LOGIC TUTOR EXPERIMENTS..

NAND GATES

TO avoid having to make, and use, three different types of circuit (AND, OR and NOT gates) to provide logic functions; the modern logic designer uses exclusively NAND gates. By suitable arrangement of these it is possible to generate any logic function that is desired and this type of gate is to be used throughout this series.

NAND VARIANTS

A NAND gate is made by inverting after an AND; the simplest possible circuit is shown in detail in Fig. 3.1 and depicts a basic form of DTL (Diode Transistor Logic). The DTL used in the Logic Tutor is only a minor variant of this circuit; do not confuse this with the basic TTL (Transistor Transistor Logic) NAND gate—Fig. 3.2—which, although it carries out the same function, has entirely different input gating and output stages.

Though NAND is a compound of AND with NOT it is so commonly used that it is considered to be yet another fundamental gate and as such has a standard symbol of its own (Fig. 3.3.). It is possible to obtain 1, 2, 3, 4 and 6 input NAND gates as standards and certain DTL gates have facility for the number of inputs to be expanded by adding extra diodes at the expander node.

Select a two input NAND on the Logic Tutor and connect the inputs to two logic level switches monitoring the output level on one of the lamps. The output will be the inverse of AND as shown in the truth table ($Q = \overline{A \cdot B}$) which means that the output will always be level 1 until all the inputs are at level 1 and at that condition the output will fall to 0. Verify this for yourself and then do the same with a 4 input gate.

INVERT FROM NAND

We demonstrated INVERT last month, but now look at the truth table and Fig. 3.4 to see how this came about. Consider a gate with inputs A and B shorted together and the shorted node called X. If X is 1 both A and B will be 1—refer to the truth table and see that two ones on the input of a NAND give an output 0. If X is made 0 both A and B will be 0 and this is shown to give an output of 1. Thus whatever X is the output is the opposite—an INVERT or NOT function.

Another way of obtaining INVERT is to make input B permanently a level 1. Whatever A is, the output will be the opposite.

A point worth noting is that in all forms of current sinking logic (DTL and TTL) a floating input to a NAND gate will be presumed (by the gate) to be at level 1 however it is bad practice to leave spare inputs of a gate disconnected as they are liable to pick up stray signals and cause severe logic ambiguities.

AND FROM NAND

If the NAND function is obtained by inverting AND we can get back to AND by inverting again after a NAND gate as shown in Fig. 3.5. The output is $\overline{A \cdot B}$ inverted—the double negate cancels and we are left with simply $A \cdot B$.

As an exercise try and simulate a six input AND using any of the NAND gates available on the Logic Tutor (expanders are not permitted!). We shall publish an answer next month and at the same time describe De Morgan's Theorem and show how to use this in conjunction with NAND logic to produce OR.

by M. Hughes

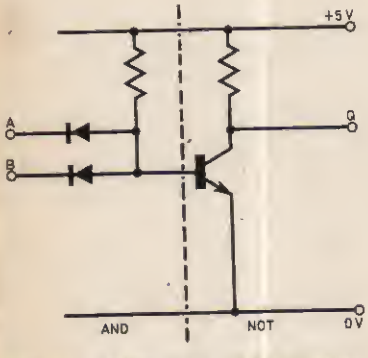

Fig. 3.1. Basic DTL NAND gate

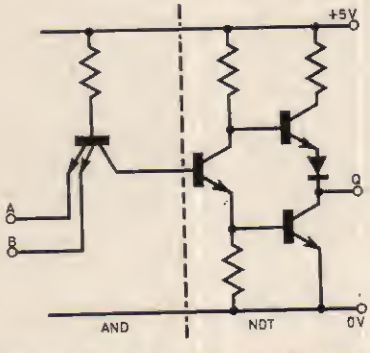

Fig. 3.2. Basic TTL NAND gate

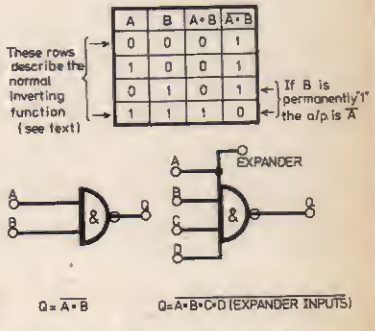

Fig. 3.3. Two forms of NAND gate together with a truth table

A	B	A·B	$\overline{A \cdot B}$
0	0	0	1
1	0	0	1
0	1	0	1
1	1	1	0

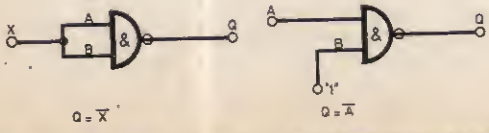

Fig. 3.4. Two ways of getting INVERT from NAND

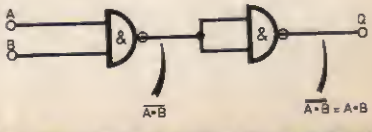

Fig. 3.5. The AND function from NAND gates