# Práctica 2 - Lógica digital (parte 1)

**Ejercicio 1**

Recordar: XOR = ⊕, NAND = |, NOR = ↓,

1. La equivalencia es verdadera.

1. Distributividad de la conjunción
2. Inverso de la conjunción
3. Identidad de la disyunción
4. Distributividad de la disyunción

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **x** | **y** | **z** |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |

Como los valores de verdad de las columnas y no so iguales, entonce la equivalencia es falsa.

**Ejercicio 2**

1. Funciones totales en {0, 1} x {0, 1}:

- Existen 4 combinaciones posibles de entradas para funciones de dos variables booleanas: (0, 0), (0, 1), (1, 0) y (1, 1).

- Cada combinación puede mapearse a 0 o 1, lo que da un total de 24 = 16 funciones booleanas diferentes.

2. Formas de expresión en álgebra de Boole:

- Las variables booleanas (p, q) pueden tomar los valores 0 o 1.

- Las operaciones básicas (AND, OR, NOT) pueden combinar estas variables para generar cualquier función lógica.

3. Teorema de funciones booleanas:

- Según el teorema de completitud del álgebra de Boole, cualquier función booleana puede ser expresada como una combinación de las operaciones AND, OR y NOT. Esto se puede hacer utilizando formas canónicas, como la forma canónica disyuntiva (Sum of Products o suma de productos, SOP) o la forma canónica conjuntiva (Product of Sums o producto de sumas, POS).

4. Construcción de todas las funciones:

- Para cada una de las 16 funciones, se puede construir su expresión en álgebra de Boole:

- Para las funciones que devuelven 1 en ciertas combinaciones de entradas, se utilizan términos en la forma de productos (AND) de las variables o sus negaciones.

- Para las funciones que devuelven 0, se pueden expresar usando la operación OR para combinar las negaciones.

Dado que todas las funciones booleanas de dos variables pueden ser representadas mediante combinaciones de las operaciones de álgebra de Boole (AND, OR, NOT), podemos afirmar que sí, todas las funciones totales f : {0, 1} x {0, 1} → {0, 1} se pueden expresar usando fórmulas del álgebra de Boole.

**Ejercicio 3**

NOT:

AND:

OR:

1. Idempotencia de la conjunción
2. Doble negación
3. De Morgan

Queda demostrado que con el operador NAND se pueden representar todas las funciones booleanas.

NOT:

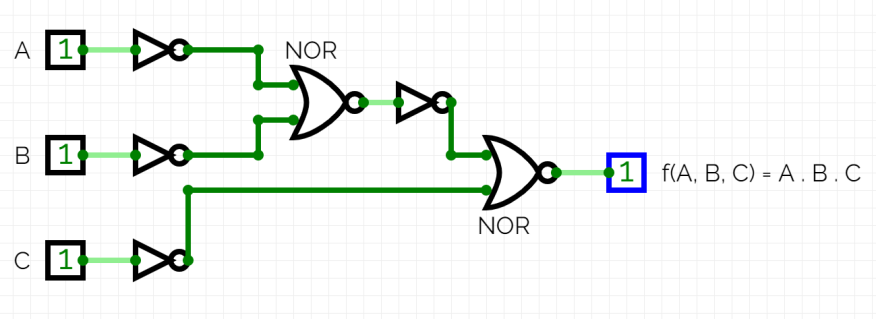
AND:

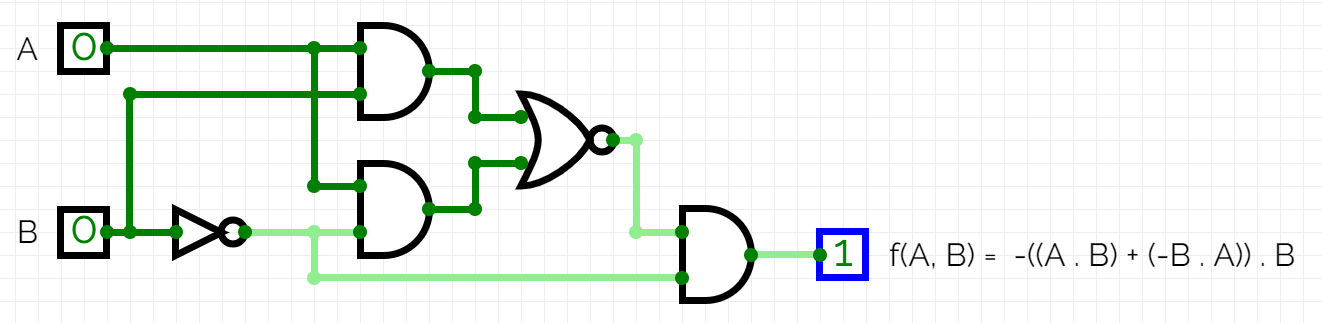
OR:

1. Idempotencia de la disyunción
2. De Morgan
3. Doble negación

Queda demostrado que con el operador NOR se pueden representar todas las funciones booleanas.

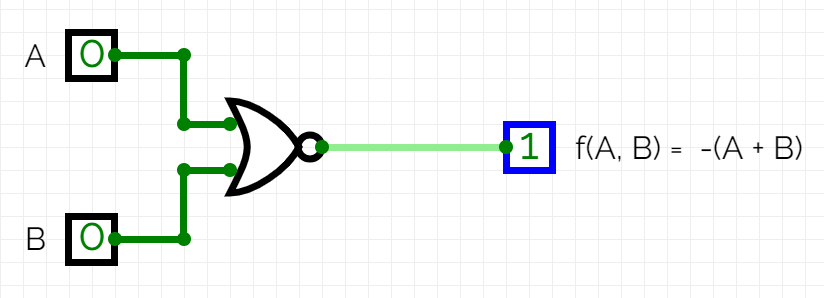
**Ejercicio 4**





La función solo devuelve 1 cuando A y B son 0. La función se puede simplificar:

1. De Morgan
2. Doble negación
3. Distributividad
4. Inverso
5. Idempotencia y conmutativa
6. Dominante
7. Identidad



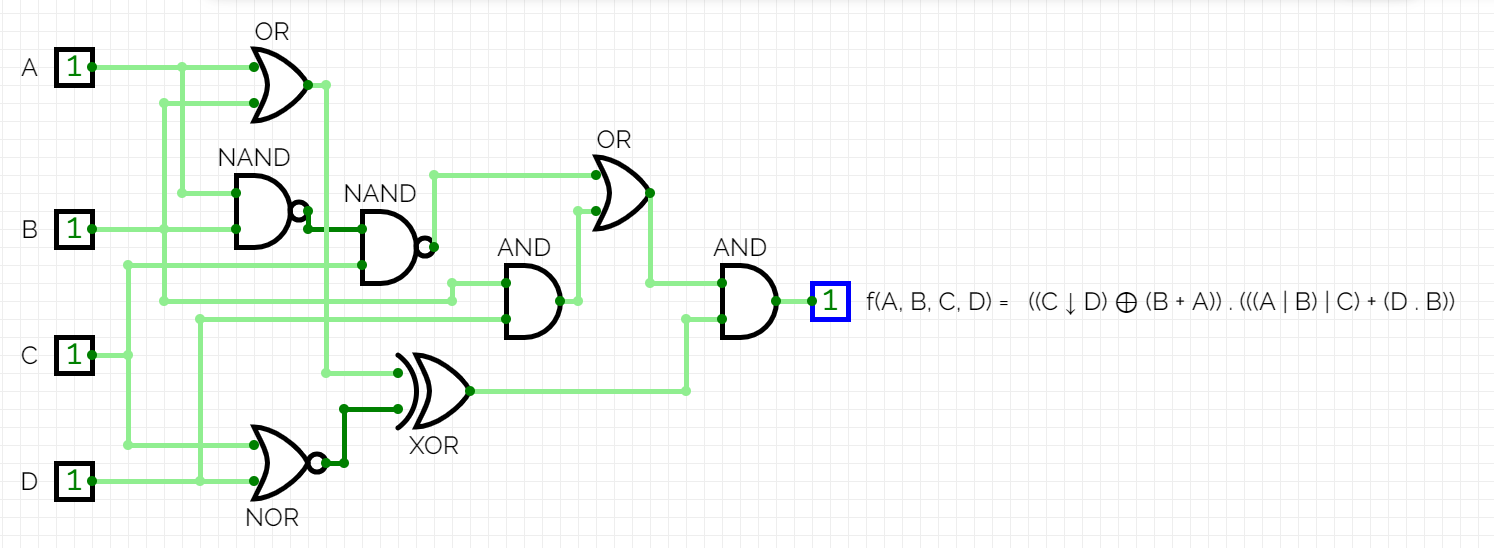


Tabla de verdad:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **C ↓ D** | **B + A** | **(C ↓ D) ⊕ (B + A)** | **A | B** | **(A | B) | C** | **D . B** | **((A | B) | C) + (D . B)** | **((C ↓ D) ⊕ (B + A)) . (((A | B) | C) + (D . B))** |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |

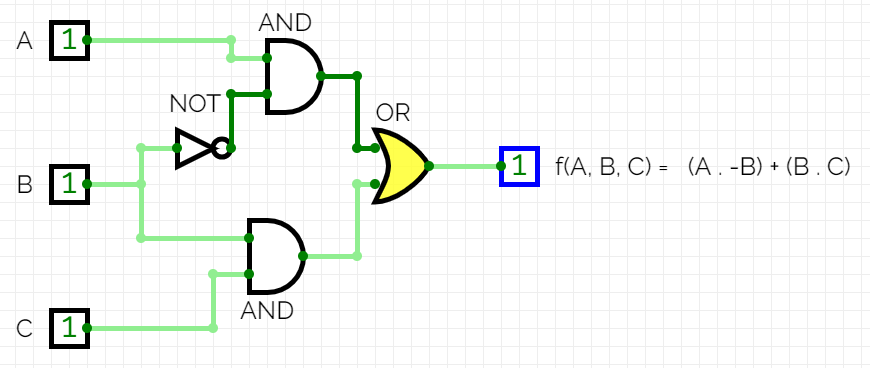
**Ejercicio 5**



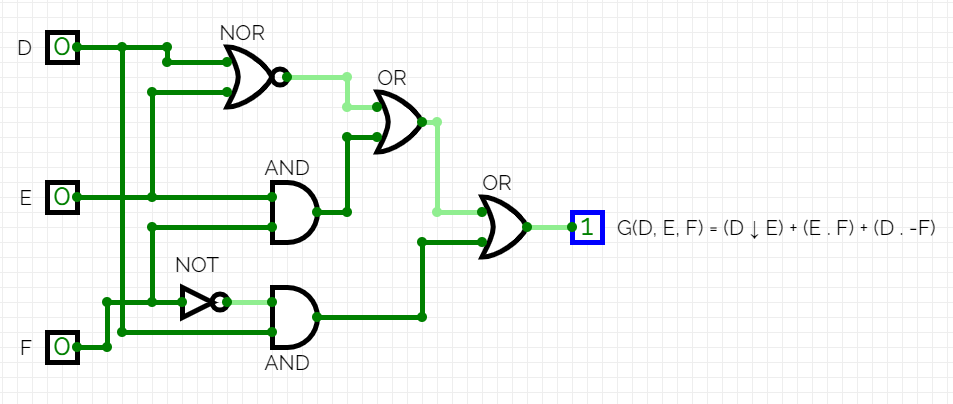
Para la función F se necesitaran 12 compuertas lógicas: 3 NOT, 3 OR, 6 AND

Para la función G se necesitaran 17 compuertas lógicas: 3 NOT, 5 OR, 9 AND

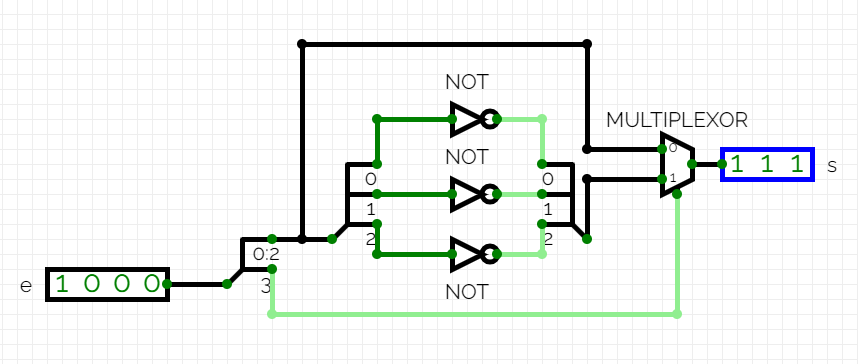
1. Distributividad
2. Inverso
3. Identidad



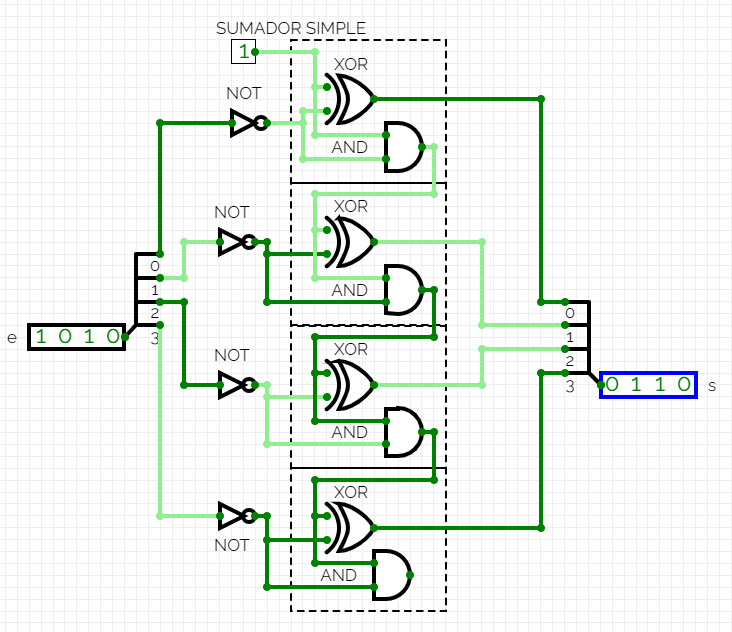
1. Distributividad y conmutatividad
2. Inverso
3. Identidad



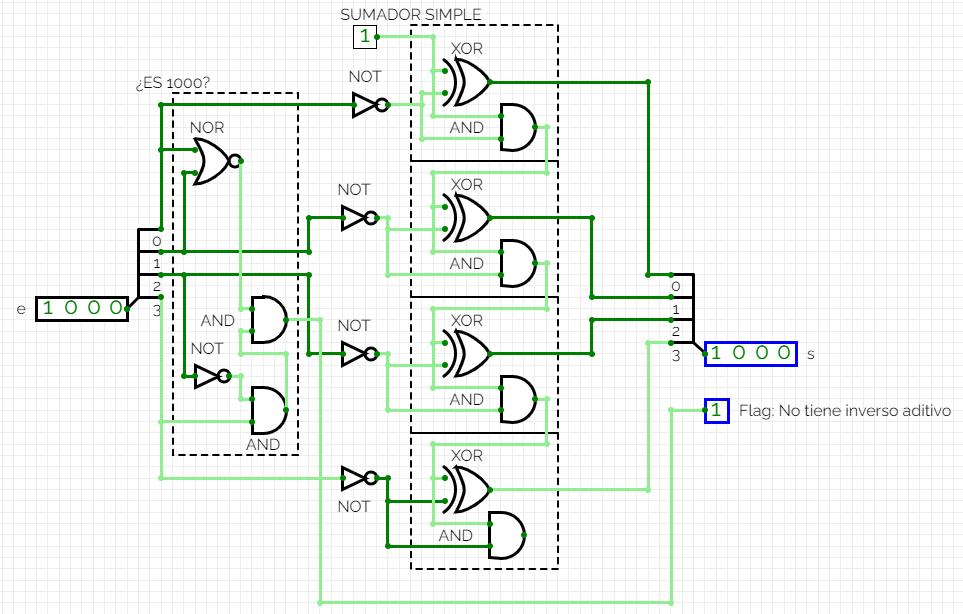
**Ejercicio 6**



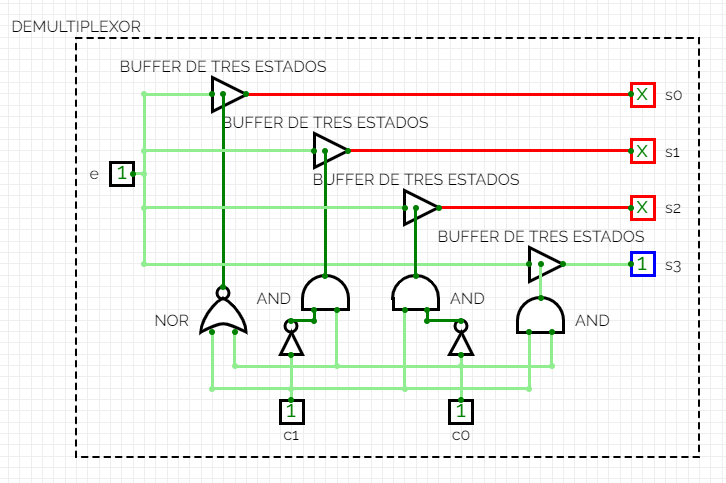
**Ejercicio 7**



1. Para 4 bits en notación de complemento a 2, el único valor que no tiene inverso aditivo es el (-8)10 = (1000)2 , ya que su inverso aditivo, el 8, y está fuera del rango de representación. RECORDAR: La suma entre inversos aditivos tiene que ser igual a 0.



**Ejercicio 8**

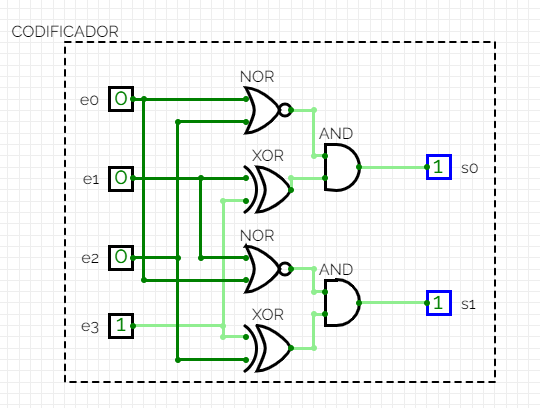


**Ejercicio 9**

a)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **e3** | **e2** | **e1** | **e0** | **s1** | **s0** |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |

1. Distributiva
2. Definición de XOR
3. Definición NOR



b)