Introduction to Digital Logic

Data and instructions are represented by **binary codes**. The computer must be able to perform arithmetic and logical operations on numerical data held in this form, and its internal circuitry must be able to interpret program instructions which are also held in this form. All of these functions are performed by **logic circuits**.

Computer logic is based on a branch of mathematics called **Boolean Algebra,** which is also known as **Propositional Calculus**. It lays down rules for the symbolic manipulation of logical variables and expressions in a manner very similar to the manipulation of 'unknowns' in ordinary algebraic expressions.

Boolean Variables

A Boolean variable can have only one of two values. These are normally referred to as TRUE or FALSE. When a logic system is implemented by an electronic circuit (such as in a computer), voltages are used to represent these values. Usually a low voltage, say 0 volts, is used to represent FALSE, and a high voltage, say 5 volts, is used to represent TRUE. It is common practice to use the symbols **0** and **1** to represent FALSE and TRUE.

Boolean variables are denoted by letters of the alphabet, thus the variables A,B,C,X,Y,Z are each able to represent a value of **0 or 1**.

Boolean Operations

Boolean Algebra describes a set of basic operations that can be carried out on Boolean variables.

NOT The NOT operation inverts the input. So NOT (False) = True, and NOT(True) = False.

The following operations all have two inputs:

AND The AND operation takes two inputs, and will only output True when both inputs are True. Otherwise it outputs False.

OR The OR operation will output True if at least one of the inputs is True. Otherwise it outputs False.

NAND The NAND operation is best thought of as an AND operation immediately followed by a NOT operation.

NOR The NOR operation may be thought of as an OR operation followed immediately by a NOT operation.

XOR The Exclusive OR operation (aka EOR) will output True if either of the inputs is True, but not if both are True. Another way of thinking of the operation of this gate is that the output is only true if the inputs are not equal, and false if both inputs are the same.