**4 ЛАБОРАТОРНАЯ РАБОТА №4**

**«ИССЛЕДОВАНИЕ МЕТОДОВ АДРЕСАЦИИ И ПРОГРАММИРОВАНИЯ АРИФМЕТИЧЕСКИХ И ЛОГИЧЕСКИХ ОПЕРАЦИЙ»**

4.1 Цель работы

Изучить основные директивы языка ассемблера, исследовать их воздействие на процесс ассемблирования и формирования листинга программы.

Исследовать особенности функционирования блоков 16-разрядного микропроцессора при выполнении арифметических и логических операций и при использовании различных способов адресации. Приобрести практические навыки программирования на языке ассемблера МП 8086 арифметических и логических операций с применением различных способов адресации.

4.2 Ход выполнения работы

4.4.1 После изучения методических указаний к лабораторной работе была написана программа в формате com, которая подсчитывает следующее выражение: Х = 3(А - 2В) + 50 – С / 2. Программа была отлажена, код программы продемонстрирован в листинге 4.1.

Листинг 4.1 – Программа в формате com

org 100h

start:

; Calculate X = 3 \* (A - 2 \* B) + 50 - C / 2

mov dx, a ; DX = A

shl b, 1 ; B = 2 \* B

sub dx, b ; DX = A - 2 \* B

mov ax, dx ; AX = A - 2 \* B

mov dx, 3

imul dx ; AX = 3 \* (A - 2 \* B)

add ax, 50 ; AX = 3 \* (A - 2 \* B) + 50

mov dx, c ; DX = C

shr dx, 1 ; DX = C / 2

sub ax, dx ; AX = 3 \* (A - 2 \* B) + 50 - C / 2

; Store result in memory for further use if needed

mov result, ax

; Exit program

mov ax, 4C00h

int 21h

result dw 0

a dw 1

b dw 2

c dw 2

4.4.2 Далее эта же программа была отредактирована для формата exe, это показано в листинге 4.2.

Листинг 4.2 – Программа в формате exe

STACKSG SEGMENT PARA STACK 'stack'

STACKSG ENDS

DATASG SEGMENT PARA 'Data'

result dw 0

a dw 1

b dw 2

c dw 2

DATASG ENDS

CODESG SEGMENT PARA 'Code'

BEGIN PROC MAIN

ASSUME CS:CODESG, DS:DATASG, SS:STACKSG, ES:DATASG

mov ax, DATASG

mov ds, ax

mov es, ax

; Calculate X = 3 \* (A - 2 \* B) + 50 - C / 2

mov dx, a ; DX = A

mov ax, b

shl ax, 1 ; AX = 2 \* B

sub dx, ax ; DX = A - 2 \* B

mov ax, dx ; AX = A - 2 \* B

mov dx, 3

imul dx ; AX = 3 \* (A - 2 \* B)

add ax, 50 ; AX = 3 \* (A - 2 \* B) + 50

mov dx, c ; DX = C

shr dx, 1 ; DX = C / 2

sub ax, dx ; AX = 3 \* (A - 2 \* B) + 50 - C / 2

; Store result in memory for further use if needed

mov result, ax

; Exit program

mov ax, 4C00h

int 21h

BEGIN ENDP

CODESG ENDS

END [BEGIN]

4.4.3 Для программ были подсчитаны их длительности выполнения. Было выделено число тактов для используемых в первой программе команд: mov - 2 такта, add – 3 такта, sub – 3 такта, div - 80 тактов, imul - 80 тактов, shl - 2 такта, int 21h - 51 такт.. Со знанием этой информации был произведен подсчет:

Общее кол-во тактов: 2 + 2 + 2 + 80 + 2 + 2 + 2 + 3 + 2 + 2 + 80 + 3 + 2 + 2 + 2 + 3 + 2 + 51 = 246 такта;

При тактовой частоте работы процессора, составляющей 5МГц (5,000,000 тактов в секунду), время, потраченное на выполнение программы будет составлять: 246/5000000 ≈ 0,05 миллисекунд.

Во второй программе используются те же операции, но присутствуют дополнительные операции для работы с регистрами.

Общее кол-во тактов: 2 + 2 + 2 + 2 + 2 + 2 + 80 + 2 + 2 + 2 + 3 + 2 + 2 + 80 + 3 + 2 + 2 + 2 + 3 + 2 + 51 = 252 такт;

Время, потраченное на выполнение программы будет составлять: 252/5000000 ≈ 0,05 миллисекунд.

**Выводы**

В ходе работы были изучены основные директивы языка ассемблера, исследовано их воздействие на процесс ассемблирования и формирования листинга программы. Также были исследованы особенности функционирования блоков 16-разрядного микропроцессора при выполнении арифметических и логических операций и при использовании различных способов адресации. Помимо этого, были приобретены практические навыки программирования на языке ассемблера МП 8086 арифметических и логических операций с применением различных способов адресации.