Тарасенко-Сивиконь Артем Олегович, КІ-СА

Лабораторна робота №2. Арифметичні операції над двійковими числами

(Код можна подивитись на GitHub)

1. Множення двійкових чисел (Зсув результату вправо):

```
Start:    Register: 00000000  Multiplier: 0110  Multiplicand: 0111

Iteration 1 :
Shift register right: 00000000
Shift multiplier right: 0011
Register:  00000000  Multiplier:  0011  Multiplicand:  0111

Iteration 2 :
Least significant multiplier bit is '1': Register = Register + Multiplicand = 01110000
Shift register right: 00111000
Shift multiplier right: 0001
Register:  00111000  Multiplier:  0001  Multiplicand:  0111

Iteration 3 :
Least significant multiplier bit is '1': Register = Register + Multiplicand = 10101000
Shift register right: 01010100
Shift multiplier right: 0000
Register:  01010100  Multiplier:  0000  Multiplicand:  0111

Iteration 4 :
Shift register right: 00101010
Shift multiplier right: 0000
Register:  00101010  Multiplier:  0000  Multiplicand:  0111


Result:  00101010 --> 42
```

2. Ділення двійкових чисел (Зсув залишку вправо):

```
Start:    Remainder: 00000111  Quotient: 0000  Divisor: 0010  Dividend: 0111

Iteration 1 :
Shift remainder register to the left: 00001110
Remainder = Remainder (left half) - divisor = 11101110
Remainder < 0:
  1) restor reminder value: Remainder = Remainder (left half) + divisor = 00001110
  2) shift quotient to the left and set the new least significant bit to 0: 0000
Remainder: 00001110  Quotient: 0000  Divisor: 0010

Iteration 2 :
Shift remainder register to the left: 00011100
Remainder = Remainder (left half) - divisor = 11111100
Remainder < 0:
  1) restor reminder value: Remainder = Remainder (left half) + divisor = 00011100
  2) shift quotient to the left and set the new least significant bit to 0: 0000
Remainder: 00011100  Quotient: 0000  Divisor: 0010

Iteration 3 :
Shift remainder register to the left: 00111000
Remainder = Remainder (left half) - divisor = 00011000
Remainder > 0:
  Shift quotient register to the left and set the new least significant bit to 1: 0001
Remainder: 00011000  Quotient: 0001  Divisor: 0010

Iteration 4 :
Shift remainder register to the left: 00110000
Remainder = Remainder (left half) - divisor = 00010000
Remainder > 0:
  Shift quotient register to the left and set the new least significant bit to 1: 0011
Remainder: 00010000  Quotient: 0011  Divisor: 0010


Quotient: 0011 --> 3    Remainder: 0001 --> 1
```

3. Робота з IEEE 754 Floating Point (Додавання):

```
X1 = 31.0 --> 01000001111110000000000000000000  ( sign: 0, exponent: 10000011, mantissa: 11111000000000000000000 )
X2 = -12.0 --> 11000001010000000000000000000000  ( sign: 0, exponent: 10000011, mantissa: 11111000000000000000000 )

Sign of the result: 1

Exponent difference: 131 - 130 = 1
Left shift decimal point of M2 by the exponent difference:
M2 before  -  11000000000000000000000
M2 after   -  01100000000000000000000

Subtrsacting M2 from M1, result mantissa: 10011000000000000000000
Normalizing result matissa: 10011000000000000000000
Correcting exponent value of the result: 10000010

Result: 01000001100110000000000000000000 --> 19.0 ( sign: 0, exponent: 10000011, mantissa: 00110000000000000000000 )
```