

ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

доцент, канд.тех.наук_

подпись, дата

Попов А.А.

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

Принципы организации программ

по дисциплине: АРХИТЕКТУРА ЭВМ И СИСТЕМ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТГР. №

4936

подпись, дата

Назаров М.Р.

фамилия, инициалы

Санкт-Петербург 2021

Лабораторная работа 1. Принципы организации программ

Цель работы:

Освоение принципов построения приложений на языке ассемблера для системы Texas Instruments, ознакомление с командами и правилами построения программ, ознакомление с методикой проектирования программ в среде программирования.

Задание на лабораторную работу:

В ходе выполнения лабораторной работы необходимо разработать программу, в соответствии с заданием. Отчет по лабораторной работе должен содержать описание индивидуального задания, граф схемы алгоритмов с их описанием, текст программы с соответствующими комментариями и пример результатов работы.

Варианты заданий:

Вариант №13:

unsigned int

13. $(A * B - C) + D$

Ход работы:

Создадим граф схему алгоритма:



Напишем код программы с комментариями:

```
.global _c_int00 ;точка входа
_c_int00:
.data
a .uint 4 ;A = 4
b .uint 2 ;B = 2
c .uint -8 ;C = -8
d .uint 20 ;D= 20
.text
MVK .S1 a,A0 ;Загрузить адрес регистра a в A0
MVK .S1 b,A1 ;Загрузить адрес регистра b в A1
MVK .S1 c,A2 ;Загрузить адрес регистра c в A2
MVK .S1 d,A3 ;Загрузить адрес регистра d в A3
LDW .D1 *A0, A0 ;Выгрузить значение регистра a в A1
LDW .D1 *A1, A1 ;Выгрузить значение регистра b в A2
LDW .D1 *A2, A2 ;Выгрузить значение регистра c в A3
LDW .D1 *A3, A3 ;Выгрузить значение регистра d в A4
NOP 4
MPY .M1 A0,A1,A1 ;умножение AxB, результат (8) в A1;
NOP 1
SUB .L1 A1, A2,A2 ;вычисляем разность AxB-C, результат (16) в A2;
ADD .L1 A2,A3,A3 ; складываем значение по модулю (AxB-C)+D; результат (36)
в A3;
```

Проверим работу программы:

Начальное состояние программы (все регистры пустые).

Name	Value	Description
Core Registers		
A0	0x00000000	Core
A1	0x00000000	Core
A2	0x00000000	Core
A3	0x00000000	Core

Записанные адреса ячеек памяти в первые регистры.

Name	Value	Description
Core Registers		
A0	0x00000040	Core
A1	0x00000044	Core
A2	0x00000048	Core
A3	0x0000004C	Core






После выгрузки значений в соответствующие регистры и ожидания четырех тактов

Name	Value	Description
Core Registers		
A0	0x00000004	Core
A1	0x00000002	Core
A2	0xFFFFFFFF8	Core
A3	0x00000014	Core

В регистр A1 записываем результат суммы значения регистров A0 * A1.






Name	Value	Description
Core Registers		
A0	0x00000004	Core
A1	0x00000008	Core
A2	0xFFFFFFFF8	Core
A3	0x00000014	Core

В регистр A2 запишем результат произведения значения регистров A1 - A2.

Name	Value	Description
▼  Core Registers		
 A0	0x00000004	Core
 A1	0x00000008	Core
 A2	0x00000010	Core
 A3	0x00000014	Core

$0x10 = 16$

В регистр A3 запишем результат суммы значений A2 + A3, что будет конечным результатом программы.

Name	Value	Description
▼  Core Registers		
 A0	0x00000004	Core
 A1	0x00000008	Core
 A2	0x00000010	Core
 A3	0x00000024	Core

$0x24 = 36$

Вывод: в ходе лабораторной работы были получены основные навыки работы с таким низкоуровневым языком программирования как assembler, так же были изучены и применены на практике основные команды и директивы языка.