

## **5 ЛАБОРАТОРНАЯ РАБОТА №5**

### **«ИССЛЕДОВАНИЕ МЕТОДОВ ПРОГРАММИРОВАНИЯ ОБРАБОТКИ СТРОКОВЫХ ДАННЫХ»**

#### **5.1 Цель работы**

Изучить основные команды языка ассемблера для обработки строковых данных и команды передачи управления, исследовать их воздействие на процесс ассемблирования и формирования листинга программы.

Исследовать особенности функционирования блоков 16–разрядного микропроцессора при выполнении команд обработки строк и передачи управления. Приобрести практические навыки программирования на языке ассемблера МП 8086 задач обработки линейных массивов.

#### **5.2 Ход выполнения работы**

5.5.1 После изучения методических указаний к лабораторной работе была оптимизирована программа из приложения. Необходимо заполнить 160 ячеек области памяти, начинающейся с адреса MAS1 рядом натуральных чисел. Программа была отлажена, код программы продемонстрирован в листинге 5.1.

##### **Листинг 5.1 – Текст программы**

```
; multi-segment executable file template.
data segment
i db 0
SUM DW 0
ELEMENT_POSITION DW 0
ELEMENT DB 0
MAS1 DB 160 DUP(0)
MAS2 DB 160 DUP(0)
last_digit DB 16
ends

stack segment
dw 128 dup(0)
ends

code segment
start:
; set segment registers:
mov ax, data
mov ds, ax
mov es, ax

LEA DI, MAS1
MOV CX, 160
```

```

MOV BX, 1

FILL_LOOP:
mov bh,i
mov MAS1[si],bh
inc i
inc si
loop FILL_LOOP

LEA SI, MAS1
LEA DI, MAS2
MOV CX, 160
REP MOVSB

MOV AL, [last_digit]
MOV CX, 160
LEA SI, MAS1

FIND_LOOP:
CMP MAS1[SI], AL
JE FOUND
INC SI
MOV CL, MAS1[SI]
MOV ELEMENT, CL
MOV ELEMENT_POSITION, SI
LOOP FIND_LOOP

JMP NOT_FOUND

NOT_FOUND:
MOV ELEMENT_POSITION, -1
MOV ELEMENT, -1

FOUND:

MOV CX, 160
LEA SI, MAS1
XOR AX, AX
XOR BX, BX
SUM_LOOP:
MOV BL, MAS1[SI]
ADD AX, BX
INC SI
LOOP SUM_LOOP

MOV SUM, AX

mov ax, 4c00h ; exit to operating system.
int 21h
ends
end start ; set entry point and stop the assembler.

```

5.5.3 Для программы была подсчитана длительность выполнения. Было выделено число тактов для используемых команд: mov - 2 такта, add – 3 такта, cmp - 2 такта, je - 2 такта, inc - 2 такта, loop - 3 такта.

Со знанием этой информации был произведен подсчет:

Инициализация массива: общее количество тактов =  $150 * (2 + 2 + 2 + 2 + 3) = 1650$  тактов.

Копирование массива: общее количество тактов =  $160 * 8 = 1280$  тактов.

Цикл поиска: общее количество тактов =  $160 * (2 + 2 + 2 + 2 + 3) = 1760$  тактов.

Суммирование элементов: общее количество тактов =  $160 * (2 + 3 + 2 + 3) = 1600$  тактов.

Общее количество тактов для программы:  $1650 + 1280 + 1760 + 1600 = 6290$  тактов.

При тактовой частоте работы процессора, составляющей 5 МГц (5,000,000 тактов в секунду), время, потраченное на выполнение программы, будет составлять:  $6000 / 5000000 = 0.001258$  секунд или 1.258 миллисекунд.

## **Выводы**

В ходе работы были изучены основные команды языка ассемблера для обработки строковых данных и команды передачи управления, было исследовано их воздействие на процесс ассемблирования и формирования листинга программы. Помимо этого, были исследованы особенности функционирования блоков 16-разрядного микропроцессора при выполнении команд обработки строк и передачи управления. Также были приобретены практические навыки программирования на языке ассемблера МП 8086 задач обработки линейных массивов.