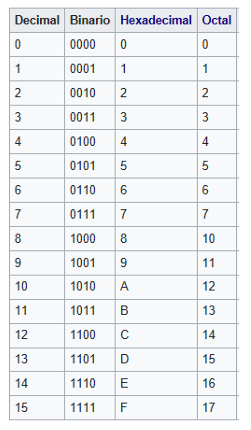
**MODULO: MATEMATICA DISCRETA.**

**Desarrollar y subir en la práctica final, mínimo 3 ejercicios.**

* **Hacer una tabla con los sistemas numéricos más relevantes (Binario, Octal, Decimal y Hexadecimal).**

****

* **Hacer ejercicios conversión con números binarios.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **256** | **128** | **64** | **32** | **16** | **8** | **4** | **2** | **1** |

1. **151**

**151-128=23-16=7-4=3-2=1-1=0**

**128**

**16**

**4**

**2**

**1**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **0** | **0** | **1** | **0** | **1** | **1** | **1** |

1. **25**

**25-16=9-8=1-1=0**

**16**

**8**

**1**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **0** | **0** | **0** | **1** | **1** | **0** | **0** | **1** |

1. **110**

**110-64=46-32=14-8=6-4=2-2=0**

**64**

**32**

**8**

**4**

**2**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **0** | **1** | **1** | **0** | **1** | **1** | **1** | **0** |

* **Hacer ejercicios de conversión a octal.**

|  |  |
| --- | --- |
| **000** | **0** |
| **001** | **1** |
| **010** | **2** |
| **011** | **3** |
| **100** | **4** |
| **101** | **5** |
| **110** | **6** |
| **111** | **7** |

* **Tarea de conjuntos cambiar números y hacer las operaciones.**  
    
  **A= 1,3,4,6,8**

**B= 2,3,6,7**

**C= 5,6,8**

## A

## Imágenes de Overlapping Infographic: descubre bancos de fotos, ilustraciones, vectores y vídeos de 9,245 | Adobe StockA**∩B∩C=6**

## 4

## 2

## B

## (A-B) U C=1,4,8,5

## 3

## 1

## B-(A**∩C) =2,3,7,8,6**

## 7

## 6

## (A-B) U(A**∩C) =1,4,8,6**

## 8

## 5

## C

**Desarrollo:**

* **A= 1, 3, 4, 6, 8**

**B= 2, 3, 6, 7**

**C= 5, 6, 8**

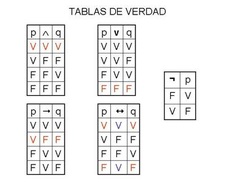
## **A∩B∩C** (A-B) U C B - (A**∩C)** (A-B) U(A**∩C)**

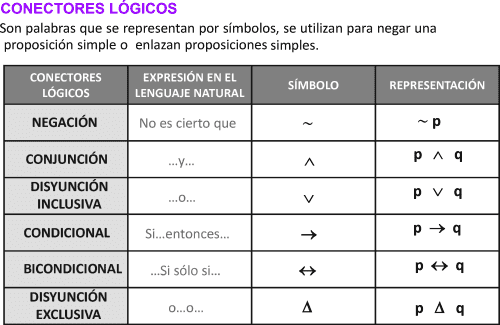
6 6 6 (1,4,8) 5 2,3,7(6,8) (1,4,8) (6,8)



* **Investigar la tabla de la verdad, simbología, conectores, condiciones, valor de verdad, regla de operaciones.**

La tabla de verdad o tabla de valores de verdad, es una tabla que muestra el valor de verdad de una proposición compuesta, para cada combinación de valores que se pueda asignar a sus componentes.



****

* La negación es una conectiva lógica que transforma un enunciado en su opuesto lógico y se le llama conectiva singular porque se aplica sobre un solo enunciado.

|  |  |  |  |
| --- | --- | --- | --- |
| **P** | **Q** | ∼**P** | ∼ **Q** |
| **V** | **V** | **F** | **F** |
| **V** | **F** | **F** | **V** |
| **F** | **V** | **V** | **F** |
| **F** | **F** | **V** | **V** |

* La conjunción es una conectiva lógica que enlaza dos enunciados dando como resultado una fórmula que será verdadera solamente cuando sus enunciados componentes son verdaderos.

|  |  |  |
| --- | --- | --- |
| **P** | **Q** | **P** ∧ **Q** |
| **V** | **V** | **V** |
| **V** | **F** | **F** |
| **F** | **V** | **F** |
| **F** | **F** | **F** |

* La disyunción es una conectiva lógica que enlaza dos enunciados dando como resultado una fórmula que será verdadera solamente cuando al menos uno de sus enunciados componentes es verdadero, siendo falsa cuando ambos son falsos.

|  |  |  |
| --- | --- | --- |
| **P** | **Q** | **P** ∨ **Q** |
| **V** | **V** | **V** |
| **V** | **F** | **V** |
| **F** | **V** | **V** |
| **F** | **F** | **F** |

* La condicional es una conectiva lógica que enlaza dos enunciados dando como resultado una fórmula que será verdadera cuando el segundo enunciado sea verdadero o tenga el mismo valor de verdad que el primero. al primer enunciado involucrado se le llama antecedente y al segundo se le llama consecuente.

|  |  |  |
| --- | --- | --- |
| **P** | **Q** | **P → Q** |
| **V** | **V** | **V** |
| **V** | **F** | **F** |
| **F** | **V** | **V** |
| **F** | **F** | **V** |

* La doble condicional o bicondicional es una conectiva lógica que enlaza dos enunciados dando como resultado una fórmula que será verdadera solamente cuando sus enunciados componentes tienen el mismo valor de verdad.

|  |  |  |
| --- | --- | --- |
| **P** | **Q** | **P** ↔ **Q** |
| **V** | **V** | **V** |
| **V** | **F** | **F** |
| **F** | **V** | **F** |
| **F** | **F** | **V** |

* **Las compuertas lógicas And, Nand, Or, Nor, Xor, Not, Xnor.**

Las **compuertas lógicas** son circuitos electrónicos diseñados para obtener resultados booleanos (0,1), los cuales se obtienen de operaciones lógicas binarias (suma, multiplicación). Dichas compuertas son AND, OR, NOT, NAND, NOR, XOR, XNOR. Además se pueden conectar entre sí para obtener nuevas funciones.

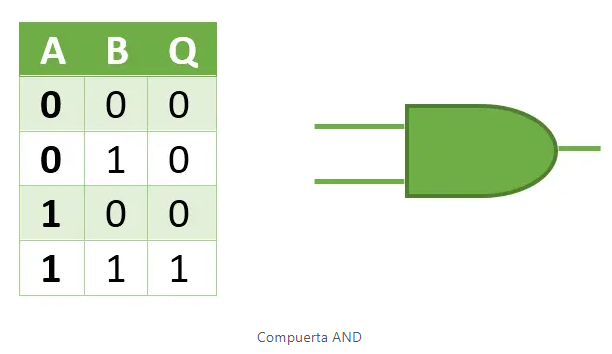
## **Compuerta AND**

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



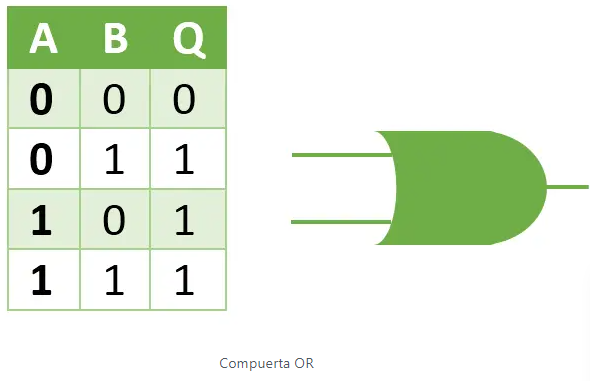
## **Compuerta OR**

la compuerta OR, la salida estará en estado alto cuando cualquier entrada o ambas estén en estado alto. De tal manera que sea una suma lógica.

* Operación

Q=A+B

* Tabla de verdad y símbolo

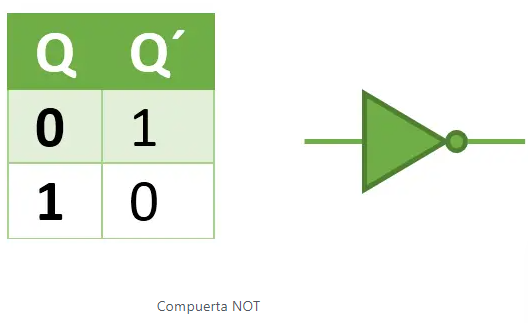


## **Compuerta NOT**

En la compuerta NOT, el estado de la salida es inversa a la entrada. Evidentemente, una negación.

Q=Q

* Tabla de verdad y símbolo



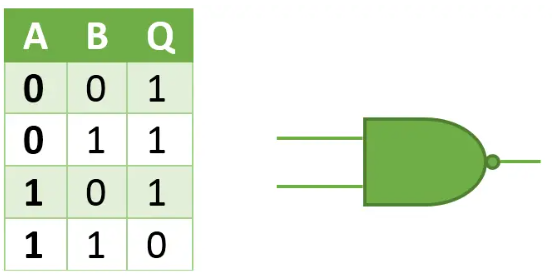
## **Compuerta NAND**

Para la compuerta NAND, cuando las dos entradas estén en estado alto la salida estará en estado bajo. Como resultado de la negación de una AND.

* Operación

Q= (A.B)

* Tabla de verdad y símbolo



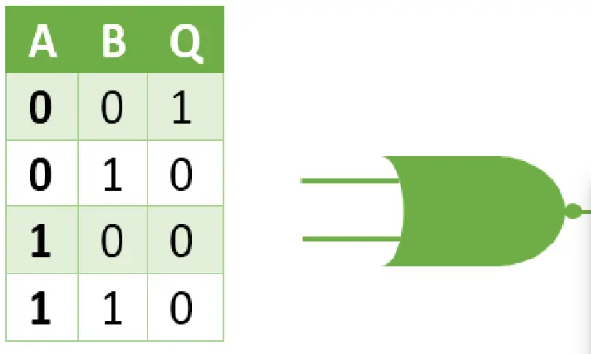
## **NOR**

En la compuerta NOR, cuando las dos entradas estén estado bajo la salida estará en estado alto. Esencialmente una OR negada.

* Operación

Q= (A+B)

* Tabla de verdad y símbolo



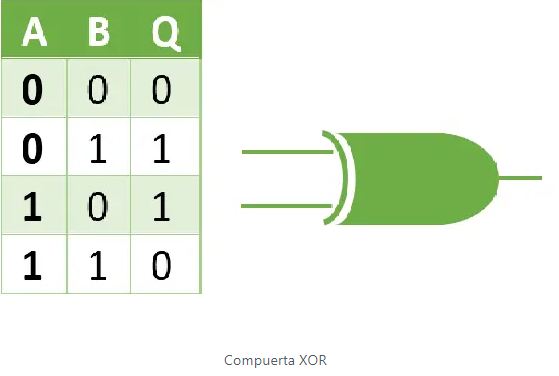
**Compuerta XOR**

La compuerta XOR Su salida estará en estado bajo cuando las dos entradas se encuentren en estado bajo o alto. Al mismo tiempo podemos observar que entradas iguales es cero y diferentes es uno.

* Operación

Q= A.B+A.B

* Tabla de verdad y símbolo



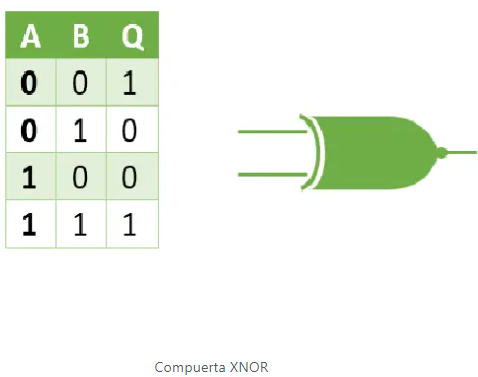
**Compuerta XNOR**

Su salida de hecho estará en estado bajo cuando una de las dos entradas se encuentre en estado alto. Igualmente, la salida de una XOR negada.

* Operación

Q=A.B+A.B

* Tabla de verdad y símbolo

****

* **Hacer los siguientes ejercicios de multiplicación binaria.**

**0x0=0, 0x1=0, 1x0=0, 1x1=1.**

**Ejercicios:**

**111100100**

**X 010**

**000000000**

**111100100**

**+ 000000000**

**01111001000**

**100010**

**X 100**

**000000**

**000000**

**+ 100010**

**10001000**

**111000**

**X 110**

**000000**

**111000**

**+ 111000**

**1000110000**