

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования

**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)**

Факультет систем управления (ФСУ)

Кафедра автоматизированных систем управления (АСУ)

Основы языка Ассемблер. Арифметические операции.

Отчет по лабораторной работе №3 по дисциплине
«Вычислительная техника»

Студент гр. 431-3
_____ Гурулёв А.В.
«24» октября 2022 г.

Руководитель
_____ Алфёров С.М.
«___» _____ 2022 г.

Томск 2022

Оглавление

ВВЕДЕНИЕ	3
1 РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ	4
ВЫВОД	5
Приложение А	6

ВВЕДЕНИЕ

Цель работы - Получение навыков работы с командами арифметики (add, adc, sub, sbb, mul, imul) add, adc, sub, sbb, mul, imul) и преобразования данных (add, adc, sub, sbb, mul, imul) cbw, cwd).

Задание:

- 7) A (add, adc, sub, sbb, mul, imul) int), B (add, adc, sub, sbb, mul, imul) dword), C (add, adc, sub, sbb, mul, imul) word).

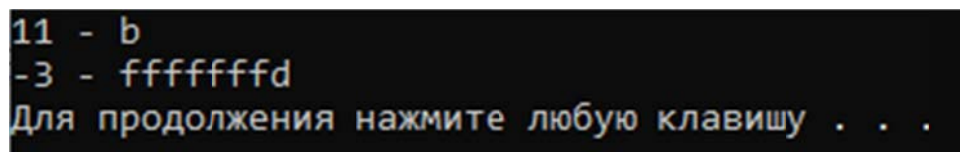
Вычислить:

$$D = (\text{add, adc, sub, sbb, mul, imul}) C/A * C + B;$$

$$E = (\text{add, adc, sub, sbb, mul, imul}) B - C / A * C.$$

1 РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ

На рисунке 1.1 представлен результат работы программы.



```
11 - b  
-3 - ffffffff  
Для продолжения нажмите любую клавишу . . .
```

Рисунок 1.1 - Результат работы программы

ВЫВОД

Я получил навыки работы с командами арифметики (add, adc, sub, sbb, mul, imul) add, adc, sub, sbb, mul, imul) и преобразования данных (add, adc, sub, sbb, mul, imul) cbw, cwd).

Приложение А

Листинг кода

```
#include <iostream>
using namespace std;
//7 Вариант
int main()
{
    _int32 a = 1;           //EAX
    _int32 b = 2;           //EAX
    _int16 c = 3;           //AX

    _int32 d = 0;           //EAX (c/a)*c+b
    _int32 e = 0;           //EAX (b-c)/a*c
    _asm //Нахождение d
    {
        mov ax, c;
        cwde;               //Помещаем C в EAX

        mov edx, 0;         //Помещаем в поле остатка 0
        div a;               //C/A
        mov d, eax;         //Сохраняем результат

        mov ax, c;          //Расширяем C для перемножения
        cwde;

        mul d;               //Перемножаем EAX на D

        add ax, word ptr b;  //Складываем младшие части
        add dx, word ptr b[2]; //Складываем старшие части

        mov word ptr d, ax;  //Сохраняем результат
        mov word ptr d[2], dx;
    }

    cout << d << " - " << hex << d << endl;

    bool f;

    _asm //Нахождение e
    {
        mov ax, c;           //Вычитаем C из B
        cwde;
        sbb b, eax;
        mov eax, b;

        cmp eax, 0;          //Если EAX меньше чем 0
        jnl pos1;

        neg eax;              //То меняем знак и ставим флаг
        mov f, 1;

        pos1:                //Иначе - нет

        mov edx, 0;          //Помещаем в поле остатка 0
        idiv a;               //Делим результат на A
        mov e, eax;          //Сохраняем результат

        mov ax, c;           //Расширяем C до dw
        cwde;

        imul e;               //Умножаем на C
    }
}
```

```

        mov word ptr e, ax;          //Сохраняем результат
        mov word ptr e[2], dx;
    }

    if (f)
    {
        cout << dec << -e << " - " << hex << -e << endl;
    }
    else
    {
        cout << dec << e << " - " << hex << e << endl;
    }

    system("pause");
    return 0;
}

```