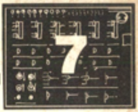
LOGIC TUTOR EXPERIMENTS..



THE HALF ADDER

by M. J. Hughes

THE circuit we left you to puzzle over last month was another form of the EXCLUSIVE OR gate. The Boolean expressions for the various nodes are shown in Fig. 1. The outputs of the two gates that are WIRED OR'd together would have been $\bar{A}+\bar{B}$ and $A+B$ respectively without the link between them but as soon as the link is made these two functions become ANDed together and then it is a simple matter of Boolean manipulation to show that the output function is EXCLUSIVE OR.

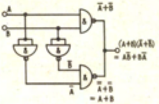


Fig. 1. Answer to last month's problem. The circuit is an EXCLUSIVE OR

0+0 = 0
1+0 = 1
0+1 = 1
1+1 = 0 (carry 1)

Exclusive OR truth table

A	B	$A\bar{B}+B\bar{A}$
0	0	0
1	0	1
0	1	1
1	1	0

Fig. 2. Comparison of binary addition with EXCLUSIVE OR truth table

Binary Addition

It is necessary to understand the basic principle of adding together two binary integers. The rules are very simple—in fact exactly the same as in conventional denary arithmetic except that whenever you get a sum greater than 1 you must carry over a digit into the next column. Let's take the simple case of adding two single digit numbers together. We'll call the digits A and B to differentiate between them and show the sum and carry—when necessary—in the following simple table. Note that we are carrying out pure addition here and in this instance + means plus and not OR

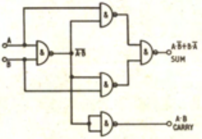
Digit A		Digit B		Sum
0	+	0	=	0
1	+	0	=	1
0	+	1	=	1
1	+	1	=	0 (carry 1 over)

In this list of sums we have used every permutation of the two numbers. Compare the arithmetic with the truth table for the EXCLUSIVE OR gate shown in Fig. 2. If we used electrical signals to represent the numbers we wanted to add together and could accept an electrical signal as an answer you can see that the output of an EXCLUSIVE OR gate gives use a true representation of the sum of the two binary integers. It does not, however, give us the carry digit when we want to add 1 and 1. In addition we only need a carry when we have 1 AND 1 so it is a simple matter to provide this output from the same pair of inputs by introducing an AND function.

Fig. 3 shows how this can be done by taking the $\overline{AB}$ function—generated at the centre node of the EXCLUSIVE OR—and inverting it. You can see that the truth table for the circuit shown in Fig. 3 is an exact replica of the answers we would wish to get when carrying out a binary addition. The circuit is called a half adder. As is often the case there are various ways of designing half adders—you now know at least three ways of making the EXCLUSIVE OR function so try making some more half adders yourself on the Logic Tutor.

You might query why this is called a half adder. The reason is that when we come to add together two multi-digit numbers (see Fig. 4) our current circuit is quite capable of dealing with the least significant column (i) but when we come on to the other columns we have to be able to add together the respective digits of numbers A and B but also have to be able to add in any carry over that was generated in the previous (lesser significance) column. Thus to add multidigit numbers together we must have a circuit that can handle three inputs.

A circuit that will do this is called the full adder and we shall deal with this next month.



A	B	SUM	CARRY
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

Fig. 3. A half adder is an EXCLUSIVE OR plus an AND function of the inputs

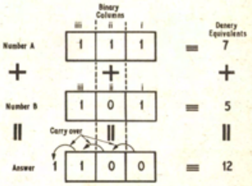


Fig. 4. When carrying out addition of two multidigit numbers a half adder is capable of dealing with column (i) but when it comes to column (ii) there must be provision for three inputs, digit (ii) of A + digit (ii) of B + possible carry over from column (i). The same applies to all higher power columns