ALTERNATIVE PATHS

It is possible to solve any logical problem using combinations of the three basic types of logic gates (AND, OR and NOT) that we have met so far in the course. In this instalment of the Logic course, we introduce two new gates — NAND and NOR — which give us alternative ways of designing circuits.

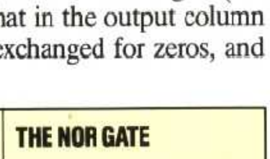
If we can solve all logic problems using AND, OR and NOT gates, why do we need to bother ourselves learning about other types of gates? The reason is that using these new gates, either in isolation or with other gates, can reduce the cost of manufacturing the circuit by simplifying the wiring required or by producing a more elegant solution to a problem. All logic problems may be solved using one of the following techniques:

a) AND, OR and NOT gates together
b) NAND gates only
c) NOR gates only
d) a combination of the above

So let's look at these two new types of gates. As with all circuits and circuit elements, the function of each gate is best described by its truth table.

THE NAND GATE

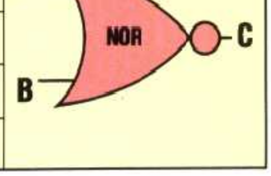
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0



NAND is short for Not AND, and comparing this truth table with the one for an AND gate (see page 8) it can be seen that in the output column all the ones have been exchanged for zeros, and vice versa.

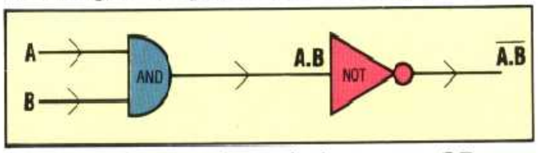
THE NOR GATE

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

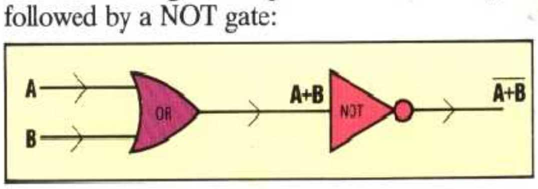


Similarly, NOR is short for Not OR, and a comparison of the output columns for this table and the table for an OR gate (see page 8) again shows that all the ones and zeros have been negated.

There are no special symbols for NAND and NOR operations in Boolean algebra but we can represent each function using the AND, OR and NOT symbols that we have already met. A NAND gate is equivalent to this simple circuit:



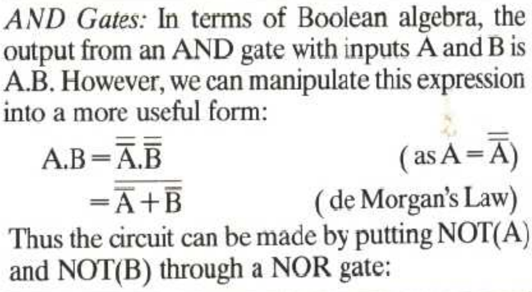
and the NOR gate is equivalent to an OR gate followed by a NOT gate:



USING NAND AND NOR

Just as it is possible to draw AND/OR/NOT circuits that are equivalent to NAND and NOR, so we can represent each of these three basic gates in terms of a series of NOR gates or a series of NAND gates.

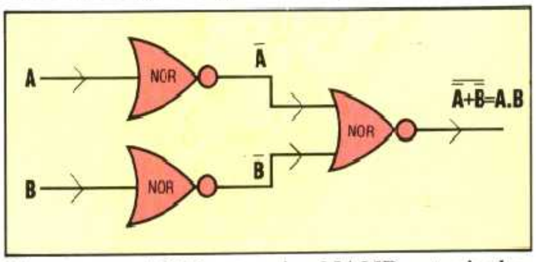
NOT Gates: Negation can be achieved by connecting both inputs together, using either a NOR gate or a NAND gate:



AND Gates: In terms of Boolean algebra, the output from an AND gate with inputs A and B is A.B. However, we can manipulate this expression into a more useful form:

$$A.B = \overline{\overline{A}}.\overline{\overline{B}} \qquad (\text{as } A = \overline{\overline{A}})$$
$$= \overline{\overline{A} + \overline{B}} \qquad (\text{de Morgan's Law})$$

Thus the circuit can be made by putting NOT(A) and NOT(B) through a NOR gate:



To create an AND gate using NAND gates is also possible. The output from a NAND gate is $\overline{A.B}$.