













How to Use Basic Boolean Functions Tutorial

Functions -> Programming -> Boolean  -> And , Or , Not , Not Or (NOR) , Not And (NAND) , Exclusive Or (XOR) , Not Exclusive Or (XNOR) , True , False  

This tutorial will explain how to use the Boolean data type and Boolean functions such as those used in Boolean algebra like AND, OR, and NOT (conjunction, disjunction, and negation respectively). These functions are similar to the basic logic gates used to design circuits and only perform two bit input operations.

Boolean data types have only two values, “True” and “False”. Booleans are have their own set of functions that deal exclusively with manipulating Booleans that are derived directly from Boolean algebra and are the primary means of manipulating Boolean data types. These functions are also known as logical operators or logic gates which are discussed below. Because of the simplicity of the Boolean data type, they are frequently used as the values used to drive conditional branching and decision making structures.

To find the logical operator gates, start by opening the Functions palette and selecting the

Programming palette. Then select the Boolean palette . This should bring up the Boolean operator functions.







Select the function that you want to use and place it in your block diagram.

Each function except the NOT, True, and False functions takes two Booleans as its inputs and, because of the rules of Boolean logic, the order of the inputs does not matter. In addition, each function outputs a Boolean value depending on whether the logical operator evaluates the two inputs as True or False.

The NOT function serves as a negation function that takes a Boolean value and outputs the opposite Boolean value (i.e. Outputs true if the input is false, and vice-versa).

You will notice, in the Boolean palette, that the basic logic functions are the AND, OR, and Exclusive Or (XOR), and their respective negations Not And (NAND), Not Or (NOR), and Not Exclusive Or (XNOR).

The output for each of these Boolean logic functions for each combination of inputs is given below:

Input 1	Input 2		AND 	OR 	XOR 	NAND 	NOR 	XNOR 
T	T		T	T	F	F	F	T
T	F		F	T	T	T	F	F
F	T		F	T	T	T	F	F
F	F		F	F	F	T	T	T

The Not function takes one Boolean input and simply inverts it. So if the input is True, the Not function outputs False and vice-versa.

Finally, the Boolean constants True and False simply output those Boolean values and are the Boolean equivalent to Numeric and String constants.