THE ALGEBRA OF DECISION MAKING

Computers carry out their given functions by passing a series of high or low voltages around electronic circuits. These voltages can be interpreted in terms of the binary digits (or bits) 1 and 0. Some functions, such as addition, require specially designed circuits to produce specific outputs for any given input. These are termed 'logic' circuits.

Boolean algebra, the branch of mathematics concerned with true/false logic, is the theoretical basis from which computer architecture is physically realised. The concepts and rules of Boolean algebra are few and easily understood.

In the first instalments of this course, we will study in detail the theoretical and practical aspects of logic circuit design, together with examples of the basic circuits at work inside your own home computer. The rules of Boolean algebra are based on three simple logical operations: AND, OR and NOT. These three logical operations conform closely to the way we use these words in everyday English. Look at this statement:

If it is fine AND it is a Saturday, David will go for a walk.

If David is to go walking or not depends on two things: whether it is fine, and whether it is a Saturday. In coming to a decision about going for a walk, David is only concerned with whether the statements 'it is fine' and 'it is a Saturday' are true or false. There are four possible combinations and only one will result in David taking a walk. A table which shows all the possible combinations of a series of statements is called a 'truth table'. Here is the truth table for our logical AND statement:

IT IS FINE	IT IS A SATURDAY	DAVID WILL GO FOR A WALK
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

A similar process can be undertaken to illustrate the function of the logical operation OR. Consider this statement:

If Jack OR Jill can go, John will go to the match.

Once again there are two conditions that will determine whether or not John goes to the match: whether Jack can go, *or* whether Jill can go. In the same way as the AND statement, we can construct a truth table for the OR statement. Since there are two conditions, each of which may be true or false, there are again four possible combinations. The truth table for the statement will look like this:

JACK CAN GO	JILL CAN GO	JOHN WILL GO TO THE MATCH
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

The third logical operation (NOT) performs a very simple function. Consider this statement:

If it is NOT dark then I will go out.

This time the only condition to consider is whether it is dark. This may be true or false; hence there are only two possible conditions for our truth table.

IT IS DARK	I WILL GO OUT
FALSE	TRUE
TRUE	FALSE

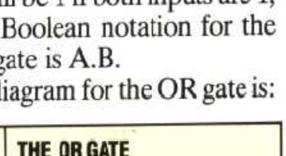
LOGIC GATES

The simple electronic devices that make up computer logic circuits are called 'logic gates'. The three simplest logic gates mimic the function of the logical operations AND, OR and NOT. These gates function by representing a TRUE condition by the binary digit 1, and the FALSE condition by the binary digit 0. So, for each logic gate we can construct a truth table showing all the input combinations together with the resulting output. Each gate has a circuit symbol associated with it and can be written as a Boolean expression.

The truth table and diagram for the AND gate with inputs A and B and output C is:

THE AND GATE

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

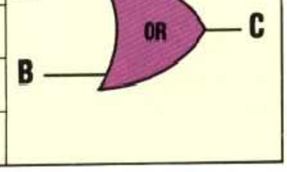


The function of the AND gate can be described in words as: 'the output will be 1 if both inputs are 1, and 0 otherwise'. The Boolean notation for the output from an AND gate is A.B.

The truth table and diagram for the OR gate is:

THE OR GATE

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1



LIZ DIXON