МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

КАФЕДРА РАДИОЭЛЕКТРОННЫХ СРЕДСТВ

Отчет по дисциплине

«Цифровые устройства и микропроцессоры»

Лабораторная работа №1

«Система команд микропроцессора х86»

Вариант 4

Выполнил: студенты группы ИНБс– 3301\_\_\_\_\_\_\_\_\_\_\_\_/ А.В. Пелых /

Проверил: профессор кафедры \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/М.А. Земцов/

Киров 2025

Цель работы: изучение системы команд и способов адресации микропроцессоров с архитектурой х86.  
  
Задание:



Рисунок 1 – Задание

Код программы:

.686

.model flat,stdcall

.stack 100h

.data

X dw 17;

Y dw 5;

Z dw 44;

M dw ?;

.code

ExitProcess PROTO STDCALL :DWORD

Start:

;расчет (Z-X-Y)

mov ax, Z

sub ax, X

sub ax, Y

;полученный результат делю на 2

cwd;

mov bx, 2

idiv bx

;расчет (X&Y)

mov bx, X

and bx, Y

;складываю (Z-X-Y)/2 c выражением (X&Y)

add ax,bx

mov M, ax ;конечный ответ

exit:

Invoke ExitProcess, 1

End Start

Вывод: в ходе выполнения лабораторной работы ознакомился с основными командами ассемблера и решил простое математическое выражение.

MOV – добавление значения в регистр или присвоение одной переменной в другую.

ADD – сложение

SUB – вычитание

AND – логическое «И»

.686

.model flat,stdcall

.stack 100h

.data

X dd 013DDh

Y dd 0715Fh

Z dd 002FEh

O dd 0007Dh

P dd 009ABh

M dd ?

R dd ?

cf dd 0

.code

ExitProcess PROTO STDCALL :DWORD

Start:

; Пункт 1

mov eax, X ; Записываем заданное значение Х

add eax, O ; Складываем заданное значение X со заданным значением O

mov X, eax ; Сохраняем результат в EAX

mov ecx, 3 ; Устанавливаем счетчик на 3 (количество вычитаний)

loop\_start:

mov ebx, Y ; Загружаем текущее значение Y в eax

sub ebx, Z ; Вычитаем Z из Y

mov Y, ebx ; Сохраняем результат обратно в Y

loop loop\_start ; Уменьшаем счетчик и продолжаем цикл, пока он не станет равным 0

; Пункт 2

; Вычисляем M = X' + X

mov edx, Y ; Загружаем новое значение Y (X')

add edx, X ; Суммируем с X

mov M, edx ; Сохраняем результат в M

; Проверяем M > OF99 (0xFF99)

cmp M, 0FF99h ; Сравниваем M с 0xFF99

jg greater\_than\_of99 ; Если M > OF99, переходим к п/п 1

; Пункт 2: R = M - 019B - cf

mov eax, M ; Загружаем M в EAX

sub eax, 019Bh ; Вычитаем 019B

sub eax, cf ; Вычитаем cf (0)

mov R, eax ; Сохраняем результат в R

jmp check\_R ; Переходим к проверке R

exit:

Invoke ExitProcess,1

End Start