

**Stony Brook University**  
**MAT 118 Spring 2013**

## **Arithmetic with binary numbers**

We take our decimal place-value notation for granted. But it only came to the West with Fibonacci around 1200 AD, and was slow to be adopted. Galileo (1564-1642) was a master of arithmetic but never wrote a decimal point, as far as we know. Binary arithmetic, besides giving insight into how computers process numbers, shows by a second example the power of place-value notation.

### **What are binary numbers**

In decimal notation each digit (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) represents a multiple of a power of 10. Which power it is depends on the *place*. Just looking at whole numbers for the moment, the rightmost digit is units (multiples of  $1 = 10^0$ ). Moving to the left the next digit represents tens (multiples of  $10 = 10^1$ ), the next is hundreds (multiples of  $100 = 10^2$ ) and so on. So  $327 = 3 \times 10^2 + 2 \times 10^1 + 7 \times 10^0 = 300 + 20 + 7$ , etc.

Binary notation works exactly the same, except that each digit (0, 1) represents a multiple of a power of 2. The power depends on the place exactly as in decimal notation. When it's necessary to distinguish binary digits from decimal we can use subscripts 2 or 10 to tell them apart. So for example  $101101_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 32_{10} + 8_{10} + 4_{10} + 1 = 45_{10}$ .

### **Conversion from decimal to binary**

There are two standard ways to convert a decimal into binary. The most simple-minded is to look for the largest power of 2 which is less than or equal to the given number. Subtract that power from your number. Then look for the largest power of 2 which is less than or equal to the difference. Subtract that power and repeat until you are left with 0 or 1. Then you will have expressed your number as a sum of powers of 2, which tells you where to put the 1s in its binary representation. For example, to convert  $81_{10}$  into binary: The largest power of 2 in 81 is  $64 = 2^6$ . The difference is  $81 - 64 = 17$ . The largest power of 2 in 17 is  $16 = 2^4$ . The difference is  $17 - 16 = 1$  so we can stop. We now know  $81 = 64 + 16 + 1 = 2^6 + 2^4 + 2^0$ . So  $81_{10} = 1010001_2$ .

## Addition

Addition is performed by stacking the numbers to be added vertically such that the binary places match. Then the digits are added column by column, with carries as usual. Note the **carries** in this example, which corresponds to the decimal  $27 + 13 = 40$ :

$$\begin{array}{r} \phantom{+} \phantom{1} \phantom{0} \phantom{1} \phantom{0} \phantom{0} \\ \phantom{+} \phantom{1} \phantom{0} \phantom{1} \phantom{0} \phantom{0} \\ + \phantom{1} \phantom{0} \phantom{1} \phantom{0} \phantom{0} \phantom{0} \\ \hline 1 \phantom{0} \phantom{1} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \end{array}$$

In the next example, which corresponds to  $27 + 29 + 5 + 15 = 76$ , the first column sums to binary 100, so a 1 gets carried into the *third* column. Etc.

$$\begin{array}{r} \phantom{+} \phantom{1} \phantom{0} \phantom{0} \phantom{1} \phantom{1} \phantom{0} \phantom{0} \\ \phantom{+} \phantom{1} \phantom{0} \phantom{0} \phantom{1} \phantom{1} \phantom{0} \phantom{0} \\ \phantom{+} \phantom{1} \phantom{0} \phantom{0} \phantom{1} \phantom{1} \phantom{0} \phantom{0} \\ + \phantom{1} \phantom{0} \phantom{0} \phantom{1} \phantom{1} \phantom{0} \phantom{0} \\ \hline 1 \phantom{0} \phantom{0} \phantom{0} \phantom{1} \phantom{1} \phantom{0} \phantom{0} \end{array}$$

## Subtraction

Subtraction is performed exploiting our place-value notation by writing the number to be subtracted below the larger number, with places lined up, and working column by column from the right, as usual, with borrowing when necessary.

In this example, the borrowings are written **small**, and what's left after the borrowing is written **regular size**. This corresponds to  $46 - 27 = 19$  in decimal notation. Note that after you have borrowed from your neighbor, he needs to borrow to cover his debit.

$$\begin{array}{r} \phantom{-} \phantom{1} \phantom{0} \phantom{0} \phantom{1} \phantom{1} \\ \phantom{-} \phantom{1} \phantom{0} \phantom{0} \phantom{1} \phantom{1} \\ - \phantom{1} \phantom{0} \phantom{0} \phantom{1} \phantom{1} \\ \hline 1 \phantom{0} \phantom{0} \phantom{0} \phantom{1} \phantom{1} \end{array}$$

In the next example, which corresponds to  $44 - 27 = 17$ , your neighbor has to borrow so you can borrow from him. Your neighbor now has two (10) so when you borrow one from him, he has 1 left over.

### Multiplication

Let's start small. These examples are chosen so that the addition part does not involve complicated carries.

$$\begin{array}{r} 101 \\ \times 101 \\ \hline 101 \\ 000 \\ 101 \\ \hline 11001 \end{array}$$

As usual we won't write in a row of 0s, but just put a zero at the right-hand end of the next line (this calculation is the decimal  $5 \times 5 = 25$ ):

$$\begin{array}{r} 101 \\ \times 101 \\ \hline 101 \\ 1010 \\ \hline 11001 \end{array}$$

Here is  $54 \times 9 = 486$  in binary:

$$\begin{array}{r} 110110 \\ \times 1001 \\ \hline 110110 \\ 11011000 \\ \hline 111100110 \end{array}$$

### Division

We do it just like regular long division. It's easier because at each step either the divisor is less than the current dividend, producing a 1, or it is smaller, producing a 0. To learn this technique it is essential to work the examples yourself, in pencil, along with the calculations in these notes.

Here is  $86 \div 3 = 28$ , remainder 2:

$$\begin{array}{r}
 11100 \\
 11 \overline{)1010110} \\
 \underline{11} \\
 100 \\
 \underline{11} \\
 11 \\
 \underline{11} \\
 010
 \end{array}$$

And here is  $94 \div 5 = 18$ , remainder 4. (I cribbed the TeX-code for these examples from the web).

$$\begin{array}{r}
 10010 \\
 101 \overline{)1011110} \\
 \underline{101} \\
 0111 \\
 \underline{101} \\
 100
 \end{array}$$

And here is  $1750 \div 26 = 67$ , remainder 8.

$$\begin{array}{r}
 1000011 \\
 11010 \overline{)11011010110} \\
 \underline{11010} \\
 101011 \\
 \underline{11010} \\
 100010 \\
 \underline{11010} \\
 1000
 \end{array}$$