

# AP COMPUTER SCIENCE

## TOPIC III: NUMBERS AND BASES

PAUL L. BAILEY

### 1. INTRODUCTION

Consider the difference between the words *number* and *numeral*, as they are used by mathematicians.

Webster's New World dictionary defines number as *a symbol or word, or a group of either of these, showing how many or which one in a series*. This is clearly not what we mean when we refer to rational or real numbers. Yet, the alternate definitions are even further from our usage. Perhaps closer would be *an idea corresponding to a quantity*.

Webster's does a better job with the second word, defining numeral as *a figure, letter, or word, or a group of any of these, expressing a number*. So if a number is an idea, a numeral is an expression of an idea.

Our standard way of writing numbers is positional, and depends on the choice of ten as a base. However, the choice of ten is arbitrary, and other cultures have made other choices.

Computers naturally use base two, which is easy to convert to base eight or sixteen. So, computer professionals are also conversant in these bases. We describe more fully what bases are, and how to convert between them.

### 2. WHAT IS BASE $b$ ?

A *natural number* is the number zero, or any number which can be obtained from zero by repeatedly adding one. That is, the natural numbers are

$$0, 1, 2, 3, 4, \dots$$

Suppose we write our numbers using a fixed number of symbols. Say we have  $b$  symbols, and each of these represents a different number between zero and  $b - 1$ . We call each of these symbols a *digit*.

Let  $n$  be a natural number, and let  $k$  be the largest integer such that  $n \geq b^k$ . We can express any natural number  $n$  in a unique way as a sequence of digits as follows:

$$n = (a_k a_{k-1} \dots a_1 a_0)_b = a_k b^k + a_{k-1} b^{k-1} + \dots + a_1 b + a_0,$$

where  $a_0, a_1, \dots, a_k$  are digits, each representing a number which is less than  $b$ . This is called the *base  $b$  representation* of the number  $n$ , and we call  $b$  the *base*.

3. CONVERTING FROM BASE  $b$  TO BASE 10

We typically write numbers using base ten, for example,

$$(25283)_{10} = 2 \times 10^4 + 5 \times 10^3 + 2 \times 10^2 + 8 \times 10 + 3.$$

However, the choice of ten as a base is arbitrary. We give some examples of writing numbers in different bases.

- $(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2 + 1 = 8 + 0 + 2 + 1 = 11$
- $(1234)_5 = 1 \times 5^3 + 2 \times 5^2 + 3 \times 5 + 4 = 125 + 50 + 15 + 4 = 194$
- $(1357)_8 = 1 \times 8^3 + 3 \times 8^2 + 5 \times 8 + 7 = 512 + 192 + 40 + 7 = 651$

**Example 1.** Convert  $(582)_7$  to decimal.

*Solution.* Let  $b = 7$ ; this is the original base. Let  $k$  be one less than the number of digits; since there are three digits,  $k = 2$ . Thus,  $b^k = 7^2 = 49$  is the highest power of seven which is less than the number.

For each digit, write that digit times the appropriate power of 7, and add together the results:

$$(582)_7 = 5 \times 7^2 + 8 \times 7 + 2 = 5 \times 49 + 8 \times 7 + 2 = 345 + 56 + 2 = 403.$$

□

Consider the set of digits  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$ . There are sixteen of these, so we can write numbers in base sixteen with them. For example,

$$\bullet (0A5F)_{16} = 0 \times 16^3 + 10 \times 16^2 + 5 \times 16 + 15 = 0 + 2560 + 80 + 15 = 2655$$

**Example 2.** Convert  $(1AF8)_{16}$  to decimal.

*Solution.* In this case, there are four digits so  $k = 3$  and  $b = 16$ . Since  $A = 10$  and  $F = 15$ , we have

$$(1A3F)_{16} = 1 \times 16^3 + 10 \times 16^2 + 3 \times 16 + 15 = 4096 + 2560 + 48 + 15 = 6719.$$

□

#### 4. CONVERTING BETWEEN BASES TWO, EIGHT, AND SIXTEEN

We have special names for the bases which are used most commonly with respect to computers:

- decimal: base ten
- binary: base two (a base two digits is called a *bit*)
- octal: base eight
- hexadecimal: base sixteen

Computers naturally work in base two. However, understanding base two numbers can be troublesome because they are so long. For this reason, programmers often rewrite binary numbers in octal or hexadecimal notation, because it is easy to convert from base two to base eight or sixteen, and back again.

The following chart shows the conversions between bases for zero through fifteen.

Dec	Bin	Oct	Hex	Dec	Bin	Oct	Hex
0	0000	00	0	8	1000	10	8
1	0001	01	1	9	1001	11	9
2	0010	02	2	10	1010	12	A
3	0011	03	3	11	1011	13	B
4	0100	04	4	12	1100	14	C
5	0101	05	5	13	1101	15	D
6	0110	06	6	14	1110	16	E
7	0111	07	7	15	1111	17	F

**4.1. Binary to Octal.** Since  $2^3 = 8$ , binary may be converted into octal by grouping the bits into chunks containing three bits each. The appropriate power of eight will then factor out of the group. For example, consider  $(101001)_2$ . We have

$$(101110)_2 = (2^5 + 0 + 2^3) + (2^2 + 2 + 0) = (2^2 + 0 + 1) \times 8 + (2^2 + 2 + 0) = 5 \times 8 + 6 = (56)_8.$$

To convert binary into octal, it suffices to convert each three bit chunk into one octal digit.

**Example 3.** Convert the twelve bit number  $(101001011110)_2$  into octal.

*Solution.* Group the 12 bits into four chunks of three bits each, and then convert each chunk into a single octal digit:

$$(101001011110)_2 \Rightarrow 101|001|011|110 \Rightarrow 5|1|3|6 \Rightarrow (5136)_8.$$

□

**4.2. Binary to Hexadecimal.** Since  $2^4 = 16$ , binary may be converted into hexadecimal by grouping the bits into chunks containing four bits each.

**Example 4.** Convert the eight bit number  $(10011101)_2$  into hexadecimal.

*Solution.* Group the eight bits into 2 chunks of four bits each, and then convert each chunk into a single hexadecimal digit:

$$(10011101)_2 \Rightarrow 1001|1101 \Rightarrow 9|D \Rightarrow (9D)_{16}.$$

□

5. CONVERTING FROM BASE 10 TO BASE  $b$ 

When we add, subtract, or multiply two integers, we obtain one integer as the result. But when we divide one integer into another, we produce two integers.

Let  $m$  and  $n$  be integers. Divide  $m$  into  $n$  to obtain the *quotient*  $q$ , which is the largest integer such that  $n \geq mq$ , and the *remainder*  $r$ . We have

$$n = mq + r \quad \text{and} \quad 0 \leq r < m.$$

We may convert from base 10 to any base  $b$  by successively dividing by  $b$  and taking the remainder. To see this, let  $n$  be the number given in base  $b$  as  $(a_k a_{k-1} \dots a_1 a_0)_b$ . Then

$$\begin{aligned} n &= (a_k a_{k-1} \dots a_1 a_0)_b \\ &= a_k b^k + a_{k-1} b^{k-1} + \dots + a_1 b + a_0 \\ &= b(a_k b^{k-1} + a_{k-1} b^{k-2} + \dots + a_1) + a_0. \end{aligned}$$

Since  $a_0 < b$ , it is clear that if we divide  $n$  by  $b$ , the quotient is  $(a_k a_{k-1} \dots a_1)_b$  and the remainder is  $a_0$ . By the same reasoning, if we now divide this quotient by  $b$ , the remainder will be  $a_1$ . In this way, we can pick the base  $b$  digits out of  $n$ .

To summarize, if we wish to convert a number  $n$  to base  $b$ , we divide by  $b$ , then divide the quotient by  $b$ , and continue until the quotient is zero. The digits of the base  $b$  expression of  $n$  are the remainders we obtain along the way. The first remainder is the least significant, and the last remainder is the most significant.

**Example 5.** Convert  $(72)_{10}$  to base five.

*Solution.* Repeatedly divide by 5.

Divide 72 by five to get  $72 = 5 \times 14 + 2$ , so  $q = 14$  and  $r = 2$ .

Divide 14 by five to get  $14 = 5 \times 2 + 4$ , so  $q = 2$  and  $r = 4$ .

Divide 2 by five and get  $2 = 5 \times 0 + 2$ , so  $q = 0$  and  $r = 2$ .

Thus  $(72)_{10} = (242)_5$ .

Check this:  $(242)_5 = 2 \times 5^2 + 4 \times 5 + 2 = 50 + 20 + 2 = 72$ . □

**Example 6.** Convert  $(463)_{10}$  to binary.

*Solution.* Repeatedly divide by two.

Divide 463 by 2 and get  $q = 231$  and  $r = 1$ .

Divide 231 by 2 and get  $q = 115$  and  $r = 1$ .

Divide 115 by 2 and get  $q = 57$  and  $r = 1$ .

Divide 57 by 2 and get  $q = 28$  and  $r = 1$ .

Divide 28 by 2 and get  $q = 14$  and  $r = 0$ .

Divide 14 by 2 and get  $q = 7$  and  $r = 0$ .

Divide 7 by 2 and get  $q = 3$  and  $r = 1$ .

Divide 3 by 2 and get  $q = 1$  and  $r = 1$ .

Divide 1 by 2 and get  $q = 0$  and  $r = 1$ . We stop here; this is the leftmost bit.

Thus  $(463)_{10} = (111001111)_2$ . □

**Example 7.** Convert  $(2607)_{10}$  to hexadecimal.

*Proof.* Divide 2607 by 16 to get  $q = 162$  and  $r = 15$ , where  $(15)_{10} = (F)_{16}$ .

Divide 162 by 16 and get  $q = 10$  and  $r = 2$ .

Divide 10 by 16 and get  $q = 0$  and  $r = 10$ , where  $(10)_{10} = (A)_{16}$ .

Thus  $(2607)_{10} = (A2F)_{16}$ . □

## 6. ADDING AND MULTIPLYING IN BINARY

Since computers internally use base two, we focus our attention on this case, to see how computers add and multiply numbers.

This is accomplished in much the same way as we add and multiply in decimal, except that the only possible digits are 0 and 1.

**6.1. Adding in Binary.** To add binary numbers:

- (a) Write one above the other, lining up the corresponding digits. Draw a line under these.
- (b) Proceeding from right to left, add each column.
- (c) The only possible results are 0, 1, and 2. If the column has a carried digit, you may get 3. Note that  $(2)_{10} = (10)_2$ , and  $(3)_{10} = (11)_2$ .
  - (0) If the result is 0, write 0 below the line.
  - (1) If the result is 1, write 1 below the line.
  - (2) If the result is 2, write 0 below the line and carry a one to the next column.
  - (3) If the result is 3, write 1 below the line and carry a one to the next column.
- (d) Continue with the next column, until all columns are depleted.

**Example 8.** Add  $(11010101)_2$  to  $(10011100)_2$ .

*Solution.* Proceed according to the algorithm to obtain:

```

Carry ->    1  111
1st # ->    11010101
2nd # ->    10011100
          + -----
          101110001

```

Note that  $(11010101)_2 = 128+64+16+4+1 = 213$ ,  $(10011100)_2 = 128+16+8+4 = 156$ , and  $(101110001)_2 = 256 + 64 + 32 + 16 + 1 = 369$ ; also  $213 + 156 = 369$ .  $\square$

**6.2. Multiplying in Binary.** Multiplying a base  $b$  number by  $b^w$  shifts all digits  $w$  positions to the left, and inserts  $w$  zeros on the right. For example,  $23456 \times 1000 = 23456000$  in base ten.

So, multiplying a binary number of by  $2^w$  shifts all of the zeros and ones  $w$  places to the right, and inserts  $w$  zeros in the vacated positions. Since every digit of a number expressed in binary represents either zero or a power of two, a computer can multiply by a sequence of appropriate shifts, and then adding together the results.

To multiply binary numbers:

- (a) Write one above the other, lining up the corresponding digits. Draw a line under these.
- (b) Proceeding from right to left on the bottom number, for each nonzero digit in the second number, shift the top number the appropriate number of places, and write down the result.
- (c) Continue until all digits of the second number have been considered. We should have one row below the line for each nonzero digit of the second number.
- (c) Add these together (using appropriate carrying of the digits).

**Example 9.** Multiply  $(10111)_2$  by  $(111)_2$ .

*Solution.* Proceed as described above (\* means multiplication):

```

1st# ->      10111
2nd# ->      101
          *  -----
              10111 (times 1, no shift)
              1011100 (times 4, shift two places left)
          +  -----
              1110011

```

Notice that  $(10111)_2 = 16 + 4 + 2 + 1 = 23$ ,  $(101)_2 = 4 + 1 = 5$ , and  $(1110011)_2 = 64 + 32 + 16 + 2 + 1 = 115$ ; also  $23 \times 5 = 115$ .  $\square$

## 7. ROWS AND COLUMNS

The division algorithm is the process which takes two integers and produces a quotient and remainder, and it has many applications in computing, using the operations of **div** and **mod**.

If  $a$  and  $n$  are integers, and we write  $a = nq + r$  with  $0 \leq r < n$ , then we define

$$a \text{ div } n = q \quad \text{and} \quad a \text{ mod } n = r.$$

For example, if  $n = 39$  and  $m = 5$ , then  $q = 7$  and  $r = 4$ . Note that  $39 = 5 \times 7 + 4$ .

We define the operations of **div** and **mod** by

$$a \text{ div } n = q \quad \text{and} \quad a \text{ mod } n = r.$$

For example,

- $39 \text{ div } 5 = 7$ , that is, 39 divided by 5 equals 7 (this is integer division).
- $39 \text{ mod } 5 = 4$ , that is, 39 modulo 5 equals 4.

The division algorithm, and the resulting operations of **div** and **mod**, are used by computer programs to find the appropriate row and column of an entry in a sequence.

Suppose we wish to display things on a computer screen in  $m$  rows and  $n$  columns, and we have  $mn$  things to display. We number the things 1 through  $mn$ . The row contains the first  $n$  things, the second contains the next  $n$  things, and so forth.

The row and column of the  $k^{\text{th}}$  thing is

- $\text{row}(k) = ((k - 1) \text{ div } n) + 1$
- $\text{col}(k) = ((k - 1) \text{ mod } n) + 1$

We are adding and subtracting one because we started our numbering of things, rows, and columns, with one instead of zero.

The number  $k = \text{num}(x, y)$  associated with column  $x$  and row  $y$  is

- $\text{num}(x, y) = (n(y - 1) + (x - 1)) + 1$

**Example 10.** Consider a calender for March with March 1 on Sunday.

- (a) Find the row and column of March 18.
- (b) Find the date of the fifth day of the fourth week.

*Solution.* Since there are seven days in a week,  $n = 7$ . Since there are 31 days,  $m = 5$  and the last week is partial.

To find the row and column of March 18, let  $k = 18$  in the above formula and compute

- $\text{row}(18) = ((18 - 1) \text{ div } 7) + 1 = (17 \text{ div } 7) + 1 = 2 + 1 = 3$
- $\text{col}(18) = ((18 - 1) \text{ mod } 7) + 1 = (17 \text{ mod } 7) + 1 = 3 + 1 = 4$

So, March 18 is the fourth day (Wednesday) of the third week.

To find the date of the fifth day of the fourth week, we let  $x = 5$  and  $y = 4$ , so

- $\text{num}(5, 4) = (7(4 - 1) + (5 - 1)) + 1 = 21 + 4 + 1 = 26$

The date is March 26. □

The point here is not that we require these formulae to figure out this calender problem; the point is that this is how computers do it.

## 8. PROBLEMS

**Problem 1.** Convert the following numbers to decimal.

- (a)  $(101001)_2$
- (b)  $(765)_8$
- (c)  $(449)_{16}$
- (d)  $(1234)_7$

**Problem 2.** Convert the following numbers from binary to octal.

- (a)  $(101001)_2$
- (b)  $(110001111)_2$
- (c)  $(010101010101)_2$

**Problem 3.** Convert the following numbers from binary to hexadecimal.

- (a)  $(10101100)_2$
- (b)  $(1001100110011001)_2$

**Problem 4.** Convert the following numbers to binary.

- (a)  $(1752)_8$
- (b)  $(1A8E)_{16}$

**Problem 5.** Perform the indicated conversion.

- (a) Convert  $(172)_{10}$  to base 5.
- (b) Convert  $(301)_{10}$  to octal.
- (c) Convert  $(123)_{10}$  to binary.
- (d) Convert  $(200)_{10}$  to hexadecimal.

**Problem 6.** Convert the roman numeral LXVII to decimal, binary, octal, and hexadecimal.

**Problem 7.** Add  $(00110011)_2$  to  $(10101010)_2$ , without converting to decimal. Then convert all numbers to decimal to check your work.

**Problem 8.** Multiply  $(00110011)_2$  by  $(10101010)_2$ , without converting to decimal. Then convert all numbers to decimal to check your work.

**Problem 9.** A terminal displays characters in a grid with 24 rows and 80 columns.

- (a) How many characters can the terminal display?
- (b) Find the row and column of the 1234<sup>th</sup> character.
- (c) Find the number of the character in row 7, column 51.