Київський національний університет імені Тараса Шевченка
радіофізичний факультет

Лабораторна робота № 2

Тема**:** «Арифметичні операції над двійковими числами»

Роботу виконав
студент 3 курсу
КІ
Герасименко Артем

Київ 2019

Хід роботи

Створити програму, що ілюструє покрокове виконання наступних алгоритмів (за варіантами в Moodle).

Під покроковим виконанням мається на увазі вивід в двійковому представленні значень регістрів, що використовуються в процесі обрахунку на кожній ітерації, а також виводу самої логіки роботи алгоритму у вигляді опису (наприклад: “Значення регістру DIVISOR > 0: додаємо біт 0 до QUOTIENT, сзуваємо….”).

Код завантажте в свій репозиторій в GitHub.
В звіті навести приклад покрокового виконання кожного з варіантів, посилання на код та завантажити в Moodle.

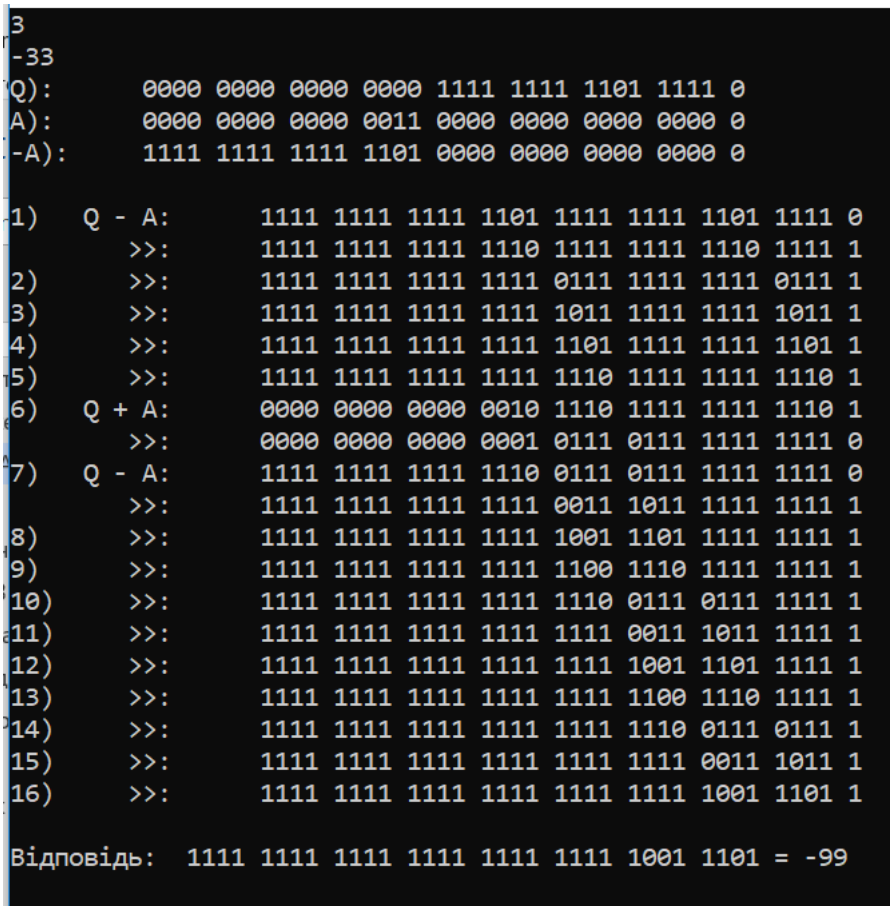
GitHub: https://github.com/Piikasooo/CS
https://github.com/Piikasooo/CS/tree/master/%D0%9A%D0%A1%D0%BB%D0%B0%D0%B1%D0%B02

Множення двійкових чисел

D: Алгоритм Бута

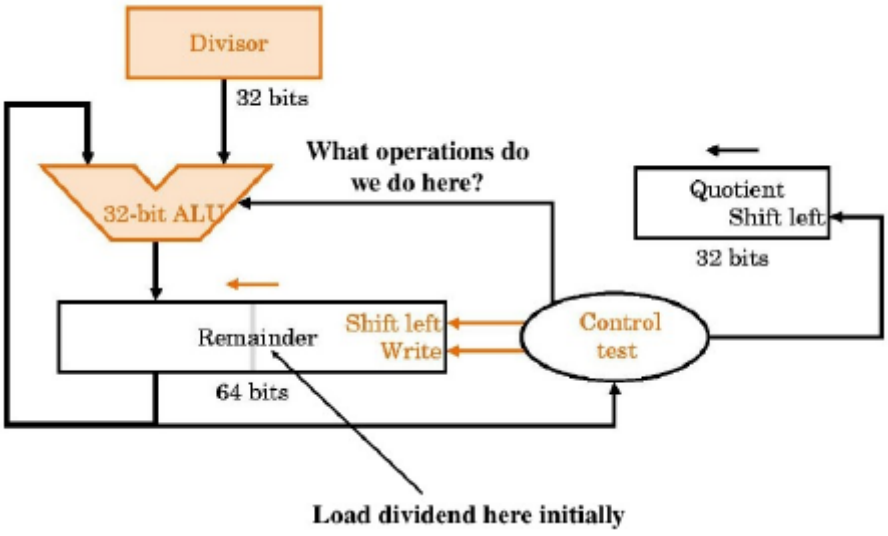
Алгоритм Бута використовується для пришвидчення множення двох двійкових знакових чисел завдяки заміні операції додавання операцією зсуву.

Приклад виконання програми для чисел 3 та -33



Ділення двійкових чисел

B: Зсув залишку вправо



Приклад виконання програми для чисел 9 та 3

```
Введіть 16-бітове число (ділене):
9
Введіть 16-бітове число (дільник):
3
Регістр:                    0 0000 0000 0000 0000 0000 0000 0000 1001
Крок 0
Зсув ліворуч:              0 0000 0000 0000 0000 0000 0000 0001 0010
Віднімання дільника:    1 1111 1111 1111 1101
Регістр:                    1 1111 1111 1111 1101 0000 0000 0001 0010
Додаємо дільник до регістру  0 0000 0000 0000 0000 0000 0000 0001 0010
Встановлюємо останній біт частки в 0:              0000 0000 0000 0000
------------------------------------------------------------------------
Крок 1
Зсув ліворуч:              0 0000 0000 0000 0000 0000 0000 0010 0100
Віднімання дільника:    1 1111 1111 1111 1101
Регістр:                    1 1111 1111 1111 1101 0000 0000 0010 0100
Додаємо дільник до регістру  0 0000 0000 0000 0000 0000 0000 0010 0100
Встановлюємо останній біт частки в 0:              0000 0000 0000 0000
------------------------------------------------------------------------
Крок 2
Зсув ліворуч:              0 0000 0000 0000 0000 0000 0000 0100 1000
Віднімання дільника:    1 1111 1111 1111 1101
Регістр:                    1 1111 1111 1111 1101 0000 0000 0100 1000
Додаємо дільник до регістру  0 0000 0000 0000 0000 0000 0000 0100 1000
Встановлюємо останній біт частки в 0:              0000 0000 0000 0000
------------------------------------------------------------------------
Крок 3
Зсув ліворуч:              0 0000 0000 0000 0000 0000 0000 1001 0000
Віднімання дільника:    1 1111 1111 1111 1101
Регістр:                    1 1111 1111 1111 1101 0000 0000 1001 0000
Додаємо дільник до регістру  0 0000 0000 0000 0000 0000 0000 1001 0000
Встановлюємо останній біт частки в 0:              0000 0000 0000 0000
------------------------------------------------------------------------
Крок 4
Зсув ліворуч:              0 0000 0000 0000 0000 0000 0001 0010 0000
Віднімання дільника:    1 1111 1111 1111 1101
Регістр:                    1 1111 1111 1111 1101 0000 0001 0010 0000
Додаємо дільник до регістру  0 0000 0000 0000 0000 0000 0001 0010 0000
Встановлюємо останній біт частки в 0:              0000 0000 0000 0000
------------------------------------------------------------------------
Крок 5
Зсув ліворуч:              0 0000 0000 0000 0000 0000 0010 0100 0000
Віднімання дільника:    1 1111 1111 1111 1101
Регістр:                    1 1111 1111 1111 1101 0000 0010 0100 0000
Додаємо дільник до регістру  0 0000 0000 0000 0000 0000 0010 0100 0000
Встановлюємо останній біт частки в 0:              0000 0000 0000 0000
------------------------------------------------------------------------
Крок 6
Зсув ліворуч:              0 0000 0000 0000 0000 0000 0100 1000 0000
Віднімання дільника:    1 1111 1111 1111 1101
Регістр:                    1 1111 1111 1111 1101 0000 0100 1000 0000
Додаємо дільник до регістру  0 0000 0000 0000 0000 0000 0100 1000 0000
Встановлюємо останній біт частки в 0:              0000 0000 0000 0000
------------------------------------------------------------------------
Крок 7
Зсув ліворуч:              0 0000 0000 0000 0000 0000 1001 0000 0000
Віднімання дільника:    1 1111 1111 1111 1101
Регістр:                    1 1111 1111 1111 1101 0000 1001 0000 0000
Додаємо дільник до регістру  0 0000 0000 0000 0000 0000 1001 0000 0000
Встановлюємо останній біт частки в 0:              0000 0000 0000 0000
------------------------------------------------------------------------
Крок 8
Зсув ліворуч:              0 0000 0000 0000 0000 0001 0010 0000 0000
Віднімання дільника:    1 1111 1111 1111 1101
Регістр:                    1 1111 1111 1111 1101 0001 0010 0000 0000
Додаємо дільник до регістру  0 0000 0000 0000 0000 0001 0010 0000 0000
Встановлюємо останній біт частки в 0:              0000 0000 0000 0000
------------------------------------------------------------------------
```

```
Крок 9
Зсув ліворуч:              0 0000 0000 0000 0000 0010 0100 0000 0000
Віднімання дільника:    1 1111 1111 1111 1101
Регістр:                   1 1111 1111 1111 1101 0010 0100 0000 0000
Додаємо дільник до регістру  0 0000 0000 0000 0000 0010 0100 0000 0000
Встановлюємо останній біт частки в 0:              0000 0000 0000 0000
--------------------------------------------------------------------
Крок 10
Зсув ліворуч:              0 0000 0000 0000 0000 0100 1000 0000 0000
Віднімання дільника:    1 1111 1111 1111 1101
Регістр:                   1 1111 1111 1111 1101 0100 1000 0000 0000
Додаємо дільник до регістру  0 0000 0000 0000 0000 0100 1000 0000 0000
Встановлюємо останній біт частки в 0:              0000 0000 0000 0000
--------------------------------------------------------------------
Крок 11
Зсув ліворуч:              0 0000 0000 0000 0000 1001 0000 0000 0000
Віднімання дільника:    1 1111 1111 1111 1101
Регістр:                   1 1111 1111 1111 1101 1001 0000 0000 0000
Додаємо дільник до регістру  0 0000 0000 0000 0000 1001 0000 0000 0000
Встановлюємо останній біт частки в 0:              0000 0000 0000 0000
--------------------------------------------------------------------
Крок 12
Зсув ліворуч:              0 0000 0000 0000 0001 0010 0000 0000 0000
Віднімання дільника:    1 1111 1111 1111 1101
Регістр:                   1 1111 1111 1111 1110 0010 0000 0000 0000
Додаємо дільник до регістру  0 0000 0000 0000 0001 0010 0000 0000 0000
Встановлюємо останній біт частки в 0:              0000 0000 0000 0000
--------------------------------------------------------------------
Крок 13
Зсув ліворуч:              0 0000 0000 0000 0010 0100 0000 0000 0000
Віднімання дільника:    1 1111 1111 1111 1101
Регістр:                   1 1111 1111 1111 1111 0100 0000 0000 0000
Додаємо дільник до регістру  0 0000 0000 0000 0010 0100 0000 0000 0000
Встановлюємо останній біт частки в 0:              0000 0000 0000 0000
--------------------------------------------------------------------
Крок 14
Зсув ліворуч:              0 0000 0000 0000 0100 1000 0000 0000 0000
Віднімання дільника:    1 1111 1111 1111 1101
Регістр:                   0 0000 0000 0000 0001 1000 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 0011 0000 0000 0000 0000
Віднімання дільника:    1 1111 1111 1111 1101
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)
```

Робота з IEEE 754 Floating Point

A: Додавання

```
Enter first float signed value:
5.865
Enter second float signed value:
-7.345
Adding -7.345 (a), to 5.865 (b)


  Convert "a" to binary (without exponent and normalization):
        1 | 00000000 | 01011000010100011110100
  Convert "b" to binary (without exponent and normalization):
        0 | 00000000 | 11011101011100001010000
  Normalize "a":
        1 | 10000001 | 11010110000101000111101
  Normalize "b" :
        0 | 10000001 | 01110111010111000010100
  Shift left "b" on 0:
        0 | 10000001 | 01110111010111000010100
  Adding "a" to "b":
        1 | 10000001 | 11010110000101000111101
      + 0 | 10000001 | 01110111010111000010100
  Answer is:
        In decemal: -1.48
        In binary: 1 | 01111111 | 01111010111000010100100
```

Висновок: у ході виконання лабораторної роботи я виконав різні арифметичні операції на двійковими числами, а саме множення алгоритмом Бута, ділення зсувом залишку вправо і також додавання чисел з плаваючою комою.