

**Київський національний університет імені Тараса Шевченка
факультет радіофізики, електроніки та комп'ютерних систем**

Лабораторна робота № 2

Тема: «Арифметичні операції над двійковими числами»

Роботу виконав
студент 3 курсу
КІ-СА
Мургашов Гліб

Київ 2020

Мета: Дослідити алгоритми, що використовуються в мікропроцесорах для множення та ділення цілих чисел та підходи до роботи з дійсними числами.

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....”).

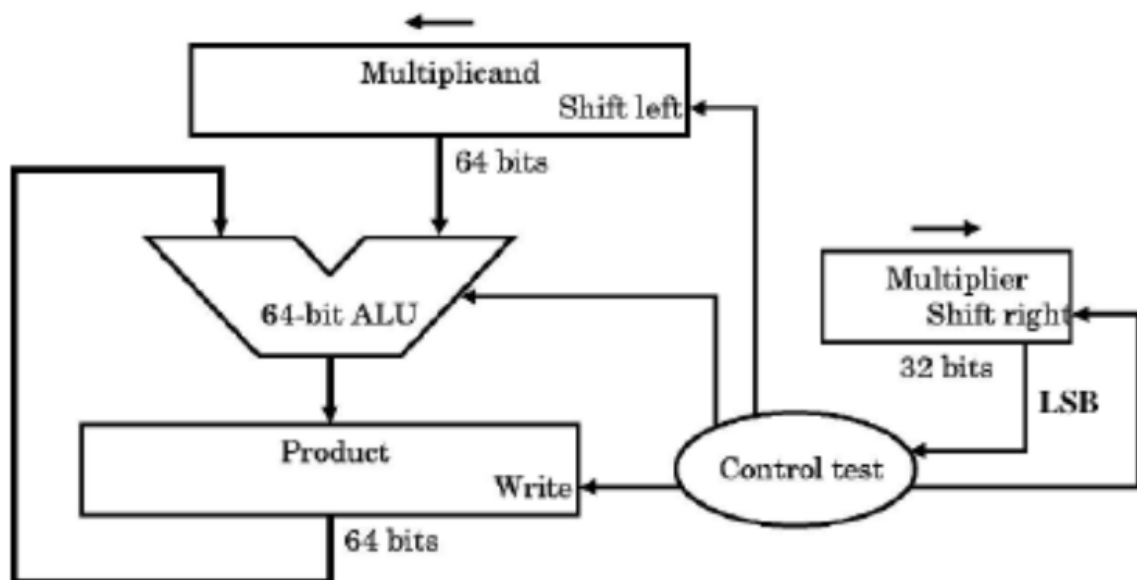
Код завантажте в свій репозиторій в GitHub.

В звіті навести приклад покрокового виконання кожного з варіантів, посилання на код та завантажити в Moodle.

HINT: використовуйте операції зсуву та додавання над змінними відповідно до алгоритму та виводьте значення змінних в бітовій формі.

1. Множення двійкових чисел

а. Множення як є



Консольний вивід програми:

```
Enter first number:756
Enter second number:456

1| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
2| Multiplicand:  0000 0000 0000 0000 0000 0000 0101 1111 0100
3| Multiplier:  0000 0000 0000 0000 0000 0001 1100 1000

4| Multiplicand leftshifted:  0000 0000 0000 0000 0000 0101 1110 1000
5| Multiplier rightshifted:  0000 0000 0000 0000 0000 0000 1110 0100
-----

6| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
7| Multiplicand:  0000 0000 0000 0000 0000 0000 0101 1110 1000
8| Multiplier:  0000 0000 0000 0000 0000 0000 1110 0100

9| Multiplicand leftshifted:  0000 0000 0000 0000 0000 1011 1101 0000
10| Multiplier rightshifted:  0000 0000 0000 0000 0000 0000 0111 0010
-----

11| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
12| Multiplicand:  0000 0000 0000 0000 0000 0000 1011 1101 0000
13| Multiplier:  0000 0000 0000 0000 0000 0000 0111 0010

14| Multiplicand leftshifted:  0000 0000 0000 0000 0001 0111 1010 0000
15| Multiplier rightshifted:  0000 0000 0000 0000 0000 0000 0011 1001
-----

16| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
17| Multiplicand:  0000 0000 0000 0000 0001 0111 1010 0000
18| Multiplier:  0000 0000 0000 0000 0000 0000 0011 1001

19| Product + multiplicand:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0111 1010 0000
20| Multiplicand leftshifted:  0000 0000 0000 0000 0000 0010 1111 0100 0000
21| Multiplier rightshifted:  0000 0000 0000 0000 0000 0000 0001 1100
-----

22| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0111 1010 0000
23| Multiplicand:  0000 0000 0000 0000 0000 0010 1111 0100 0000
24| Multiplier:  0000 0000 0000 0000 0000 0000 0001 1100

25| Multiplicand leftshifted:  0000 0000 0000 0000 0101 1110 1000 0000
26| Multiplier rightshifted:  0000 0000 0000 0000 0000 0000 0000 1110
-----

27| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0111 1010 0000
28| Multiplicand:  0000 0000 0000 0000 0101 1110 1000 0000
29| Multiplier:  0000 0000 0000 0000 0000 0000 0000 1110

30| Multiplicand leftshifted:  0000 0000 0000 0000 1011 1101 0000 0000
31| Multiplier rightshifted:  0000 0000 0000 0000 0000 0000 0000 0000 0111
-----

32| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0111 1010 0000
33| Multiplicand:  0000 0000 0000 0000 1011 1101 0000 0000
34| Multiplier:  0000 0000 0000 0000 0000 0000 0000 0000 0111

35| Product + multiplicand:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1101 0100 1010 0000
36| Multiplicand leftshifted:  0000 0000 0000 0001 0111 1010 0000 0000
37| Multiplier rightshifted:  0000 0000 0000 0000 0000 0000 0000 0011
-----

38| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1101 0100 1010 0000
39| Multiplicand:  0000 0000 0000 0001 0111 1010 0000 0000
40| Multiplier:  0000 0000 0000 0000 0000 0000 0000 0000 0011

41| Product + multiplicand:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010 0100 1110 1010 0000
42| Multiplicand leftshifted:  0000 0000 0000 0010 1111 0100 0000 0000
43| Multiplier rightshifted:  0000 0000 0000 0000 0000 0000 0000 0001
-----

44| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010 0100 1110 1010 0000
45| Multiplicand:  0000 0000 0000 0010 1111 0100 0000 0000
46| Multiplier:  0000 0000 0000 0000 0000 0000 0000 0000 0001

47| Product + multiplicand:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0101 0100 0010 1010 0000
48| Multiplicand leftshifted:  0000 0000 0000 0101 1110 1000 0000 0000
49| Multiplier rightshifted:  0000 0000 0000 0000 0000 0000 0000 0000
-----

50| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0101 0100 0010 1010 0000
51| Multiplicand:  0000 0000 0000 0101 1110 1000 0000 0000
52| Multiplier:  0000 0000 0000 0000 0000 0000 0000 0000

53| Multiplicand leftshifted:  0000 0000 0000 1011 1101 0000 0000 0000
54| Multiplier rightshifted:  0000 0000 0000 0000 0000 0000 0000 0000
-----

55| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0101 0100 0010 1010 0000
56| Multiplicand:  0000 0000 0000 1011 1101 0000 0000 0000
57| Multiplier:  0000 0000 0000 0000 0000 0000 0000 0000

58| Multiplicand leftshifted:  0000 0000 0001 0111 1010 0000 0000 0000
59| Multiplier rightshifted:  0000 0000 0000 0000 0000 0000 0000 0000
-----

60| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0101 0100 0010 1010 0000
61| Multiplicand:  0000 0000 0001 0111 1010 0000 0000 0000
62| Multiplier:  0000 0000 0000 0000 0000 0000 0000 0000

63| Multiplicand leftshifted:  0000 0000 0010 1111 0100 0000 0000 0000
64| Multiplier rightshifted:  0000 0000 0000 0000 0000 0000 0000 0000
-----
```

[illegible]

2.

1

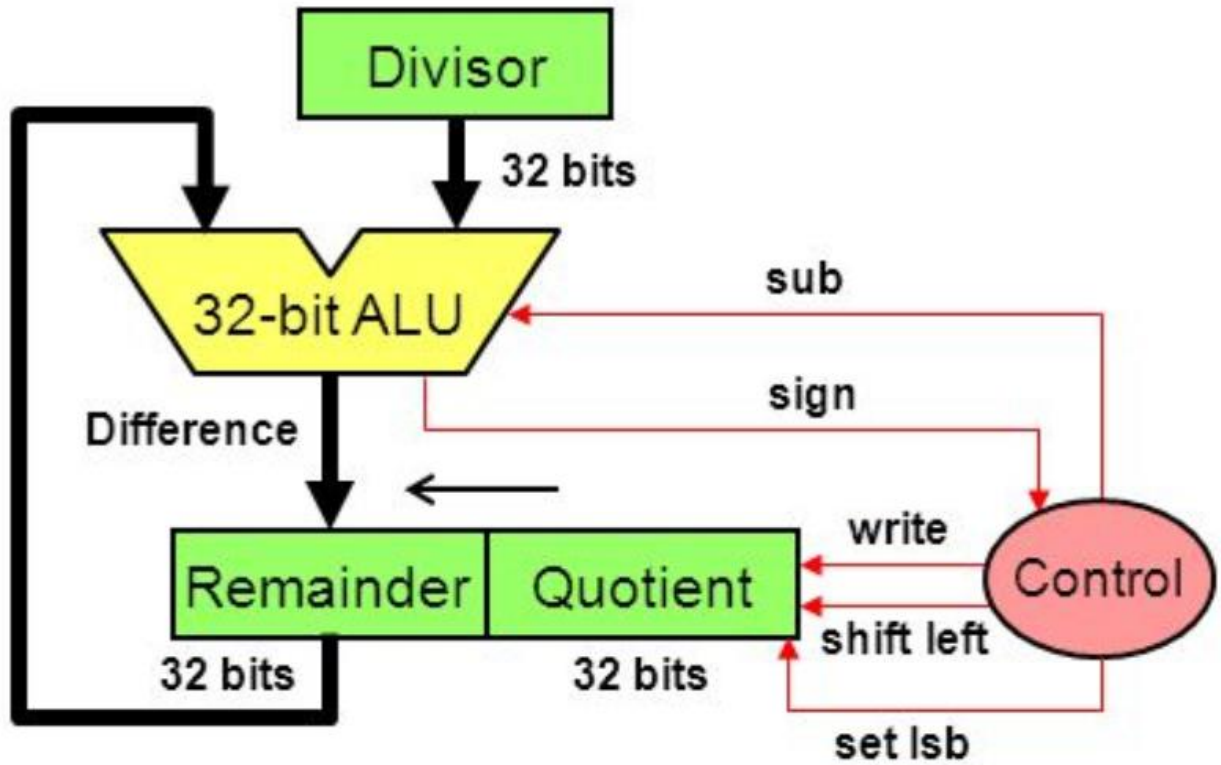
```

-----
125| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0101 0100 0010 1010 0000
126| Multiplicand: 1111 0100 0000 0000 0000 0000 0000 0000 0000
127| Multiplier:  0000 0000 0000 0000 0000 0000 0000 0000
-----
128| Multiplicand leftshifted: 1110 1000 0000 0000 0000 0000 0000 0000
129| Multiplier rightshifted: 0000 0000 0000 0000 0000 0000 0000 0000
-----
130| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0101 0100 0010 1010 0000
131| Multiplicand: 1110 1000 0000 0000 0000 0000 0000 0000 0000
132| Multiplier:  0000 0000 0000 0000 0000 0000 0000 0000
-----
133| Multiplicand leftshifted: 1101 0000 0000 0000 0000 0000 0000 0000
134| Multiplier rightshifted: 0000 0000 0000 0000 0000 0000 0000 0000
-----
135| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0101 0100 0010 1010 0000
136| Multiplicand: 1101 0000 0000 0000 0000 0000 0000 0000 0000
137| Multiplier:  0000 0000 0000 0000 0000 0000 0000 0000
-----
138| Multiplicand leftshifted: 1010 0000 0000 0000 0000 0000 0000 0000
139| Multiplier rightshifted: 0000 0000 0000 0000 0000 0000 0000 0000
-----
140| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0101 0100 0010 1010 0000
141| Multiplicand: 1010 0000 0000 0000 0000 0000 0000 0000 0000
142| Multiplier:  0000 0000 0000 0000 0000 0000 0000 0000
-----
143| Multiplicand leftshifted: 0100 0000 0000 0000 0000 0000 0000 0000
144| Multiplier rightshifted: 0000 0000 0000 0000 0000 0000 0000 0000
-----
145| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0101 0100 0010 1010 0000
146| Multiplicand: 0100 0000 0000 0000 0000 0000 0000 0000 0000
147| Multiplier:  0000 0000 0000 0000 0000 0000 0000 0000
-----
148| Multiplicand leftshifted: 1000 0000 0000 0000 0000 0000 0000 0000
149| Multiplier rightshifted: 0000 0000 0000 0000 0000 0000 0000 0000
-----
150| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0101 0100 0010 1010 0000
151| Multiplicand: 1000 0000 0000 0000 0000 0000 0000 0000 0000
152| Multiplier:  0000 0000 0000 0000 0000 0000 0000 0000
-----
153| Multiplicand leftshifted: 0000 0000 0000 0000 0000 0000 0000 0000
154| Multiplier rightshifted: 0000 0000 0000 0000 0000 0000 0000 0000
-----
155| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0101 0100 0010 1010 0000
156| Multiplicand: 0000 0000 0000 0000 0000 0000 0000 0000 0000
157| Multiplier:  0000 0000 0000 0000 0000 0000 0000 0000
-----
158| Multiplicand leftshifted: 0000 0000 0000 0000 0000 0000 0000 0000
159| Multiplier rightshifted: 0000 0000 0000 0000 0000 0000 0000 0000
-----
160| Product:  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0101 0100 0010 1010 0000
161| Multiplicand: 0000 0000 0000 0000 0000 0000 0000 0000 0000
162| Multiplier:  0000 0000 0000 0000 0000 0000 0000 0000
-----
163| Multiplicand leftshifted: 0000 0000 0000 0000 0000 0000 0000 0000
164| Multiplier rightshifted: 0000 0000 0000 0000 0000 0000 0000 0000
-----
165| result: 344736
Binary: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0101 0100 0010 1010 0000
Для продолжения нажмите любую клавишу . . .

```

Ділення двійкових чисел

с. Частка та залишок в одному регістрі



Консольний вивід програми:


```
Enter first number:4094
Enter second number:89
```

[illegible][illegible]

```

25| Divisor:  0000 0000 0000 0000 0000 0000 0101 1001 0000 0000 0000 0000 0000 0000 0000 0000
Remainder + quotient: 0000 0000 0000 0000 0000 0000 0001 1111 1111 1100 0000 0000 0000 0000 0000 0000
-----

26| Divisor:  0000 0000 0000 0000 0000 0000 0101 1001 0000 0000 0000 0000 0000 0000 0000 0000
Remainder + quotient: 0000 0000 0000 0000 0000 0000 0011 1111 1111 1000 0000 0000 0000 0000 0000 0000
-----

27| Divisor:  0000 0000 0000 0000 0000 0000 0101 1001 0000 0000 0000 0000 0000 0000 0000 0000
Remainder + quotient: 0000 0000 0000 0000 0000 0000 0111 1111 1111 0000 0000 0000 0000 0000 0000 0000
-----

28| Divisor:  0000 0000 0000 0000 0000 0000 0101 1001 0000 0000 0000 0000 0000 0000 0000 0000
Remainder + quotient: 0000 0000 0000 0000 0000 0000 0100 1101 1110 0000 0000 0000 0000 0000 0000 0001
-----

29| Divisor:  0000 0000 0000 0000 0000 0000 0101 1001 0000 0000 0000 0000 0000 0000 0000 0000
Remainder + quotient: 0000 0000 0000 0000 0000 0000 0000 1001 1011 1100 0000 0000 0000 0000 0000 0010
-----

30| Divisor:  0000 0000 0000 0000 0000 0000 0101 1001 0000 0000 0000 0000 0000 0000 0000 0000
Remainder + quotient: 0000 0000 0000 0000 0000 0000 0000 1000 0101 1000 0000 0000 0000 0000 0000 0101
-----

31| Divisor:  0000 0000 0000 0000 0000 0000 0101 1001 0000 0000 0000 0000 0000 0000 0000 0000
Remainder + quotient: 0000 0000 0000 0000 0000 0000 0000 0101 1001 0000 0000 0000 0000 0000 0000 1011
-----

32| Divisor:  0000 0000 0000 0000 0000 0000 0101 1001 0000 0000 0000 0000 0000 0000 0000 0000
Remainder + quotient: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0111
-----

33| Divisor:  0000 0000 0000 0000 0000 0000 0101 1001 0000 0000 0000 0000 0000 0000 0000 0000
Remainder + quotient: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010 1110
-----

34| Quotient: 0000 0000 0000 0000 0000 0000 0010 1110[46]
35| Remainder: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000[0]
Для продолжения нажмите любую клавишу . . .

```

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

Консольний вивід програми:


```

Enter first number:13
Enter second number:56
adding 56 (a) to 13 (b)

Convert "a" to binary:
0 . 00000000 . 000000000000000000000000
Convert "b" to binary:
0 . 00000000 . 000000000000000000000000
Normalize "a":
0 . 10000100 . 110000000000000000000000
Normalize "b":
0 . 10000010 . 101000000000000000000000
Left carry "b" on 2:
0 . 10000010 . 011010000000000000000000
Add "a" to "b":
  0 . 10000100 . 110000000000000000000000
+0 . 10000010 . 011010000000000000000000
Product :
decimal: 69
binary: 0 . 10000101 . 000101000000000000000000
Для продовження натисніть будь-яку клавішу . . . |

```

Висновок: В даній лабораторній роботі були освоєні навички роботи з алгоритмами, які використовуються для множення, ділення, додавання в мікропроцесорах. Були написані програми, які виконують певні операції та покроково виводять результат на консоль.

Git: https://github.com/HleBASS/CS_labs