Lesson 2 - Computer Arithmetic

Learning Outcomes

By the end of this lesson, students will be able to;

- apply basic arithmetic operations.
- describe two's complement in integer representation.
- describe sign-magnitude representation.
- describe floating point representation mechanism.
- compute arithmetic operations using integers and floating point numbers.

2.1 Binary Arithmetic

We are familiar with different arithmetic operations such as addition, subtraction, multiplication and division in a decimal system. Arithmetic in a computer are done using binary numbers and it is similar to the decimal arithmetic.

The sum of two binary numbers is calculated by the same rules as in decimal, except that the digits of the sum in any significant position can be only 0 or 1 . Any carry obtained in a given significant position is used by the pair of digits one significant position higher. Subtraction is slightly more complicated. The rules are still the same as in decimal, except that the borrow in a given significant position adds 2 to a minuend digit. Multiplication is simple: The multiplier digits are always 1 or 0: therefore, the partial products are equal either to the multiplicand or to 0. The rules of binary arithmetic operations are shown in Figure 2.1.

Binary addition:-

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Binary Multiplication:-

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Binary subtraction:-

A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Binary Division:-

A	B	Output
0	1	0
1	1	1
Division by zero is meaning less		

Figure 2.1: The rules of binary arithmetic

Example 2.1

Add the binary numbers 110010 and 110111.

Solution.

Carries:		1	1		1	1				
Augend:		0	1	1	0	0	1	0	=	50
Addend:	+	0	1	1	0	1	1	1	=	55
Sum:		1	1	0	1	0	0	1	=	105

Example 2.2

Subtract the binary numbers 101101 and 100111.

Solution.

Borrows:						2		2			
Minuend:		1	0	~~1~~	0	~~1~~	0	0	1	=	45
Subtrahend:	-	1	0	0		1		1	1	=	39
Difference:		0	0	0		1		1	0	=	6

Example 2.3

Multiply the binary numbers 1011 and 101.

Solution.

Multiplicand:			1	0	1	1	=	11
Multiplier:		×		1	0	1	=	5
			1	0	1	1		
		0	0	0	0			
	1	0	1	1				
Product:	1	1	0	1	1	1	=	55

Example 2.4

Divide the binary numbers 11001 and 101.

Solution.

$$
\begin{array}{cccccccc}
 & & & & & 1 & 0 & 1 \\ \cline{4-8}
1 & 0 & 1 & \multicolumn{1}{|c}{1} & 1 & 0 & 0 & 1 \\
 & & & 1 & 0 & 1 & \downarrow & \downarrow \\ \cline{4-6}
 & & & 0 & 0 & 1 & 0 & 1 \\
 & & & & & 1 & 0 & 1 \\ \cline{6-8}
 & & & & & 0 & 0 & 0
\end{array}
$$

Therefore the answer is 101.

2.2 Signed Binary Numbers

In tradition we use the '+' sign before a number to indicate it as a *positive* number and a '-' sign to indicate it as a *negative* number. This method of representation of numbers is called *sign-magnitude* representation. However, on a computer this $+$ and $-$ representation is not convenient. Hence computer use three different techniques to denote signed binary numbers. They are sign-magnitude representation, one's-complement representation and two's-complement representation.

2.2.1 Sign-Magnitude Representation

In this representation we use an additional bit to denote the sign of the number. This additional bit is called as the *sign bit*. The sign bit is usually placed as the MSB. Generally we replace "+" sign with "0" and "-" with "1". For instance, an 8-bit signed binary number 01100101 represents a positive number whose magnitude is $(1100101)_2 = (101)_{10}$. In the given number 01100101 MSB is 0, which indicates that the number is positive. Further, in the signed binary form, 11100101 represents a negative number whose magnitude is $(1100101)_2 = (101)_{10}$. The 1 in the MSB position indicates that the given number is negative.

2.2.2 One's Complement Representation

One's complement operation is very simple to carry out. Using one's complement representation, the positive numbers are directly represented by their equivalent binary form. To find the one's complement of a negative number convert all of the 1's in the number to 0's, and all of the 0's to 1's. Then add the sign bit. Therefore, in the one's complement representation the leftmost bit is 0 for positive numbers and 1 for negative numbers, as it is for the signed magnitude representation. The process of changing 1's to 0's and changing 0's to 1's, is called as *complementing* the bits.

Example 2.5

(a) What is the one's complement binary representation of decimal number -75?

(b) What is the one's complement binary representation of decimal number 11?

Solution.

(a)

Binary value of 75	=	1001011
Complement of 1001011	=	0110100
One's complement of -75	=	10110100

(b)

Binary value of 11	=	1011
One's complement of 11	=	1011

2.2.3 Two's Complement Representation

Using two's complement representation, the positive numbers are directly represented by their equivalent binary form. For the negative numbers, add 1 to the one's complement of the corresponding number.

Example 2.6

(a) What is the two's complement binary representation of decimal number -75?

(b) What is the two's complement binary representation of decimal number 11?

Solution.

(a)

Binary value of 75	=	1001011
Complement of 1001011	=	0110100
One's complement of -75	=	10110100
Two's complement (add 1)	=	+ 1
Two's complement of -75	=	10110101

(b)

Binary value of 11	=	1011
Two's complement of 11	=	1011

2.3 Subtract with Two's Complement

Two's complement is used in digital computers to simplify the subtraction operation. The subtraction of two positive numbers $(M - N)$, both of base 2, can be done as follows:

1. Add minuend M to the two's complement of the subtrahend N.
2. Inspect the result obtained in step 1 for an end carry.
   (a) If an end carry occurs discard it.

(b) If an end carry does not occurs, take the two's complement of the number obtained in step 1 and place a negative sign in front.

Example 2.7

Using two's complement do the following subtraction.

(a) 17 – 10

(b) 26 – 75

Solution.

(a)
$$\begin{array}{llr} \text{Binary value of } 17 & = & 10001 \\ \text{Two's complement of } -10 & = & +\ 10110 \\ \hline & = & \mathbf{1}00111 \\ \text{End carry} & = & \mathbf{1} \\ \text{Result (Discard end carry)} & = & 00111 \end{array}$$

(b)
$$\begin{array}{llr} \text{Binary value of } 26 & = & 00011010 \\ \text{Two's complement of } -75 & = & +\ 10110101 \\ \hline & = & 11001111 \\ \text{Therefore no carry bit} & & \\ \text{Two's complement of } 11001111 & = & 110001 \end{array}$$

Therefore result is -110001.

Self-Assessment Exercises

1. Do the following binary arithmetics.

   (a) $1101.01 + 101.01$

   (b) 110001.110 – 10111.010

   (c) 110001.11×101.1

   (d) 1101.1×11.1

   (e) 10111.110/1011.11

2. Give the representation of the following negative numbers on a computer using two's compliment.

   (a) -197_{10}

   (b) -1987_{10}

   (c) -49321_{10}

3. Using two's compliment do the following subtractions.

   (a) $29 - 12$

   (b) $9457 - 6785$

   (c) $5830 - 876$

Suggested Reading

Chapter 19: William Stallings, (2010) Computer Organization and Architecture: Designing for Performance, 9^{th} Edition, Prentice Hall.