**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**«Вятский Государственный Университет»**

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра РЭС

**Отчёт по лабораторной работе №1:**

**Система команд микропроцессора х86**

Дисциплина:

**«Цифровые устройства и микропроцессоры»**

Вариант 8

Выполнил: студент ИНБс-3301-01-00 Куковикин И.А.

(должность) (группа) (ФИО) (подпись)

Проверил: доцент РЭС Земцов М.А.

(должность) (кафедра) (ФИО) (подпись)

Киров 2025

**Цель работы:** изучение системы команд и способов адресации микропроцессоров с архитектурой x86.



Рисунок 1 – Вариант задания

**Решение:**

.686 ; Указываем, что используем архитектуру x86

.model flat, stdcall ; Определяем модель памяти и стандарт вызовов

.stack 100h ; Определяем стек размером 256 байт

.data ; Начало секции данных

X dw 8 ; Объявляем переменную X и инициализируем её значением 8

Y dw -7 ; Объявляем переменную Y и инициализируем её значением -7

Z dw -81 ; Объявляем переменную Z и инициализируем её значением -81

M dw ? ; Объявляем переменную M (не инициализирована)

.code ; Начало секции кода

ExitProcess PROTO STDCALL : DWORD ; Прототип функции ExitProcess

Start: ; Метка начала программы

mov ax, Z ; Загружаем значение Z в регистр AX

rol ax, 3 ; Сдвигаем значение в AX влево на 3 бита (умножение на 8)

mov Z, ax ; Сохраняем измененное значение обратно в Z

mov ax, X ; Загружаем значение X в регистр AX

imul Y ; Умножаем AX на Y (результат сохраняется в AX)

mov bx, ax ; Перемещаем результат умножения из AX в регистр BX

mov ax, Z ; Загружаем обновленное значение Z в AX

add ax, bx ; Добавляем значение BX (результат умножения) к AX

mov cx, X ; Загружаем значение X в регистр CX

add cx, Y ; Добавляем значение Y к CX

xor ax, cx ; Выполняем побитовую операцию XOR между AX и CX

mov M, ax ; Сохраняем результат операции XOR в переменную M

exit: ; Метка выхода

Invoke ExitProcess, 1 ; Вызываем функцию завершения процесса с кодом возврата 1

End Start ; Конец программы

**Вывод:** в ходе выполнения лабораторной работы ознакомился с основными командами ассемблера и выполнил простой арифметический пример.

Я изучил следующие операции:

MOV: добавление числа в регистр,

ROL: выполняет циклический сдвиг,

IMUL: выполняет умножение со знаком

ADD: сложение двух чисел,

XOR: исключающее ИЛИ.