BINARY ARITHMETIC

James T. Smith
San Francisco State University

A *numeral* is the name of a number. We construct *base n* numerals for the natural numbers by counting as usual with the first n digits starting at 0. For *binary* numerals, $n = 2$:

Decimal numerals	*Binary numerals*
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
⋮	⋮

The two binary digits 0 and 1 are called *bits*.[1] Often we use a subscript to indicate the base:

$$1504_{10} = 1\times10^3 + 5\times10^2 + 0\times10^1 + 4\times10^0$$
$$1101_2 = 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0.$$

(We write the base and the exponents with decimal numerals.)

You can use equations like the previous one to convert from binary to decimal numerals:

$$1101_2 = 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times10^0 = 8 + 4 + 0 + 1 = 13.$$

---

[1] This term is due to John Tukey. Click here for his biographical sketch and portraits on the St. Andrews website.

Decimal to binary conversion requires repeated short division with remainders:

2)13		
2) 6	remainder	1
2) 3	remainder	0
2) 1	remainder	1
0	remainder	1

Read the list of remainders *upward*.

Adding binary numerals is like adding decimals:

binary	decimal
101011	43
+1011	+11
= 110110	= 54

Note the rhythm of the carries: 1 + 1 = 0 carry 1, 1 + 1 + 1 = 1 carry 1. Adding columns of more than two binary numerals is usually impractical because of the profusion of carries.

You can *subtract* binary numerals as you subtract decimals, but borrowing is perplexing:

binary	decimal
01001	
1~~10110~~	54
–1011	–11
= 101011	= 43

To *multiply* a binary numeral by 2^p you just shift it left p places, filling on the right with zeros. Multiplication in general consists of shifting and adding:

$$\begin{array}{r} 101011 \\ \times 1011 \\ \hline 101011 \\ 101011 \\ 101011 \\ \hline 111011001 \end{array}$$

You can also think of multiplying a binary numeral by 2^p as moving the binary point *rightward* p places. *Dividing* by 2^p moves the point *leftward* p places. Long division works as it does with decimal numerals:

$$\begin{array}{r} .1 \\ 10\overline{)1.0} \end{array} \qquad (\tfrac{1}{2})_{10} = (.1)_2$$

$$\begin{array}{rl} & .0101 \\ 11 & \overline{)1.0000} \\ & \underline{11} \\ & 100 \\ & \underline{11} \\ & \ddots \end{array} \qquad (\tfrac{1}{3})_{10} = (.\overline{01})_2$$

$$(\tfrac{1}{4})_{10} = (.01)_2$$

$$\begin{array}{rl} & .00110011 \\ 101 & \overline{)1.00000000} \\ & \underline{101} \\ & 110 \\ & \underline{101} \\ & 1000 \\ & \underline{101} \\ & 110 \\ & \underline{101} \\ & \ddots \end{array} \qquad (\tfrac{1}{5})_{10} = (.\overline{0011})_2$$

For some fractions, the decimal expansion terminates but the binary expansion does not.

Exercises

1. Add these binary numerals by hand:

```
  1011 0101 0001 1101
+ 0110 1001 1110 1001
```

2. Convert the two addends in exercise 1 and their sum to decimal and use the results to check your addition in exercise 1.

3. Subtract these binary numerals by hand and verify by hand addition:

```
  1011 0101 0001 1101
– 0110 1001 1110 1001
```

4. Multiply these binary numerals by hand:

```
  1011
×1101
```

5. Convert the two factors in exercise 4 and their product to decimal and check your multiplication in exercise 4.

6. Divide these binary numerals by hand, obtaining quotient and remainder: 1011 1101 1100 0011 / 1100 1001.

7. Convert the dividend, divisor, quotient, and remainder in exercise 6 to decimal and check your division in exercise 6.

8. Convert the decimal numeral 123456 to binary by hand.