

Лабораторна робота № 2

Тема: «Арифметичні операції над двійковими числами»

Роботу виконав
студент 3 курсу КІ
Системний адміністратор
Костюченко Данило
Олександрович

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 2 та -6

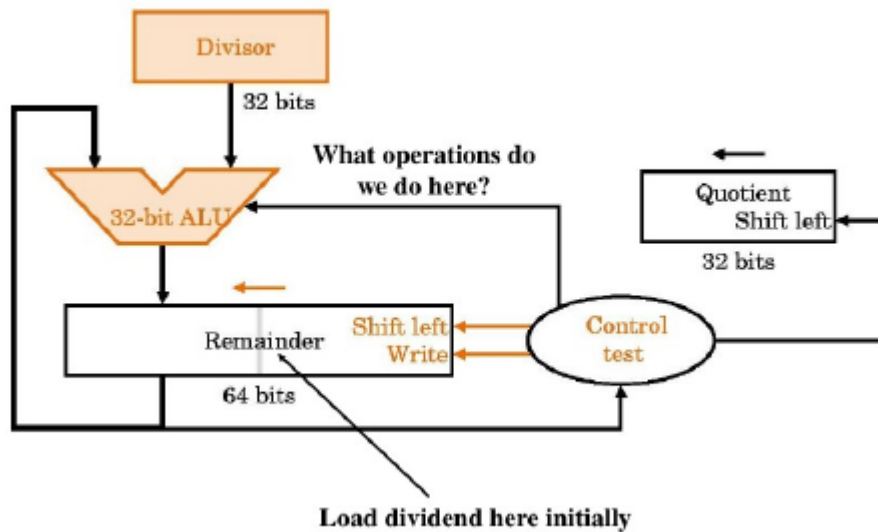
```
Enter first number:
2
Enter second number:
-6
Booth's algorithm:
Step 1:
  Shift right:      0000 0000 0000 0000 1111 1111 1111 1010 0
                   0000 0000 0000 0000 0111 1111 1111 1101 0
Step 2:
  Add S:   1111 1111 1111 1110 0000 0000 0000 0000 0
  To P:    0000 0000 0000 0000 0111 1111 1111 1101 0
  Shift right: 1111 1111 1111 1110 0111 1111 1111 1101 0
                   1111 1111 1111 1111 0011 1111 1111 1110 1
Step 3:
  Add A:    0000 0000 0000 0010 0000 0000 0000 0000 0
  To P:     1111 1111 1111 1111 0011 1111 1111 1110 1
  Shift right: 0000 0000 0000 0001 0011 1111 1111 1110 1
                   0000 0000 0000 0000 1001 1111 1111 1111 0
Step 4:
  Add S:     1111 1111 1111 1110 0000 0000 0000 0000 0
  To P:      0000 0000 0000 0000 1001 1111 1111 1111 0
  Shift right: 1111 1111 1111 1110 1001 1111 1111 1111 0
                   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
```

```

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. Зсув залишку вправо



```
*Ділення двійкович чисел (Зсув залишку вправо)*
-----
Введіть 16-бітове число (ділене):
6
Введіть 16-бітове число (дільник):
2
-----
Регістр:                0 0000 0000 0000 0000 0000 0000 0000 0110
Крок 0
Зсув ліворуч:           0 0000 0000 0000 0000 0000 0000 0000 1100
Віднімання дільника:    1 1111 1111 1111 1110
Регістр:                1 1111 1111 1111 1110 0000 0000 0000 1100
Додаємо дільник до регістру 0 0000 0000 0000 0000 0000 0000 0000 1100
Встановлюємо останній біт частки в 0:      0000 0000 0000 0000
-----
Крок 1
Зсув ліворуч:           0 0000 0000 0000 0000 0000 0000 0001 1000
Віднімання дільника:    1 1111 1111 1111 1110
Регістр:                1 1111 1111 1111 1110 0000 0000 0001 1000
Додаємо дільник до регістру 0 0000 0000 0000 0000 0000 0000 0001 1000
Встановлюємо останній біт частки в 0:      0000 0000 0000 0000
-----
Крок 2
Зсув ліворуч:           0 0000 0000 0000 0000 0000 0000 0011 0000
Віднімання дільника:    1 1111 1111 1111 1110
Регістр:                1 1111 1111 1111 1110 0000 0000 0011 0000
Додаємо дільник до регістру 0 0000 0000 0000 0000 0000 0000 0011 0000
Встановлюємо останній біт частки в 0:      0000 0000 0000 0000
-----
Крок 3
Зсув ліворуч:           0 0000 0000 0000 0000 0000 0000 0110 0000
Віднімання дільника:    1 1111 1111 1111 1110
Регістр:                1 1111 1111 1111 1110 0000 0000 0110 0000
Додаємо дільник до регістру 0 0000 0000 0000 0000 0000 0000 0110 0000
Встановлюємо останній біт частки в 0:      0000 0000 0000 0000
-----
```



```

Крок 13
Зсув ліворуч:      0 0000 0000 0000 0001 1000 0000 0000 0000
Віднімання дільника: 1 1111 1111 1111 1110
Регістр:           1 1111 1111 1111 1111 1000 0000 0000 0000
Додаємо дільник до регістру 0 0000 0000 0000 0001 1000 0000 0000 0000
Встановлюємо останній біт частки в 0:      0000 0000 0000 0000
-----
Крок 14
Зсув ліворуч:      0 0000 0000 0000 0011 0000 0000 0000 0000
Віднімання дільника: 1 1111 1111 1111 1110
Регістр:           0 0000 0000 0000 0001 0000 0000 0000 0000
Зсув ліворуч частки 0000 0000 0000 0000
Встановлюємо останній біт частки в 1:      0000 0000 0000 0001
-----
Крок 15
Зсув ліворуч:      0 0000 0000 0000 0010 0000 0000 0000 0000
Віднімання дільника: 1 1111 1111 1111 1110
Регістр:           0 0000 0000 0000 0000 0000 0000 0000 0000
Зсув ліворуч частки 0000 0000 0000 0010
Встановлюємо останній біт частки в 1:      0000 0000 0000 0011
-----
Регістр:      0 0000 0000 0000 0000 0000 0000 0000 0000
-----
Результат:
-----
Залишок:      0 0000 0000 0000 0000 (В десятковій системі: 0)
Частка:      0000 0000 0000 0011 (В десятковій системі: 3)

```

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

а. Додавання

- i. Align binary points
- ii. Add significands
- iii. Normalize result

```
C:\Users\conta\Desktop\KS\Kostiuchenko_LAB_2\CS-Lab-2-3\CS-Lab-2\CS-Lab-2-3>
a = 3
b = 1
Result exponent: 10000000
shifted a-mantisa: 00000000110000000000000000000000
shifted a-mantisa: 00000000010000000000000000000000
Add significands: 00000001000000000000000000000000
Exponent = 10000001
Mantisa = 0
My bits: 010000001000000000000000000000000000
Real bits: 010000001000000000000000000000000000
Real = 4
My = 4
```

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

<https://github.com/Lagranje/ComputerSystems>