Two's Company

Although computers give speedy answers to complex arithmetical problems, they deal with them in the simplest way

In the last part of our course on binary, we discovered how computers could add. Now we look at the process of multiplication.

If you had to multiply 14 by 12, a simple way would be to do a multiple addition, for example 14+14+14+14+ . . . (12 times). Since multiplication is in one sense a form of repeated addition, this approach would certainly work, and is how the first computers dealt with multiplication. However, the method is awkward and time-consuming, so computer designers evolved a more efficient method.

When two numbers are multiplied, the operation is usually written down on paper like this:

```
  14
 ×12
  28
+14      (a final 0 is often written
 168      to keep the digits in
          the correct column)
```

Exactly the same process works in any base of numbers. Let's look at an example in base two or binary:

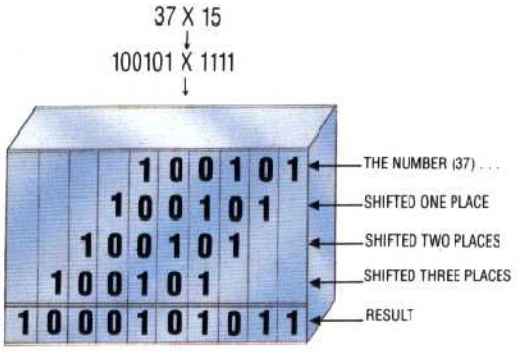
```
  101
 ×11
  101
+101
 1111
```

With larger numbers the method is exactly the same, so let's go back to the example of 14 × 12 and do it in binary:

```
    1110    (14)
   ×1100    (12)
    0000
   0000
  1110
 1110
10101000    (168)
```

Multiplication is even simpler in binary than in decimal because there can never be a carried digit. When you multiply a number by 1 the number is unchanged, 14 × 1 = 14 and when you multiply a number by 0 the answer is 0, 14 × 0 = 0. This is true in binary, decimal and all number systems.

When mathematicians looked at similar calculations to the one above they saw a simple pattern emerging: binary multiplication consists of only two operations, 'shifting' and addition. And this is exactly how the computer does a multiplication. First it shifts 'copies' of the upper line into their correct position (determined by the 1s and 0s in the lower line) and then it adds all the 'copies' together.

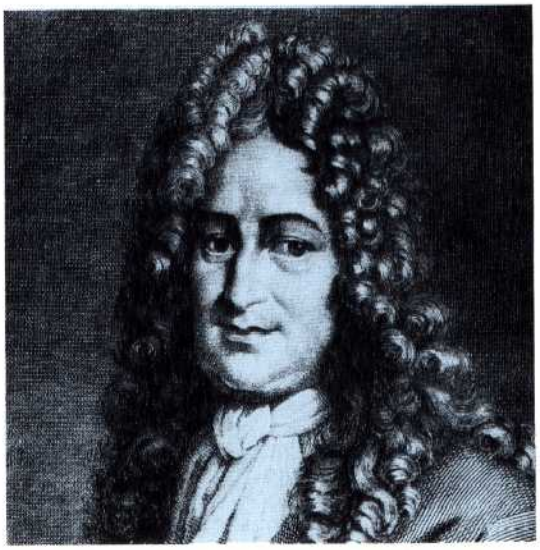


KEVIN JONES

Shifting Into Multiplication
Multiplication is much easier in binary than it is in decimal. The same process of long multiplication used in the decimal system is applied, but because there are only two numbers involved in the multiplying (0 and 1), the operation is very simple. When a number is multiplied by 1 the result is obviously the same number. In the illustration, where 100101 (37) is multiplied by 1111 (15), four copies of 100101 appear. Each copy is shifted to the left according to the position of the 1 that multiplies it. Finally all the copies are added together to give the answer 1000101011 (555)

The computer needs to have a large digit capacity to perform multiplications. When the 4-digit number 1110 was multiplied by the 4-digit number 1100 the answer contained 8 digits (10101000) and in general the result of a multiplication can be up to twice the length of the largest number.

It may come as a surprise to learn that a computer's multiplication can return an incorrect result. Nearly all these errors can be traced back to the amount of space the machine's designer has allowed to hold the answer. If insufficient space has been allocated, 'overflow' will occur, the least significant digits will be lost, and the result will thus be erroneous.



Gottfried Wilhelm Leibniz
Leibniz (1646–1716) was a contemporary of Isaac Newton and made contributions to mathematics, science and philosophy. He invented a machine that could multiply and divide. He also investigated the possibilities for using binary arithmetic in calculating devices, though the first known reference to binary numbers was by Francis Bacon in 1623. In his later years he fell into dispute with Newton over the invention of the calculus