

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО
ПРИБОРОСТРОЕНИЯ»

ИНСТИТУТ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И ПРОГРАММИРОВАНИЯ

КАФЕДРА компьютерных технологий и программной инженерии

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Старший преподаватель
должность, уч. степень, звание

подпись, дата

Николаев Д.А.
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3
по курсу: Архитектура ЭВМ и систем

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4932

подпись, дата

Н.С. Иванов
инициалы, фамилия

Санкт-Петербург 2021

Цель работы

Освоение принципов построения приложений на языке ассемблера для системы Texas Instruments, ознакомление с командами и правилами построения программ в соответствии с особенностями конвейерного и параллельного выполнения команд.

Задание:

7. Факториал

Текст программы:

; 7. factorial ; 68 takt

```
.ref _c_int00
_c_int00:
```

```
.data
```

```
array1: .int 1, 4;
size .set 2
```

```
.text
```

```
    MVK .S1 1,      A0 ; result for element array
    MVK .S1 size,   A3 ; index array
    MVKL .S1 array1, A4 ; load in left part
    MVKH .S1 array1, A4 ; load in right part

    MVK .S1 0,      A5 ; a
```

```
LOOP:
```

```
    SUB .S1 A3, 1, A3    ; A3 = A3 - 1 ; index array

    ; A3 != 0
    SUB .S1 A1, A1, A1    ; reset
    ADD .S1 A3, 0,  A1    ; move A3 -> A1
    [!A1] B .S1 FINAL    ;
    NOP 5

    LDW .D1 *A4[A3], A5 ; load element in A5
    NOP 4

    SUB .S1 A0, A0, A0 ; A0 := 1
```

```

    ADD .S1 A0, 1, A0 ; A0 := 1
WHILE:
    ; A5 != 1
    SUB .S1 A1, A1, A1 ; reset
    ADD .S1 A5, 0, A1 ; move A5 -> A1
    SUB .S1 A1, 1, A1 ; A1 - 1
    [!A1] B .S1 LOOP ;
    NOP 5

    MPY .M1 A0, A5, A0 ; *=
|| SUB .S1 A5, 1, A5 ; A5 -= 1

    ; A5 != 1
    SUB .S1 A1, A1, A1 ; reset
    ADD .S1 A5, 0, A1 ; move A5 -> A1
    SUB .S1 A1, 1, A1 ; A1 - 1
    [!A1] B .S1 LOOP ;
    NOP 5

    B .S1 WHILE

```

FINAL:

Выводы

- Использование параллельного режима выполнения команд может значительно сократить количество тактов процессора, необходимых для исполнения алгоритмов, однако во время выполнения работы количество тактов удалось сократить лишь на 2, поскольку алгоритм достаточно прост.