[Electronics 1](#_Toc16089)

[Signal 1](#_Toc11591)

[Frequency 1](#_Toc27273)

[Attenuation 1](#_Toc13)

[Distortion 1](#_Toc25303)

[Eye Diagram 2](#_Toc28395)

[Truth table for all operations 2](#_Toc4618)

[Exclusive OR 2](#_Toc25738)

# Electronics

## Signal

### Types of signal

Continuous, discrete, analog and digital signals

### Frequency

Number of occurrences of a repeating event per unit of time.

**f=1/T (Hertz)**

F=frequency

T=Time period of the wave. Time to complete one cycle.

1Hz = One cycle per second

If f=1Khz, T=1/1000 = 0.001 Seconds

### Attenuation

Refers to loss of reduction in strength of signal as it travels through a medium.

### Distortion

Alteration or change of received signal from original signal. Its often considered as a form of interference or noise.

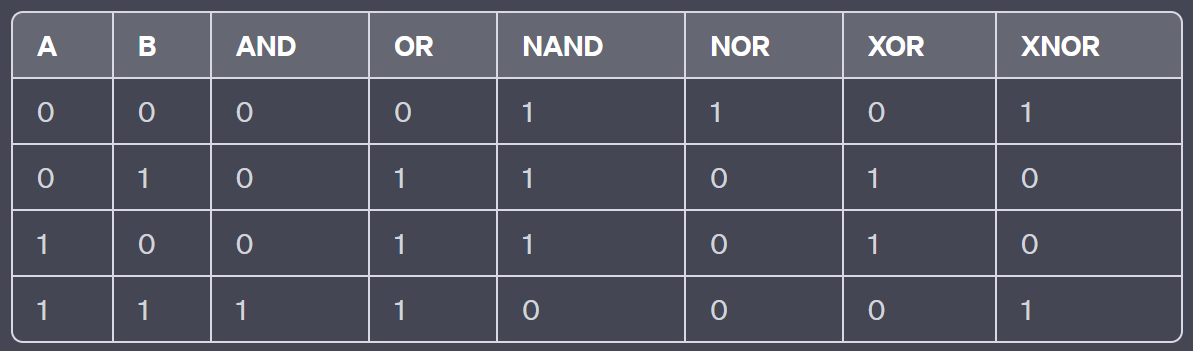
### Eye Diagram

## Truth table for all operations

### Exclusive OR

Returns true if exactly one of the operands is true.

A XOR B=(A AND NOT B) OR (NOT A AND B)



Used for Parity checking, if number of 1 is odd, return 1, odd parity. Else, return 0, even parity.

Swap using ExOR.

Warning: This swap leads to aliasing problem

1: x = x xor y

2: y = x xor y

3: x = x xor y

On line 1 we combine x and y (using XOR) to get this “hybrid” and we store it back in x. XOR is a great way to save information, because you can remove it by doing an XOR again.

So, this is exactly what we do on line 2. We XOR the hybrid with y, which cancels out all the y information, leaving us only with x. We save this result back into y, so now they have swapped.

On the last line, x still has the hybrid value. We XOR it yet again with y (now with x’s original value) to remove all traces of x out of the hybrid. This leaves us with y, and the swap is complete!