Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт инженерной физики и радиоэлектроники

Кафедра «Радиоэлектронная техника информационных систем»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

Дисциплина «Цифровые устройства и микропроцессоры»

Лабораторная работа №1

Изучение языка ассемблера, ядра Cortex-M3 и среды

разработки Keil MDK

Вариант 10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Руководитель |  |  |  |  | Батурин Т. Н. |
|  |  |  | подпись, дата |  | инициалы, фамилия |
|  |  |  |  |  |  |
| Студент | РФ20-32Б 051939486 |  |  |  | Левенец А.А. |
|  | номер группы, зачетной книжки |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2023

1. **Цель работы**

Освоение инструкций ассемблера микропроцессора Cortex-M3 и программной среды Keil MDK.

1. **Задание к лабораторной работе**

В одномерном массиве найти последний положительный элемент и заменить его на среднее арифметическое значение.

1. **Ход выполнения задания**

Для выполнения задания была составлена блок-схема алгоритма нахождения последнего положительного элемента в массиве и заменой его на среднее арифметическое. Блок-схема представлена на рисунке 1. Реализация блок-схемы на языке ассемблера представлена в приложениях А, Б и В.



Рисунок 1 – Блок-схема программы для реализации задания

В результате выполнения программы заданный произвольный массив заменяется на массив, в котором последний положительный элемент будет заменён на среднее арифметическое. В данном случае массив mas=[-1, 7, 9, 9, -3, 0, 4, 5, -1, -1] заменяется на массив, представленный на рисунке 2.

****

Рисунок 2 – Результат работы программы

**ПРИЛОЖЕНИЕ А**

Листинг 1. Файл main.s

GET stm32\_EQU.s

MAX\_SIZE EQU 0xA

; Объявляем сегмент констант

AREA CONSTANT\_FLASH, DATA, READONLY

; Здесь объявляются константы

mas DCB -0x01, 0x07, 0x09, 0x09, -0x03, 0x00, 0x04, 0x05, -0x01, -0x01

; Объявляем сегмент переменных

negative DCB -0x05, 0x05

AREA VERIABLE\_RAM, DATA, READWRITE

; Здесь объявляются переменные

result SPACE 0xA

; Объявляем сегмент кода

AREA MAIN, CODE, READONLY

THUMB

; Объявляем функцию main

main PROC

MOV R0, #NULL

MOV R1, #ONE

LDR R2, =negative

LDRB R7, [R2,R0]

LDRB R4, [R2,R1]

ADD R7, R4

; Здесь пишется ваша программа

; Пример программы

LDR R0, =mas ; МАССИВ

LDR R1, =result ; АДРЕССА МАССИВА

MOV R2, #MAX\_SIZE ;КОЛИЧЕСТВО ЭЛЕМЕНТОВ

MOV R3, #NULL ; ИСПОЛЬЗУЕТСЯ КАК СЧЕТЧИК

MOV R4, #NULL ; АДРЕС ПОСЛЕДНЕГО ПОЛОЖИТЕЛЬНОГО ЭЛЕМЕНТА

MOV R5, #NULL ; ТЕКУЩИЙ ЭЛЕМЕНТ

MOV R6, #NULL ; СУММУ МАССИВ

CHECK

CMP R3, #MAX\_SIZE ; ПРОВЕРКА

BLT SYM

B EXIT

SYM

LDRB R5, [R0,R3] ;В R5 КЛАДЁМ ЧИСЛО С МАССИВА

SXTB R8, R5

ADD R6, R8 ;СУММА МАССИВА

STRB R5, [R1,R3] ;ЗАПИСЫВАЕМ ЧИСЛО С R5 В МАССИВ R1

ADD R3, #ONE

CMP R8, #NULL ;СРАВНЕНИЕ ОТРИЦАТЕЛЬНОЕ ИЛИ ПОЛОЖИТЕЛЬНОЕ

BLT CHECK

SUB R3, #ONE

MOV R4, R3

ADD R3, #ONE

B CHECK

EXIT

SDIV R6, R6,R2

STRB R6, [R1,R4]

ENDP ; Конец функции main

END ; Конец файла

**ПРИЛОЖЕНИЕ Б**

Листинг 2. Файл stm32\_EQU.s

; Определение макроподстановок

STACK EQU 0x20000000 ; адрес стека

STACK\_SIZE EQU 0x500 ; размер стэка

STACK\_TOP EQU STACK+STACK\_SIZE ; вершина стека

NULL EQU 0x00

ONE EQU 0x01

FOUR EQU 0x04

MAX EQU 0x100

END

**ПРИЛОЖЕНИЕ В**

Листинг 3. Файл startup.s

GET main.s ; Вставка файла main.s

; Директива PRESERVE8 указывает, что текущий файл требует или сохраняет восьми байтовое выравнивание стека

PRESERVE8

; Директива THUMB инструктирует ассемблер интерпретировать последующие инструкции,как инструкции Thumb используя синтаксис UAL.

THUMB

; Объявляем стартовый сегмент кода

AREA RESET, CODE, READONLY

; Таблица векторов прерываний

DCD STACK\_TOP ; Указатель на вершину стека

DCD startup ; Вектор сброса

; Точка входа

ENTRY

; Startup код

startup PROC ; Начало startup кода

; Инициализация МК

; Переход в программу main

B main

ENDP ; Конец программы

END ; Конец файла