|  |  |
| --- | --- |
| ГУАП  КАФЕДРА № 43  ОТЧЕТ  ЗАЩИЩЕН С ОЦЕНКОЙ  ПРЕПОДАВАТЕЛЬ |  |
|  |  |
| доцент, канд.тех.наук подпись, дата | Попов А.А. |

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2

Использование циклов

по дисциплине: АРХИТЕКТУРА ЭВМ И СИСТЕМ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТГР. № 4936 Петровнина Д.В.

подпись, дата фамилия, инициалы

Санкт-Петербург 2021

**Лабораторная работа 2. Использование циклов**

**Цель работы:**

Освоение принципов построения приложений на языке ассемблера для системы Texas Instruments, ознакомление с командами и правилами построения программ в соответствии с особенностями организации циклов.

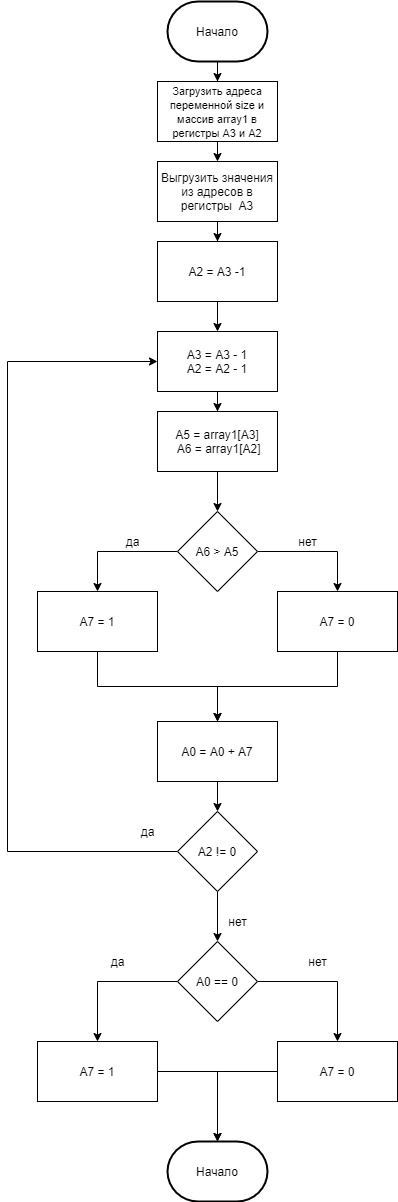
**Задание на лабораторную работу:**

освоение принципов построения приложений на языке ассемблера для системы Texas Instruments, ознакомление с командами и правилами построения программ в соответствии с особенностями организации циклов.

**Варианты заданий:**

Вариант №16: **signed bit**  
Проверить является ли массив упорядоченным по возрастанию. Результат проверки записать в регистр.

**Ход работы:**

Создадим граф схему алгоритма:

Напишем код программы с комментариями:

**.global** \_c\_int00 ;точка входа

**\_c\_int00:**

**.data** ;секция данных

size **.set** 4 ;размер массива(>1)(препроцессорная константа)

**array1:** **.byte** 1,2,3,4 ;создаем массив 32 разрядных чисел

**.text** ;секция кода

MVK .S1 size,A3 ;загружаем колво элементов массива в A3

MVK .S1 0, A2

MVKL .S1 array1,A4 ;загружаем адрес массива1 в A4

MVKH .S1 array1,A4

**NOP** 2

**SUB** .L1 A3,1,A2 ; индекс следующего элемента массива

**LOOP:**

**SUB** .L1 A3, 1, A3 ;A2 := A2 - 1

**SUB** .L1 A2, 1, A2 ;A3 := A3 - 1

LDB .D1 \*A4[A3], A5 ;загружаем текущий элемент в A5

LDB .D1 \*A4[A2], A6 ;загружаем следующий элемент в A6

**NOP** 4 ;4х тактовая задержка загрузки

CMPGT .L1 A6, A5, A7 ;Если A6 > A5, то записать 1-> A7, иначе 0-> A7

**ADD** .L1 A0,A7,A0 ; А0 + А7

[A2] B .S1 LOOP ;переход если A2 <> 0

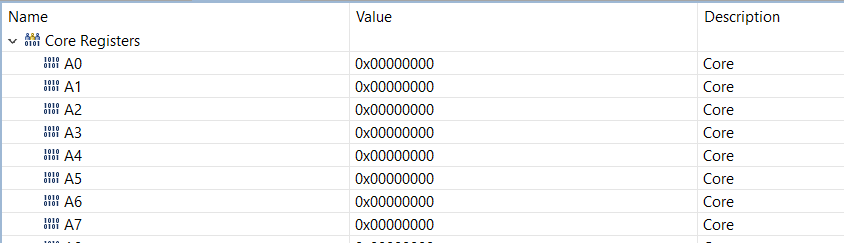
**NOP** 5

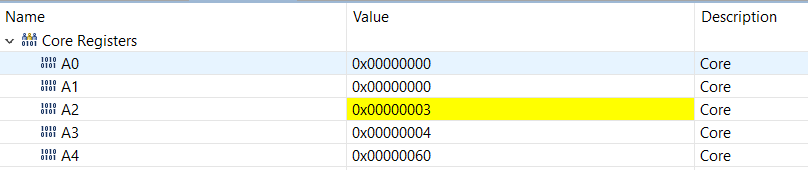
CMPEQ A0, 0, A7 ;если А0 == 0, то записать 1 -> А7 - массив упорядочен. Иначе записать 0 -> A7 - массив неупорядочен

**NOP** 5

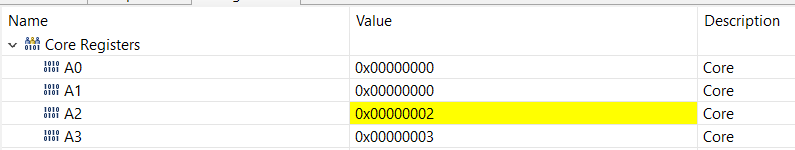
**Проверка массива, упорядоченного по возрастанию.**

Массив: **array1: .byte 1,2,3,4**

Начальное состояние программы (все регистры пустые)

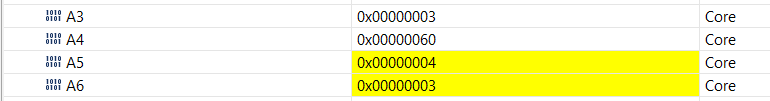
Записанные значения и адреса ячеек памяти в первые регистры  


Уменьшаем значения индексов текущего(регистр А3) и следующего(регистр А2) элемента на 1

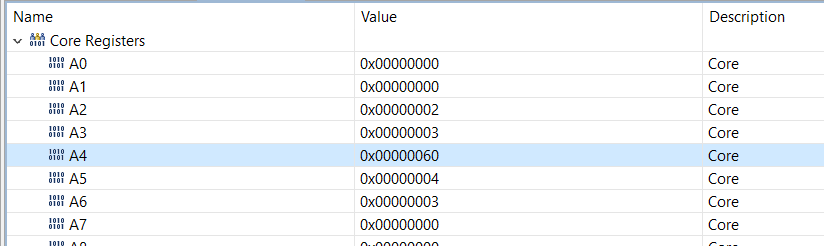


В регистр А5 записываем значение текущего элемента, в регистр А6 — следующего  
array1[A3] = 4

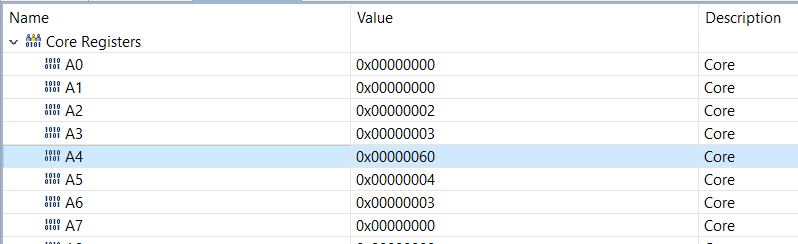
array1[A2] = 3



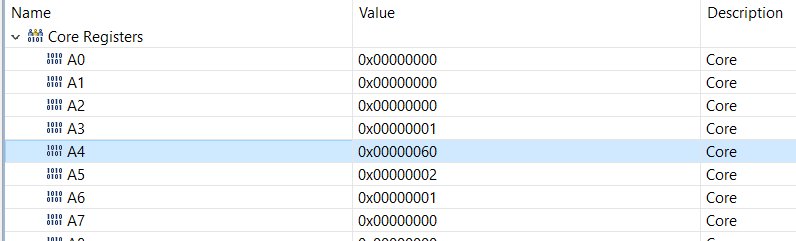
В регистр А7 запишем результат сравнения текущего и следующего элемента: если следующий элемент больше текущего, записываем 1 в А7 (элементы не упорядочены), если меньше — записываем 0 (элементы упорядочены)

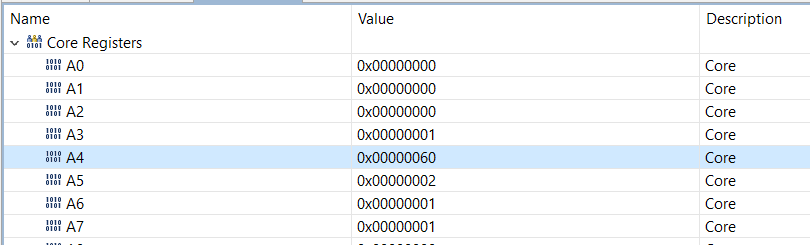


Суммируем значения из регистров А0 и А7, и записываем в А0



Пройдём по цикле ещё несколько раз, пока индекс следующего элемента в регистре А2 не станет равным нулю.



Проверим результат, полученный в ячейке А0: если А0 = 0, то записываем 1 в А7 — массив упорядочен, если А0 > 0, записываем 0 в А7 — массив не упорядочен  


Результат: А7 = 1 => массив упорядочен **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.