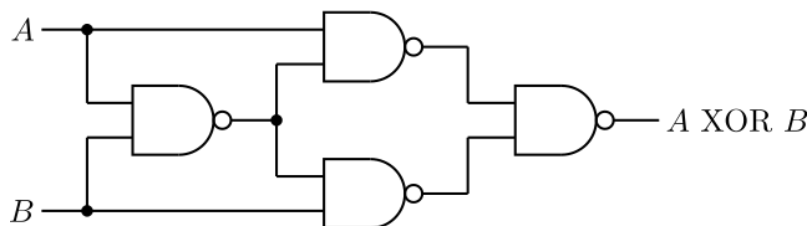
:

En este circuito podemos observar diferentes tipos de compuertas, entre ellas AND, XOR y OR, para un mejor entendimiento éstas se verán mejor identificadas con los colores morado, rojo y azul respectivamente, como se observa en la siguiente imagen:

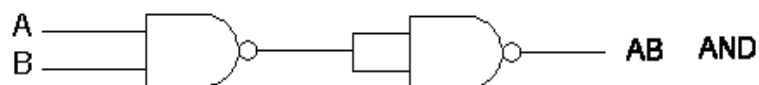


Para llegar a nuestro objetivo que es realizar un sumador completo con 4 bits usando sólo compuertas NAND, realizamos una conversión de la siguiente forma:

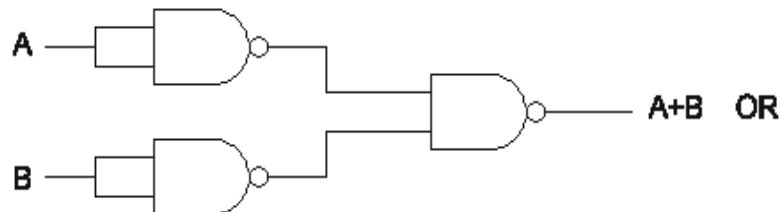
- Las compuertas XOR (Rojas) fueron convertidas a



- Las compuertas AND (Moradas) fueron convertidas a

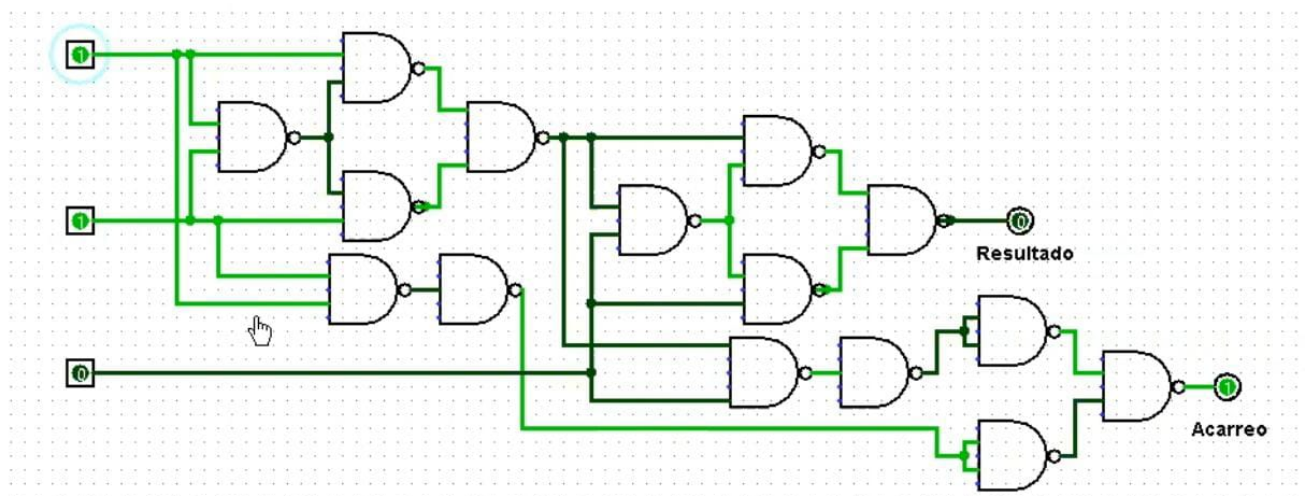


- Las compuertas OR (Azul) fueron convertidas a



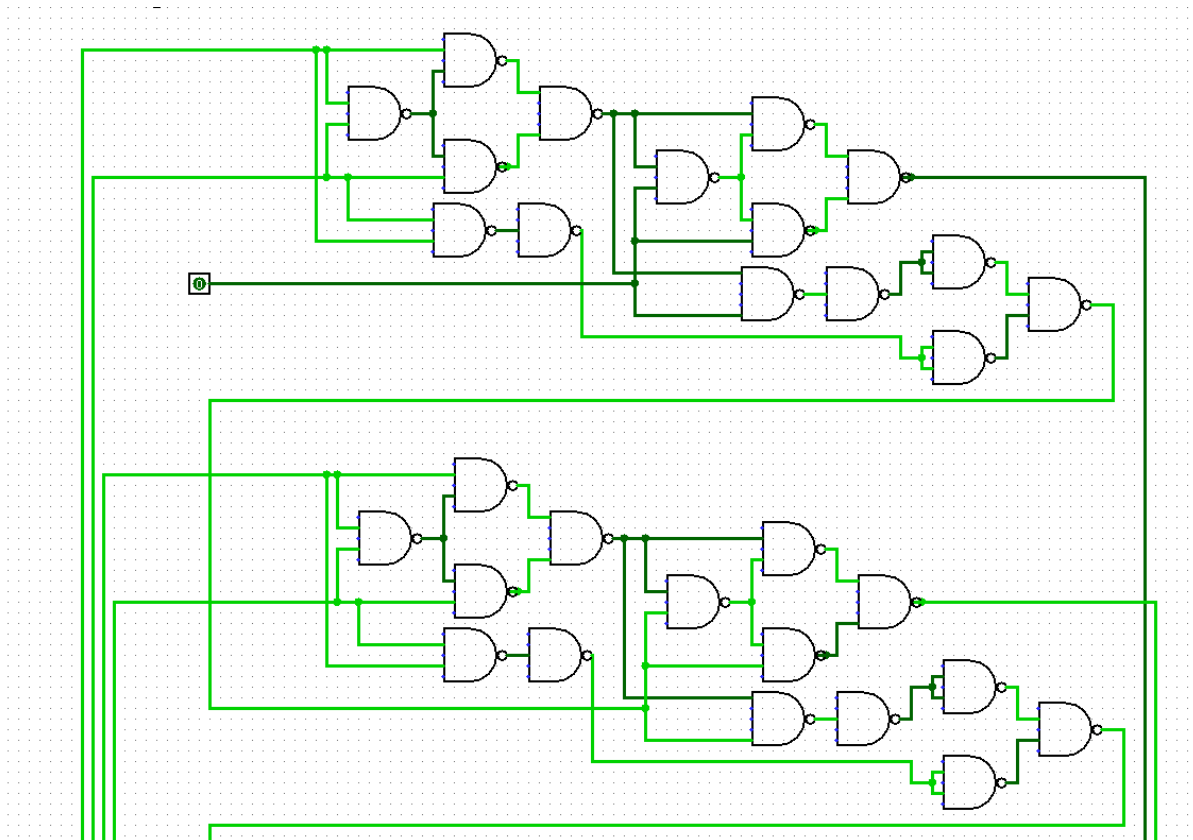
Y así obtuvimos nuestro sumador completo hecho únicamente con compuertas NAND cuyo circuito se puede ver a continuación:

### Circuito 1 bit



### Circuito de 4 bits.

A partir del circuito para sumar un solo bit podemos construir el sumador de 4 bits, para ello debemos tener en cuenta el acarreo que genera la suma del bit anterior, por esta razón el circuito empieza a operar desde el bit menos significativo, donde inicialmente el acarreo es cero (es decir que este no posee corriente como vemos en la figura), hasta el bit más significativo.



Para sumar los demás bits que se deseen replicaremos el circuito principal de un solo bit cuantas veces se requiera, donde el bit anterior pasará corriente al bit siguiente si se produce un acarreo.

Para evitar un sobreflujo definimos un "Acarreo final" y añadimos un bit más a la salida del circuito, de esta manera el sumador tendrá la capacidad de representar resultados mayores a 15(1111) pudiendo así representar hasta una suma de 30(11110)

