CS 251 — Lecture 8

Bartosz Antczak *Instructor: Stephen Mann* *January 26, 2017*

Shift Left and Right Operations

The **shift left** operation multiplies a number by 2

$0010 = 2$	
$0100 = 4$	Shifted to the left; multiplied by 2
$1000 = 8$	Shifted to the left again; multiplied by 2

This is very useful because it's a much faster process than simple multiplying. And intuitively, the **shift right** operation divides a number by 2 (for this operation, we must also duplicate the top bit when shifting)

$0110 = 6$	
$0011 = 3$	Shift to the right and duplicated top bit; divided by 2

This division approach drops any remainders and returns only the value (e.g., dividing $0011 = 3$ by two will yield $0001 = 1$. Dividing $0101 = 5$ by two will yield $0010 = 2$).

Multiplication on Binary Numbers

When we multiply binary numbers, we use the same approach we use to multiply base-10 numbers. Working with our ALU, we'll follow an algorithm to easily multiply numbers (an example is shown):

Multiplier = 1011, Multiplicand = 1101

Iteration	Step	Multiplier	Multiplicand	Product
0	Initial Values	1011	0000 1101	0000 0000
1	Add mpcd to prod Shift left mpcd Shift right mplr	 0101	 0001 1010	0000 1101
2	Add mpcd to prod Shift left mpcd Shift right mplr	 0010	 0011 0100	0010 0111
3	No operation Shift left mpcd Shift right mplr	 0001	 0110 1000	
4	Add mpcd to prod Shift left mpcd Shift right mplr	 0000	 1101 0000	1000 1111

Figure 8.1: Source: Multiplying 1011×1101. Courtesy of Prof. Mann's slides.

The algorithm works using these following steps:

1. Consider three initial values: your multiplier, multiplicand, and final product (which is initially 0000 0000)
2. Add the multiplicand to the product, then shift the multiplicand to the left and the multiplier to the right
3. Repeat this process until the multiplier is zero

Representing Numbers that aren't Integers

We'll represent these kinds of numbers using a pseudo scientific notation. Our scientific notation will be in base 2. We'll store this in our binary string using the IEEE 754 standard:

Figure 8.2: Source: The template for representing a number in scientific notation using the IEEE 754 standard for a 32-bit string. Courtesy of Prof. Mann's slides.

The above value is represented as $(-1)^{\mathrm{S}} \times (1.\text{significand}) \times 2^{(\text{Exponent} - \text{Bias})}$