**2 ЛАБОРАТОРНАЯ РАБОТА №2**

**«ИССЛЕДОВАНИЕ АРХИТЕКТУРЫ И СИСТЕМЫ КОМАНД ВОСЬМИРАЗРЯДНОГО МИКРОПРОЦЕССОРА»**

2.1 Цель работы

Исследовать архитектуру и основные блоки 8-разрядного процессора. Исследовать взаимодействие основных блоков процессора при выполнении команд разных типов. Приобрести навыки написания и отладки ассемблерных программ в эмуляторе KP580 Emulator.

2.2 Ход выполнения работы

2.2.1 В начале лабораторной работы была изучена архитектура МП КР580ВМ80. Помимо этого был изучен основной набор команд для этого микропроцессора, а также изучены возможности эмулятора и экранного отладчика kp580\_new. Далее было исследовано изменение в основных блоках процессора в ходе выполнения команд различных типов. В соответствии с заданным вариантом была составлена блок-схема алгоритма, которая показана на рисунке 2.1.

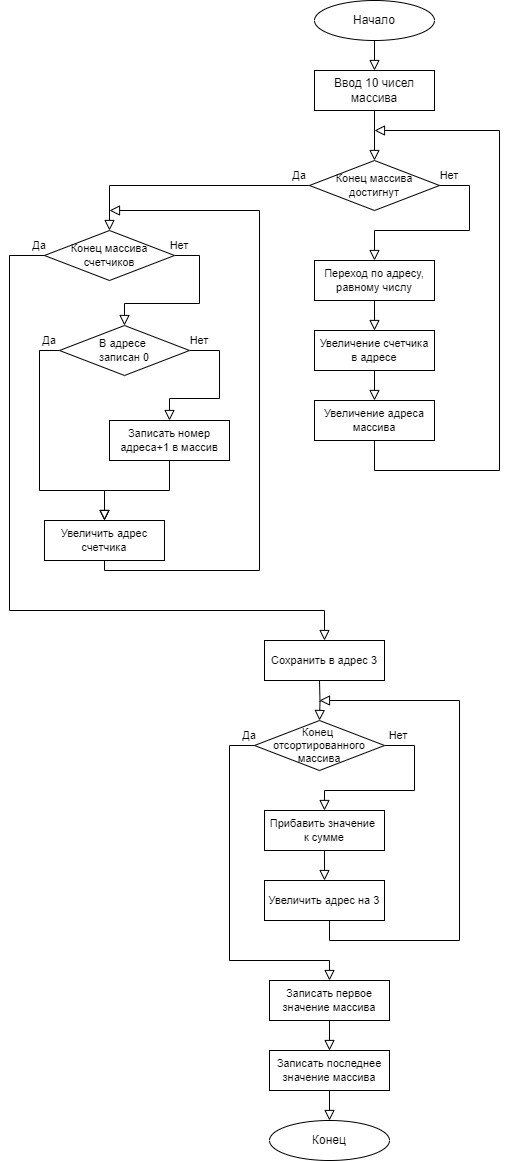


Рисунок 2.1 – Блок-схема алгоритма

2.2.2 Далее была реализована ассемблерная программа, которая сортирует введенный массив методом подсчета, выводит сумму элементов, стоящих на позициях, кратных трём, а также выводит максимальный элемент и минимальный. Код программы продемонстрирован в листинге 2.1.

Листинг 2.1 – Программа сортировки массива

mvi h, 02H

mvi l, 00H

mvi m, F0H

inr l

mvi m, 0BH

inr l

mvi m, 01H

inr l

mvi m, 09H

inr l

mvi m, 0CH

inr l

mvi m, 02H

inr l

mvi m, E1H

inr l

mvi m, 02H

inr l

mvi m, 03H

inr l

mvi m, 07H

mvi h, 03H

mvi b, FFH

mvi d, 00H

mvi e, 0AH

count:

mvi h, 02H

mov l, d

mov a, m

mvi h, 03H

mov l, a

dcr l

inr d

inr m

dcr b

dcr e

jnz count

mvi b, 00H

mvi d, 00H

mvi l, 00H

mvi c, 00H

mvi a, FFH

sort:

mvi a, 0AH

mvi h, 03H

mov l, c

mov e, m

inr c

cmp d

jz sort\_loop\_out

jmp print

print:

mvi a, 00H

cmp e

jz e\_is\_null

mvi h, 04H

mov l, d

mov m, c

inr d

dcr e

jnz print

e\_is\_null:

dcr c

inr c

jnz sort

sort\_loop\_out:

mvi h, 04H

mvi l, 03H

mvi b, 00H

mvi e, FFH

sum:

mov a, b

add m

mov b, a

inr l

inr l

inr l

mov a, l

cmp e

jc sum

mvi h, 05H

mvi l, 02H

mov m, b

mvi h, 04H

mvi l, 00H

mov a, m

mvi h, 05H

mov m, a

mvi h, 04H

mvi l, 09H

mov a, m

mvi h, 05H

mvi l, 01H

mov m, a

hlt

2.2.3 Для написанной программы была подсчитана ее длительность выполнения. Было выделено число тактов для используемых в программе команд: MVI r, data – 7 тактов, INR r – 5 тактов, MOV r1, r2 – 5 тактов, DCR r – 5 тактов, CMP r – 4 такта, JZ addr – 10 тактов (если переход выполняется) или 7 тактов (если переход не выполняется), JMP addr – 10 тактов, ADD r – 4 такта, JC addr – 10 тактов (если переход выполняется) или 7 тактов (если переход не выполняется), HLT – 7 тактов. Со знанием этой информации был произведен подсчет:

Кол-во тактов для инициализации массива: 12\*7 + 5\*9 = 129;

Кол-во тактов для заполнения массива счетчиков: 7\*4 + (7\*3 + 5\*6 + 10) \*255 = 15583;

Кол-во тактов для вывода отсортированного массива: 7\*5 + ( 7\*4 + 5\*2 + 4 + 10)\*10 + ( 7\*3 + 4 + 10 + 5\*3)\*10 + (7 + 4 + 10)\*245 + (5\*2 + 10)\*246 = 74575;

Кол-во тактов для подсчета суммы элементов: 7\*4+(5\*6 + 4\*5 + 10) \*3 = 208;

Кол-во тактов для вывода суммы, максимального и минимального элементов:

7\*11 + 5\*4 = 97;

Общее кол-во тактов: 129 + 15583 + 74575 + 208 + 97 = 90592;

При тактовой частоте работы процессора, составляющей 2МГц (2,000,000 тактов в секунду), время, потраченное на выполнение программы будет составлять: 90592/2000000 ≈ 40 миллисекунд.

**Выводы**

В ходе работы была исследована архитектура и основные блоки 8-разрядного процессора. Также было исследовано взаимодействие основных блоков процессора при выполнении команд разных типов. Помимо этого, были приобретены навыки написания и отладки ассемблерных программ в эмуляторе KP580 Emulator.