**Universidad de El Salvador**

**Facultad Multidisciplinaria Oriental**

**Departamento de ingeniería y arquitectura**

**Tema:**

Compuertas Lógicas

**Grupo #2:**

Arana Díaz, Wilbert Alexander AD18005

Argueta Del Cid, Darwin Otoniel AD17008

Arteaga Segovia, Cristian Alexis AS18036

Ascencio Ramirez, Jim Kimpatrick AR19078

Ayala Bejarano, Miguel Ángel AB19015

Velasquez, Victoria Gabriela VV19020

**Asignatura:**

Sistemas Digitales

**Docente:**

David Alonso Mendoza Artiga

**Ciclo y año:**

01-2021

**Lugar y fecha:**

Universidad de el salvador Facultad Multidisciplinaria Oriental

Departamento de ingeniería y arquitectura.

Miércoles 21 de abril de 2021

**Introducción:**

El objetivo del presente trabajo es investigar acerca de las compuertas lógicas existentes, para que sirven, su simbología, tablas de verdad aplicación de estas en la creación de circuitos lógicos por medio de ejemplos realizados por los estudiantes para la mejor comprensión de las compuertas lógicas y métodos para la creación de circuitos lógicos. También queremos reforzar todos los conocimientos adquiridos en el horario de clases, acerca de las compuertas lógicas que son de mucha utilidad en la vida diaria, realizamos 3 ejemplos de las compuertas lógicas Compuerta AND. Compuerta OR .Compuertas NOT. Compuerta NOR, Compuerta NAND, Compuerta OR-exclusiva.

**Resumen**

En el presente trabajo se realizó una investigación acerca de los concentos de las compuertas lógicas AND, OR, NOT, NOR, NAND, OR-Exclusiva y el método del mapa de Karnaugh, realizando tres ejemplos de circuitos lógicos por cada una de las compuertas investigadas por el grupo de trabajo.

Se realizaron los ejemplos a partir de las clases impartidas de la materia de Sistemas y Procedimientos y a partir de lo investigado por el grupo de trabajo.

**Marco teórico**

**Multiplicación binaria** En general los sumadores que dispondremos en las computadoras son capaces de sumar dos números por lo tanto debemos adoptar el algoritmo e ir acumulando el resultado parcial y sucesivamente sumar el multiplicando corrido si el bit del multiplicador así lo indica.

**Compuertas universales** La compuerta NOR se dice que es una compuerta universal porque cualquier función booleana puede implementarse con ella. El procedimiento de diagrama de bloques para implementar funciones booleanas con compuertas NOR es similar al procedimiento para compuertas NAND:

**Circuito** Un **circuito** es una interconexión de componentes eléctricos (como baterías, resistores, inductores, condensadores, interruptores, transistores, entre otros) que transporta la corriente eléctrica a través de una trayectoria cerrada.

**Complemento lógico** es una operación sobre proposiciones, valores de verdad, o en general, valores semánticos. Intuitivamente, la negación de una proposición es verdadera cuando dicha proposición es falsa, y viceversa.

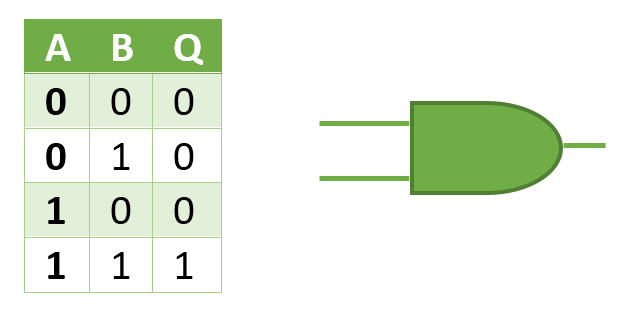
**Compuerta AND**

Para la compuerta AND, La salida estará en estado alto de tal manera que solo si las dos entradas se encuentran en estado alto. Por esta razón podemos considerar que es una multiplicación binaria.

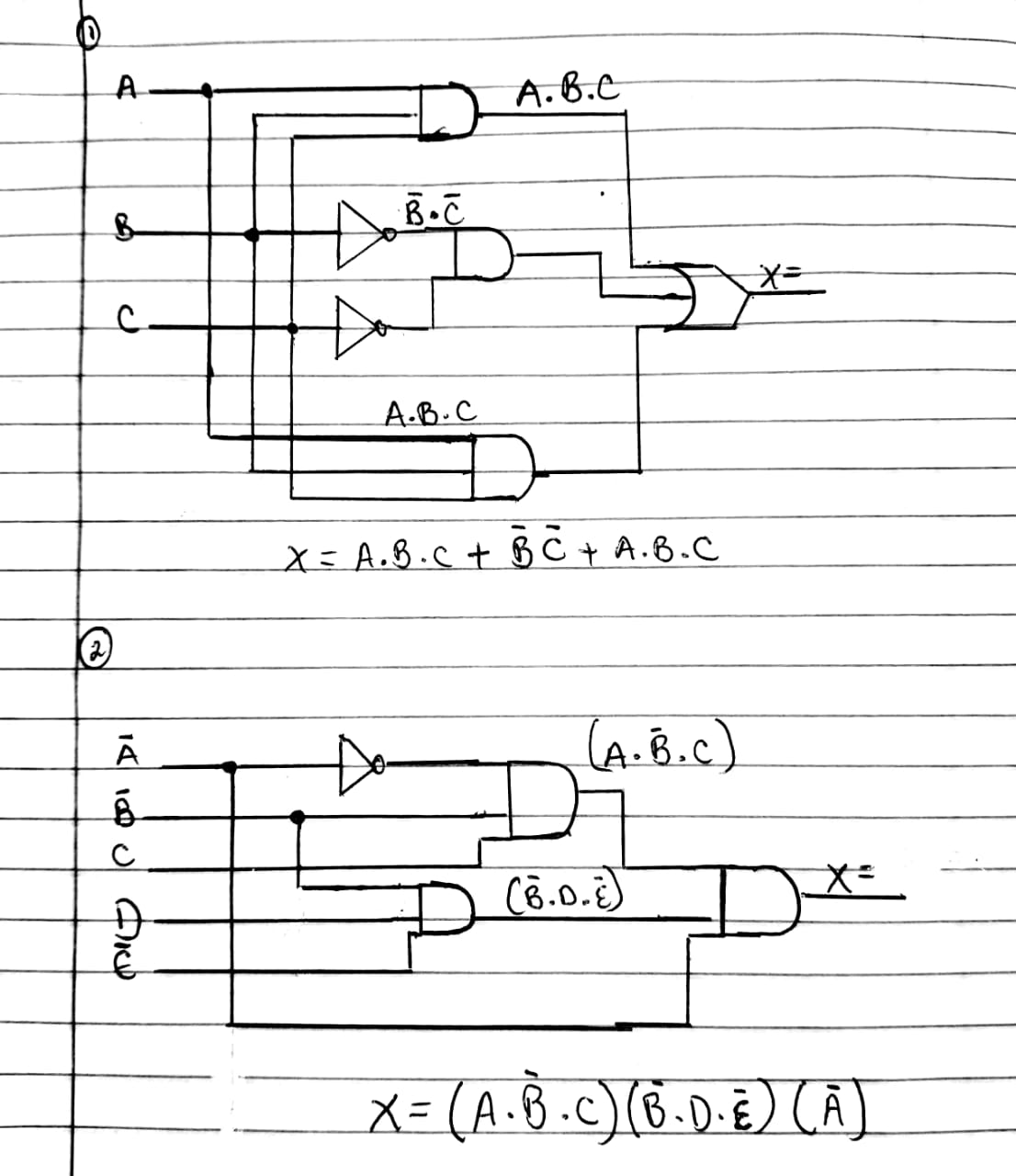
* Operación

Q=A.B

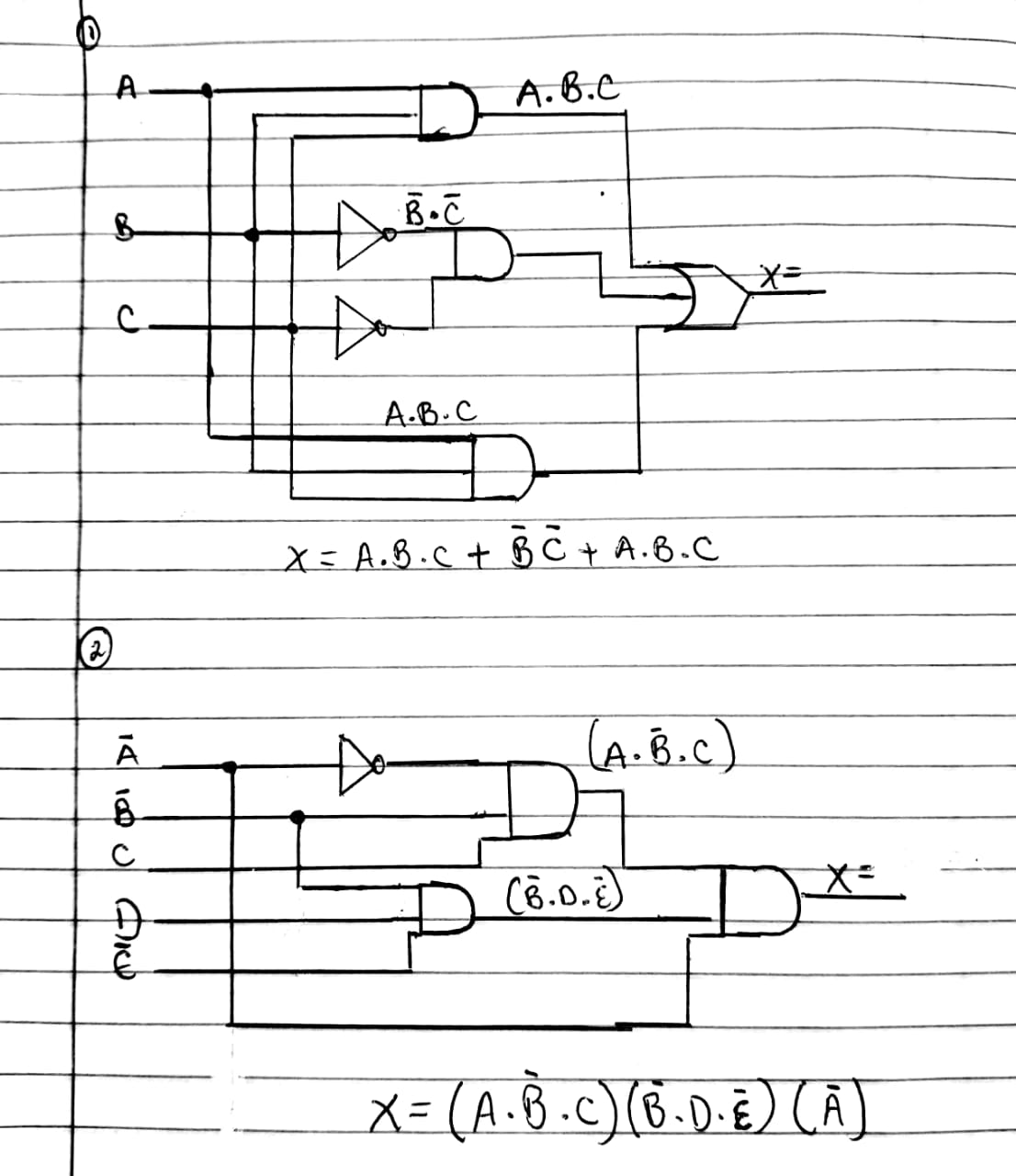
* Tabla de verdad y símbolo



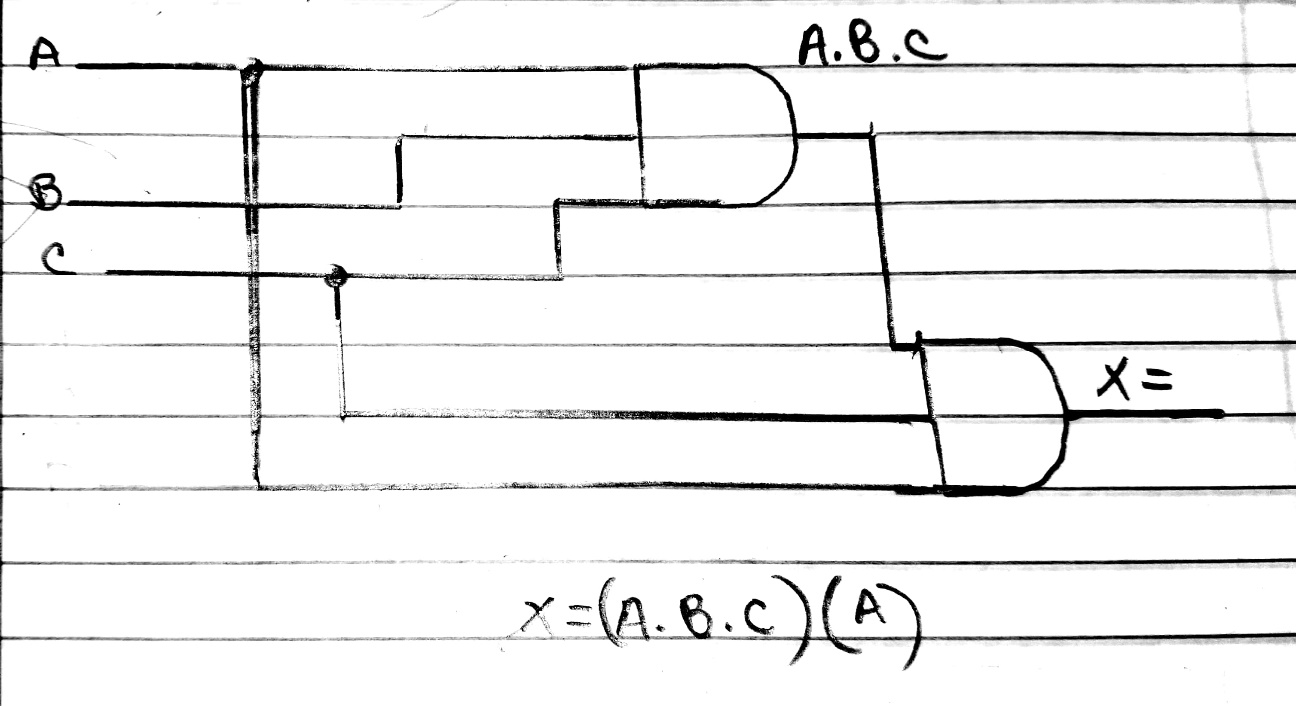
Ejercicio#1



Ejercicio#2



Ejercicio#3



**Tablas de Verdad**

AND (**A** \* **B**\***C**)

**Tablas de Verdad**

OR (**A + B**+**C**)

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | c | A + B+C |

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | c | A \* B\*C |
| a | b | c | (a,b,c) |
| b | c |  | (b,c) |

**Compuerta NOT**

Esta compuerta que también puede ser llamada compuerta inversora es una puerta lógica que es usada para una negación lógica. Puede ser indicada por medio de un apostrofe en el lado superior derecho de la variable

Ejemplo: X = 0, la negación lógica y como se representa seria X´ = 1.

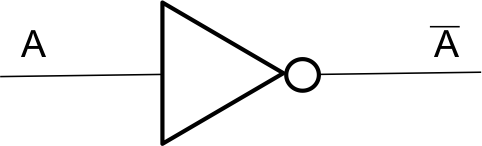
También se puede representar con una barra sobre la variable.

Ejemplo: X = 0, la negación lógica y como se representa seria

En una tabla de verdad se mostraría el resultado lógico inverso.

|  |  |  |
| --- | --- | --- |
| Ecuación | Entrada | Salida |
| A=A´  o  A = | 0 | 1 |
| 1 | 0 |

Su representación grafica es la siguiente

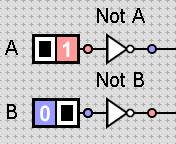


Ejercicios que utilizan la compuerta not:

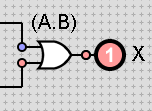
Ejercicio#1

1. Representar en circuito la siguiente expresión: X = Not A or Not B

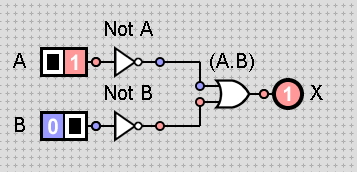
Cambiamos el encendido de la variable (A) ha apagado y el apagado de la variable (B) ha encendido.



Luego se multiplica A\*B



Quedaría de la siguiente forma:



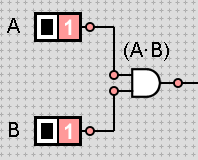
Y su tabla de verdad seria:

|  |  |  |
| --- | --- | --- |
| A | B | X |
| 0 | 0 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 1 | 0 |

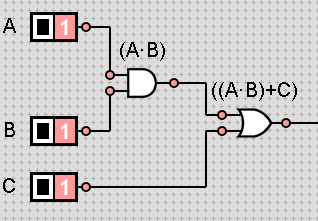
Ejecicicio#2

1. Representar en circuito la siguiente expresión: X = NOT ((A and B) or C)

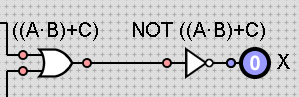
Multiplicamos A por B.



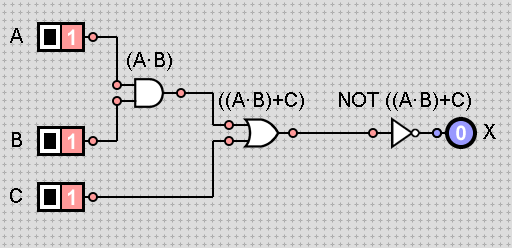
Luego le sumamos C “((A \* B) + C)”



Y negamos la expresión “NOT ((A´\* B) + C)”



El circuito completo seria:



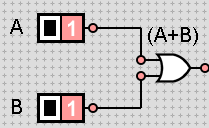
Su tabla de verdad seria:

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | X |
| 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |

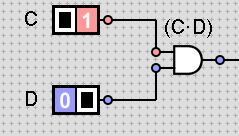
Ejercicio#3

1. Representar en circuito la siguiente expresión: X= Not((A or B) and (C and D))

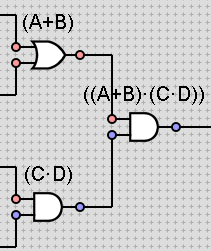
Primero sumamos (A+B)



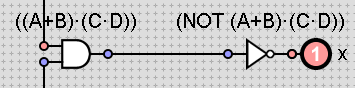
Segundo multiplicamos (C\*D)



Luego multiplicamos ((A+B) \*(C\*D))



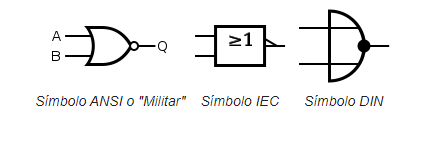
Por último, negamos todo (NOT (A+B) \*(C\*D))



**Compuerta NOR**

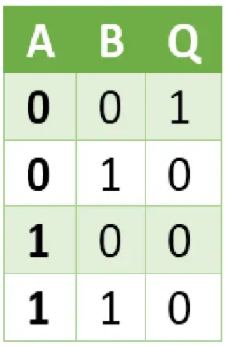
Las compuertas NOR son el complemento lógico de las compuertas OR, Esencialmente una OR negada. Las compuertas NOR y NAND son consideradas **compuertas universales** porque a partir de estas es posible construir todas las compuertas básicas (NOT, AND y OR).

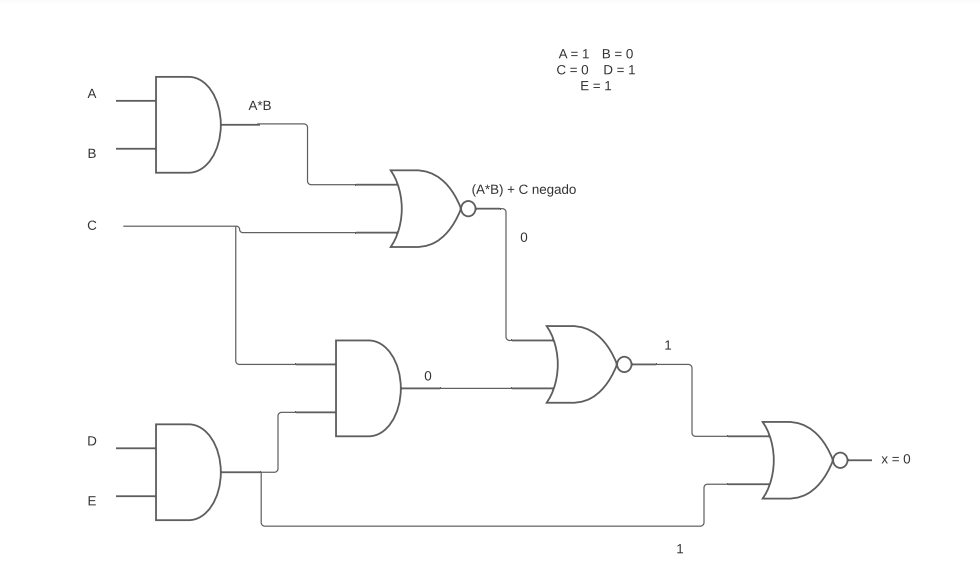
**Símbolos**

****

**Operaciones Q= (A+B)**

**Tabla de Verdad**

****

Ejercicio**#1**

**Tablas de Verdad**

AND (A\*B)

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **X** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

NOR ((A\*B) +C) negado

|  |  |  |  |
| --- | --- | --- | --- |
| **(A\*B)** | **C** | **(A\*B) + C** | **X** |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

AND (D\*E) +C

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **X** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

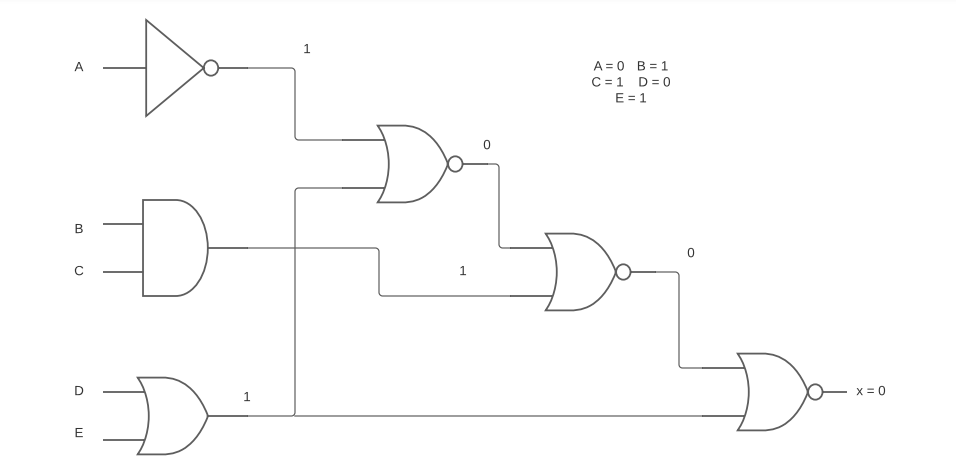
NOR ((A\*B + C) + (D\*E) +C)) negado

|  |  |  |  |
| --- | --- | --- | --- |
| **C** | **(D\*E)** | **C + (D\*E)** | **X** |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

NOR ((A\*B + C) + (D\*E) +C)) negado + (D\*E) negado

|  |  |  |  |
| --- | --- | --- | --- |
| **((A\*B + C) + (D\*E) +C)) negado** | **(D\*E)** | **((A\*B + C) + (D\*E) +C)) negado + (D\*E)** | **X** |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

**Ejercicio#2**



**Tablas de Verdad**

OR (D+E)

|  |  |  |
| --- | --- | --- |
| **D** | **E** | **D + E** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

NOR (D+E) + A negado

|  |  |  |  |
| --- | --- | --- | --- |
| **(D+E)** | **A** | (D+E) + A negado | **X** |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

AND (B\*C)

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **X** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

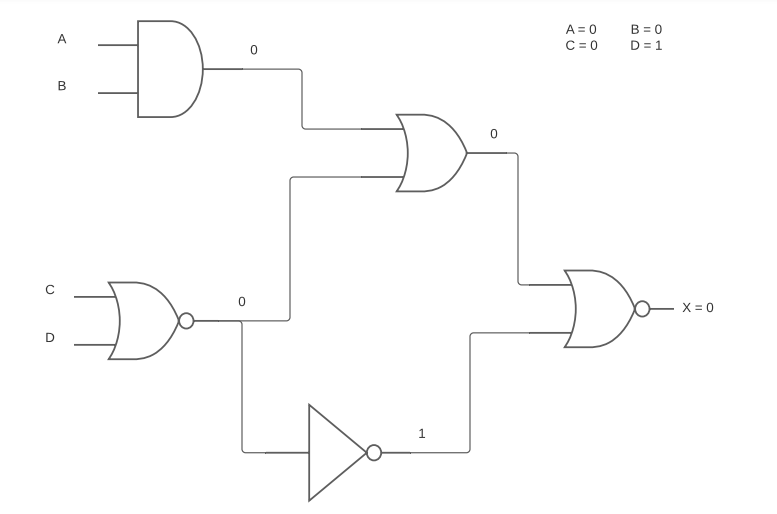
NOR (D+E) + A negado + (B\*C) negado

|  |  |  |  |
| --- | --- | --- | --- |
| **(D+E) + A negado** | **(B\*C)** | **((D+E) + A negado) + (B\*C) negado** | **X** |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

NOR ((D+E) + A negado + (B\*C) negado) + (D\*E) negado

|  |  |  |  |
| --- | --- | --- | --- |
| **(D+E) + A negado** | **(B\*C)** | **((D+E) + A negado) + (B\*C) negado** | **X** |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

**Ejercicio#3**



**Tablas De Verdad**

AND (A\*B)

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **X** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

NOR (C + D) negado

|  |  |  |  |
| --- | --- | --- | --- |
| **C** | **D** | **C + D** | **X** |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

OR ((C + D) negado + (A\*B))

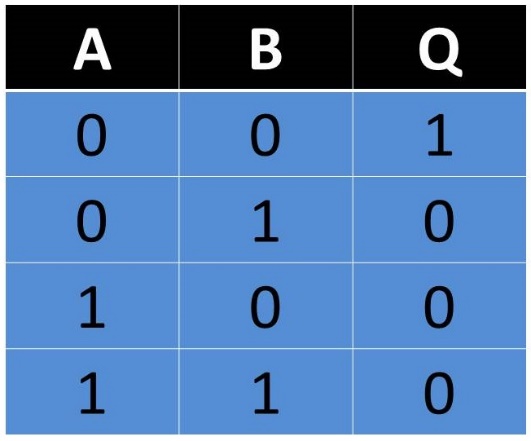
|  |  |  |
| --- | --- | --- |
| **(C + D) negado** | **A\*B** | **((C + D) negado + (A\*B))** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

NOR (((C + D) negado + (A\*B)) + (C\*D) negado)

|  |  |  |  |
| --- | --- | --- | --- |
| **C** | **D** | **C + D** | **X** |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

**Compuerta lógica NAND**

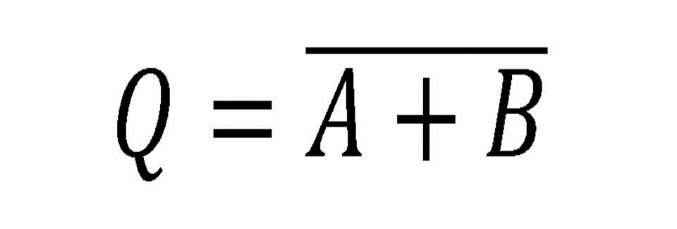
También denominada como AND negada, esta compuerta trabaja al contrario de una AND ya que al no tener entradas en 1 o solamente alguna de ellas, esta concede un 1 en su salida, pero si esta tiene todas sus entradas en 1 la salida se presenta con un 0.



TABLA

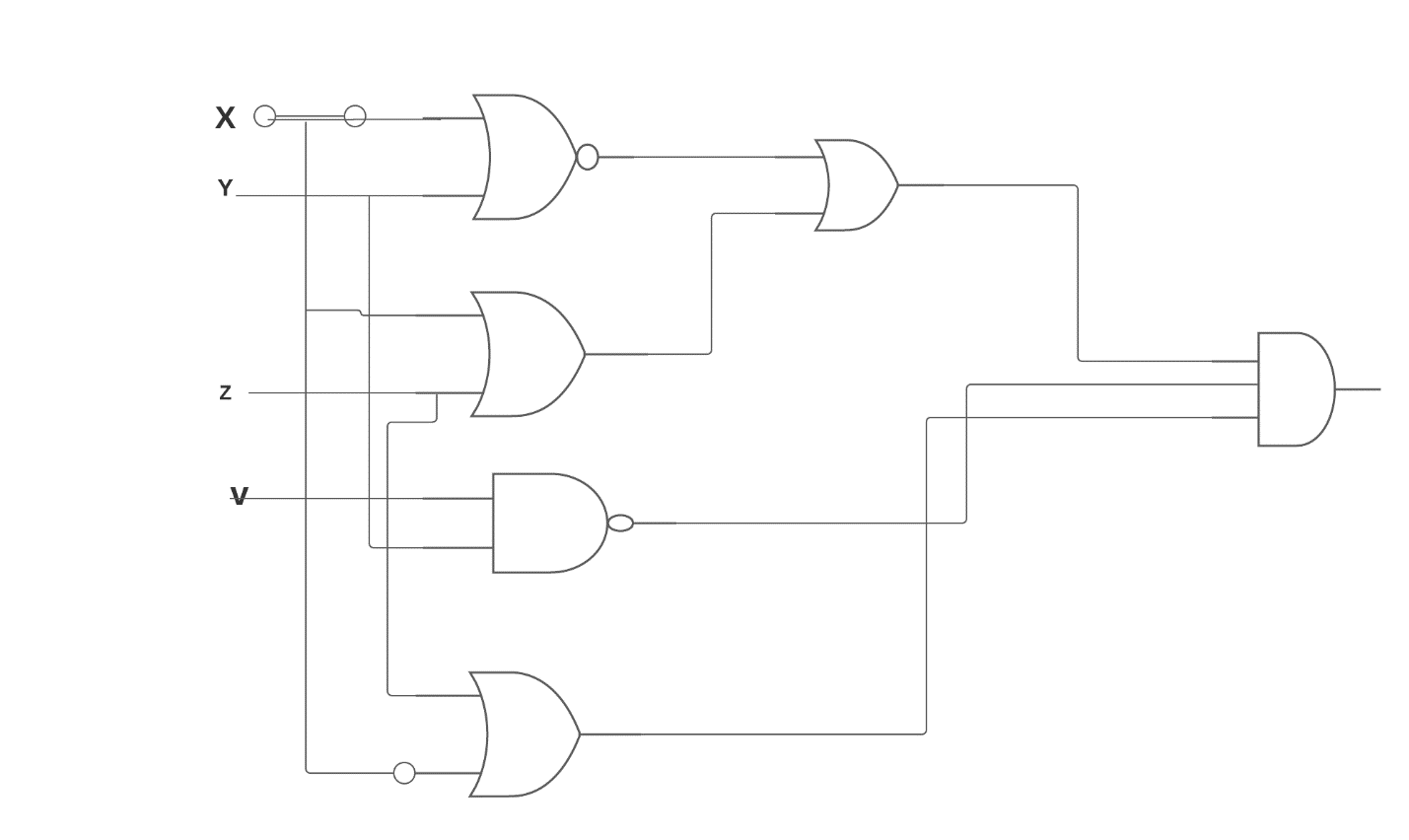


REPRESENTACION



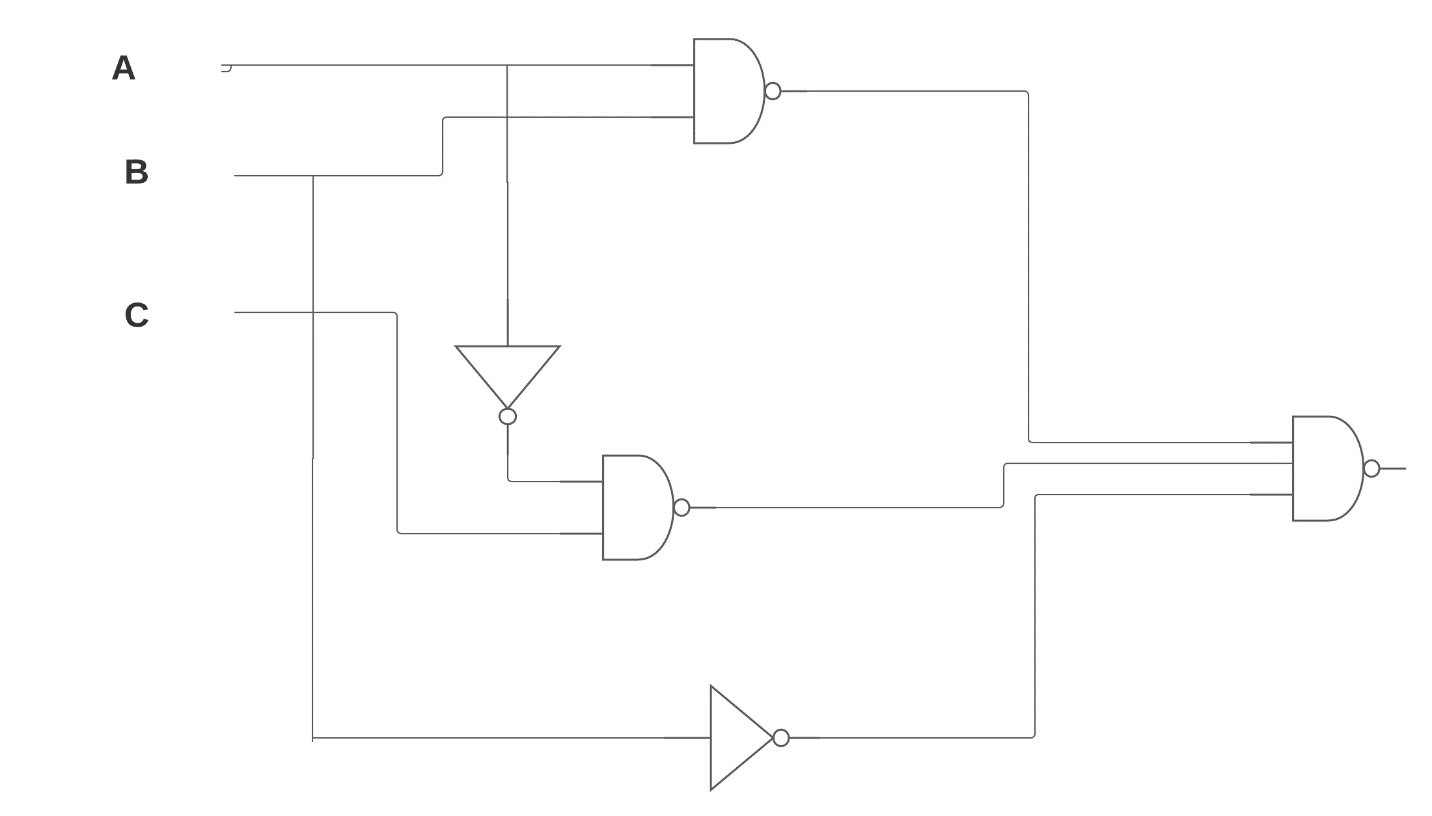
FORMULA

Ejemplo #1



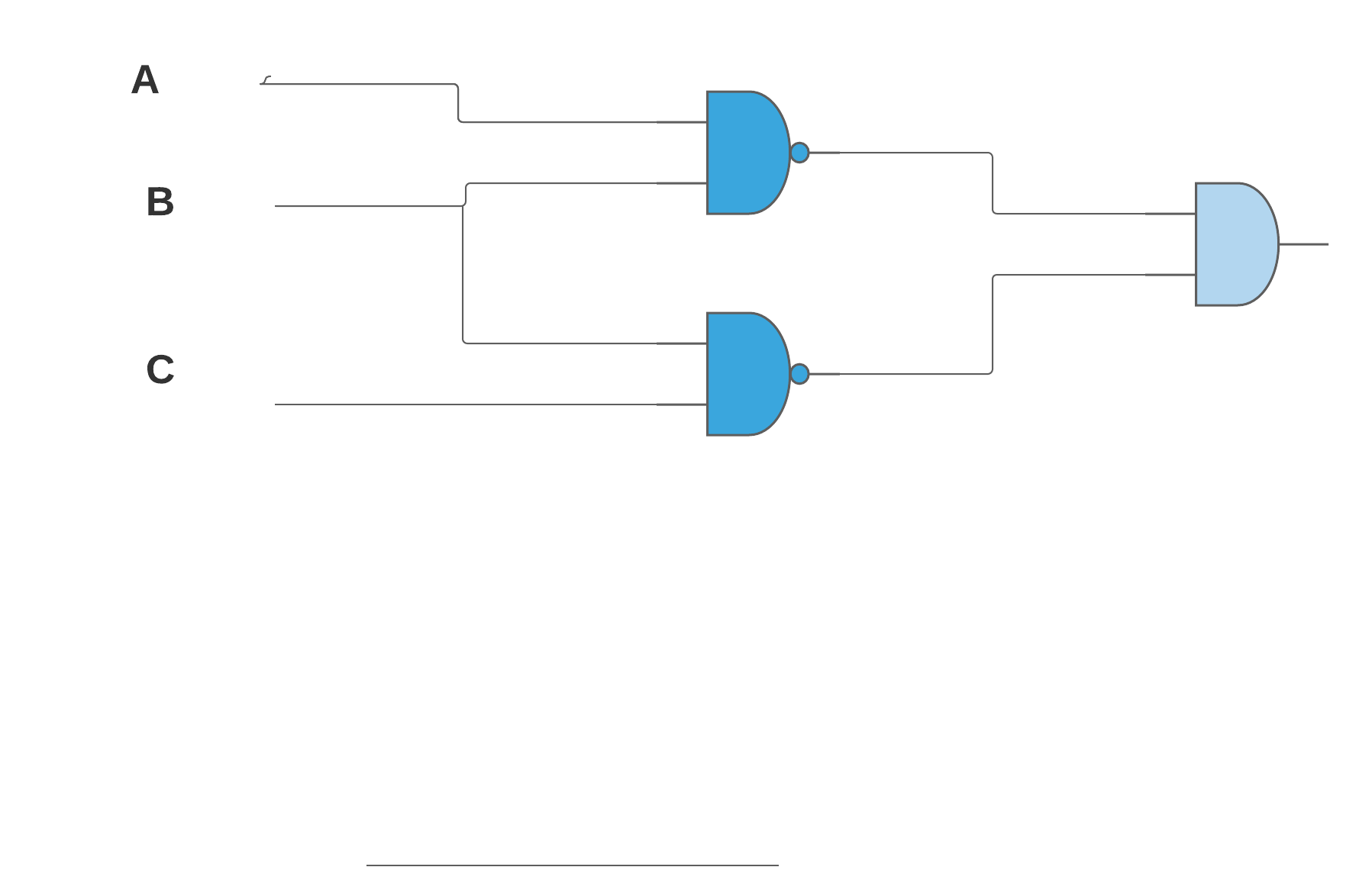
Resultado = [ + ( x + z) ] )

EJEMPLO #2



Resultado =

EJEMPLO #3



Resultado =

**Compuerta lógica OR-Exclusiva (XOR).**

La compuerta lógica OR-Exclusiva conocida por si nombre en inglés XOR funciona de tal manera que nos da una salida verdadera (1) si solo una de las entradas es verdadera (1) entradas diferentes, y nos dará una salida en cero o falsa si las entradas tienen el mismo tipo ósea entradas iguales verdadero con verdadero falso con falso, lo cual se muestra en la siguiente tabla de verdad. (Wikipedia, s.f.)

**Tabla se verdad de la compuerta lógica OR-Exclusiva.**

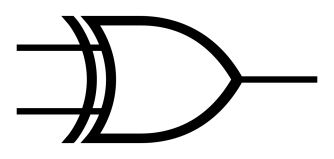
|  |  |  |
| --- | --- | --- |
| **A** | **B** | **x** |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |
|  |  |  |
|  |  |  |

**Expresión algebraica:**

**C:\Users\toshiba\Desktop\Sin título.png**

X= A.’B+’A.B

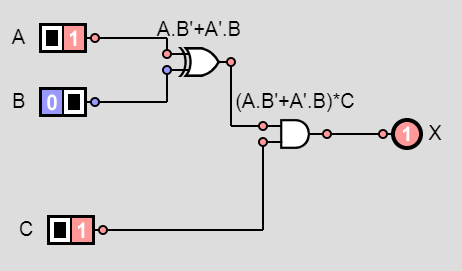
**Símbolo de la compuerta lógica OR-Exclusiva:**



**Ejercicio#1**

**Ejemplo 1 circuito de la compuerta lógica OR-Exclusiva:**

A=1, B= 0, C=1



Como A y B entran a la compuerta XOR tomas la forma de esa ecuación A.’B+’A.B, luego entran en una compuerta AND junto con C por lo que nuestra ecuación queda de la siguiente manera:

(A.’B+’A.B)\*C= X

Al reemplazar estos valores tenemos que:

(1)\*1=1

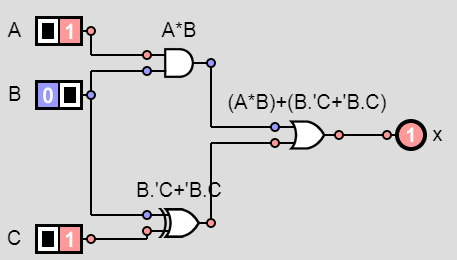
Tabla de verdad:

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | X |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 |

Ejercicio#2

**Ejemplo 2 circuitos de la compuerta lógica OR-Exclusiva:**

A= 1, B=0, C=1



Primero A y B entran a una compuerta AND lo que nos da A\*B

Luego B y C entran en una compuerta XOR lo que nos deja la ecuación B.’C+’B.C

Y por último estas entran en una compuerta OR por lo que obtenemos la ecuación (A\*B)+ (B.’C+’B.C)=X

Reemplazando los datos:

(1\*0)=0

(B.’C+’B.C)=1

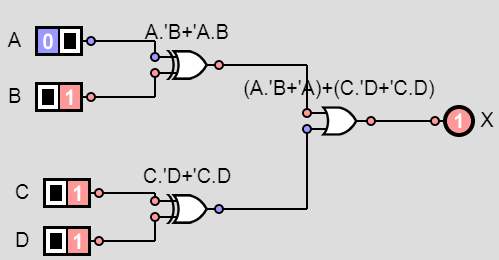
0+1=1

Tabla de verdad:

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | X |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |

**Ejercicio#3**

**Ejemplo 3 circuitos de la compuerta lógica OR-Exclusiva:**



Primero A y B entran a una compuerta XOR lo que nos da el resultado de A.’B+’A+B como A = 0 y B=1 el resultado de A.’B+’A+B=1.

Luego C y D entran a otra compuerta XOR por lo que obtenemos la ecuación C.’D+’C\*D como C=1 y D=1 el resultado de la ecuación C.’D+’C\*D=0

Luego estas ecuaciones entran en una compuerta OR por lo que tenemos lo siguiente

(A.’B+’A+B) + (C.’D+’C\*D)=x

Al sustituir tenemos 1+0=1

Tabla de verdad:

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

**Bibliografía**

Torres, H. (2021, 7 marzo). *Compuertas Lógicas*. HETPRO/TUTORIALES. https://hetpro-store.com/TUTORIALES/compuertaslogicas/#:%7E:text=Para%20la%20compuerta%20AND%2C%20La,que%20es%20una%20multiplicaci%C3%B3n%20binaria.

Hetpro. (2021). *Hetpro-Store*. Obtenido de https://hetpro-store.com/TUTORIALES/compuertas-logicas/

unknow. (18 de Abril de 21). *Compuertas Logicas*. Obtenido de http://163.10.22.82/OAS/compuertas\_logicas/compuerta\_nor.html

Wiki. (21). *Wikipedia*. Obtenido de https://es.wikipedia.org/wiki/Puerta\_NOR

Wikipedia. (s.f.). *wikipedia*. Recuperado el 17 de Abril de 2021, de wikipedia: https://es.wikipedia.org/wiki/Puerta\_XOR

colaboradores de Wikipedia. (2021, 5 abril). *Puerta NOT*. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Puerta\_NOT

Hetpro. (2021). *Hetpro-Store*. Obtenido de https://hetpro-store.com/TUTORIALES/compuertas-logicas/

unknow. (18 de Abril de 21). *Compuertas Logicas*. Obtenido de http://163.10.22.82/OAS/compuertas\_logicas/compuerta\_nor.html

Wiki. (21). *Wikipedia*. Obtenido de https://es.wikipedia.org/wiki/Puerta\_NOR

Wikipedia. (s.f.). *wikipedia*. Recuperado el 17 de Abril de 2021, de wikipedia: https://es.wikipedia.org/wiki/Puerta\_XOR

*Course hero*. (S. f.). Course hero. Recuperado 18 de abril de 2021, de <https://www.coursehero.com/file/88375399/Trabajo-1-Compuertas-l%C3%B3gicasdocx/>

L. (s. f.). *Las Compuertas Lógicas y sus Operaciones Lógicas (AND, OR, NOT, NAND, NOR, XOR, XNOR)*. Logicbus S.A. de C.V https://www.logicbus.com.mx/compuertas-logicas.php