# Gates And Adders

## Binary numbers, 1s and 0s, can be added together using the simple logic of AND, OR, and NOT

We have seen in a previous article (see page 68) how relatively simple transistor circuits can be used to make logical decisions such as AND, OR and NOT. The surprising thing is that these same 'logic gates' are also the building blocks used to perform arithmetic inside the computer. In logic, the inputs to the gates are either zero volts, to represent 'false', or a positive voltage, to represent 'true'. The absence of voltage is usually symbolised by a zero (0) and a positive voltage is usually symbolised by a one (1). When logic gates are used to perform arithmetic, the same zeros and ones are used, but this time they literally represent the ones and zeros being added.

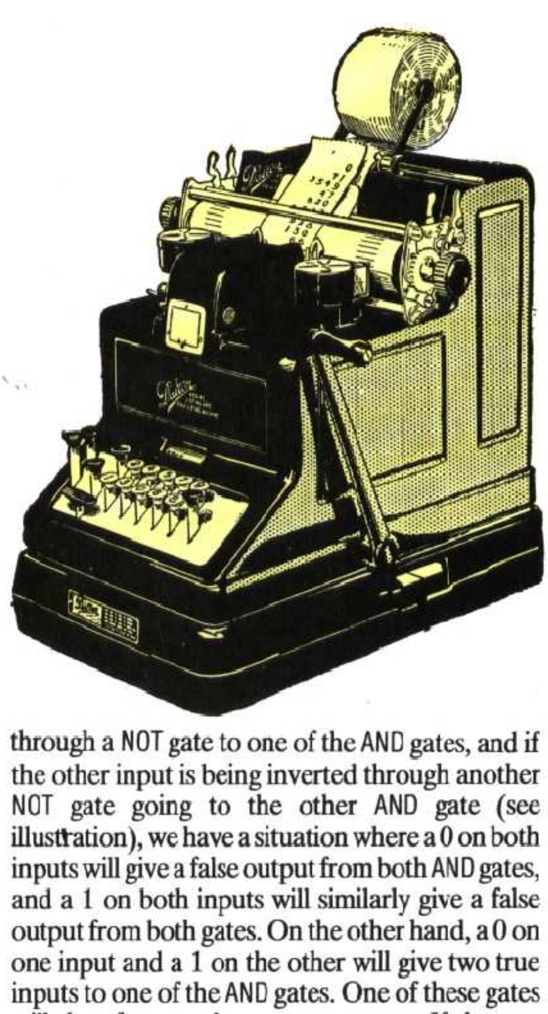
If we want to add two binary digits, there will be only two inputs to the adding circuit, and there can be only four combinations of input — 0 + 0, 0 + 1, 1 + 0 and 1 + 1. From our studies of binary arithmetic, we have learnt that 0 plus 0 equals 0 (as in decimal arithmetic). We also know that 0 plus 1 (or 1 plus 0) equals 1 (again as in decimal arithmetic). The difference from the arithmetic we learnt in school is that in binary, 1 plus 1 equals 0 carry 1. Showing these four additions arithmetically, they would look like this:

| X | | Y | | Z |
|---|---|---|---|---|
| 0 | + | 0 | = | 0 |
| 0 | + | 1 | = | 1 |
| 1 | + | 0 | = | 1 |
| 1 | + | 1 | = | 10 |

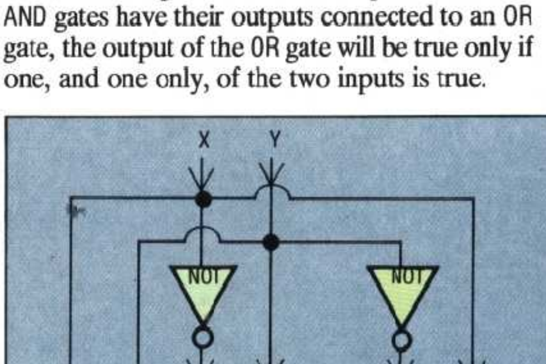
If we were to use an OR gate to do the addition, we would get a false output (0) if both inputs were false (0), and a true output (1) if either of the inputs were true (0 + 1 or 1 + 0).

So far, using a simple OR gate would seem perfectly adequate for adding two binary digits. But wait. If both the inputs are true, the output of a simple OR gate would also be true, but that would be the wrong answer in binary arithmetic. The answer should be a 0 and a carry 1. A simple OR gate would get it right for three out of the four possible input combinations, but three right out of four isn't good enough.

What is needed is a circuit that will give an answer of 0 if both inputs are 0, and an answer of 1 if either of the inputs is 1 and the other is 0, and an answer of 0 if both inputs are 1 (as in the truth table above). This is not as hard as it may seem. If we have two AND gates, with the two inputs going to both gates, but with one input being inverted through a NOT gate to one of the AND gates, and if the other input is being inverted through another NOT gate going to the other AND gate (see illustration), we have a situation where a 0 on both inputs will give a false output from both AND gates, and a 1 on both inputs will similarly give a false output from both gates. On the other hand, a 0 on one input and a 1 on the other will give two true inputs to one of the AND gates. One of these gates will therefore produce a true output. If the two AND gates have their outputs connected to an OR gate, the output of the OR gate will be true only if one, and one only, of the two inputs is true.



**The Desk-Top Adder**
Until the recent invention of the electronic calculator, the mechanical adding machine (or cash register) was a common feature in shops and offices. With the exception of a few refinements, it has remained essentially unchanged for 300 years, working through an arrangement of toothed wheels and cogs. The commercial potential for a calculator was quick to be seen. Pascal invented the first adding machine for use in his father's tax office. With division and multiplication developed by Leibnitz, the calculator was ready for business




## Half Adder

The half adder is a device for adding two binary numbers. It uses an arrangement of logical gates to do this. It is named the half adder because it cannot cope with the carried digit that often results when adding numbers. Try adding 1 and 1. Remember that a NOT gate inverts a 1 to a 0 and a 0 to a 1. Both inputs to an AND gate must be 1 for an output of 1. The output of an OR gate will be 1 if one or both inputs are 1. The output will only be 0 if both inputs are 0. The journey of the digits is illustrated here

MARK WATKINSON