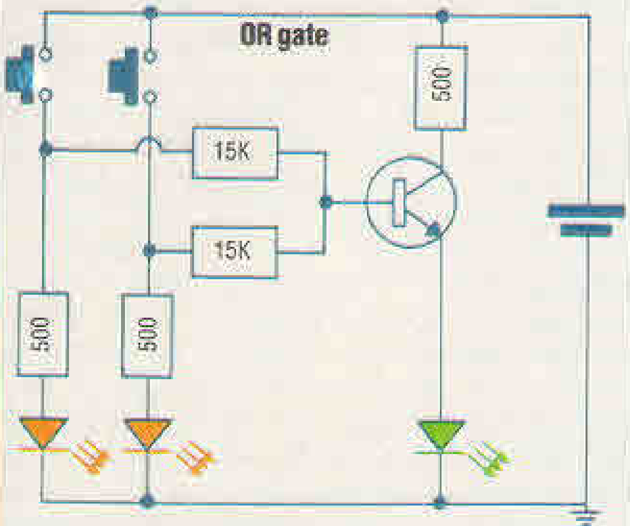
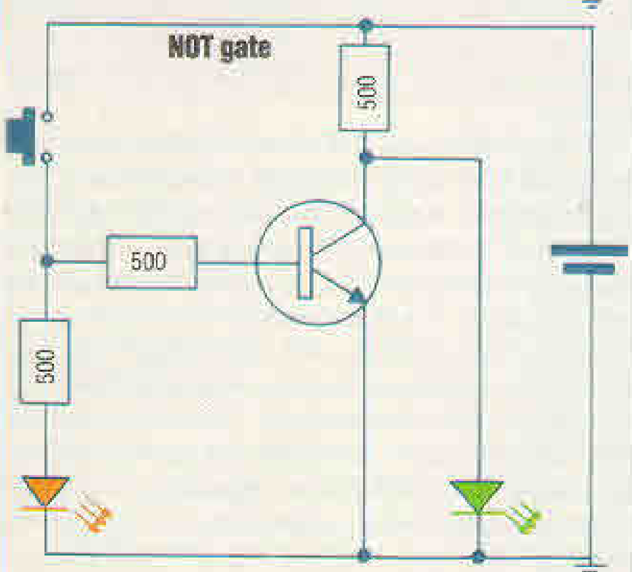
2) NOR Gate

On page 145, we constructed NOT, OR and AND gates using transistors. This first practical exercise is to build a NOR gate using a similar circuit to those for NOT, OR and AND. As a helpful hint, the circuits for OR and NOT appear below. There are two approaches to this problem. You could use your knowledge of logic circuits to construct a NOR gate by combining an OR and a NOT gate. Alternatively, you may spot a short-cut method if you study the NOT gate circuit closely.





3) Decimal To Binary Converter

Construct a circuit that converts decimal to binary. In order to keep the circuit simple, we'll restrict this exercise to two-bit binary numbers, that is, decimal numbers from zero to three. Your circuit should have four input switches labelled zero, one, two and three. When you press one of these switches, the corresponding two binary bits should be shown on a pair of LEDs. A partial truth table for the converter looks like this:-

ZERO BUTTON	ONE BUTTON	TWO BUTTON	THREE BUTTON	HI-BIT	LO-BIT
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

4) BCD Or Not BCD?

We have studied the binary-coded decimal (BCD) numbering system in the Computer Science course. This is a numbering system halfway between decimal and binary — each decimal digit is converted to its binary equivalent and the resulting four-bit groups are joined together to make the BCD number. For example, 53 is coded as 01010011, 5 is represented by 0101 and three is represented by 0011. This means that any valid BCD digit is a group of bits from 0000 to 1001 corresponding to zero to nine in decimal. Any codes in the range 1010 to 1111 are illegal in BCD.

Build a circuit to test whether a given four-bit input is a legal digit. Your circuit should have four switches: B0, B1, B2 and B3, representing the number being tested. There should be two outputs: a green LED if the number is a legal BCD digit and a red LED otherwise. The truth table looks like this:

Decimal Equivalent	B3	B2	B1	B0	Valid BCD	Invalid BCD
0	0	0	0	0	1	0
1	0	0	0	1	1	0
2	0	0	1	0	1	0
3	0	0	1	1	1	0
4	0	1	0	0	1	0
5	0	1	0	1	1	0
6	0	1	1	0	1	0
7	0	1	1	1	1	0
8	1	0	0	0	1	0
9	1	0	0	1	1	0
10	1	0	1	0	0	1
11	1	0	1	1	0	1
12	1	1	0	0	0	1
13	1	1	0	1	0	1
14	1	1	1	0	0	1
15	1	1	1	1	0	1

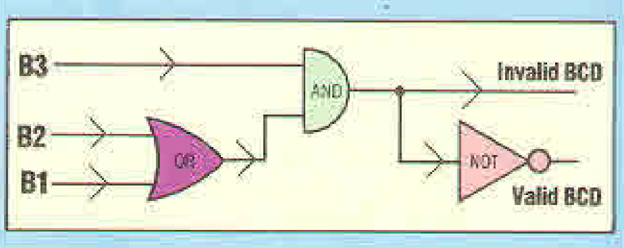
From this truth table, we can see that the BCD signal is given by $\overline{B3}+\overline{B2}.\overline{B1}$. The invalid BCD signal is the NOT of this:

$$\overline{\overline{B3}+\overline{B2}.\overline{B1}}$$

which can be simplified as:

$$\overline{\overline{B3}}\cdot\overline{\overline{B2}.\overline{B1}}$$
$$B3.(B2+B1)$$

So the circuit to test for an invalid BCD digit is quite simple and only involves three of the inputs. A suggested logic diagram for the BCD validator circuit is:



Answers in the next part of Workshop.