

Exploring Binary

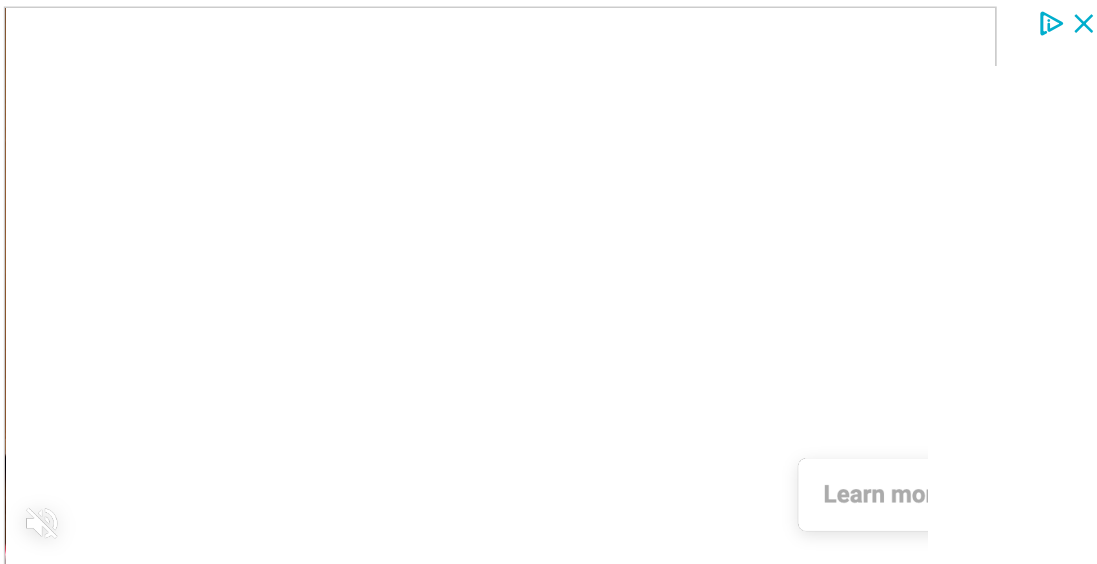
Binary Addition

By [Rick Regan](#) January 30th, 2012

This is the first of a four part series on “pencil and paper” binary arithmetic, which I’m writing as a supplement to my [binary calculator](#). This article introduces binary arithmetic, and then discusses binary addition.

$$\begin{array}{r} 11.1 \\ + 1.1 \\ \hline 110.11 \end{array}$$

Example of Binary Addition



Binary Arithmetic

Binary arithmetic is of interest because that’s how computers do math. When implemented in computers, many things must be taken into account: format (fixed-point, floating-point, etc.), word size (8-bit, 16-bit, 32-bit, etc.), sign representation (sign-magnitude, ones’ complement, two’s complement, etc.), overflow (when numbers are too big), and underflow

(when numbers are too small). Furthermore, there is the design of the circuits to perform the algorithms (full adders, half adders, ripple carry adders, carry-lookahead adders, restoring dividers, non-restoring dividers, etc.). These are not inherent properties of binary arithmetic — they're just implementation issues.

The truth is, in its purest form, binary arithmetic is very simple. The same pencil-and-paper algorithms you learned in grade school work for binary numbers. All you do differently is apply the “facts” for binary numerals, the (small) set of rules for manipulating 0s and 1s. And binary numbers on paper are written as you'd expect: without leading zeros, and with a minus sign (‘-’) if negative.

In my series of articles, I will explain the pure forms of the four basic operations of binary arithmetic: addition, [subtraction](#), [multiplication](#), and [division](#). In this first article, I will discuss binary addition.

Decimal Addition

To add two multiple-digit decimal numbers, you first need to know how to add two single-digit decimal numbers. This requires the memorization of 100 facts, or 55 facts if you exclude the commutative or “turnaround” facts. Also, because of carries, you need to know ten additional facts: $10 + 0 = 10$, $10 + 1 = 11$, ..., $10 + 9 = 19$. The latter apply when there's a carry (always 1) and the “top” digit is 9.

Decimal Addition Facts

+	0	1	2	3	4	5	6	7	8	9	10
0	0	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10	11
2	2	3	4	5	6	7	8	9	10	11	12
3	3	4	5	6	7	8	9	10	11	12	13
4	4	5	6	7	8	9	10	11	12	13	14
5	5	6	7	8	9	10	11	12	13	14	15
6	6	7	8	9	10	11	12	13	14	15	16
7	7	8	9	10	11	12	13	14	15	16	17
8	8	9	10	11	12	13	14	15	16	17	18
9	9	10	11	12	13	14	15	16	17	18	19
10	10	11	12	13	14	15	16	17	18	19	20

9	9	10	11	12	13	14	15	16	17	18	19
10	10	11	12	13	14	15	16	17	18	19	–

For example, let's add 19.7 and 12.8:

$$\begin{array}{r}
 11 \\
 19.7 \\
 + 12.8 \\
 \hline
 32.5
 \end{array}$$

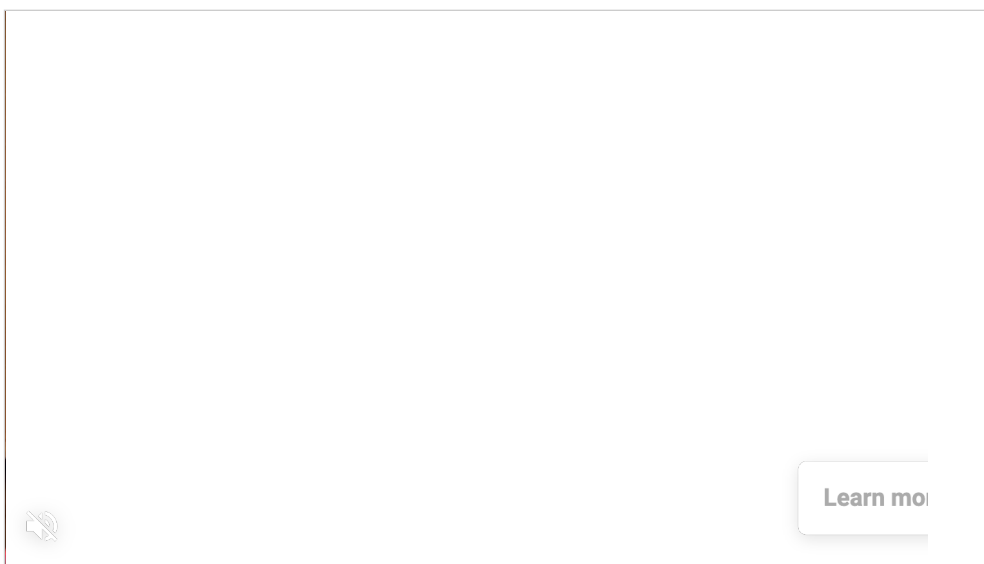
*Example of
Decimal Addition*

Let's think about what a carry does. It turns our neat "add two single-digit numbers per column" problem into "add *three* single-digit numbers per column." But addition is a binary (in a different sense) operation, meaning it operates on two numbers at a time. So what we do is add the carry to the "top" digit, and then add that result to the "bottom" digit. The first addition may result in a double digit answer (10), which means we'd have to use one of the ten extra facts to do the second addition.

In our example, the carry is added to 9, giving 10, and then 10 is added to 2, giving 12. The second addition used the extra "10 + 2 = 12" fact.

$$\begin{array}{r}
 11 \\
 19.7 \\
 210 \\
 + 12.8 \\
 \hline
 32.5
 \end{array}$$

*Decimal Addition
Showing
Intermediate
Additions*



Binary Addition

Binary addition works the same way as decimal addition, except it uses a different — and much smaller — set of facts. There are only four single-digit facts, or three if you exclude the commutative fact. To handle carries, you need to know two additional facts, $10 + 0 = 10$ and $10 + 1 = 11$. ([Resist the urge to pronounce 10 as ten](#) and 11 as eleven; either call them one-zero and one-one, or two and three.) The latter apply when there's a carry (always 1) and the “top” digit is 1.

Binary Addition Facts

+	0	1	10
0	0	1	10
1	1	10	11
10	10	11	—

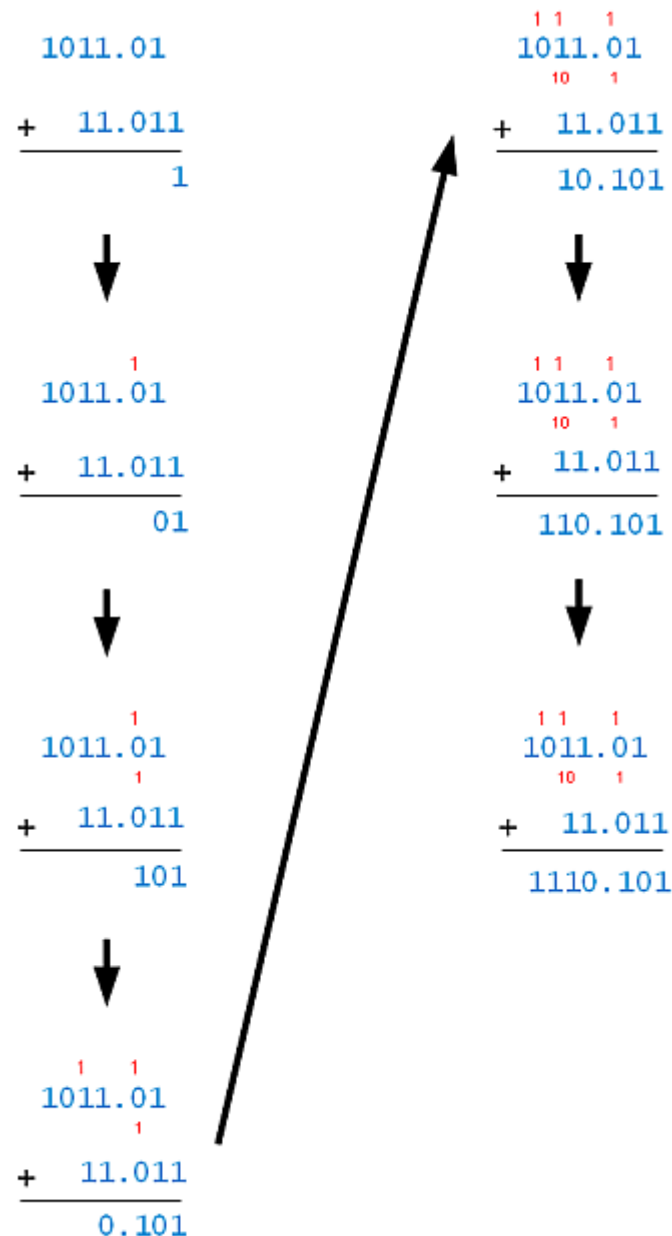
Of the five order-independent facts, two are the same as in decimal ($0 + 0 = 0$ and $1 + 0 = 1$), and one is trivial since it just adds zero ($10 + 0 = 10$). **This leaves only two facts to memorize: $1 + 1 = 10$ and $10 + 1 = 11$!**

To add two binary numbers, proceed as in decimal addition:

- If one or both numbers has a fractional part, line up the [radix points](#).

- Proceeding from right to left, add the digits in each “column,” according to the facts table.
- If the result has two-digits, write down the least significant digit; carry the most significant digit to the next column.

Let's return to the example in the introduction, $1011.01 + 11.011$, this time showing it step-by-step:



Steps of Binary Addition

Eventually you won't need to write down the intermediate additions, and you might even start doing '1 + 1 + 1' as one step.

Checking the Answer

You can verify the answer by [converting the operands to decimal](#), doing decimal addition, and then [converting the decimal answer to binary](#) (of course you can do it that way period, avoiding binary arithmetic entirely!). $1011.01 = 11.25$, and $11.011 = 3.375$. They sum to 14.625 , which is 1110.101 , the answer we got using binary addition.

You can also check the answer using my [binary calculator](#).

Discussion

We didn't need to know the place value of columns, or that a carry represents a power of the base; the algorithm is base-independent. We *did* however need a base-dependent set of facts, which allowed us to manipulate binary numerals according to the basic rules of binary addition.

EB

Related

- [The Binary Marble Adding Machine](#)



**Binary
Subtraction**

Binary Division

**Why 0.1 Does
Not Exist In
Floating-Point**

**Powers Of Two
In The
Josephus...**

**Nine Ways to
Display a
Floating-...**

**My Fascination
with Binary
Numbers**

**Floating-Point
Will Still Be
Broken In...**

**\
(
i**

[Get articles by RSS \(What Is RSS?\)](#)[Get articles by e-mail](#)

Binary numbers / Binary arithmetic, Binary integers, Convert to binary, Convert to decimal

5 comments

1. Miguel Dutra

October 22, 2015 at 10:47 am

Excellent! Thank you! Just a question: why .01 binary converts to 25 decimal?

2. Rick Regan

October 23, 2015 at 8:49 pm

@Miguel,

Do you mean why does .01 convert to .25? Because .01 is 1/4.

3. A_pl

December 21, 2015 at 12:11 pm

Its very easy, but you described some facts i've forgotten about. Thank you for that work, greetings from Poland! 😊

4. O

June 13, 2016 at 5:02 am

I'd like to ask what if the result of addition in one column has 3 digits?

5. [Rick Regan](#)

June 13, 2016 at 9:20 am

@O,

This comes up in binary multiplication — see this [question](#) and my [answer](#).

Comments are closed.

Copyright © 2008–2020 Exploring Binary

[Privacy policy](#)

Powered by [WordPress](#)