variables ($\overline{A+B+C} = \bar{A}.\bar{B}.\bar{C}$ *and* $\overline{A.B.C} = \bar{A} + \bar{B} + \bar{C}$). De Morgan's laws may also be applied in stages:

$(\overline{A+B}).\bar{C}$
$= \bar{A}.\bar{B}.\bar{C}$ (using de Morgan's Law on the brackets)
$= \overline{A+B+C}$ (recombining using de Morgan's Law)

In common with normal algebra there are three further rules that may be applied to Boolean algebra. The *associative law* allows brackets to be moved:

$(A.B).C = A.(B.C) = A.B.C$
$(A+B)+C = A+(B+C) = A+B+C$

The order in which the letters are written may be changed according to the *commutative law* :

$A.B = B.A$
$A+B = B+A$

The *distributive law* allows brackets to be multiplied out:

$A.(B+C) = A.B + A.C$

EXAMPLES OF SIMPLIFICATION

1) Simplify $(\overline{A+B} + \bar{A}.B).B$
$= (\bar{A}.\bar{B} + \bar{A}.B).B$ (de Morgan)
$= \bar{A}.\bar{B}.B + \bar{A}.B.B.$ (distributive law)
$= 0 + \bar{A}.B$ ($\bar{B}.B = 0$, $B.B = B$)
$= \bar{A}.B$

2) Simplify $\bar{A}.\bar{B} + \bar{A}.B + A.B$
$= \bar{A}.(\bar{B}+B) + A.B$ (distributive law)
$= \bar{A} + A.B$ ($\bar{B}+B = 1$)
$= \bar{A} + B$ (dual of relation 6)

3) Simplify $\overline{\bar{A}+B} + \overline{\bar{A}+\bar{B}} + \bar{A}.B$
$= \bar{\bar{A}}.\bar{B} + \bar{\bar{A}}.\bar{\bar{B}} + \bar{A}.B$ (de Morgan)
$= A.\bar{B} + A.B + \bar{A}.B$ ($\bar{\bar{A}} = A$)
$= A.(\bar{B}+B)+\bar{A}.B$ (distributive law)
$= A + \bar{A}.B$ ($\bar{B}+B = 1$)
$= A + B$ (dual of relation 6)

A SIMPLIFIED XOR GATE

In the previous instalment we looked at an unsimplified circuit for an Exclusive-OR gate. Let us now examine the same problem again, but this time armed with the ability to simplify the Boolean expression and hence the circuit. The truth table for the Exclusive-OR gate is:

INPUT		OUTPUT
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

From the truth table we have previously decided that $C = \bar{A}.B + A.\bar{B}$. There is little simplification that can be made here and a five gate circuit would be required to implement this expression. However, there is an alternative way of approaching the problem. From the truth table, C can be said to be 1 if A and B aren't both 1 or both 0. In Boolean terms we can write down an alternative expression for C:

$C = \overline{A.B + \bar{A}.\bar{B}}$

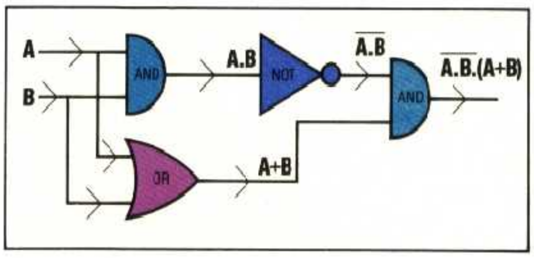
Using de Morgan's Laws repeatedly we can simplify the circuit to get:

$C = (\overline{A.B}).(\overline{\bar{A}.\bar{B}})$

and, finally:

$C = \overline{A.B}.(A+B)$

This expression requires only four gates:



A FULL ADDER CIRCUIT

Previously, we looked at the process of binary addition and designed a simple circuit to add two bits together and produce two outputs for the sum and carry digits in the answer. This circuit we called a half adder. If we call the first input X and the second input Y, we can then verify from the truth table for a half adder (see page 33) that the sum (or answer) output (S) can be represented by the expression: $S = \bar{X}.Y + X.\bar{Y}$. Using de Morgan's law this expression simplifies to: $S = \overline{X.Y}.(X+Y)$. The carry output (C) is simply: $C = X.Y$.

In binary arithmetic there are, in fact, three digits to be added in any one column of the addition sum. As well as the two digits to be added there is also a carry over from the previous column to be included. To be able to reproduce the process of binary addition we must design a circuit with three inputs and two outputs. If we call the carry from the previous column P then the truth table for a full adder will be:

INPUTS			OUTPUTS	
P	X	Y	C	S
0	0	0	0	. 0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1