

Лабораторна робота № 2

Тема: «Арифметичні операції над двійковими числами»

Роботу виконав
студент 3 курсу КІ
Системний адміністратор
Бабяк Андрій Миколайович

Хід роботи

1. Алгоритм Бута

d. Алгоритм Бута

00 – NOP
10 – SUB
11 – NOP
01 – ADD

Перевірка роботи програми :

На вхід програми було подано числа 3 та -4.

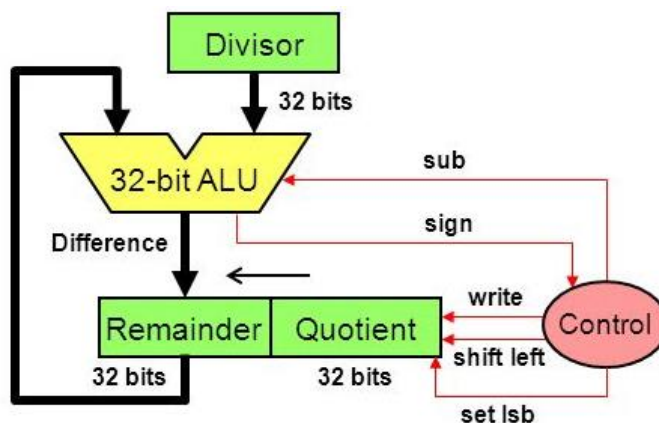
```
Enter first number:
3
Enter second number:
-4
Booth's algorithm:
Step 1:
  Shift right:  0000 0000 0000 0000 1111 1111 1111 1100 0
                0000 0000 0000 0000 0111 1111 1111 1110 0
Step 2:
  Shift right:  0000 0000 0000 0000 0111 1111 1111 1110 0
                0000 0000 0000 0000 0011 1111 1111 1111 0
Step 3:
  Add S:   1111 1111 1111 1101 0000 0000 0000 0000 0
  To P:    0000 0000 0000 0000 0011 1111 1111 1111 0
  Shift right:  1111 1111 1111 1101 0011 1111 1111 1111 0
                1111 1111 1111 1110 1001 1111 1111 1111 1
Step 4:
  Shift right:  1111 1111 1111 1110 1001 1111 1111 1111 1
                1111 1111 1111 1111 0100 1111 1111 1111 1
Step 5:
  Shift right:  1111 1111 1111 1111 0100 1111 1111 1111 1
                1111 1111 1111 1111 1010 0111 1111 1111 1
Step 6:
  Shift right:  1111 1111 1111 1111 1010 0111 1111 1111 1
                1111 1111 1111 1111 1101 0011 1111 1111 1
Step 7:
  Shift right:  1111 1111 1111 1111 1101 0011 1111 1111 1
                1111 1111 1111 1111 1110 1001 1111 1111 1
Step 8:
  Shift right:  1111 1111 1111 1111 1110 1001 1111 1111 1
                1111 1111 1111 1111 1111 0100 1111 1111 1
Step 9:
  Shift right:  1111 1111 1111 1111 1111 0100 1111 1111 1
                1111 1111 1111 1111 1111 1010 0111 1111 1
Step 10:
  Shift right:  1111 1111 1111 1111 1111 1010 0111 1111 1
                1111 1111 1111 1111 1111 1101 0011 1111 1
Step 11:
  Shift right:  1111 1111 1111 1111 1111 1101 0011 1111 1
                1111 1111 1111 1111 1111 1110 1001 1111 1
Step 12:
  Shift right:  1111 1111 1111 1111 1111 1110 1001 1111 1
                1111 1111 1111 1111 1111 1111 0100 1111 1
```

```

Step 13:
  Shift right:    1111 1111 1111 1111 1111 1111 0100 1111 1
                  1111 1111 1111 1111 1111 1111 1010 0111 1
Step 14:
  Shift right:    1111 1111 1111 1111 1111 1111 1010 0111 1
                  1111 1111 1111 1111 1111 1111 1101 0011 1
Step 15:
  Shift right:    1111 1111 1111 1111 1111 1111 1101 0011 1
                  1111 1111 1111 1111 1111 1111 1110 1001 1
Step 16:
  Shift right:    1111 1111 1111 1111 1111 1111 1110 1001 1
                  1111 1111 1111 1111 1111 1111 1111 0100 1
Answer is:
  In decimal: -12
  In binary: 1111 1111 1111 1111 1111 1111 1111 0100

```

2. Частка та залишок в одному регістрі



```

Enter first number:
23
Enter second number:
8

Register:
      0 0000 0000 0000 0000 0000 0000 0001 0111

Shift left:
      0 0000 0000 0000 0000 0000 0000 0010 1110
Subtract divisor: 1 1111 1111 1111 1000
Register:
      1 1111 1111 1111 1000 0000 0000 0010 1110
Set last quotient bit to 0:
      1 1111 1111 1111 1000 0000 0000 0010 1110
Shift left:
      1 1111 1111 1111 0000 0000 0000 0101 1100
Add divisor: 0 0000 0000 0000 1000
Register:
      1 1111 1111 1111 1000 0000 0000 0101 1100
Set last quotient bit to 0:
      1 1111 1111 1111 1000 0000 0000 0101 1100
Shift left:
      1 1111 1111 1111 0000 0000 0000 1011 1000
Add divisor: 0 0000 0000 0000 1000
Register:
      1 1111 1111 1111 1000 0000 0000 1011 1000
Set last quotient bit to 0:
      1 1111 1111 1111 1000 0000 0000 1011 1000
Shift left:
      1 1111 1111 1111 0000 0000 0001 0111 0000
Add divisor: 0 0000 0000 0000 1000
Register:
      1 1111 1111 1111 1000 0000 0001 0111 0000

```

```

Set last quotient bit to 0:
  1 1111 1111 1111 1000 0000 0001 0111 0000
Shift left:
  1 1111 1111 1111 0000 0000 0010 1110 0000
Add divisor: 0 0000 0000 0000 1000
Register:
  1 1111 1111 1111 1000 0000 0010 1110 0000
Set last quotient bit to 0:
  1 1111 1111 1111 1000 0000 0010 1110 0000
Shift left:
  1 1111 1111 1111 0000 0000 0101 1100 0000
Add divisor: 0 0000 0000 0000 1000
Register:
  1 1111 1111 1111 1000 0000 0101 1100 0000
Set last quotient bit to 0:
  1 1111 1111 1111 1000 0000 0101 1100 0000
Shift left:
  1 1111 1111 1111 0000 0000 1011 1000 0000
Add divisor: 0 0000 0000 0000 1000
Register:
  1 1111 1111 1111 1000 0000 1011 1000 0000
Set last quotient bit to 0:
  1 1111 1111 1111 1000 0000 1011 1000 0000
Shift left:
  1 1111 1111 1111 0000 0001 0111 0000 0000
Add divisor: 0 0000 0000 0000 1000
Register:
  1 1111 1111 1111 1000 0001 0111 0000 0000
Set last quotient bit to 0:
  1 1111 1111 1111 1000 0001 0111 0000 0000
Shift left:
  1 1111 1111 1111 0000 0010 1110 0000 0000
Add divisor: 0 0000 0000 0000 1000
Register:
  1 1111 1111 1111 1000 0010 1110 0000 0000
Set last quotient bit to 0:
  1 1111 1111 1111 1000 0010 1110 0000 0000
Shift left:
  1 1111 1111 1111 0000 0101 1100 0000 0000
Add divisor: 0 0000 0000 0000 1000
Register:
  1 1111 1111 1111 1000 0101 1100 0000 0000

```

```

Set last quotient bit to 0:
  1 1111 1111 1111 1000 0101 1100 0000 0000
Shift left:
  1 1111 1111 1111 0000 1011 1000 0000 0000
Add divisor: 0 0000 0000 0000 1000
Register:
  1 1111 1111 1111 1000 1011 1000 0000 0000
Set last quotient bit to 0:
  1 1111 1111 1111 1000 1011 1000 0000 0000
Shift left:
  1 1111 1111 1111 0001 0111 0000 0000 0000
Add divisor: 0 0000 0000 0000 1000
Register:
  1 1111 1111 1111 1001 0111 0000 0000 0000
Set last quotient bit to 0:
  1 1111 1111 1111 1001 0111 0000 0000 0000
Shift left:
  1 1111 1111 1111 0010 1110 0000 0000 0000
Add divisor: 0 0000 0000 0000 1000
Register:
  1 1111 1111 1111 1010 1110 0000 0000 0000
Set last quotient bit to 0:
  1 1111 1111 1111 1010 1110 0000 0000 0000
Shift left:
  1 1111 1111 1111 0101 1100 0000 0000 0000
Add divisor: 0 0000 0000 0000 1000
Register:
  1 1111 1111 1111 1101 1100 0000 0000 0000
Set last quotient bit to 0:
  1 1111 1111 1111 1101 1100 0000 0000 0000
Shift left:
  1 1111 1111 1111 1011 1000 0000 0000 0000
Add divisor: 0 0000 0000 0000 1000
Register:
  0 0000 0000 0000 0011 1000 0000 0000 0000
Set last quotient bit to 1:
  0 0000 0000 0000 0011 1000 0000 0000 0001
Shift left:
  0 0000 0000 0000 0111 0000 0000 0000 0010

```

```

Subtract divisor: 1 1111 1111 1111 1000
Register:
  1 1111 1111 1111 1111 0000 0000 0000 0010
Set last quotient bit to 0:
  1 1111 1111 1111 1111 0000 0000 0000 0010
Add divisor: 0 0000 0000 0000 1000
Register:
  0 0000 0000 0000 0111 0000 0000 0000 0010
Answer is:
  Remainder:      0 0000 0000 0000 0111 (in decimal: 7)
  Quotient:       0000 0000 0000 0010 (in decimal: 2)

```

3. Робота з IEEE 754 Floating Point (Представити лише ключові кроки при виконанні операцій)

b. Множення

- i. Compute exponents
- ii. Multiply significands
- iii. Normalize result
- iv. Set sign

```

Enter first number:
25,01
Enter second number:
8,321
Mantissa multiplication:
Mantissa1      0100100000001010001111011
                x
Mantissa2      000001010010001011010001
                =
Mantissa       01101000000011011101100111110000000111001101011

Normalization is not needed:

101000000001101110110011

Sign:
0 XOR 0 = 0

Exponent:
exp1          10000011 ( 131 )
              +
exp2          10000010 ( 130 )
              - 127 + 0
              = 10000110 ( 134 )
Result: 01000011010100000001101110110011 ( 208,1082 )

```

Висновок: Під час виконання даної лабораторної роботи були досліджені алгоритми, що використовуються в мікропроцесорах для множення та ділення цілих чисел та підходи до роботи з дійсними числами. Були створені програми, що ілюструють покрокове виконання алгоритму Бута, ділення (частка та залишок в одному регістрі) та множення дійсних чисел за допомогою IEEE 754 Floating Point