Министерство науки и высшего образования РФ

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №2

по дисциплине «ЭВМ и переферийные устройства»

на тему «Работа процессора в линейных программах»

Выполнили: студенты группы 22ВВП1

Беляев Д. И.

Демин М. С.

Сергунов М. Р.

Приняли:

Никишин К. И.

Патунин Д. В.

Пенза 2025

**Цель работы**

Изучение способов адресации и арифметико-логических и посылочных операций.

**Задание**

Задание1: Составить программу вычисления по формуле (см. варианты). Операнды и результат разместить в памяти. Один из операндов A,B, С адресовать, используя базовую адресацию, остальные - базовую со смещением, константу адресовать непосредственно, результат - прямо.



Задание2: Используя листинг программы, полученный при выполнении задания1, разобрать первой записанной в сегменте кода команды add (четные варианты) или команды sub (нечетные варианты) и записать бинарные коды всех полей ее формата.

**Ход работы**

Задание 1

Написали программу удовлетворяющую условию:

* Y=A-B+(C-11) = Y=5−(−11)+12−11=17
* Базовая адрессация для операнда A
* Базовая адрессация со смещением для операндов B и C
* Непосредственная адрессация для константы – ‘-11’
* Прямая адрессация для результата

Листинг:

stseg segment para stack

dw 16 dup(?)

stseg ends

dseg segment para

a dw 5 ; A = N = 5

b dw -11 ; B = N - 16 = -11

c dw 12 ; C = N + 7 = 12

y dw ? ; Результат вычисления

dseg ends

cseg segment para

lab1 proc far

assume cs:cseg, ds:dseg, ss:stseg

push ds

mov ax, dseg

mov ds, ax

; Устанавливаем базовый адрес для операндов A, B, C

mov bx, offset a

; Сначала вычисляем (C - 11)

mov dx, [bx+4] ; DX = C

sub dx, 11 ; DX = C - 11

; Затем вычисляем (A - B)

mov ax, [bx] ; AX = A

sub ax, [bx+2] ; AX = A - B

; Складываем результаты: (A - B) + (C - 11)

add ax, dx ; AX = (A - B) + (C - 11)

; Записываем результат в переменную y

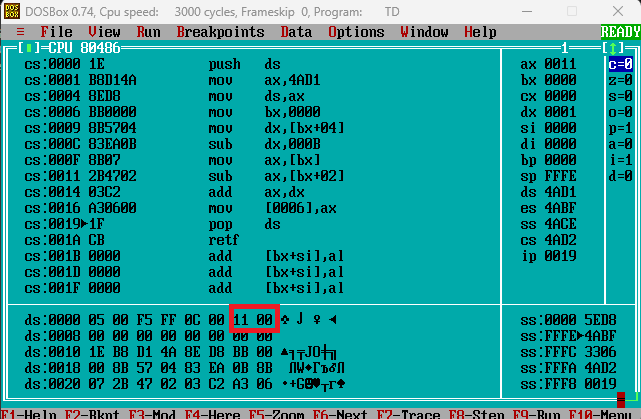
mov y, ax

pop ds

ret

lab1 endp

cseg ends

end lab1

Задание 2

Рассмотрим машинный код инструкции sub *83 EA 0B* побайтово.

1. Первый байт: 83

83 – это opcode для группы инструкций, работающих с 16-битным операндом и 8-битным немедленным значением.

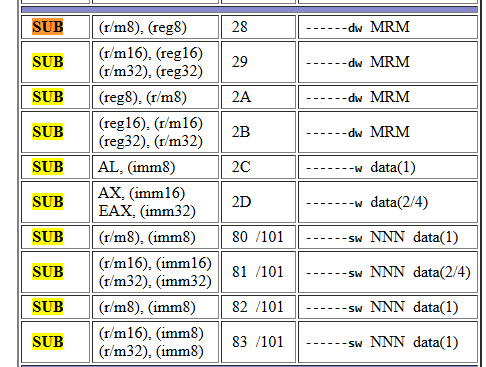
2. Второй байт: EA

Этот байт содержит поле mod-reg/rm, которое разбивается на три части:  
EA₁₆ = 1110 1010₂

* mod (2 старших бита) - Значение 11₂ означает, что операнд находится в регистре (без обращения к памяти).
* reg (следующие 3 бита) - Для инструкции SUB с opcode 83 ожидается, что поле reg = 101
* r/m (3 младших бита) - Код 010₂ соответствует регистру DX (в 16-битной адресации: 000 = AX, 001 = CX, 010 = DX, 011 = BX, и т.д.).

3. Третий байт: 0B

0B – это 8-битный немедленный операнд, который будет знаково расширен до 16 бит. В данном случае это число 0x0B, то есть десятичное 11.

****

**Вывод**

Изучили способы адресации и арифметико-логические и посылочные операций.